Audio file

Your Recording 21.wav

Transcript

00:00:01 Speaker 2

In this.

00:00:01 Speaker 3

Lesson We're going to look at some machine learning basics and the fundamentals of model training. So we're going to look at a little bit behind what's happening during the training process, a little bit about how machine learning actually works. We're going to spend some time defining linear regression and understanding the maths behind it and in particular minimizing loss. Then we're going to look at the role.

00:00:14 Speaker 3

The algorithm in model training, so we'll talk about algorithms like linear learner, Xgboost. Then we'll talk about training where we have more than one input feature. We talked about an example previously where we were looking at car sales data and we maybe were looking at a car being older, costing less than the car that was newer. That would be a single input feature of age. And of course our data is going to be much more rich than that. That will have many input features.

00:00:31 Speaker 3

And lastly, we'll talk about the training process in detail, what is happening during training and how we are minimizing loss. So let's dive in. So what's happening during machine learning? Well, we need to start out with some training data. Now during this course, we are going to use the example of house price data. We're using a data set from Kaggle of London house prices. So in that data set, we have essentially a CSV file, tabular data where we have input features such as number of bedrooms, number of bathrooms, the square footage of the property, the post code, the area where there is suburban, rural, city. So those are what we would call our input features. But crucially, we need to know what price was achieved for that combination of features. You need some historical data because that way you have your input features and your.

00:01:01 Speaker 3

Target. In other words, the value you want your model to predict upon. So I need to teach the model how a certain set of input features resulted in a target feature, so that when I present the model with data it's not seen before, it should be able to derive what a reasonable prediction would be. So your training data must include the target value that ultimately you want your model to predict.

00:01:15 Speaker 3

Now before we start the training process, we're going to need to select an algorithm. Now algorithms are already developed and of course you could write your own. That would be a very advanced thing to do for a lot of data scientists. We leverage well known, well understood off the shelf algorithms. You might have heard of the names like Xgboost, LGBM, linear learner, KNN. These are all well known algorithms. So the data scientist's job would be to select an algorithm that is appropriate for the problem that you are trying to solve. So in this example, we've chosen xgoost. Now comes the training process. Now to perform training, what's going to happen is we run a training job and that training job will take your algorithm and ingest the data or the algorithm will ingest the data and it will look to build up what we call.

00:01:45 Speaker 3

During that training process, what's the algorithm is trying to do is identify the patterns and correlations and relationships in that data so that it would be able to produce a meaningful prediction. Keep in your head that idea of the car sale example where we know that an older car, when you get high prices, a newer car, and that linear regression line of best fit. The training process is trying to draw that line. It's trying to draw a line that best fits the training data, but also act as a guide for generating predictions.

00:02:04 Speaker 3

So that of the training process will be a training model artifact, and that model artifact literally is a file like model GZ, and on its own it doesn't do very much. What we need to do is host it on a compute platform like a virtual machine or a physical server, and then we would be able to perform inference against it. Now, where you host the model is up to you. We in this course are going to concentrate on using Sagemaker itself to host the model for us. That way we don't need to provision a whole other server to do the job. We'll just let Sagemaker take care of it. We're then at the point where we'd be able to handle inference requests. Now, an inference request is where you want to generate a prediction. So you need to supply input data, have your data.

00:02:34 Speaker 3

Just be one rule of data where you supply the number of bedrooms, number bathrooms, square footage that locate the coastal location of the house. But crucially, you would not have the target value because you want the model to predict that for you. So you present the data the same way as the training data was presented to the training process. You're presenting the same data but excluding the target to the model. And that's an inference request. And the model will produce is a prediction, an inference output based on the features we predict house will be worth £320,000. Great. And that's it. That's the process. And for a large majority of enterprises who perform machine learning today, they're using tabular datasets like CSV and they're using it for either regression or something called logistic regression.

00:03:04 Speaker 3

I covered a huge proportion of what companies are doing today. We know there's a lot of excitement around things like image classification and large language models, deep learning. But right now, if you've got the skills to do machine learning, tabular data for logistic regression or linear regression, you will be ahead of the game and a great place to secure your new machine learning position.

00:03:18 Speaker 3

So machine learning is going to produce a model, and that model has been trained to recognize the pattern data so that we may make predictions when we present data that hasn't been seen by the model before. Because the model has learned from the training data what the patterns are, it's then able to create the predictions autonomously after that.

00:03:31 Speaker 3

Now machine learning has a wide variety of use cases, but we see a time and time again the organizations are using machine learning for common use cases like classification objects. But that could be something as simple as classifying a transaction is fraudulent or not. Or it could be classifying whether medical image corresponds to a patient having a particular condition or not. It can be ideal for forecasting trends. Maybe we're looking to forecast what sales figures will be like in the next month based on historical data. But crucially, machine learning is about identifying relationships in your data that might not be obvious to you simply from eyeballing the data that we can uncover deep relationships and use that for business intelligence and to predict things that we can act upon. In our example, we are going to be using machine learning model to predict house prices.

00:04:01 Speaker 3

On input features such as size, age, and location. Now I use those terms linear regression and logistic regression. When I say linear regression, I'm talking about the line of best fit. And in the introduction module, we gave you an example of maybe looking up the prices of used cars and we might plot the values of used cars for a particular year. Maybe a 2024 car would cost 90 thousand, 2023 car may cost 70,000, and a 2020 car might cost 20,000. And what we can do if we plot those data points is we could ourselves draw a line of best fit. Now that line of best fit is an approximation. OK, we're trying to draw a line that goes through as many dead points as possible. That's linear regression at its simplest. Now what that would mean is that when we're thinking about using.

00:04:31 Speaker 3

Prediction, you could take a value along the X axis there, the horizontal axis for where you don't have data point, go vertically up until you intersect with the line of best fit and then look across to the Y axis and see what the value is on Y there. You've done it. You've generated a prediction using your line of best fit. That's linear regression, where you have a single input feature and a single target value. But when we understand that, it helps us build our knowledge when we start thinking about this in multiple dimensions, which is exactly what our machine learning model is going to be doing. So a model that's solving a linear regression problem is calculating the line of best fit. And that line of best fit will have a mathematical equation. And when you understand a little bit more about the mathematical equations, the described lines.

00:05:01 Speaker 3

Now you may remember when you did maths at school when you were drawing lines to a particular equation, you may see something like this, might have something like equals X + b. Now FX might be written as Y and looked in school. In other words, if I'm trying to draw a line, then it's the line. A spectator will have a slope and that may be positive as shown in the diagram here going up One Direction, if it goes down the direction, negative slope. So that is have an angle. Now that one there looks to be about 45°. It could be slow, it's 2° or it could be as high as 90°. So the slope is determined by the value or coefficient against the X in that formula. So F of X = 4 X would be a super line than F of X = 2 X. The plus B would indicate where does that line intersect the Y axis.

00:05:31 Speaker 3

Now here it looks like the line will go through the point of origin 0 for X&Y. That doesn't have to be the case. Maybe the line is for the graph cuts through where y = 10, so that B value is the point at which the line intersects the Y axis. Now keep that in mind as we go through more about linear regression because all of this is ultimately building on these points. Here's a line of best fit for our example where we're looking at used car sales prices for a number of data points there and withdrawing a line of best fit. Again, at this point we just have a single input feature that is the age of the car. And we have a single target feature which is price, so that we know for a particular age of car will determine a particular sale price. And if we.

00:06:02 Speaker 3

The point shows training data, but if we were to pick something like January 2022, we could follow that volume X axis up to our line and then go across the axis and terminal sale price would be basic linear regression. But here's an interesting point. My line of best fit is a best fit.

It's not perfect. It doesn't go through every single data point. So there's really an accuracy of that line, isn't there? How right did I get it? How best fit did I go? And there's a way that we can quantify how well our line of best fit fitted the data. The way we do that is we just measure the distance between our known sale prices and the line of best fit. Hmm. OK, So what we're going to do is we're going to calculate what we call residuals, and that's the values of the.

00:06:32 Speaker 3

The data point and the line by side we can see there for 2017 the residual value is quite small, for 2018 is quite big and 2020 big as well. So we wanted to express how well our line of best fit matched the data we could come up with some kind of expression.

00:06:43 Speaker 3

So what we're going to do is we are going to sum up all of these residual values, but we need to overcome slight problem first because the residual values above the line are expressed as positives and the residual values below the line are expressed as negatives. And if we simply sum them all together, they would sort of cancel each other out. And we get the impression our line of misfit was perfect and it's not. So what we should be thinking about is not the sign positive or negative, but simply the magnitude. How big is the value between the line and the data point? So the way we get around this is to square these residual values because if you square a negative number becomes a positive number and then you can sum it up and get the total magnitude of error. So we call this the sum of squared residuals sounds very complicated, but it's really not. We know why we're doing this. We're doing this to.

00:07:13 Speaker 3

How well our line fits the data. So if we sum up all of those squared residuals will get a number and the lower the number that that equals the better. So during the training process we will be constantly adjusting the formula that draws the line and then we'll be summing the square for the residuals to say is that better than the last time? No, let's adjust the formula again and rerun it. If you do that over and over and over, that's what the training process is, then you'll end up with a line of best fit with some of the squared residuals is the lowest it can possibly be. So when we start our model training, we know we need our training data that has our input features and things like number bedrooms, number bathroom square footage and our target feature, the thing we want to predict upon in our.

00:07:43 Speaker 3

Hush parts. Then we need our algorithm, our chosen algorithm, the Xgboost linear learner or whatever. But whatever algorithm we choose is the algorithm that's going to extract patterns from that training data. And it can do this not just in like 1 input feature, but you might have 100 input features. So a multidimensional space, something is very hard for us as humans to visualize because beyond 3 dimensions, But mathematically we absolutely can. And the algorithm should be able to generalize. We don't want to learn exactly what is in the training data set because that's not what will be presented to model. The model will be presented with brand new data that the model has never seen before. So we must be able to generalize, extract the relationships, but not precise values. And that's a clever part of what happens.

00:08:13 Speaker 3

So we're all about trying to generate accurate predictions when the model CS data is not seen before. Now the algorithm is during the training process is iteratively trying to come up with that line of best fit, OK, that line of best fit where the sum of the squared residuals would be as low as possible. So the training process is an iterative process. It's looping over and over trying to self improve with each iteration lowering that sum of squared residuals value, assuming it's a linear regression problem. So to take our house price example, we will have a set of features like room sizes, number of rooms, the postal zip code, whether or not there's a garden. And again, different data sites for house prices could have 10 input values, input features, or maybe 100.

00:08:43 Speaker 3

We say the more features you have, richer data is, but then the training process might take much longer to perform and more complex data is the relationships. It might not be to generalize and infer what those relationships are. So it's having enough features that it can work out a good accurate prediction model, but not too many features. That means that the model struggles to come up with a generalization of the relationships within our chosen target value. In this case, it's house price. So how do we come up combining those features and target and come up with them using the algorithm in a way that expresses our kind of line of best fit? But again, think back to that linear equation formula that F of X equals ax plus B OK X + b slope of the line where intersects from.

00:09:13 Speaker 3

We're trying to come up with a line of best fit. So ultimately we're just trying to come up with a mathematical formula that best describes that line, albeit a line or surface that works in a multi dimensional space. So keep in your mind 2 dimensional example. But we know that we are working beyond just single 2 dimensions.

00:09:25 Speaker 3

Keep in mind the idea of residuals, an idea that what you come up with is a language based fit through your training data, will have a loss. There will be not an exact match. And what we're trying to do is minimize that loss, minimize the sum of squared residuals.

00:09:36 Speaker 3

So the model is going to be the tool that makes predictions. So ultimately in the model, the model will ultimately store a long and complex formula so that when we present the input data, it can take that input data, apply it to the mathematical formula equals the line of best fit and derive a prediction value from it. OK, remember F of X, we look at ax plus B. Notice we're now expressing this as WX plus B. Why missing W instead of a? I'm saying W because I introduced the concept of weight. Weight is my coefficient the same thing as it was ax in the previous example? When we look machine learning, we talk about weights and biases. Weight with coefficient on the input feature and bias will be like the intercept on the Y axis. OK, now just so that we.

00:10:06 Speaker 3

In our heads about linear algebra, when I look at the lines and our equations give ones here. Just an example, y = 2, X plus four. Okay, so it's quite a steep line because 2X there's a coefficient against the X value that's determining the slope of the line plus 4. the Y intercept is at 4 on the Y axis y = 2 X. That's at the same slope as y = 2 four. But there's no bias or intercept value for zero y = X + 2. OK, well then the coefficient value of W would be 1, so it's less steep. So it's a steeper slope as y = 2, X +4, and the intercept point and Y axis is 2 compared to y = X where coefficient is 1. Therefore it's relatively steep slope and there's no bias.

00:10:36 Speaker 3

Intersection goes from zero. So I always think it helps. I know these are very simple straight line equations, but it helps us think about slopes of lines and intercepts because we're trying to get around what's happening during training in multiple dimensions. There's kind of some of this that's going on.

00:10:46 Speaker 3

Now during the training process, your algorithm will be making use of a number of different methods to self optimize on each iteration it performs to self improve its accuracy. Now one common method by machine learning models is something called gradient descent. And gradient descent is looking at the loss function for linear regression will be using the sum of the squared residuals and then thinking how could I adjust the values in the formula F of X equals WX plus B? How can I adjust the weights and biases so that the line that's created has a lower sum of squared residuals? And once I've done that, I don't want to do again, I want to adjust them again. I want to adjust the values in the right direction. So if I'm starting to get a lower sum of squared residuals, what did I do to do that? Because I want to do more of that.

00:11:16 Speaker 3

That's increasing the coefficient against a particular input feature. So gradient descent is a technique used by the algorithm to address the coefficients and biases in the function that the model is building.

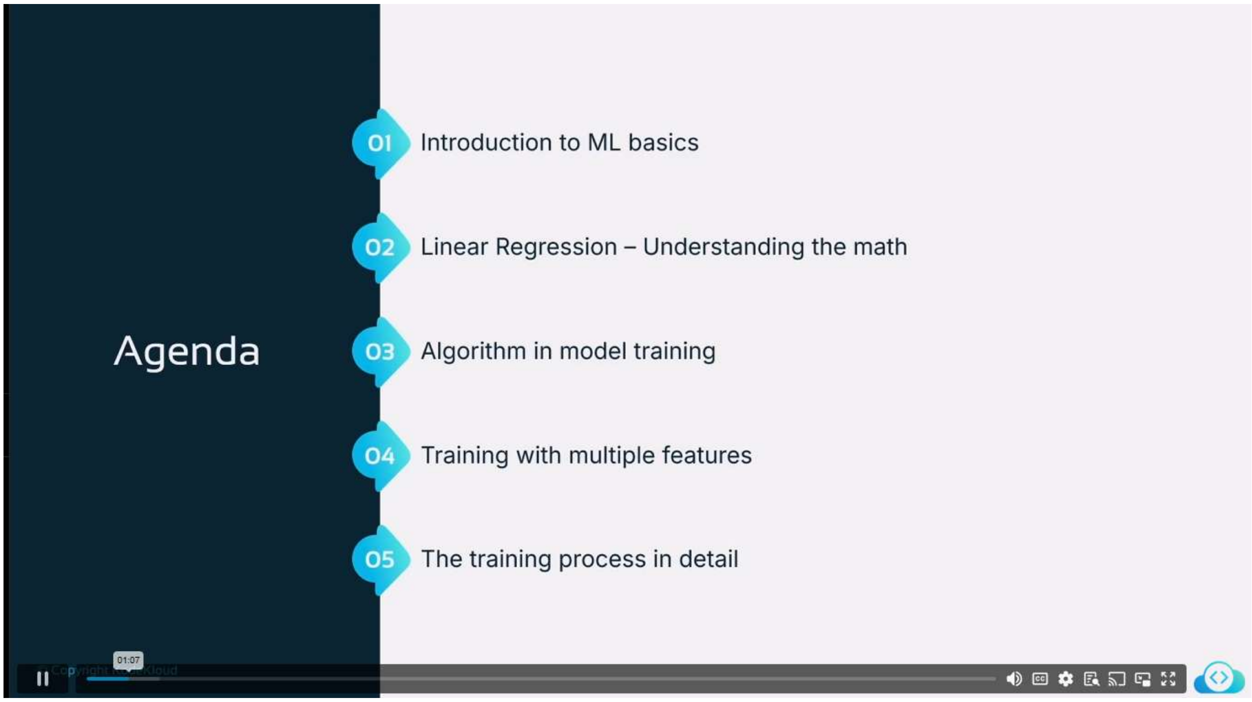
00:11:25 Speaker 3

Learning rate is about, well, how quickly ultimately the model is going to train. But what this really defines is the size of steps are taken each time. So if it needs to adjust the weights or bias of the formula, then does it do that in very small steps or does it take it in larger steps 1st and then smaller steps afterwards? So we can control the learning rate. That's something that we can influence during the training process. And we can also determine stopping criteria. So rather than iterating over the data with this algorithm 1000 times, maybe we don't get any particular benefit in accuracy or improvement in the loss function beyond the 1st 200 times. So training beyond 200 iterations over the data set is not yielding you a more accurate model, so.

00:11:55 Speaker 3

Under what criteria should we stop training? We've done as much as we're going to do to get a good model with reasonable accuracy. So we have control over that during the training process and we'll see how to do that later on in the course.

00:12:07 Speaker 2



Agenda

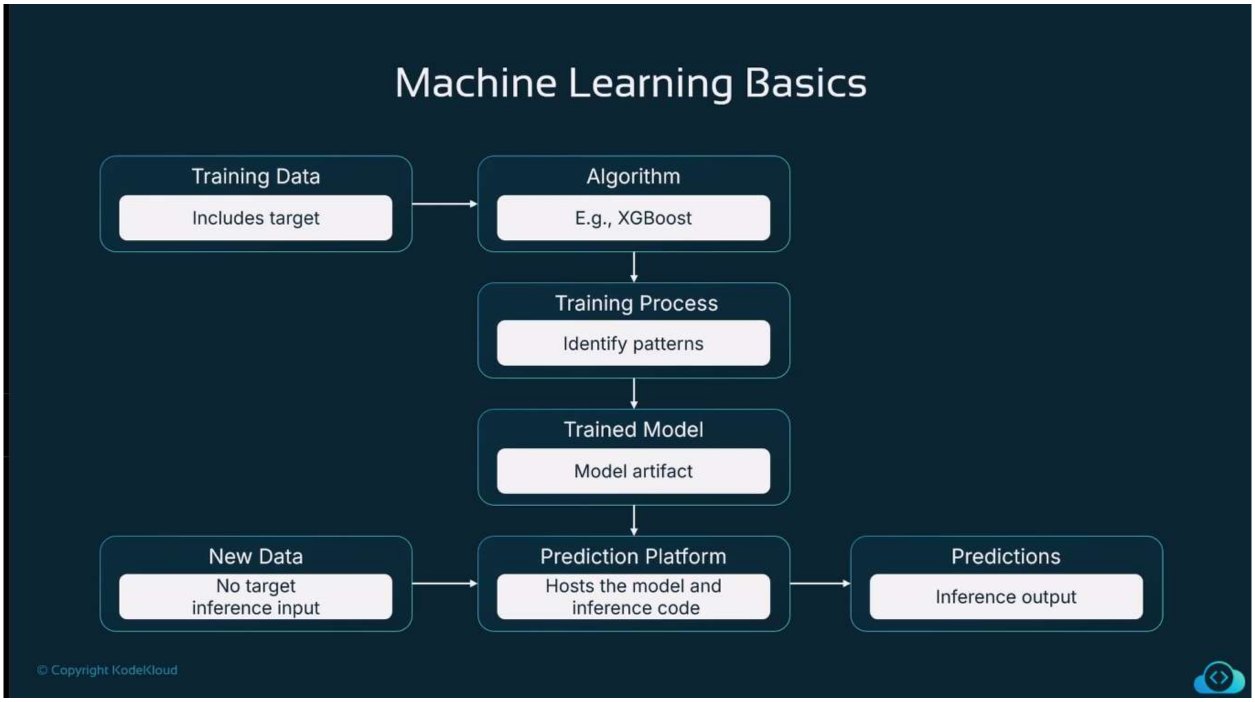
01 Introduction to ML basics

02 Linear Regression – Understanding the math

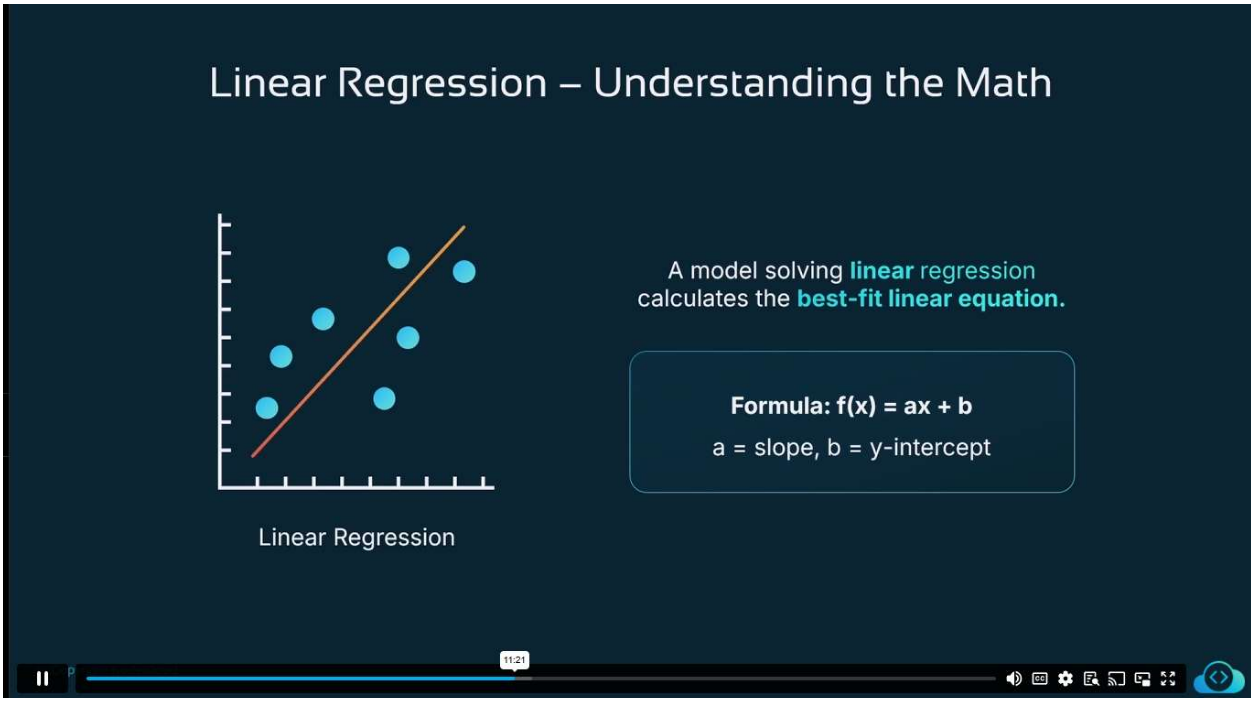
03 Algorithm in model training

04 Training with multiple features

05 The training process in detail



Machine Learning Basics



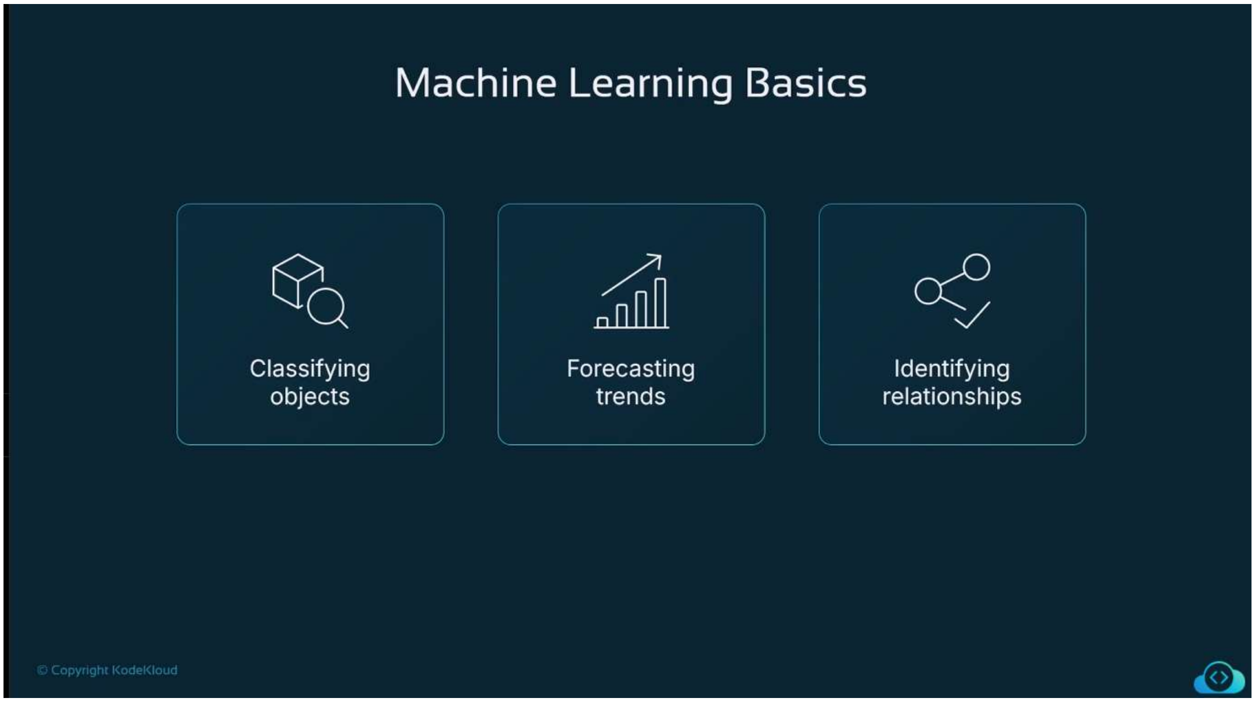
Linear Regression – Understanding the Math

Linear Regression

A model solving **linear** regression calculates the **best-fit linear equation.**

Formula: f(x) = ax + b

a = slope, b = y-intercept

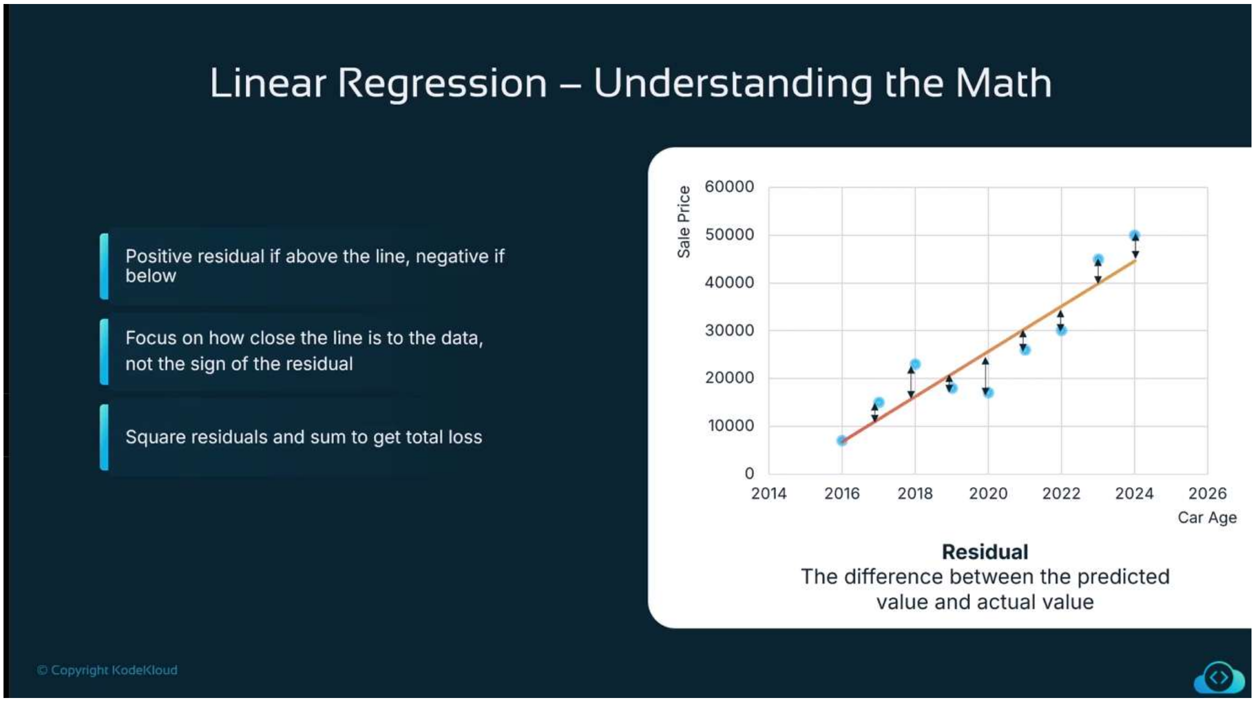


Machine Learning Basics

Classifying objects

Forecasting trends

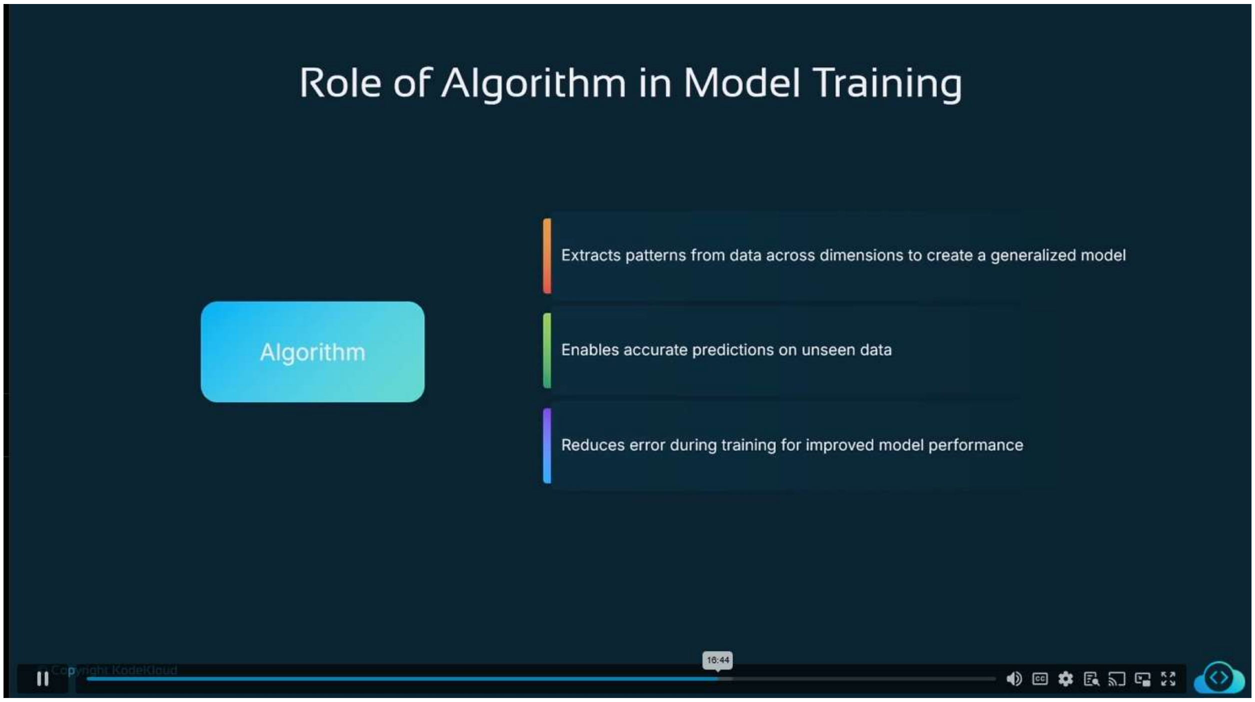
Identifying relationships



Linear Regression – Understanding the Math

- Positive residual if above the line, negative if below
- Focus on how close the line is to the data, not the sign of the residual
- Square residuals and sum to get total loss

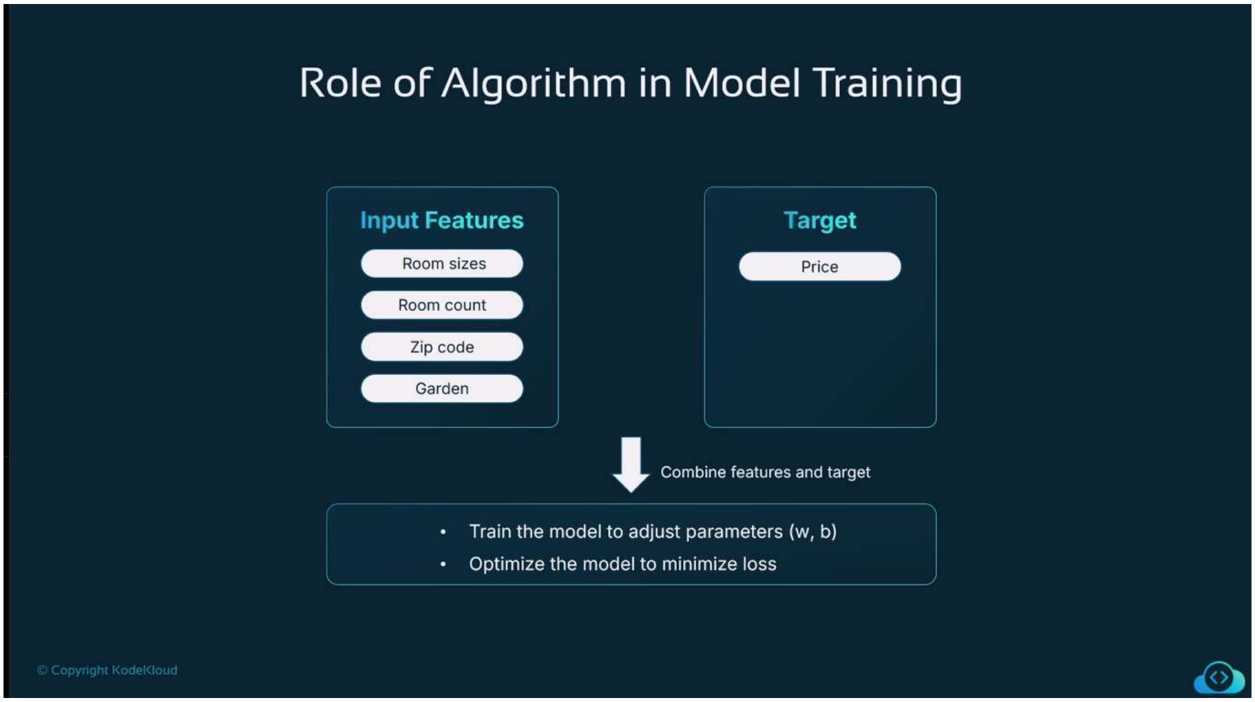
Residual
The difference between the predicted value and actual value



Role of Algorithm in Model Training

Algorithm

- Extracts patterns from data across dimensions to create a generalized model
- Enables accurate predictions on unseen data
- Reduces error during training for improved model performance



Role of Algorithm in Model Training

Input Features

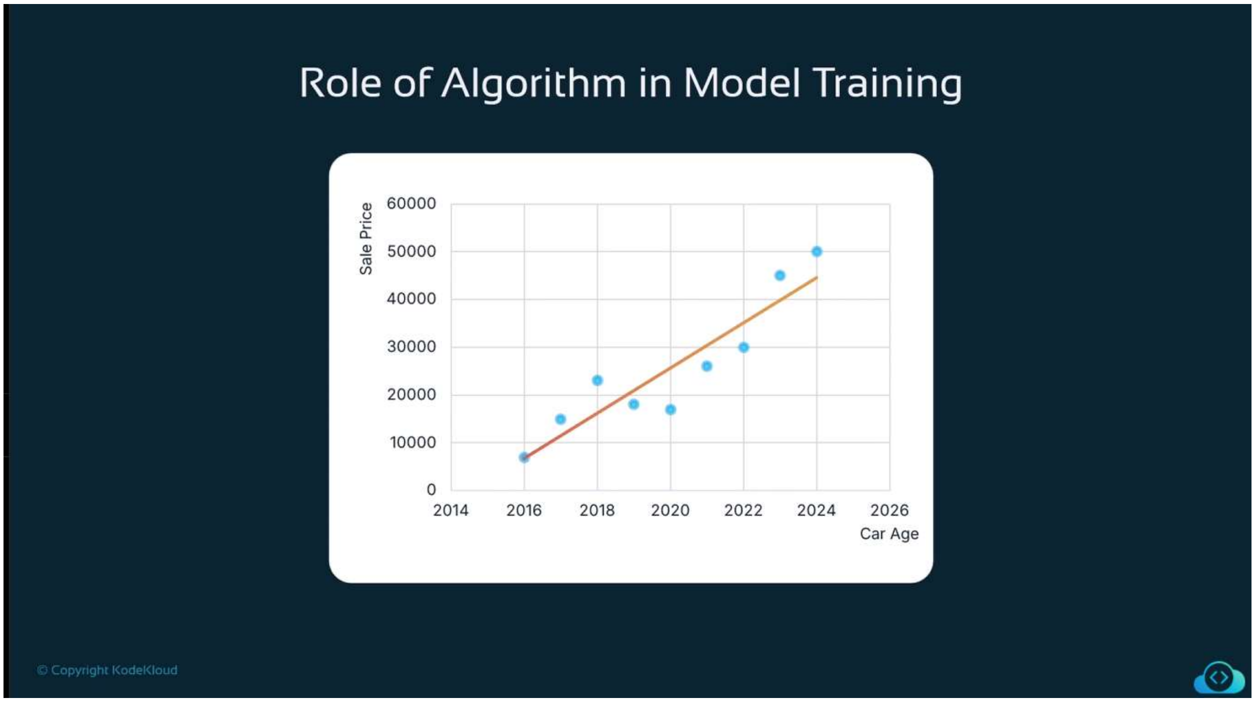
- Room sizes
- Room count
- Zip code
- Garden

Target

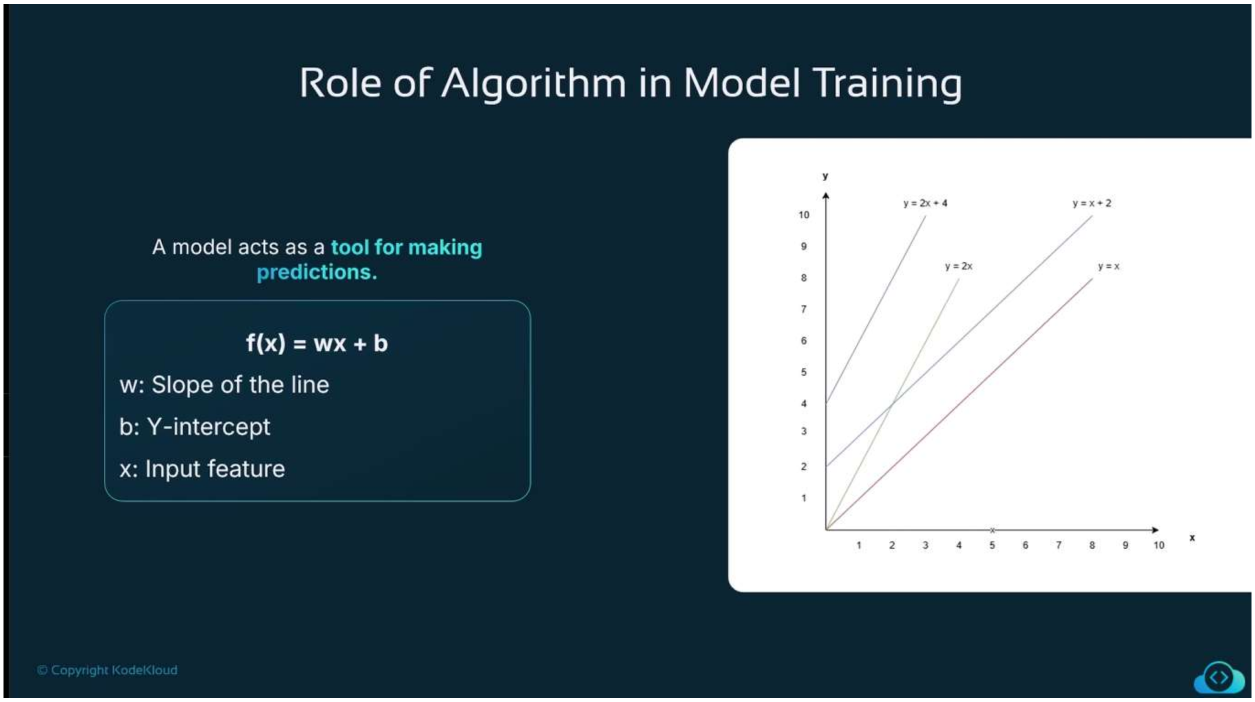
- Price

Combine features and target

- Train the model to adjust parameters (w, b)
- Optimize the model to minimize loss



Role of Algorithm in Model Training



Role of Algorithm in Model Training

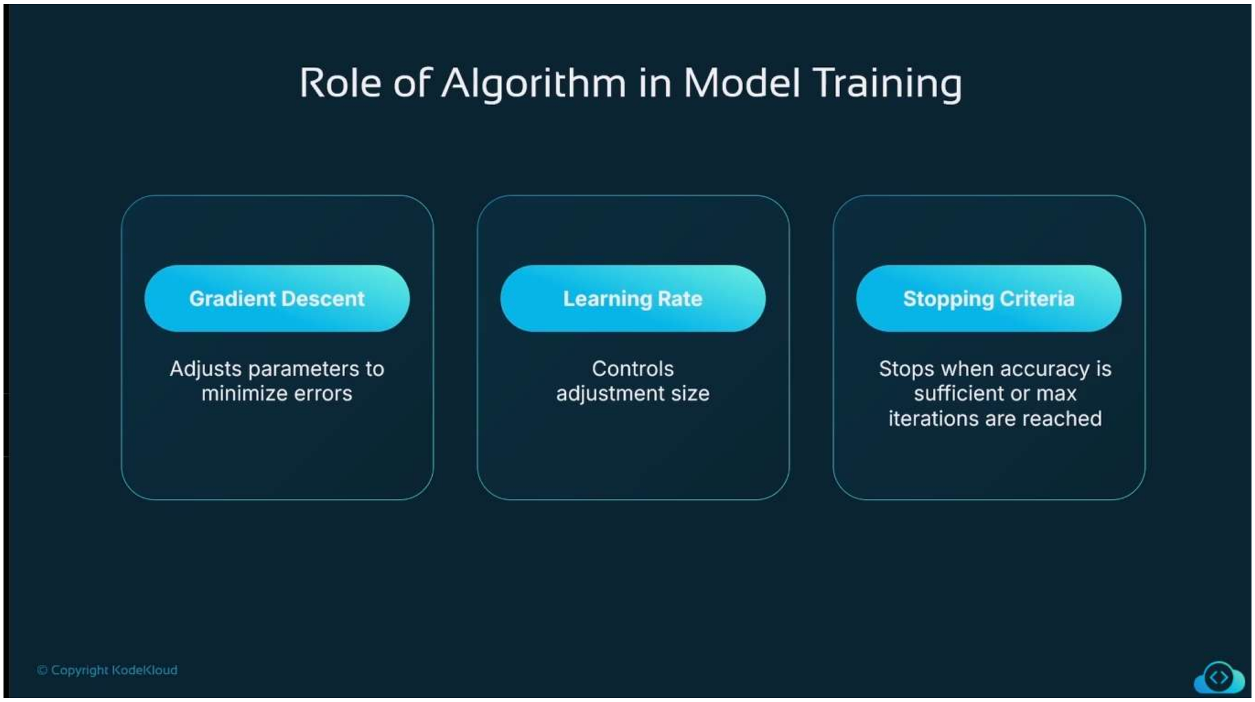
A model acts as a **tool for making predictions.**

f(x) = wx + b

w: Slope of the line

b: Y-intercept

x: Input feature



Role of Algorithm in Model Training

Gradient Descent

Adjusts parameters to minimize errors

Learning Rate

Controls adjustment size

Stopping Criteria

Stops when accuracy is sufficient or max iterations are reached