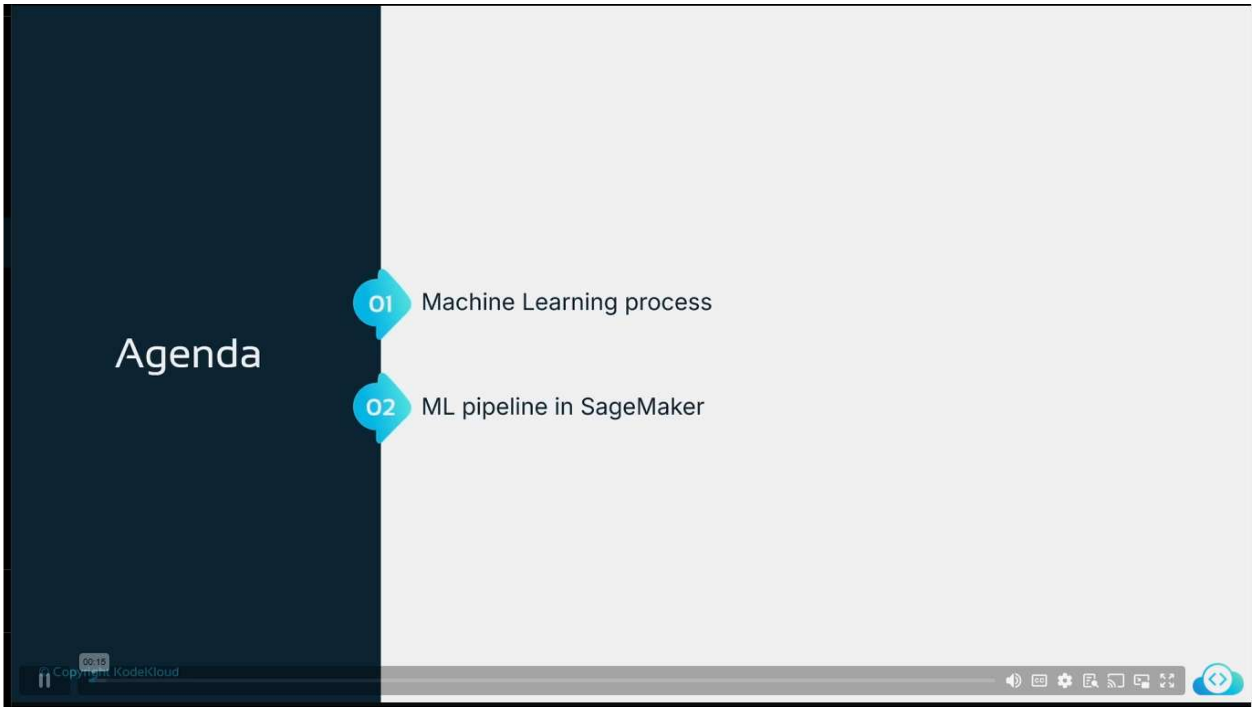
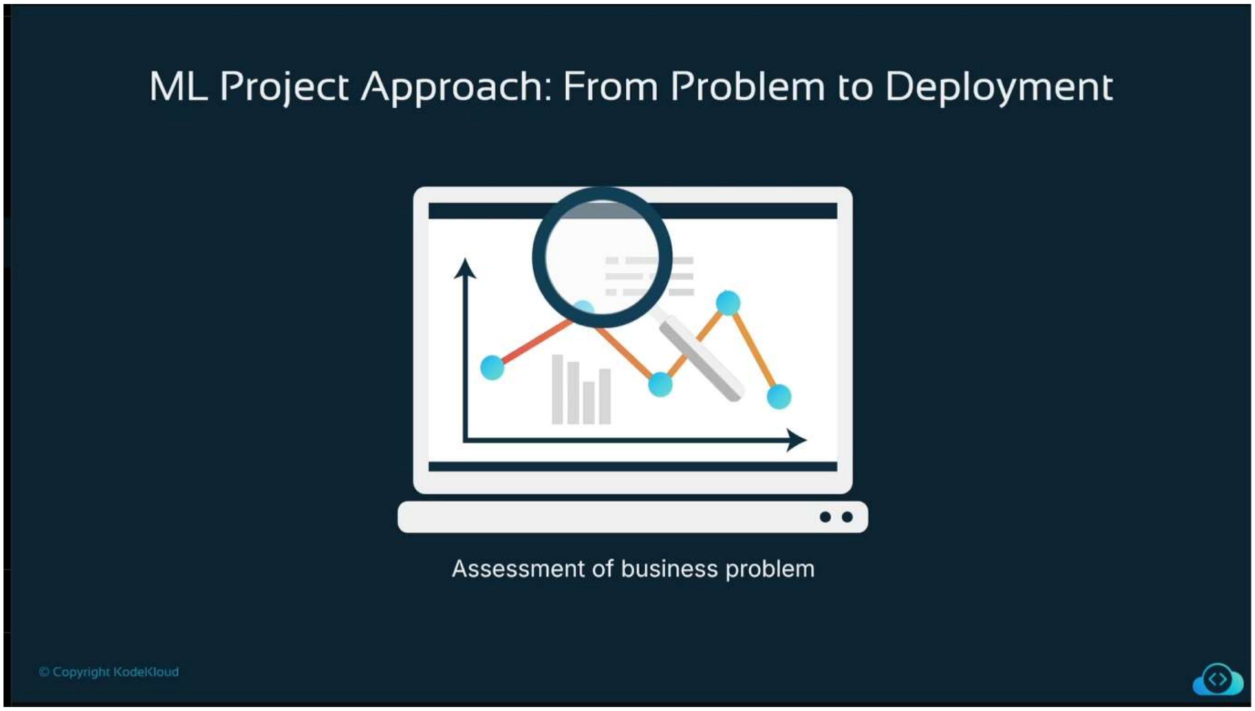
ML PipeLine



## Agenda

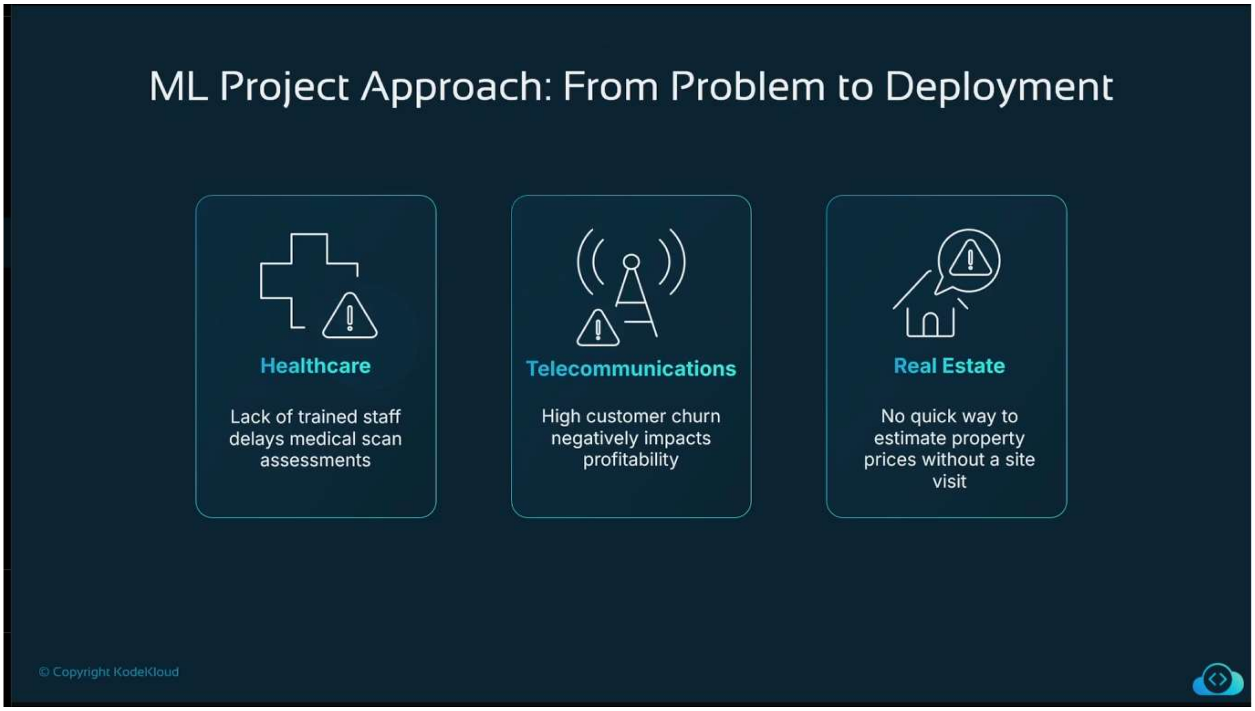
01 Machine Learning process

02 ML pipeline in SageMaker



## ML Project Approach: From Problem to Deployment

Assessment of business problem



## ML Project Approach: From Problem to Deployment

**Healthcare**

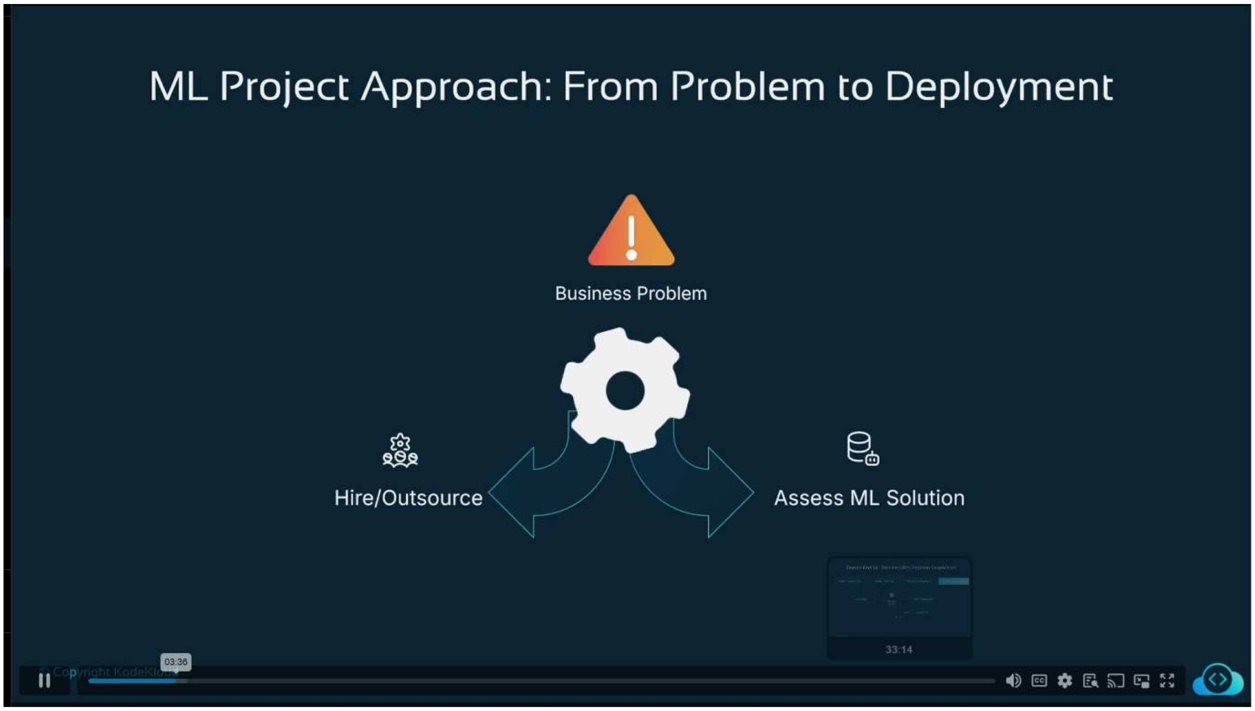
Lack of trained staff delays medical scan assessments

**Telecommunications**

High customer churn negatively impacts profitability

**Real Estate**

No quick way to estimate property prices without a site visit

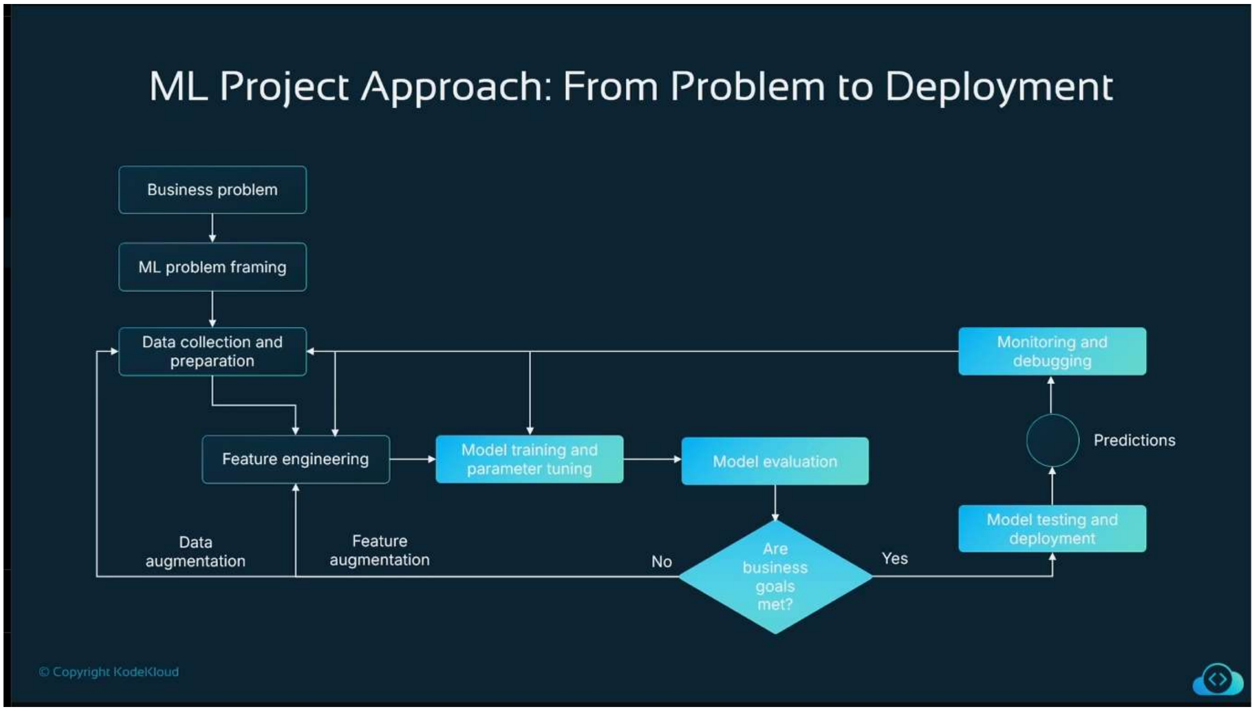


## ML Project Approach: From Problem to Deployment

Business Problem

Hire/Outsource

Assess ML Solution



## ML Project Approach: From Problem to Deployment

Business problem

ML problem framing

Data collection and preparation

Feature engineering

Model training and parameter tuning

Model evaluation

Are business goals met?

No

Yes

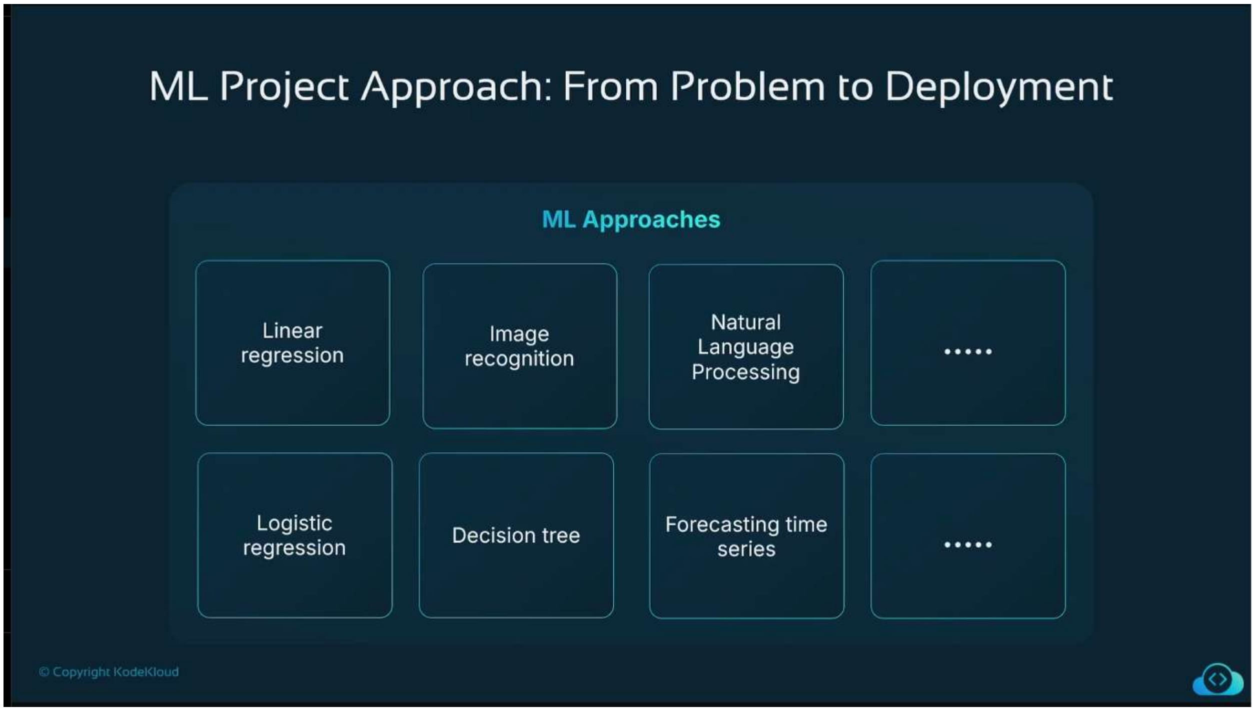
Data augmentation

Feature augmentation

Model testing and deployment

Predictions

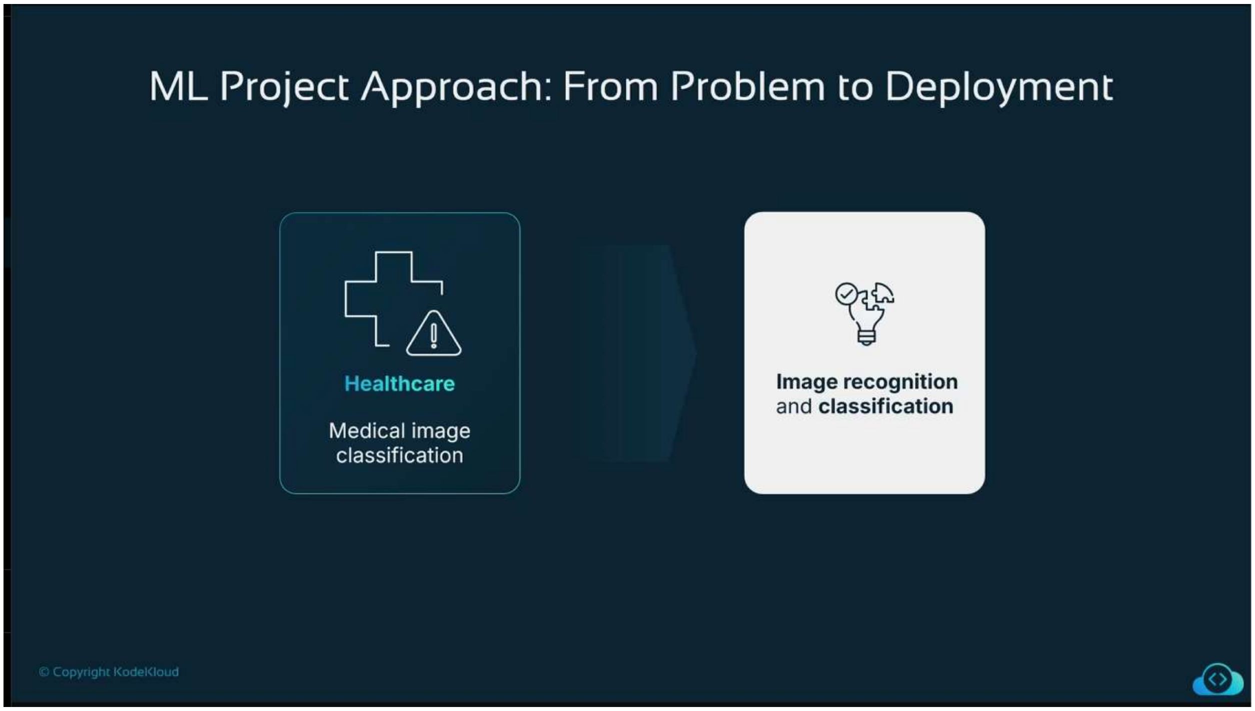
Monitoring and debugging



## ML Project Approach: From Problem to Deployment

### ML Approaches

| | | | |
|---|---|---|---|
| Linear regression | Image recognition | Natural Language Processing | ...... |
| Logistic regression | Decision tree | Forecasting time series | ...... |

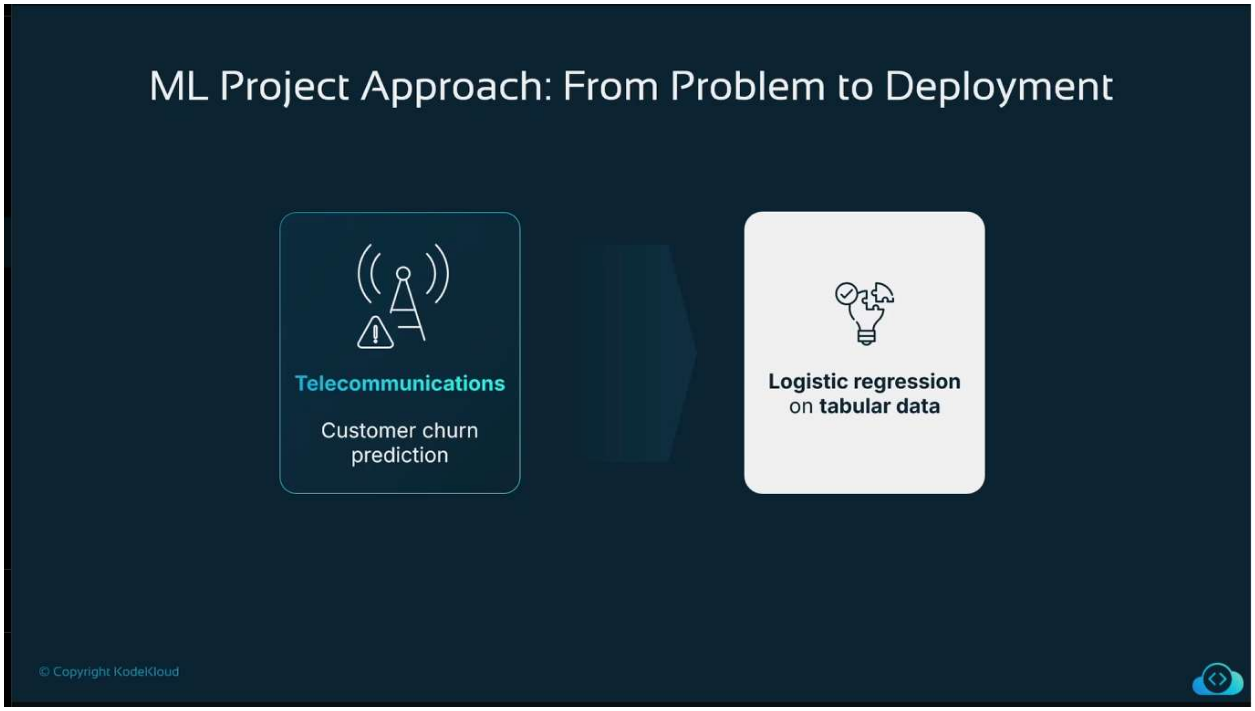


# ML Project Approach: From Problem to Deployment

**Healthcare**

Medical image classification

**Image recognition** and **classification**

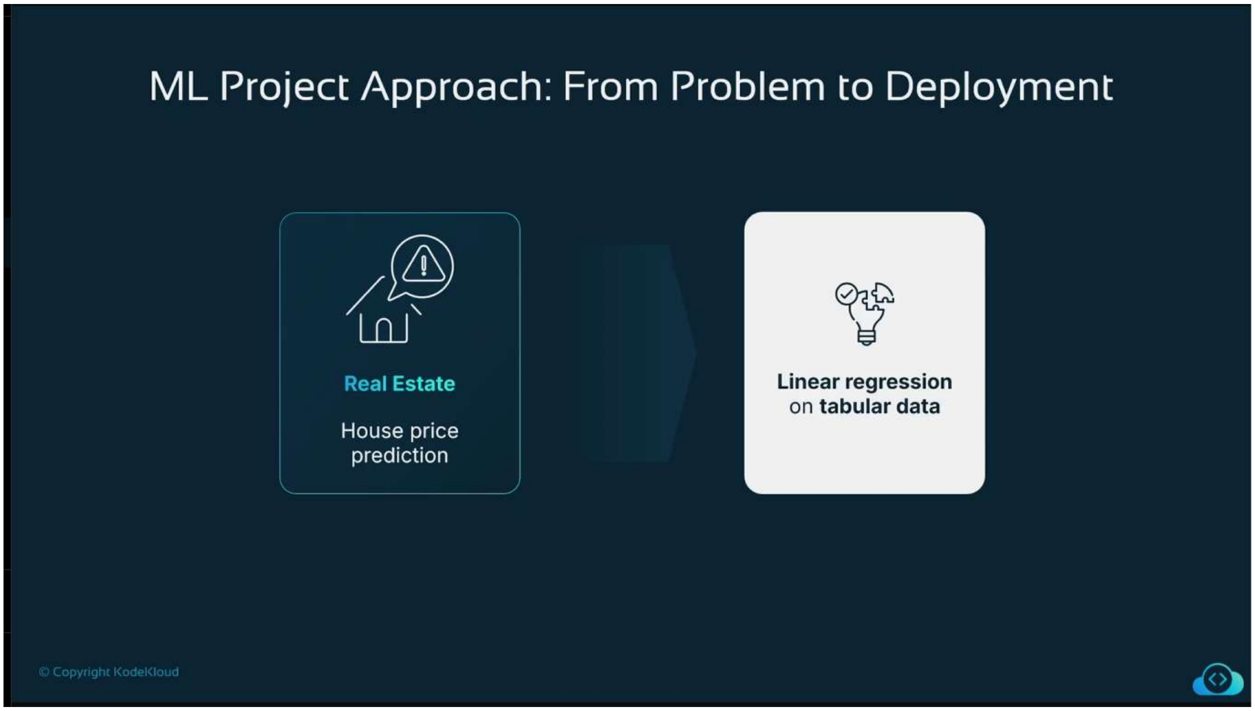


# ML Project Approach: From Problem to Deployment

**Telecommunications**

Customer churn prediction

**Logistic regression** on **tabular data**

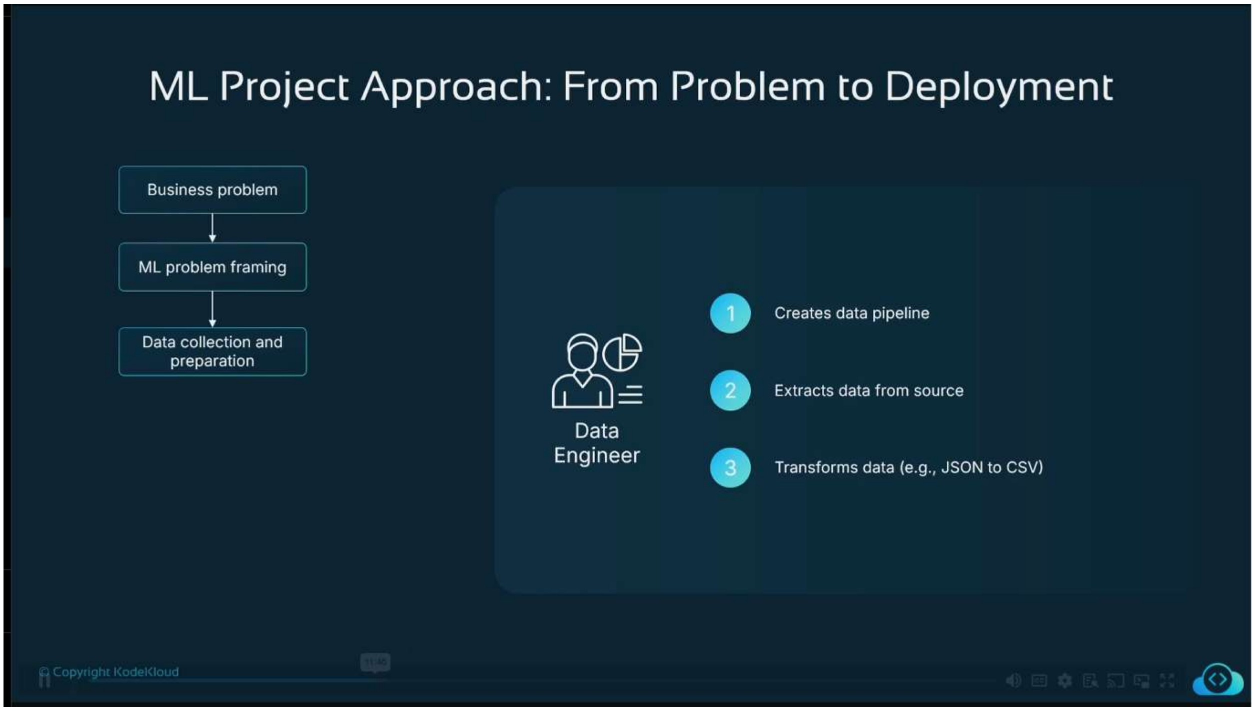


## ML Project Approach: From Problem to Deployment

**Real Estate**

House price prediction

**Linear regression** on **tabular data**



## ML Project Approach: From Problem to Deployment

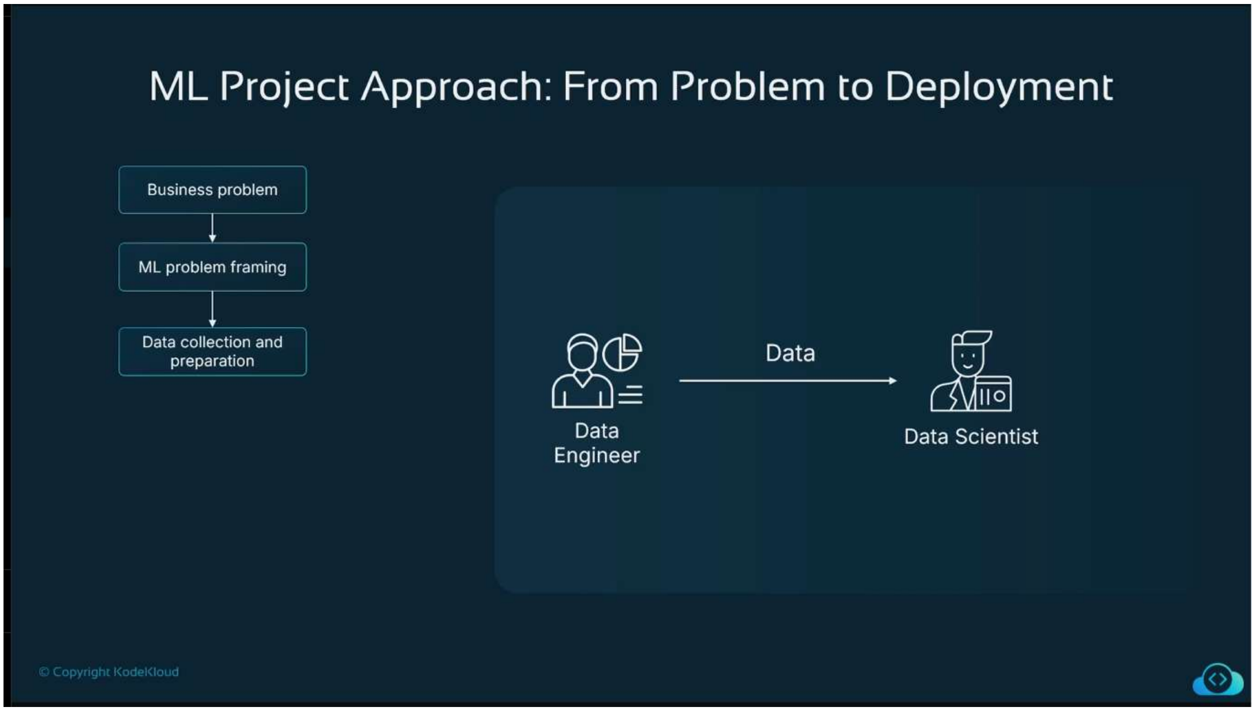
Business problem

ML problem framing

Data collection and preparation

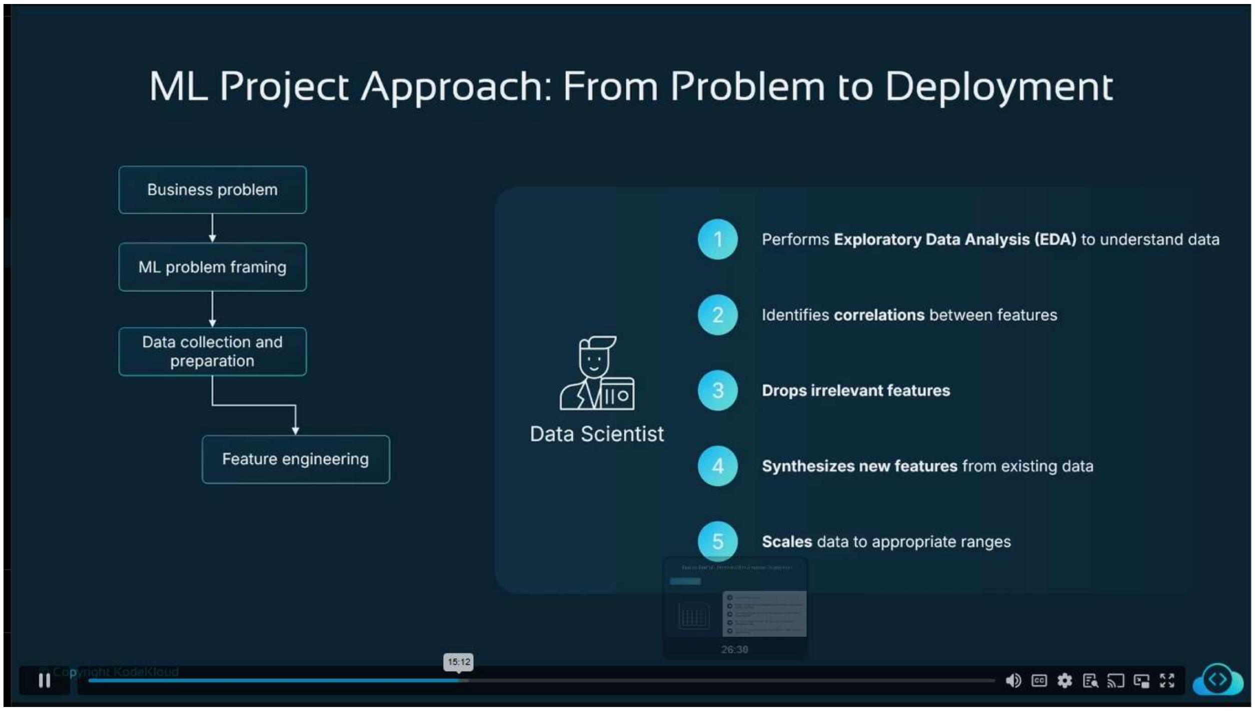
Data Engineer

1. Creates data pipeline
2. Extracts data from source
3. Transforms data (e.g., JSON to CSV)



## ML Project Approach: From Problem to Deployment

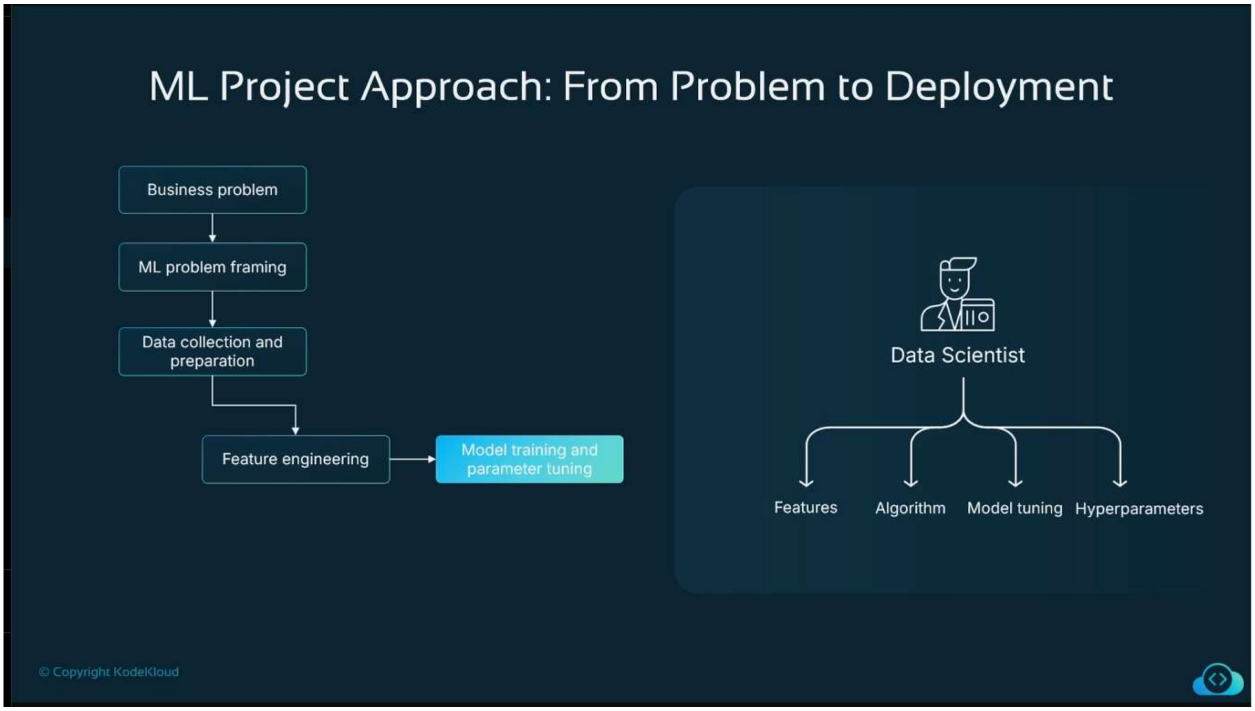
Business problem → ML problem framing → Data collection and preparation



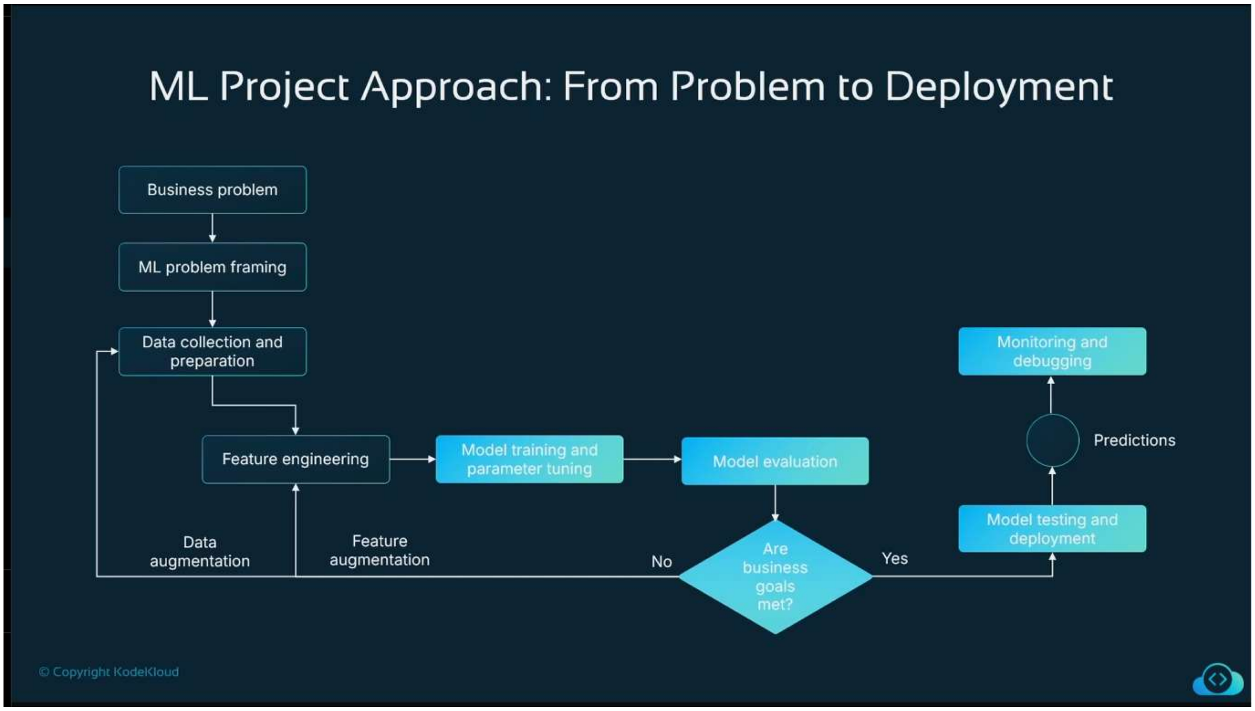
## ML Project Approach: From Problem to Deployment

Business problem → ML problem framing → Data collection and preparation → Feature engineering

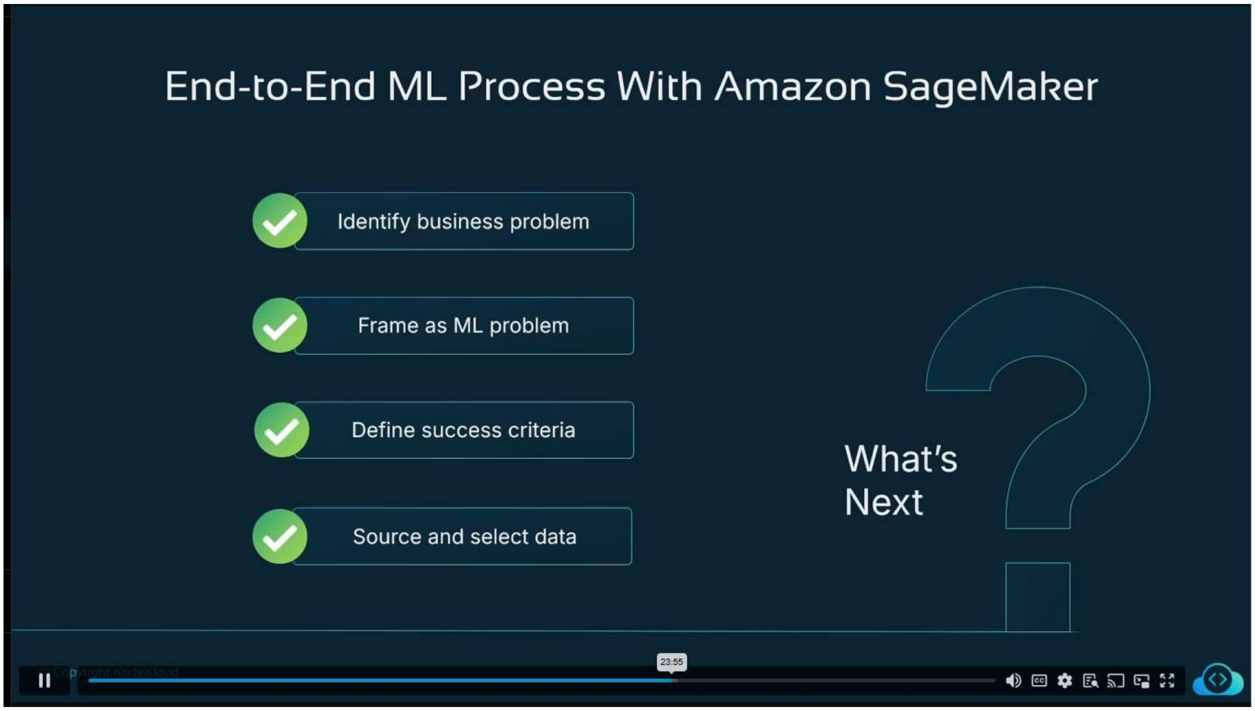
1. Performs **Exploratory Data Analysis (EDA)** to understand data
2. Identifies **correlations** between features
3. **Drops irrelevant features**
4. **Synthesizes new features** from existing data
5. **Scales** data to appropriate ranges



# ML Project Approach: From Problem to Deployment



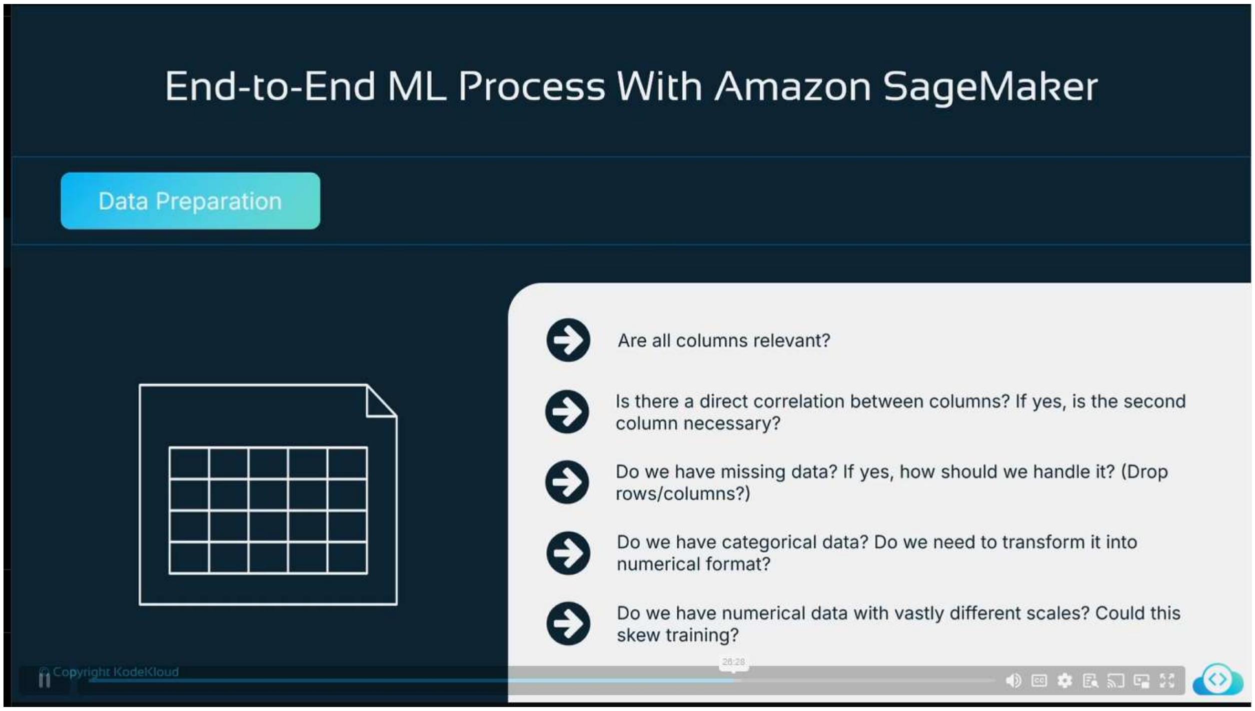
# ML Project Approach: From Problem to Deployment



## End-to-End ML Process With Amazon SageMaker

- ✓ Identify business problem
- ✓ Frame as ML problem
- ✓ Define success criteria
- ✓ Source and select data

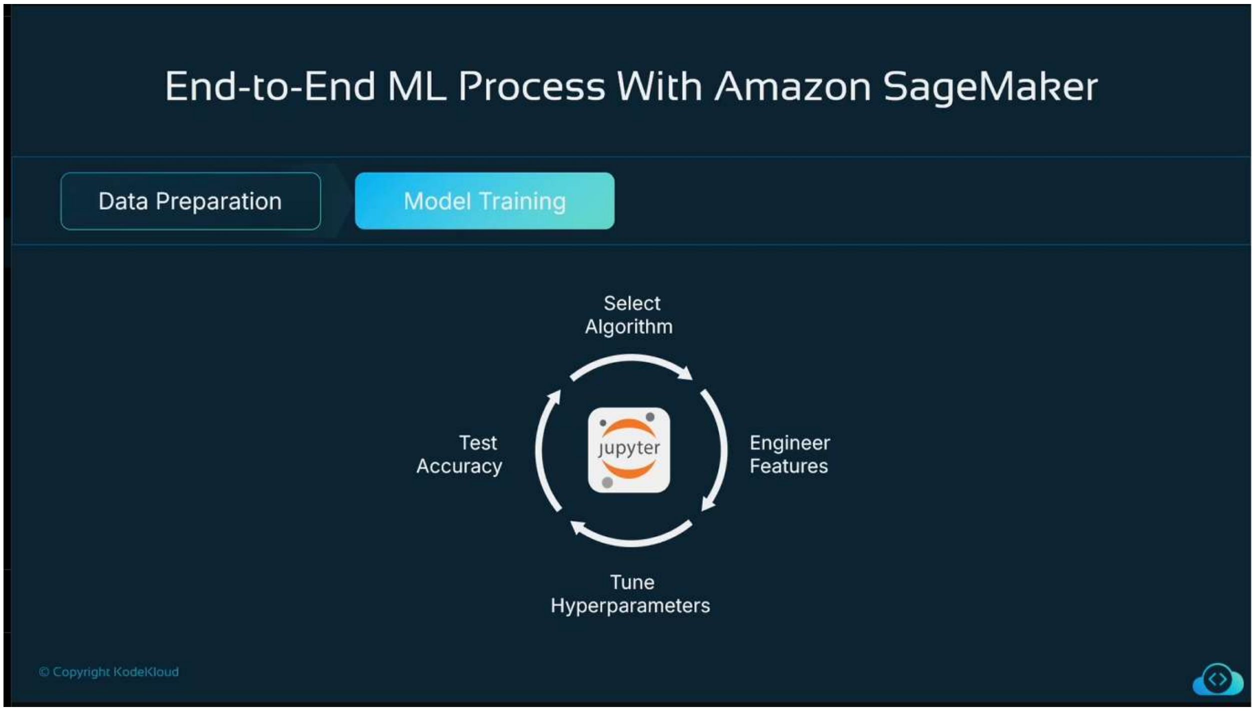
What's
Next



## End-to-End ML Process With Amazon SageMaker

Data Preparation

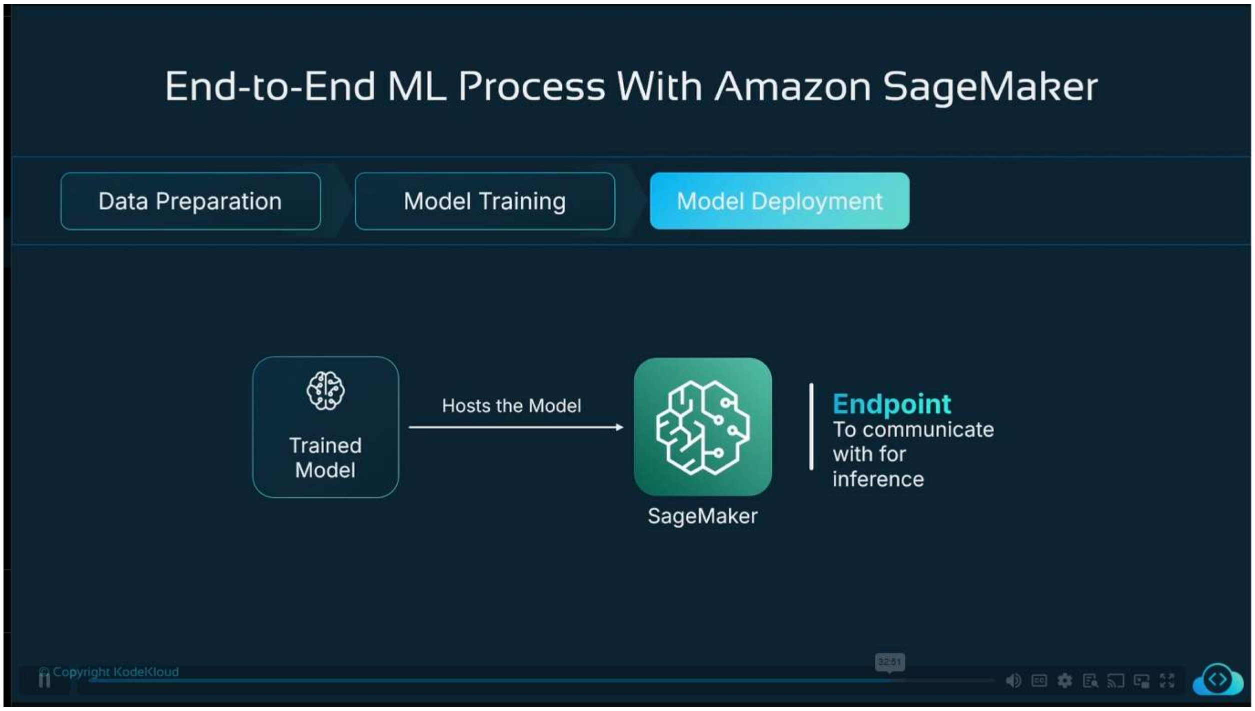
- Are all columns relevant?
- Is there a direct correlation between columns? If yes, is the second column necessary?
- Do we have missing data? If yes, how should we handle it? (Drop rows/columns?)
- Do we have categorical data? Do we need to transform it into numerical format?
- Do we have numerical data with vastly different scales? Could this skew training?



## End-to-End ML Process With Amazon SageMaker

Data Preparation

**Model Training**



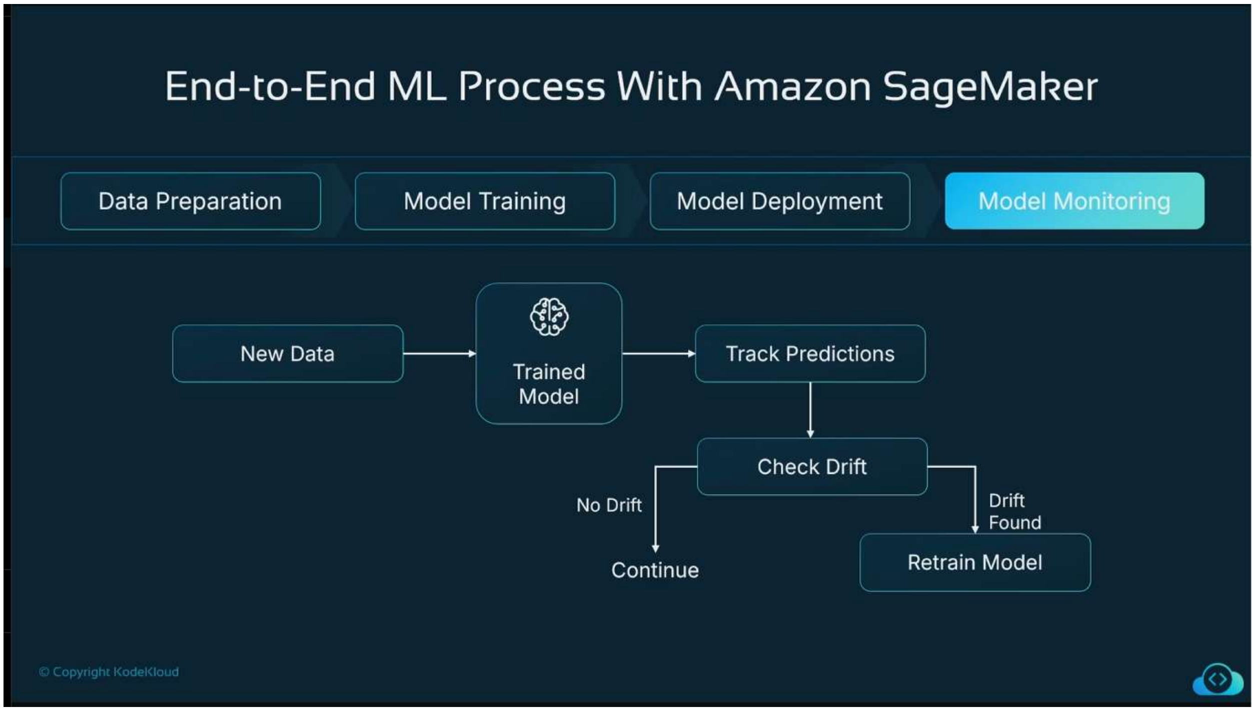
## End-to-End ML Process With Amazon SageMaker

Data Preparation

Model Training

**Model Deployment**

**Endpoint**
To communicate with for inference



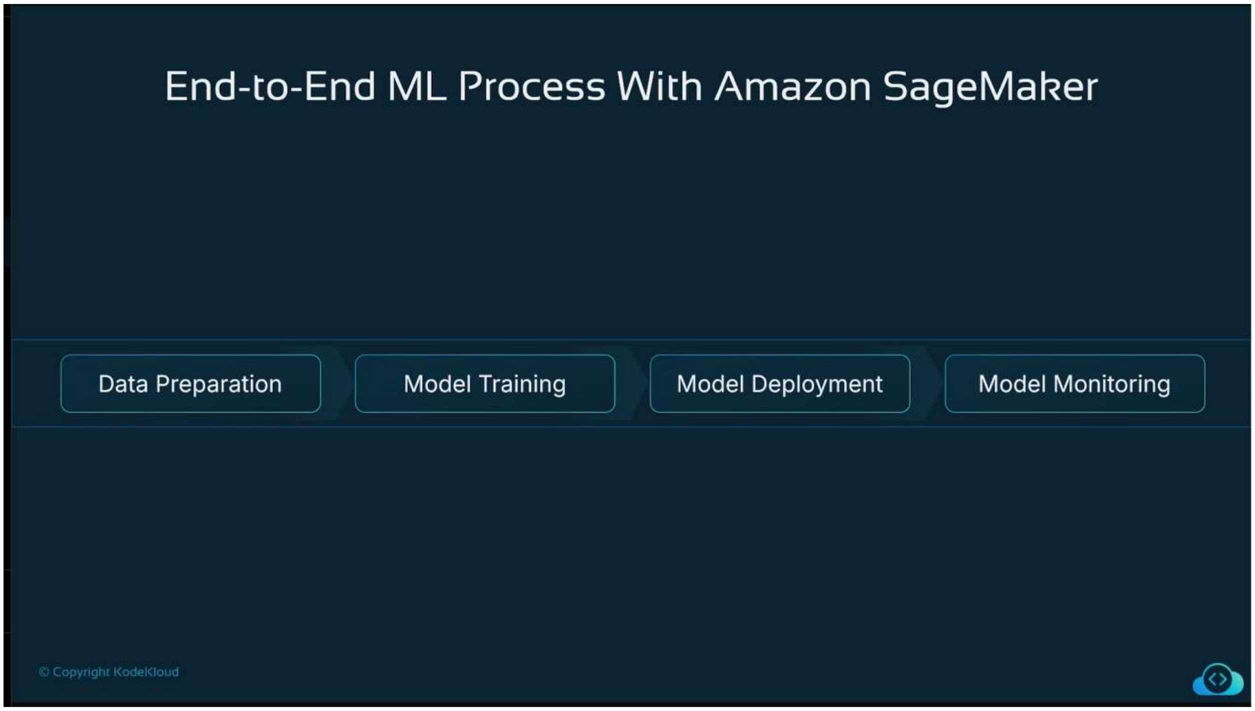
## End-to-End ML Process With Amazon SageMaker

Data Preparation → Model Training → Model Deployment → Model Monitoring

New Data → Trained Model → Track Predictions → Check Drift

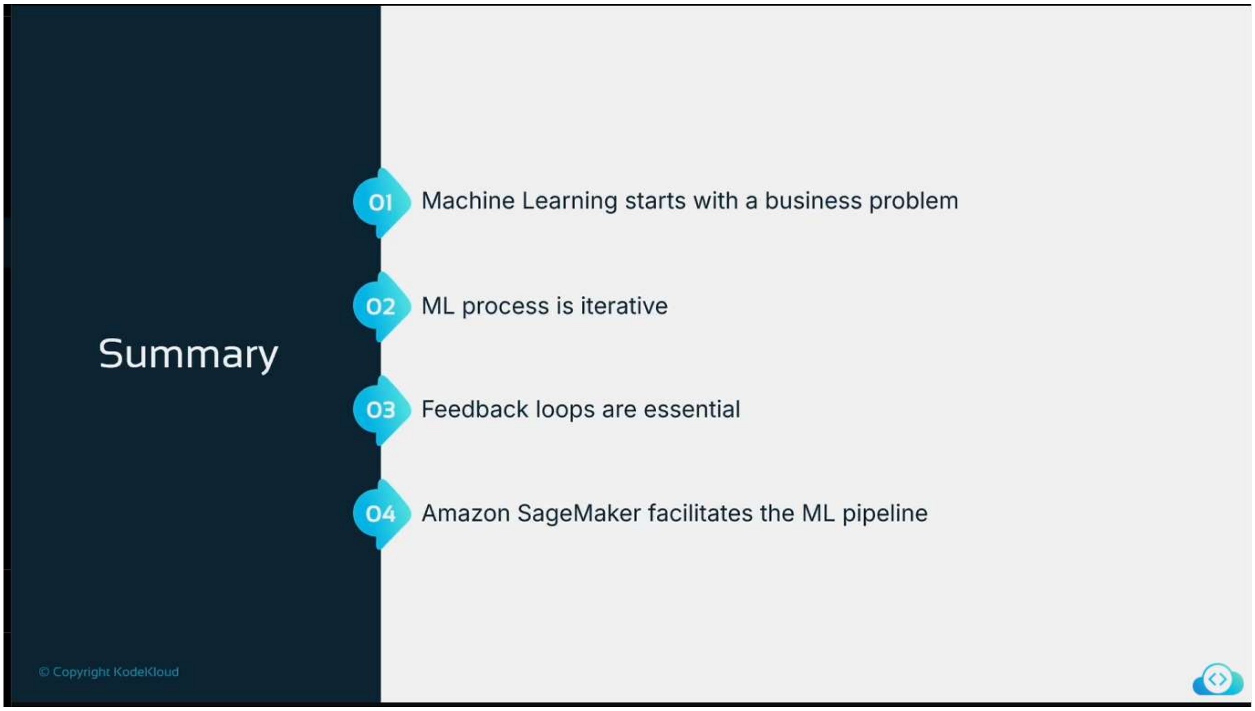
No Drift → Continue

Drift Found → Retrain Model



## End-to-End ML Process With Amazon SageMaker

Data Preparation → Model Training → Model Deployment → Model Monitoring



## Summary

01 Machine Learning starts with a business problem

02 ML process is iterative

03 Feedback loops are essential

04 Amazon SageMaker facilitates the ML pipeline

# Audio file

Your Recording 12.wav

## Transcript

00:00:06 Speaker 2

Let's now look at machine learning pipeline.

00:00:09 Speaker 2

In this lesson, we're going to look at what the process is of moving from ideation all the way through to hosting a model. And we're going to think about this as a pipeline, a sequence of activities, and how that maps onto how we use the different features of the Amazon Sagemaker AI product. So let's think about this from problem perspective. We need to think about, is ML the right approach?

00:00:39 Speaker 2

To solving our business problem.

00:00:43 Speaker 2

So oftentimes we see people decide we need to use ML because maybe ML seems very cool and new, but really what you have is a business problem that may or may not be an ML problem. It might be that traditional programming methods with a number of conditional evaluations might make more sense for a particular use case, but in other use cases.

00:01:08 Speaker 2

Then maybe an ML approach would make sense.

00:01:11 Speaker 2

But we start from the perspective of what problem is it that we are trying to solve? Now imagine we were working in the healthcare sector and maybe we don't have enough professional analysts who are able to look at medical images and determine whether or not there are particular symptoms of particular diseases or conditions. And because of that, you've got delays, delays of patients waiting to find out if they indeed have something wrong with them.

00:01:40 Speaker 2

So in that case, there your problem is that you don't have enough people in order to assess the images. And of course that doesn't necessitate machine learning solution. You could simply hire more medical professionals, but that might not be cost effective. So if we frame our problem in a way to say there's a lack of staff and insufficient funds to hire new medical professionals to assess images.

00:02:07 Speaker 2

Therefore, our problem is that we need to assess those images.

00:02:10 Speaker 2

In a way that doesn't require additional spend or additional people.

00:02:15 Speaker 2

Imagine I'm working for a telecommunications company. Maybe I'm a large mobile phone operator and I have a number of customers who take contracts with me. But then maybe as they get towards the end of the contract, ideally I want them to renew with me and not change to another provider. If I have a high churn rate, then that means my customers are leaving me and going to someone else. Now if I lose that customer, well, what was the cost of keeping that customer versus the cost of acquiring a new one to replace that?

00:02:45 Speaker 2

If we could accurately predict churn to customers who are likely to leave us, we can come up with the incentives to keep them at a price point that still is profitable for ourselves.

00:02:57 Speaker 2

If we're maybe a real estate agent, maybe we want to provide a guideline price for someone who wants to sell their property and we don't have, again, maybe enough people to go out in a timely period, determine the value of that property by physically looking at it. But if we have the characteristics of the property, we can provide an initial estimate that can maybe at least start the process of marketing that property. So again, maybe if I could get some of the details.

00:03:26 Speaker 2

About that property, I'd be able to generate a predicted price and avoid the need to visit the property initially.

00:03:34 Speaker 2

So we are always starting from a business problem. What is the problem I'm trying to solve?

00:03:40 Speaker 2

And it might be the solution is to hire more people, or to source a particular agency that might be the solution, or to come up with a traditional coding approach. Or machine learning might have value to resolving my business problem in a way that is cost-effective to do so.

00:03:59 Speaker 2

Now when we think about the machine learning pipeline, we're talking about the sequence of activities that will take us from having a business problem through to a model that can actually generate predictions for us and that we can monitor and ensure that those predictions are still being generated with a degree of accuracy that we require. So we've take our business problem, we need to determine if it is ML problem and then we start the main steps of the machine learning.

00:04:30 Speaker 2

Pipeline. So over the next few slides we'll walk through this pipeline and we'll look and see what steps are performed by what personas and how does that lead me from the point of having a business problem to actually generating accurate predictions.

00:04:47 Speaker 2

Now when we, if we are trying to frame a business problem as an ML problem, we need to start thinking about well, what ML approach would be appropriate. Now the approach that we're going to take for solving our use case that we use on this training course around predicting house prices. We're clearly using linear regression. We're coming up with a line of best fit through a series of data points in order to be able to predict a price. So the prediction.

00:05:17 Speaker 2

Will be a numeric value, but that's not the only type of ML we have. We have many more types of ML available to us. For example, we might have logistic regression. Now, logistic regression is similar to linear regression in that you are analyzing tabular data, but what we're trying to do is really categorize something as happening or not happening.

00:05:39 Speaker 2

So this might be, for example, is the customer going to churn away from us or not? Or does that image contain a medical condition, it requires further attention or not? So it's more getting into a kind of binary classification. Yes, it's in this category or it's in that category.

00:05:59 Speaker 2

But we can go further. We can go into ML approaches around image data. So for image recognition and identifying specific objects. Or maybe I'm identifying that there's a particular object type recognized within an image. Or maybe I'm coming up with some other decision tree approach based on a number of input signals. There are natural language processing approaches we could take, or forecasting data based on time series.

00:06:26 Speaker 2

But the machine learning approach you take will be really a function of.

00:06:30 Speaker 2

What suited to solving the business problem that you have in the 1st place? We don't start out saying this is an NLP or a logistic regression ML approach. We start out with what's the business problem and then which ML approach would help me solve that business problem.

00:06:49 Speaker 2

So the question when we ask ourselves are the business problem, is it an ML problem? Be prepared for a number of times. The answer to that question being no, this is not an ML problem and we can solve it in another or more traditional way. That doesn't make it bad, just means it maybe it wasn't the best candidate for being resolved by machine learning.

00:07:12 Speaker 2

Now, if we take our example of in healthcare trying to determine maybe if a patient has a particular medical condition, we might want to do a classification. A classification would be where maybe it's not just a patient has a tumor or not tumor. It might be more they have one of six possible conditions and we're either going to say they have no condition or they have these conditions. So we might the output isn't just going to be a straight binary response.

00:07:41 Speaker 2

So in that case, because of the input data being of type Image and the fact that we want to be able to classify that image as belonging to one or more categories, then the ML problem we are faced with is image recognition and classification of object within image. As such, when we start considering, well, given that that's the ML problem we have, we can then start investigating which algorithms and approaches within image classification make the most sense for that data.

00:08:10 Speaker 2

In our telecommunications example where we're trying to predict if the customer will leave us or not, then in that case, really that's a logistical regression because what we're doing there is saying based on all of the input signals that's going to likely generate an output signal that we're saying, yes, we are likely that they will churn. So that likelihood might be expressed as a percentage. And maybe we say, well, if it's more than 70% likely, then we're placing them into.

00:08:40 Speaker 2

Of likely to churn versus not likely to churn. So we can play around with thresholds when we talk about the logistic regression because you're either saying they are going to churn or they are not.

00:08:53 Speaker 2

And in our real estate example where we're taking a number of input features like square footage, number of rooms and so forth, what we want out of that is a price and that numeric value that would be based upon those input signals, that most definitely is a linear regression use case. And again, our input data there would be tabular data type again when you start working with machine learning in any large organization.

00:09:19 Speaker 2

It's very likely that you will be working with tabular data initially.

00:09:24 Speaker 2

Image classification. Image object recognition is very exciting, but unfortunately for a lot of organizations they have vast amounts of tabular data and therefore we end up working with tabular data more than we do with image data.

00:09:39 Speaker 2

So we started with our business problem. You determine whether it was an ML problem. Is it a linear regression? Could a logistic regression? Could an image classification? Could any of these ML approaches solve our business problem for us?

00:09:55 Speaker 2

That point, if we've determined that it does, yes, this is definitely a logistic regression problem or a linear regression problem, then at the next stage, we need to start thinking about sourcing our data. Now when we start thinking about sourcing our data, we need to start thinking about who is going to be performing these steps. And the size of your organization will of course influence that. You may be a single practitioner that is responsible for sourcing data, cleaning data, performing the machine learning model development.

00:10:25 Speaker 2

Training as well as the hosting an inference a small company. Or you might work for a very large enterprise where you have teams of data engineers and data scientists and machine learning engineers as well. But someone one will have to collect the data and prepare it for use and very rarely is the raw data ready for use.

00:10:47 Speaker 2

So let's say we have a data engineer available to us within our organization.

00:10:53 Speaker 2

Engineer a data pipeline, a method of extracting the data from its source. Now maybe the source data for your particular ML problem is residing in a key value store or a relational database. So the data engineer would be responsible for getting the data extracted from that source. And maybe that needs to be done on a regular basis for regular model retraining, or maybe it needs to be done as a one time thing. But generally the data engineer wants to set up.

00:11:23 Speaker 2

Repeatable mechanisms so that we can do this again and again in a consistent fashion for that data extraction and any transformations that are required. And when we say transformations, what might be that the format of the data that they extract from a data source might be let's say in Jason format, maybe if it came out of a key value store. But the format that I want the data to be in prior to model training might be CSV comma separated value. So again, the data engineer.

00:11:53 Speaker 2

Suited to come up with automating those transforms.

00:11:57 Speaker 2

So at that point, the data engineer has got the data of the data source, transformed it in a way that makes it useful to the data scientist. Data scientists could take over from this point. And again, we are making assumptions here that you have both personas in your organization. It might be the data scientists are responsible for the data source and transformation as well. So at that point.

00:12:22 Speaker 2

We're then into the data scientists activity and we can see here in our ML pipeline feature engineering is going to be one of the.

00:12:29 Speaker 2

Main activities in the pipeline that the data scientists will undertake. But it's more than just feature engineering. The data scientist is going to perform exploratory data analysis. You may have heard the term EDA. Now this is about the data scientist getting to know their data set, understanding it. What is it made-up of? What features are available to me? What format are those features in? Are there any categorical data fields that are non numeric?

00:12:59 Speaker 2

What is the obvious correlations between any of these features? If I have two features that are highly correlated, do I need to keep them both? Or could I drop one of those features and just keep one and still not lose any of the value in that data? And I may liaise at this point as a data scientist with a domain expert, somebody who understands from the business precisely what that data means, and I might use that to help me better understand what those data points mean.

00:13:29 Speaker 2

Rather than performing that in isolation, not understanding the data domain.

00:13:34 Speaker 2

So correlations between features is a key requirement to understand. We'll drop irrelevant features, things that are just not going to help the model determine a pattern of influence on the target value. This might also involve dropping features that have personal identifiable information, for example.

00:13:59 Speaker 2

Is something like a personal account number for a bank account in a financial organization.

00:14:05 Speaker 2

That number doesn't hold any intrinsic value and doesn't relate to whether or not a transaction is fraudulent. So we would drop irrelevant features. And again, this comes back to the data scientists understanding the data domains so that we can know what is an irrelevant feature. Now the data scientist might generate new features from existing data. Maybe the existing data has for a field. That would be something like when was this account opened, account creation date.

00:14:35 Speaker 2

Maybe what we think will be more important to the model will be how old is this account? So I might want to derive a new feature like number of days this account has existed, in which case you might derive a new feature, a new, essentially a new input field to the model, which would be the number of days since the account was created. So synthesizing new features is very much a kind of one of the many tools available to the data scientist that they can engineer.

00:15:05 Speaker 2

The data to be in the best shape for the model to performance training upon. Now the data scientist might also need to scale data. Now we're going to talk more scaling in an upcoming lesson, but for now, we just need to be aware that steak house prices is a good example. Number of bedrooms is probably going in a house price data set is probably going to vary between the number one and maybe the number six.

00:15:34 Speaker 2

But the cost of a house might range from 100,000 to 100 million very large numbers. And depending on the algorithm that we use to train our model, it might place more importance on very large magnitude numbers compared to the low magnitude numbers that you have for a number of bedrooms. So it might be that we want to scale the data so the model doesn't hope for emphasize on the wrong thing. We'll talk more about that later, but.

00:16:04 Speaker 2

Just be aware for now that this is going to be part of what the data scientists will decide to do with that data.

00:16:11 Speaker 2

Now once the data has been engineered, we can start the data scientist will start the model training process. Now we know from our last lesson that during the model training process, the algorithm and A methods, typically gradient descent is being used to tune the parameters of the model. If you remember that F of X equals WX plus B, those weights that are associated with each of the input features and how much weighting we should apply.

00:16:41 Speaker 2

To them in determining its effect on the target variable. So the model is constantly tuning those values during the model training process.

00:16:53 Speaker 2

Now the data scientist can influence that training process. They'll influence it through the features that they chose to engineer and present to the model. They influence it through their selection of what algorithm they use.

00:17:07 Speaker 2

And we saw in the last lesson that the algorithm that we use would really be a function of what problem you're faced with. Are you faced with a linear regression problem? Well, the linear learner algorithm would probably be best at that. Are you faced with a logistic regression problem? Well, again, linear learner would be good at that, but maybe Xgboost would also be good at. So different algorithms are better suited for different jobs.

00:17:33 Speaker 2

Data scientists might choose to experiment with one or more algorithms.

00:17:37 Speaker 2

And look and compare the results that they get from their model. Now when we think about model tuning, we can affect how the model training process occurs, typically through the adjustment of what we call hyperparameters. And this would be things that will influence how that training occurs. We saw before that in gradient descent, that's where we're adjusting those weights that are associated with each input feature to determine their influence on.

00:18:07 Speaker 2

Market value, but how does it adjust those? Does it increase the number? Does it decrease the number on each iteration? Well, we can influence that by adjusting what we call hyperparameters. And that could say, well, how big a step should we take in each direction as we change those weights? How long should we train for? How many iterations? What point does the level of accuracy that we produce by the model, Is it good enough?

00:18:36 Speaker 2

How many iterations of the data would be the number of what we call?

00:18:39 Speaker 2

Epochs. So this is all in the control of the data scientists, and they are likely to perform a number of experiments with different algorithms and different hyperparameter combinations and different feature engineers.

00:18:53 Speaker 2

In order to determine OK, that model candidate was better than that one. So it is a very scientific approach to essentially create a hypothesis, test their hypothesis, note the result and move on.

00:19:06 Speaker 2

They're a critical part of this is evaluating the model performance. Now, when you have a large training set that you are using to train your model in, what we would typically do is hold back a percentage of that data. So imagine your training data set had 1,000,000 rows of data. Well, maybe I might train my model on 700,000 rows of randomly selected data, and then maybe 300,000 rows. I'm going to hold back and I'm not going to train the.

00:19:36 Speaker 2

With that instead, I'm going to hold that back.

00:19:40 Speaker 2

And use it for evaluating my model.

00:19:42 Speaker 2

Because once I've built a model, I can then pass in the input the input features from those 300,000 samples into my model, look what the model predicted, and then compare that to what the actual target value was. In other words, you have the ground truth value in that held back data set, so you can actually compare how well your model performs against known data. And because you use, you held back some data.

00:20:13 Speaker 2

From the training process, you know that your model has never seen that data before, so it is a good way of determining how well your model is performing. Once you do that, you can then determine, well, is this producing good predictions? Is this telling us that yes, the customer is going to leave us and Oh yeah, the customer would have left us in that circumstance the model was doing well. Or is the classification of that image correctly showing us?

00:20:42 Speaker 2

That particular patient has a problem on their left lung or is it telling us that the house price predicted is within 5% deviation of the actual house price that the house sold for? So we need to determine that point with the business stakeholder. Have we met the level of accuracy that is required that would resolve that business problem? And if we haven't, then we don't just give up, we then iterate back through the process.

00:21:12 Speaker 2

When this might be.

00:21:13 Speaker 2

That we didn't collect sufficient data in the 1st place that maybe represented enough of the different classes of data. Or maybe it means that we need to revisit feature engineering and that we need to maintain that data, enrich it somehow. Maybe we drop too many features. We need to iterate again through this. So we're not going to move forward with hosting our model in production until we know that that model.

00:21:39 Speaker 2

Can actually produce value, produce accurate predictions?

00:21:44 Speaker 2

Now, assuming it is, we can then move to testing our model. I at this point, we would want to deploy our model onto a target infrastructure and then again test our model. Now this again comes back to how you might treat your data set. I gave an example of maybe a million rows of data that you've got in your tabular data set, and you may be held back 30% of that for evaluation. Well, actually what we often see is people hold back even more, so they might maybe.

00:22:14 Speaker 2

Their model on 70% of their data and then maybe hold back 20% for evaluating and then hold back a last 10% for final testing against a production environment.

00:22:27 Speaker 2

But our job doesn't end there. Once our model is generating predictions, we need to monitor that model to ensure that it keeps on producing accurate predictions. Now we'll talk more about how to do that in a later lesson, but we need to have monitoring in place to make sure that our model is still accurate. And as it starts to drift into inaccurate predictions, we have a feedback loop and that.

00:22:57 Speaker 2

Might typically take us back into needing to retrain our model, maybe on a newer data set.

00:23:04 Speaker 2

So you've seen we identify our business problem.

00:23:08 Speaker 2

To frame it as an AML problem. And again, the answer at this point might be no, it's not an ML problem, but can we frame it as an ML problem? And if we can, what's the best ML approach that fits that particular problem?

00:23:21 Speaker 2

We need to define our success criteria. For example, if we are able to predict what our sales figures are going to be within a 5% accuracy for next quarter or if we can identify the medical images, we can identify conditions within half hour period of our patient receiving a scan. Whatever the criteria is of what good looks like, we define that.

00:23:48 Speaker 2

To determine if the model should go into production.

00:23:50 Speaker 2

We then eat to source and select our data, and we've seen that that might be the job of the data engineer or the data scientist, or a combination of them both.

00:24:00 Speaker 2

So when we think about what's next, we know from data preparation.

00:24:06 Speaker 2

That we've determined if the columns are relevant.

00:24:10 Speaker 2

Who determined is the correlation between data in those columns? Now, when we think about that, remember that if you have, let's say, 2 input fields and they are very correlated with one another, so let's say when one increases, the other increases absolutely in line with them. Do we need to keep both columns? Is there a relationship that's independent of each of those features with another feature? Or can we easily drop one column and simplify our data set?

00:24:40 Speaker 2

If we do that, then if you're dealing particularly with data that maybe have 100 features, reducing the number of features might be beneficial to you. Because if you have too many features, it might be hard for the model to be able to generalize from your data set, or it might train for an extremely long period of time and not have to level of accuracy that you require. There is something called the curse of dimensionality, and this is the idea that we need to have enough dimensions of our data.

00:25:11 Speaker 2

To show the complex relationships between them and their impact on our target feature.

00:25:18 Speaker 2

But at the same time, not have too many features. That means the model can't actually identify patterns. There's just too much noise. So if you can identify features that highly correlated with one another and no other column, then it might be that that's a good candidate to drop one of them and still not lose any relationship data. Now on data preparation, we also have the challenge of missing data. You might be missing data for particular features.

00:25:47 Speaker 2

And it's the data scientist's job to think about how we handle that.

00:25:52 Speaker 2

Do we drop that data or do we maybe invent some new data? We call that imputation and we're going to talk about that more in an upcoming lesson, but we have to have an approach for dealing with that situation. We know that if we have categorical data like color of car, blue, red, gold, then our algorithm is not going to be able to deal with categorical data.

00:26:19 Speaker 2

So what we want to do is transform that into a numerical value.

00:26:23 Speaker 2

That it can deal with and do we have any data that has vastly different scales like the cost of a house versus the number of bedrooms? Is my the algorithm that I'm using, is it very sensitive to that? And is it likely to place more importance on those higher scaler values And if they are, then it might be on us to do some scaling of our those numeric features.

00:26:48 Speaker 2

Before we start training.

00:26:52 Speaker 2

In the model training process, we're likely to perform as a data scientist model training from within what we call a Jupyter notebook. Now we're going to go into what Jupyter notebooks are in an upcoming lesson, but for now, we need to be aware that a Jupyter notebook is an interactive Python environment that allows us to experiment and see the output of our experiments as sort of in line as a working document.

00:27:19 Speaker 2

Now I could absolutely run a Jupyter notebook on my laptop and train a model.

00:27:26 Speaker 2

But my laptop is going to get very hot and it's going to take many hours to do that, particularly with a large data set.

00:27:33 Speaker 2

But this is where we start to see where Amazon's Sagemaker AI is going to help us out.

00:27:38 Speaker 2

Because what I'm going to be able to do with Amazon Sagemaker is create what we call a training job.

00:27:46 Speaker 2

So Amazon Sagemaker is going to give us access to an unlimited compute platform. So if I think the training of my job is going to require a 96 CPU system with a TB of RAM and four NVIDIA graphics cards.

00:28:05 Speaker 2

Then I could create a training job with those properties, and the job will run in a compute environment that's dynamic and spun up by Sagemaker to do that job, and then I simply get the model artifact produced by that job and given back to me.

00:28:22 Speaker 2

So we kind of can decouple the training process away from the Interactive Data exploration type jobs that I'm doing in Jupiter. I'm not needing to use the compute resources on my laptop to do it. I'm using the compute resources that are dynamically available from the Amazon Cloud platform.

00:28:43 Speaker 2

So think about the resources you are going to need to perform training. Is that just memory, CPU and GPU?

00:28:52 Speaker 2

We have got unlimited amount of compute resource on tap in Sagemaker, so we know do model training will need to select an algorithm, KNN, Xgboost, LGBM, Linear Learner and numerous algorithms out there that are actually included in the Amazon Sagemaker product that will allow you to simply consume them and start to train your model.

00:29:20 Speaker 2

I need to engineer my features, and that might mean simply dropping certain features because you're it's either not relevant or you can drop it because it's highly correlated with another feature. Or maybe we can generate new features based on the existing data, maybe adding two features together or something like that.

00:29:40 Speaker 2

Then we would tune the hyperparameters for our training job and that could be number of iterations that we go over the data. What's our stopping criteria?

00:29:50 Speaker 2

Are we, how big a step should we take each time we determine that our weights and bias is wrong and we need to adjust that on the next iteration? Our biggest step should be taken each time, so our learning rate. So again, we've got to adjust that before we run another training job. Once we've run a training job and produce the model, we would want to test our accuracy against that model. How can I test accuracy? Well, I do that by holding back some of my data from my training set.

00:30:20 Speaker 2

So rather than using all of my training data for training the model, I would maybe only use 70% of my training data set for actually training the model and hold back some evaluation and tests because it's all about presenting that model with data it's never seen before. So you know how well the model behaves when presented with that unseen data, but yet data that you know what the target value was, you know what the ground truth value was. So that you.

00:30:50 Speaker 2

Can compare and contrast what your model predicted versus what the current truth value was. Thus determine accuracy. So Jupiter is going to be at the heart of what the data scientist is going to be performing.

00:31:06 Speaker 2

Think of it like if I am software developer, I might use Visual Studio Code to write my project, or if I'm a Java developer I might use Eclipse to perform this. Or if I'm maybe a JavaScript developer I might use Intelli J. But as a data scientist, I'm probably going to use an environment that supports running Python scripts but helps me in terms of providing a more interactive environment rather than just a standard developer IDE.

00:31:36 Speaker 2

And that's where Jupiter notebooks fit, and we're going to see that the data scientist is going to be used to working in Jupiter to perform these steps.

00:31:46 Speaker 2

Now, when I perform a deployment, I want to deploy my model artifact out onto a compute platform, physical server, a virtual machine, a container. This is where Amazon Sagemaker helps us out, because Amazon Sagemaker allows us to host the model on Siegemaker itself.

00:32:08 Speaker 2

So Sagemaker can spin up.

00:32:11 Speaker 2

A target virtual machine, and it actually spins up a target container as well and places your model inside with some surrounding code so it can accept inference requests from you and generate inference responses from you.

00:32:26 Speaker 2

So you don't need another virtual machine or physical server in order to host your model. You can use Sagemaker as the hosted platform itself.

00:32:38 Speaker 2

We call that a Sagemaker endpoint and it is literally just a managed virtual machine and a managed container on the virtual machine with your model artifacture model TGZ file deployed within that container with thumb surrounding inference code as a handler for those inference requests.

00:33:00 Speaker 2

And lastly, monitoring. I said that we are going to have to monitor our model to ensure that it continually produces accurate requests. So what we can do is as new data is presented to our model, we can track the predictions that were generated by that and compare that to what we call ground truth values. Thus determine is the model still producing accurate predictions.

00:33:29 Speaker 2

And we call that drift.

00:33:32 Speaker 2

If there's no drift, everything's good. If there is some drift found and where maybe the model is no longer producing accurate predictions, it's probably because the data that's been presented in the inference requests maybe doesn't look quite like the data we trained it on. And that's just going to happen over time. Data becomes historical and user patterns change over time. So we're probably going to need to retrain our model with newer data to.

00:34:02 Speaker 2

Our model back to producing more accurate predictions, but that can become an automated feedback loop.

00:34:11 Speaker 2

So we now have an ML process where Amazon Sagemaker can be used every step along the way. Data preparation, we can use Jupyter Notebooks. Model training. We can use Jupyter Notebooks with Sagemaker training jobs. So we can offload the actual training process, which might need quite a lot of compute resource to actually running the model. So where are we going to host it? We need to deploy it somewhere.

00:34:39 Speaker 2

In the Sagemaker endpoint.

00:34:40 Speaker 2

And ultimately, you need to monitor that model. You need to check that it's predictions it's producing and compare that to ground truth values. So what have we seen in this module?

00:34:52 Speaker 2

We've seen that machine learning starts always with a business problem. What is it we're trying to solve for the business? And we then need to determine is that a machine learning problem or not?

00:35:06 Speaker 2

We've seen the machine learning pipeline is not something that we just do the sequence of activities and then stop. It is iterative. We start with extracting data. We start with preparing that data so it can be used. We then engineer that data. We run through experiments. We determine through evaluation if that is producing the level of accuracy and solving the business problem. Is it not? And we feedback, we iterate through until it does or we abandon it.

00:35:36 Speaker 2

Because ML is maybe not providing the level of accuracy that we require that solves a business problem.

00:35:43 Speaker 2

Even if we get a model produced and we host, it produces valuable predictions. We still need to monitor it. We still need to see does it continuously give us good predictions or is it going to start to drift over time, in which case we have that feedback loop that might help us determine when is a good time to retrain the model on newer data.

00:36:07 Speaker 2

So these feedback loops are essential, and if we don't have this feedback loop in it, then your model can only ever be good for a point in time. You need to have the feedback loop to determine if the model can even move to production. And then even once it's in production, is it still relevant?

00:36:25 Speaker 2

We've seen that Amazon Sagemaker is going to be used every stage of the machine learning pipeline during the preparation stages of data, during training, hosting, and ultimately monitoring as well. But you can see now that Sagemaker only helps us when we've got a model to develop and host and monitor. That's why Sagemaker seems so mysterious that unless you have a use case for.

00:36:54 Speaker 2

Developing a model and hosting it, then Sagemaker doesn't appear to do very much. So that wraps up our short lesson on the machine learning pipeline. In our next lesson, we're going to look at how much math do you really need to know to get started with Sagemaker? And it's a lot less than you think. See you there.