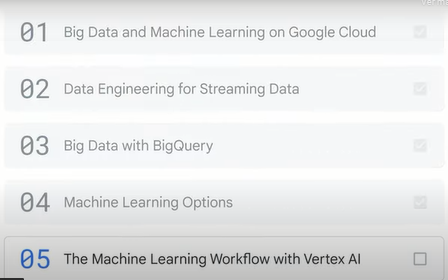
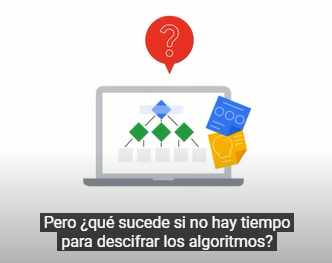
**The Machine Learning Workflow with Vertex AI.**

**Introduction.**

in the previous section of this course you explored the machine learning options available on google cloud now let's switch our focus to the machine learning workflow with vertex ai from data preparation to model training and finally model deployment

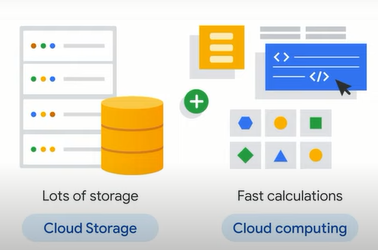


Vertex ai google's ai platform provides developers and data scientists one unified environment to build custom ml models this process is actually not too different from serving food in a restaurant starting with preparing raw ingredients through to serving dishes to a table later in this section you'll get hands-on practice building a machine learning model end-to-end using automl on vertex ai but before we get into the details let's look at the basic differences between machine learning and traditional programming in traditional programming simply put 1 plus 1 equals 2. data plus rules otherwise known as algorithms lead to answers and with traditional programming a computer can only follow the algorithms that a human has set up but what if we're just too lazy to figure out all the algorithms

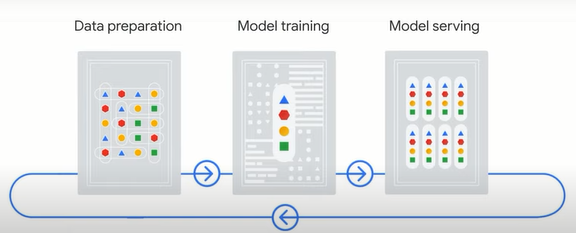


or what if the algorithms are too complex to figure out? **this is where machine learning comes in**

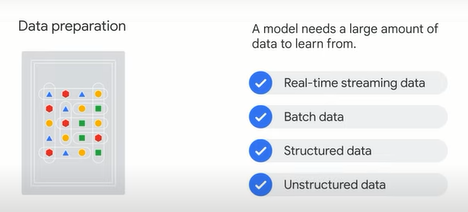
with machine learning you feed a machine a large amount of data along with answers that you would expect a model to conclude from the data then you select a machine learning model from there you expect the machine to learn from the provided data and examples to solve the puzzle on its own so instead of telling a machine how to do addition you give it pairs of numbers and the answers for example one one and two two three and five you then ask it to figure out how to do addition on its own but how is it possible that a machine can actually learn to solve puzzles for machine learning to be successful you'll need lots of storage like what's available with cloud storage and the ability to make fast calculations like with cloud computing



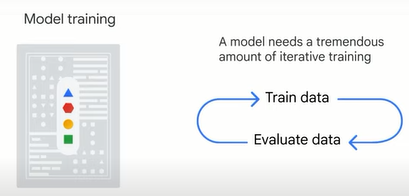
there are many practical examples of this capability for example by feeding google photos lots of pictures with tags you can teach the software to associate and then automatically attach tags to new pictures tags can then be used for search functionality or even to automatically create photo albums can you come up with any other examples to apply machine learning capabilities take a moment to think about it there are three key stages to this learning process



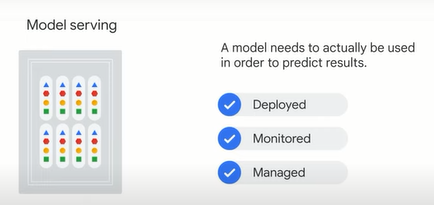
the first is **data preparation** a model needs a large amount of data to learn from data used in machine learning can either be real-time streaming or batch data and it can either be structured which is numbers and text normally saved in tables or unstructured which is data that can't be put into tables like images and videos.



the second stage is **model training** a model needs a tremendous amount of iterative training this is when training and evaluation form a cycle to train data then evaluate the data and then train the data some more



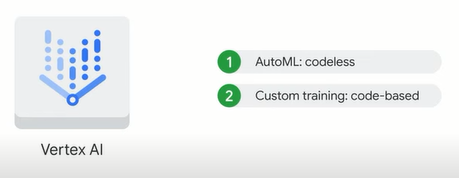
the third and final stage is **model serving** a model needs to actually be used in order to predict results this is when the machine learning model is deployed monitored and managed if you don't move an ml model into production then it has no use it remains only a theoretical model



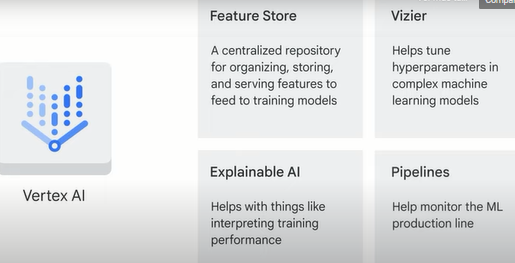
now it was mentioned earlier that the machine learning workflow on vertex ai is not too different from serving food in a restaurant so if you compare these steps to running a restaurant data preparation is when you prepare the raw ingredients model training is when you experiment with different recipes and model serving is when you finalize the menu to serve the meal to lots of hungry customers

it's important to note that an ml workflow isn't linear it's iterative for example during model training you may need to return to dig into the raw data and generate more useful features to feed the model when monitoring the model during model serving you might find data drifting or the accuracy of your prediction might suddenly drop you might need to check the data sources and adjust the model parameters fortunately these steps can be automated with machine learning operations or mlops we'll go into more detail on this soon.

so how does vertex ai support this workflow? you'll recall that vertex ai provides two options to build machine learning models automl which is a codeless solution and custom training which is a code based solution

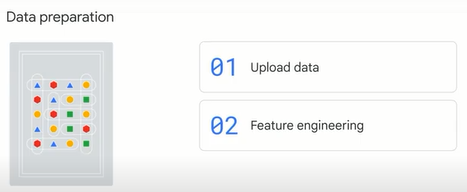


vertex ai provides many features to support the ml workflow all of which are accessible through either automl or vertex ai workbench examples include feature store which provides a centralized repository for organizing storing and serving features to feature training models vizier which helps you tune hyper parameters in complex machine learning models explainable ai which helps with things like interpreting training performance and pipelines which help you monitor the ml production line.

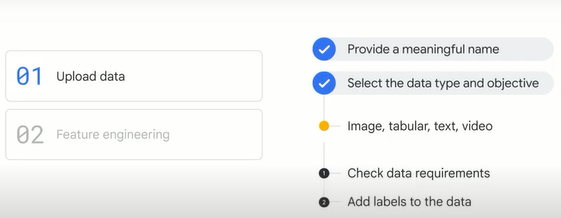


**Data preparation.**

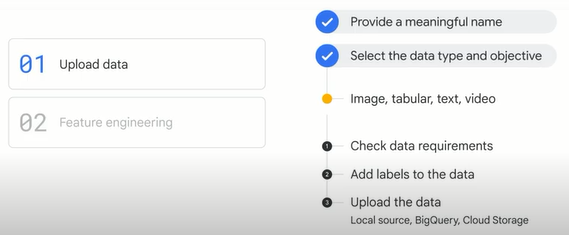
now let's look closer at an automl workflow the first stage of the automl workflow is data preparation during this stage you must upload data and then prepare the data for model training with feature engineering



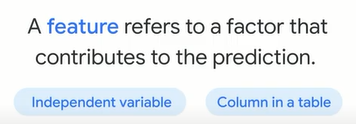
when you upload a data set in the vertex ai user interface you'll need to provide a meaningful name for the data and then select the data type and objective automl allows four types of data image tabular text and video to select the correct data type and objective you should start by checking data requirements we've included a link to these requirements in the resources section of this course next you'll need to add labels to the data if you haven't already.



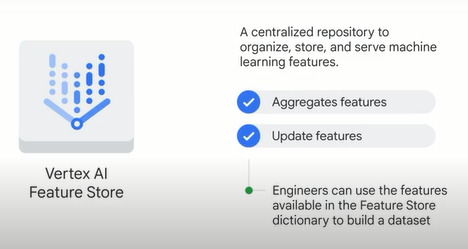
a label is a training target so if you want a model to distinguish a cat from a dog you must first provide sample images that are tagged or labeled either cat or dog a label can be manually added or it can be added by using google's paid label service via the vertex console these human labelers will manually generate accurate labels for you the final step is to upload the data data can be uploaded from a local source bigquery or cloud storage you'll practice these steps in the lab.



after your data is uploaded to automl the next step is preparing the data for model training with feature engineering imagine you're in a kitchen preparing a meal your data is like your ingredients such as carrots onions and tomatoes before you start cooking you need to peel the carrots chop the onions and rinse the tomatoes this is what feature engineering is like the data must be processed before the model starts training a feature which we explored in the bigquery section of the course refers to a factor that contributes to the prediction it's an independent variable in statistics or a column in a table

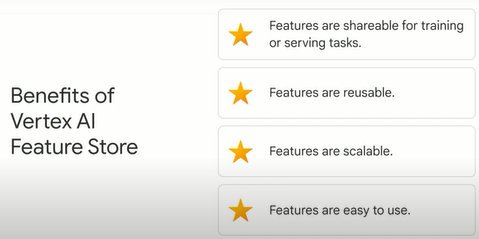


preparing features can be both challenging and tedious to help vertex ai has a function called feature store feature store is a centralized repository to organize store and serve machine learning features it aggregates all the different features from different sources and updates them to make them available from a central repository then when engineers need to model something they can use the features available in the feature store dictionary to build a data set vertex ai automates the feature aggregation to scale the process



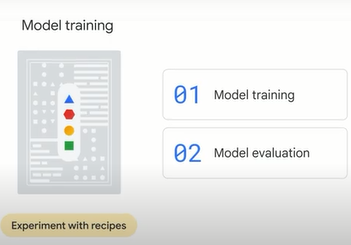
so what are the benefits of vertex ai?

feature store first features are shareable for training or serving tasks features are managed and served from a central repository which helps maintain consistency across your organization second features are reusable this helps save time and reduces duplicative efforts especially for high value features third features are scalable features automatically scale to provide low latency serving so you can focus on developing the logic to create the features without worrying about deployment and fourth features are easy to use feature store is built on an easy to navigate user interface.

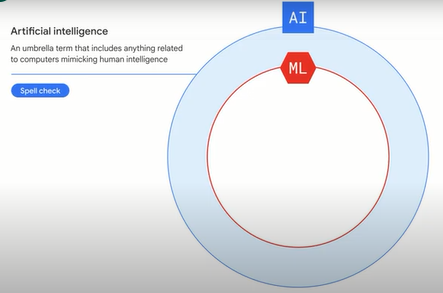


**Model training.**

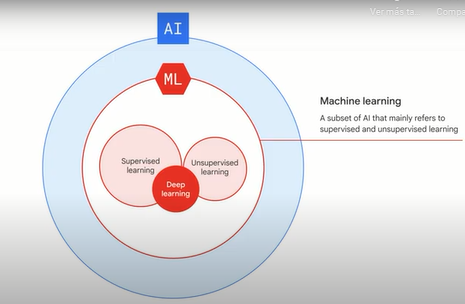
now that our data is ready which if we return to the cooking analogy is our ingredients it's time to train the model this is like experimenting with some recipes this stage involves two steps model training which would be like cooking the recipe and model evaluation which is when we taste how good the meal is this process might be iterative



before we get into more details about this stage let's pause to clarify two terms artificial intelligence and machine learning artificial intelligence or ai is an umbrella term that includes anything related to computers mimicking human intelligence for example in an online word processor robots performing human actions all the way down to spell check

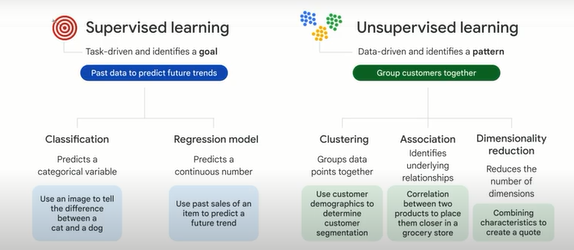


machine learning is a subset of ai that mainly refers to supervised and unsupervised learning you might also hear the term deep learning or deep neural networks it's a subset of machine learning that adds layers in between input data and output results to make a machine learn at more depth



so what's the difference between supervised and unsupervised learning?

supervised learning is task driven and identifies a goal unsupervised learning however is data driven and identifies a pattern an easy way to distinguish between the two is that supervised learning provides each data point with a label or an answer while unsupervised does not for example if we were given sales data from an online retailer we could use supervised learning to predict the sales trend for the next couple of months and use unsupervised learning to group customers together based on common characteristics there are two major types of supervised learning the first is classification which predicts a categorical variable like using an image to tell the difference between a cat and a dog the second type is a regression model which predicts a continuous number like using past sales of an item to predict a future trend and then there are three major types of unsupervised learning the first is clustering which groups together data points with similar characteristics and assigns them to clusters like using customer demographics to determine customer segmentation the second is association which identifies underlying relationships like a correlation between two products to place them closer together in a grocery store for a promotion and the third is dimensionality reduction which reduces a number of dimensions or features in a data set to improve the efficiency of a model for example combining customer characteristics like age driving violation history or car type to create an insurance quote if too many dimensions are included it can consume too many compute resources which might make the model inefficient



although google cloud provides four machine learning options with automl and pre-built apis you don't need to specify a machine learning model

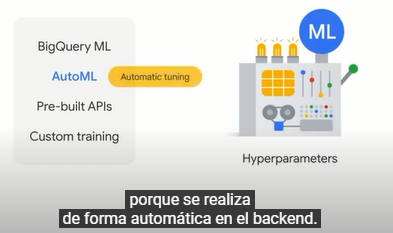


instead you'll define your objective such as text translation or image detection then on the back end google will select the best model to meet your business goal.

with the two other options bigquery ml and custom training you'll need to specify which model you want to train your data on and assign something called hyper parameters



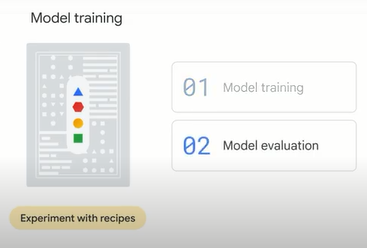
you can think of hyper parameters as user-defined knobs in a machine that helps guide the machine learning process for example one parameter is a learning rate which is how fast you want the machine to learn with automl you don't need to worry about adjusting these hyperparameter knobs because the tuning happens automatically on the back end



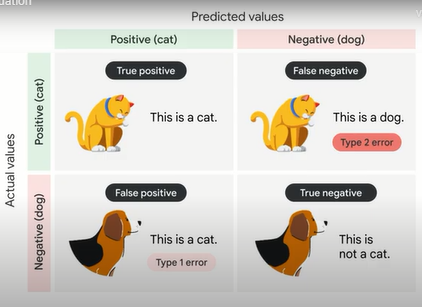
this is largely done by a neural architect search which finds the best fit model by comparing the performance against thousands of other models.

**Model evaluation.**

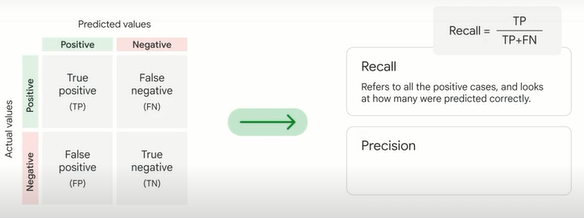
while we're experimenting with a recipe we need to keep tasting it constantly to make sure it meets our expectations this is the model evaluation portion of the model training stage



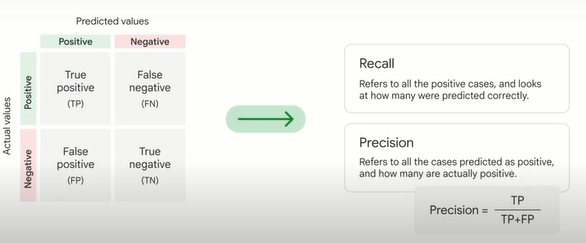
vertex ai provides extensive evaluation metrics to help determine a model's performance among the metrics are two sets of measurements the first is based on the confusion matrix for example recall and precision the second is based on feature importance which we'll explore later in this section of the course a confusion matrix is a specific performance measurement for machine learning classification problems it's a table with combinations of predicted and actual values to keep things simple we can assume the output includes only two classes let's explore an example of a confusion matrix the first is a true positive combination which can be interpreted as the model predicted positive and that's true the model predicted that this is an image of a cat and it actually is the opposite of that is a true negative combination which can be interpreted as the model predicted negative and that's true the model predicted that a dog is not a cat and it actually isn't and then there is a false positive combination otherwise known as a type 1 error which can be interpreted as the model predicted positive and that's false



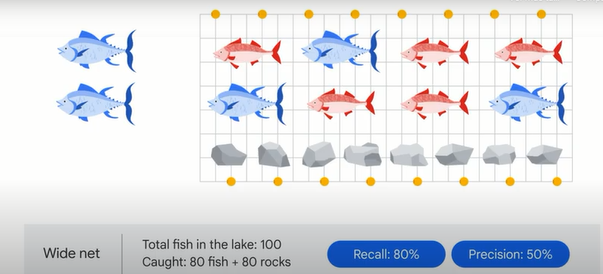
the model predicted that a dog is a cat but it actually isn't finally there is the false negative combination otherwise known as a type 2 error which can be interpreted as the model predicted negative and that's false the model predicted that a cat is not a cat but it actually is a confusion matrix is the foundation for many other metrics used to evaluate the performance of a machine learning model let's take a look at the two popular metrics recall and precision that you'll encounter in the lab recall refers to all the positive cases and looks at how many were predicted correctly this means that recall is equal to the true positives divided by the sum of the true positives and false negatives.



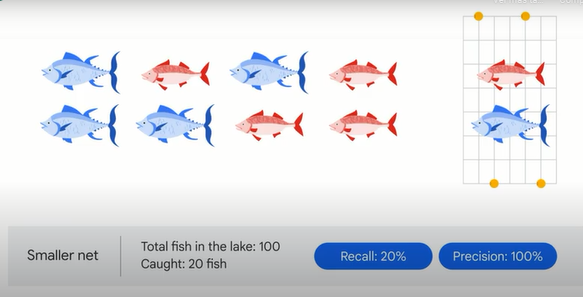
precision refers to all the cases predicted as positive and how many are actually positive this means that precision is equal to the true positives divided by the sum of the true positives and false positives



imagine you're fishing with a net using a wide net you caught both fish and rocks 80 fish out of 100 total fish in the lake plus 80 rocks the recall in this case is eighty percent which is calculated by the number of fish caught eighty divided by the total number of fish in the lake one hundred the precision is fifty percent which is calculated by taking the number of fish caught 80 and dividing it by the number of fish and rocks collected 160.



let's say you wanted to improve the precision so you switch to a smaller net this time you caught 20 fish in zero rocks the recall becomes 20 percent 20 out of 100 fish collected and the precision becomes 100 percent 20 out of 20 total fish and rocks collected.



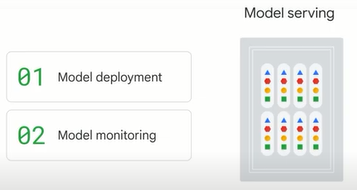
precision and recall are often a trade-off depending on your use case you may need to optimize for one or the other consider a classification model where gmail separates emails into two categories spam and not spam if the goal is to catch as many potential spam emails as possible gmail may want to prioritize recall in contrast if the goal is to only catch the messages that were definitely spam without blocking other emails gmail may want to prioritize precision in vertex ai the platform visualizes the precision in the recall curve so they can be adjusted based on the problem that needs solving you'll get the opportunity to practice adjusting precision and recall in the automl lab in addition to the confusion matrix and the metrics generated to measure model effectiveness such as recall and precision the other useful measurement is feature importance in virtuxii feature importance is displayed through a bar chart to illustrate how each feature contributes to a prediction the longer the bar or the larger the numerical value associated with a feature the more important it is this information helps decide which features are included in a machine learning model to predict the goal you'll observE the feature importance chart in the lab as well.



feature importance is just one example of vertex ai's comprehensive machine learning functionality called explainable ai explainable ai is a set of tools and frameworks to help understand and interpret predictions made by machine learning models.

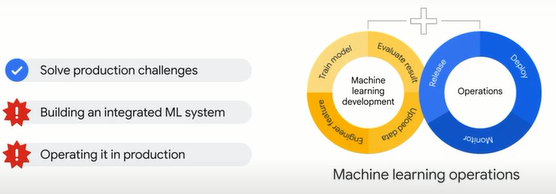
**Model deployment and monitoring.**

the recipes are ready and now it's time to serve the meal this represents the final stage of the machine learning workflow model serving model serving consists of two steps first model deployment which we can compare to serving the meal to a hungry customer and second model monitoring which we can compare to checking with the wait staff to ensure that the restaurant is operating efficiently.

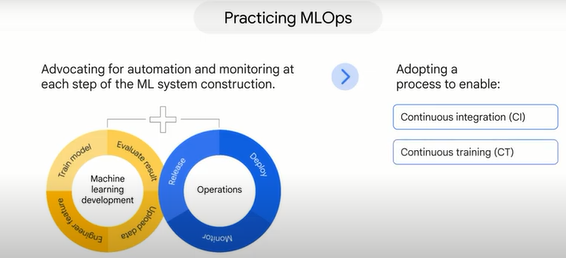


it's important to note that model management exists throughout this whole workflow to manage the underlying machine learning infrastructure this lets data scientists focus on what to do rather than how to do it machine learning operations or mlops play a big role mlops combines machine learning development with operations and applies similar principles from devops to machine learning models which is short for development and operations.

mlops aims to solve production challenges related to machine learning in this case this refers to building an integrated machine learning system and operating it in production these are considered to be some of the biggest pain points by the ml practitioners community because both data and code are constantly evolving in machine learning.

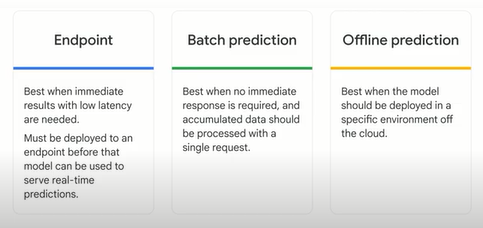


practicing mlops means advocating for automation and monitoring at each step of the ml system construction this means adopting a process to enable continuous integration continuous training and continuous delivery.

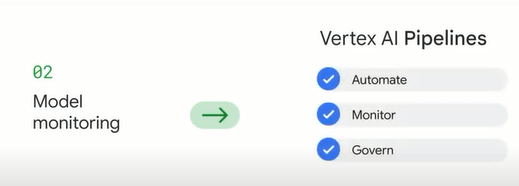


so what does mlaps have to do with model serving?

well let's start with model deployment which is the exciting time when our model is implemented in our restaurant analogy it's when the food is put on the table for the customer to eat mlops provides a set of best practices on the back end to automate this process there are three options to deploy a machine learning model **the first is to deploy to an endpoint** this option is best when immediate results with low latency are needed such as making instant recommendations based on a user's browsing habits whenever they're online a model must be deployed to an endpoint before that model can be used to serve real-time predictions **the second option** is to deploy using batch prediction this option is best when no immediate response is required and accumulated data should be processed with a single request for example sending out new ads every other week based on the user's recent purchasing behavior and what's currently popular on the market **and the final option is to deploy using offline prediction** this option is best when the model should be deployed in a specific environment off the cloud.



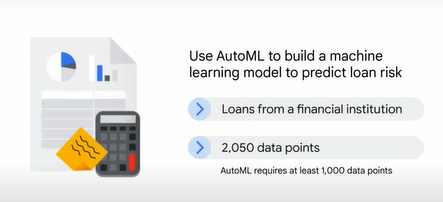
in the lab you'll practice predicting with an endpoint now let's shift our focus to model monitoring the backbone of ml ops on vertex ai is a tool called vertex ai pipelines it automates monitors and governs machine learning systems by orchestrating the workflow in a serverless manner.



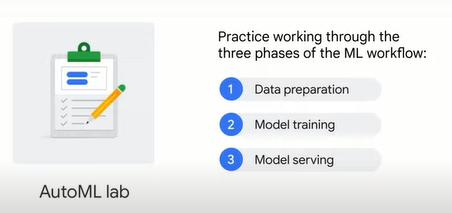
imagine you're in a production control room and vertex ai pipelines is displaying the production data on screen if something goes wrong it automatically triggers warnings based on a predefined threshold with vertex ai workbench you can define your own pipeline you can do this with pre-built pipeline components which means that you primarily need to specify how the pipeline is put together using components as building blocks and it's with these final two steps model deployment and model monitoring that we complete our exploration of the machine learning workflow the restaurant is open and operating smoothly bon appetite.

**Lab introduction: Predicting loan risk with AutoML.**

let's put what you've learned into practice with a hands-on lab in this lab you'll use automl a codeless tool to build a machine learning model to predict loan risk the data set used in the lab relates to loans from a financial institution and has 2050 data points automl requires at least 1 000 data points in a data set.



the goal is to practice working through the three phases of the machine learning workflow data preparation model training and model serving let's get started.



**Lab: Vertex AI: Predicting Loan Risk with AutoML.**

## Introduction to Vertex AI

This lab uses [Vertex AI](https://cloud.google.com/ai-platform-unified/docs?utm_source=codelabs&utm_medium=et&utm_campaign=CDR_sar_aiml_vertexio_&utm_content=-), the unified AI platform on Google Cloud to train and deploy a ML model. Vertex AI offers two options on one platform to build a ML model: a codeless solution with **AutoML** and a code-based solution with **Custom Training** using Vertex **Workbench**. You use **AutoML** in this lab.

In this lab you build a ML model to determine whether a particular customer will repay a loan.

## Task 1: Prepare the training data

The initial Vertex AI dashboard illustrates the major stages to train and deploy a ML model: prepare the training data, train the model, and get predictions. Later, the dashboard displays your recent activities, such as the recent datasets, models, predictions, endpoints, and notebook instances.

### **Create a dataset**

1. In the Google Cloud Console, on the **Navigation menu**, click **Vertex AI**.
2. Click **Create dataset**.
3. On the Datasets page, give the dataset a name.
4. For the data type and objective, click **Tabular**, and then select **Regression/classification**.
5. Click **Create**.

### **Upload data**

Three options to import data in Vertex AI:

* Upload a local file from your computer.
* Select files from Cloud Storage.
* Select data from BigQuery.

For convenience, the dataset is already uploaded to Cloud Storage.

1. For the data source, select **Select CSV files from Cloud Storage**.
2. For **Import file path**, enter

spls/cbl455/loan\_risk.csv

Copied!

content\_copy

1. Click **Continue**.

You can also configure this page by clicking **Datasets** on the left menu and then selecting the dataset name on the Datasets page.

### **(Optional) Generate statistics**

1. To see the descriptive statistics for each column of your dataset, click **Generate statistics** . Generating the statistics might take a few minutes, especially the first time.
2. When the statistics are ready, click each column name to display analytical charts.

## Task 2: Train your model

With a dataset uploaded, you're ready to train a model to predict whether a customer will repay the loan.

* Click **Train new model**.

### **Training method**

The dataset is called LoanRisk.

1. For **Objective**, select **Classification**. Select classification instead of regression because you are predicting a distinct number (whether a customer will repay a loan: 0 for repay, 1 for default/not repay) instead of a continuous number.
2. Click **Continue**.

### **Model details**

Specify the name of the model and the target column.

1. Give the model a name, such as **LoanRisk**.
2. For **Target column**, select **Default** .
3. (Optional) Explore **Advanced options** to determine how to assign the training vs. testing data and specify the encryption.
4. Click **Continue**.

### **Training options**

Specify which columns you want to include in the training model. For example, ClientID might be irrelevant to predict loan risk.

1. Click the minus sign on the **ClientID** row to exclude it from the training model.
2. (Optional) Explore **Advanced options** to select different optimization objectives. For more information about optimization objectives for tabular AutoML models, see https://cloud.google.com/vertex-ai/docs/training/tabular-opt-obj.
3. Click **Continue**.

### **Compute and pricing**

1. For **Budget**, which represents the number of node hours for training, enter **1**. Training your AutoML model for 1 compute hour is typically a good start for understanding whether there is a relationship between the features and label you've selected. From there, you can modify your features and train for more time to improve model performance.
2. Leave early stopping enabled.
3. Click **Start training**.

Depending on the data size and the training method, the training can take from a few minutes to a couple of hours. Normally you would receive an email from Google Cloud when the training job is complete. However, in the Qwiklabs environment, you will not receive an email.

To save the waiting for the model training, you download a pre-trained model in **task 5** to get predictions in **task 6**. This pre-trained model is the training result following the same steps from **task 1** to **task 2**.

## Task 3: Evaluate the model performance (demonstration only)

Veretex AI provides many metrics to evaluate the model performance. You focus on three:

* **Precision/Recall curve**
* **Confusion Matrix**
* **Feature Importance**

If you had a model trained, you could navigate to the **Models** tab in Vertex AI.

* 1. Navigate to the **Models**.
* 2. Click on the model you just trained.
* 3. Browse the **Evaluate** tab.

However in this lab, you canskip this step since you use a pre-trained model.

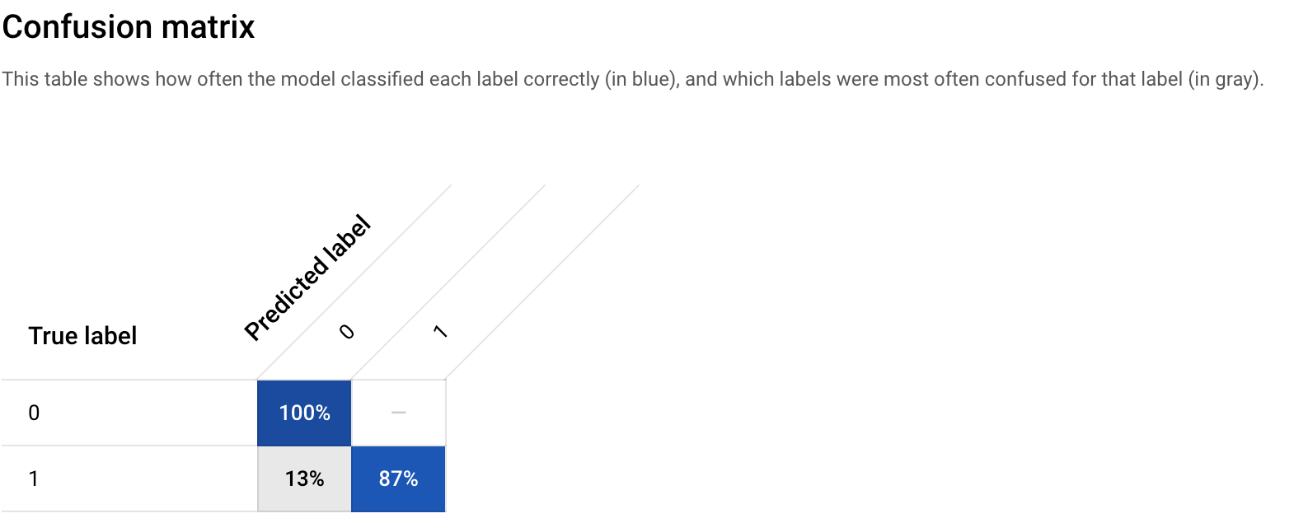
### **The Precision/Recall curve**



The confidence threshold determines how a ML model counts the positive cases. A higher threshold increases the precision, but decreases recall. A lower threshold decreases the precision, but increases recall. You can manually adjust the threshold to observe its impact on precision and recall and find the best tradeoff point between the two to meet your business needs.

### **The confusion matrix**

A [confusion matrix](https://developers.google.com/machine-learning/glossary#confusion-matrix) tells you the percentage of examples from each class in your test set that your model predicted correctly.

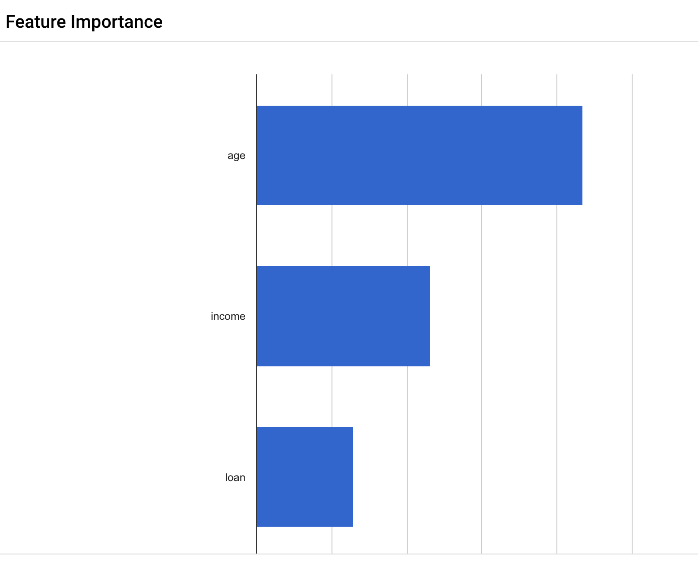


The confusion matrix shows that your initial model is able to predict 100% of the repay examples and 87% of the default examples in your test set correctly, which is not too bad.

You can improve the precentage by adding more examples (more data), engineering new features, and changing the training method, etc.

### **The feature importance**

In Vertex AI, feature importance is displayed through a bar chart to illustrate how each feature contributes to a prediction. The longer the bar, or the larger the numerical value associated with a feature, the more important it is.



These feature importance values could be used to help you improve your model and have more confidence in its predictions. You might decide to remove the least important features next time you train a model or to combine two of the more significant features into a [feature cross](https://developers.google.com/machine-learning/glossary#feature-cross) to see if this improves model performance.

Feature importance is just one example of Vertex AI’s comprehensive machine learning functionality called **Explainable AI**. Explainable AI is a set of tools and frameworks to help understand and interpret predictions made by machine learning models.

## Task 4: Deploy the model (demonstration only)

You will not deploy the model to an endpoint because the model training can take an hour. Here you can review the steps you would perform in a production environment.

Now that you have a trained model, the next step is to create an endpoint in Vertex. A model resource in Vertex can have multiple endpoints associated with it, and you can split traffic between endpoints.

### **Create and define an endpoint**

1. On your model page, on the **Deploy and test** tab, click **Deploy to endpoint**.
2. For **Endpoint name**, enter a name for your endpoint, such as **LoanRisk**.
3. Click **Continue**.

### **Model settings and monitoring**

1. Leave the traffic splitting settings as-is.
2. As the machine type for your model deployment, under **Machine type**, select **n1-standard-8, 8 vCPUs, 30 GiB memory**.
3. Leave the remaining settings as-is.
4. Click **Deploy**.

Your endpoint will take a few minutes to deploy. When it is completed, a **green check mark** will appear next to the name.

Now you're ready to get predictions on your deployed model.

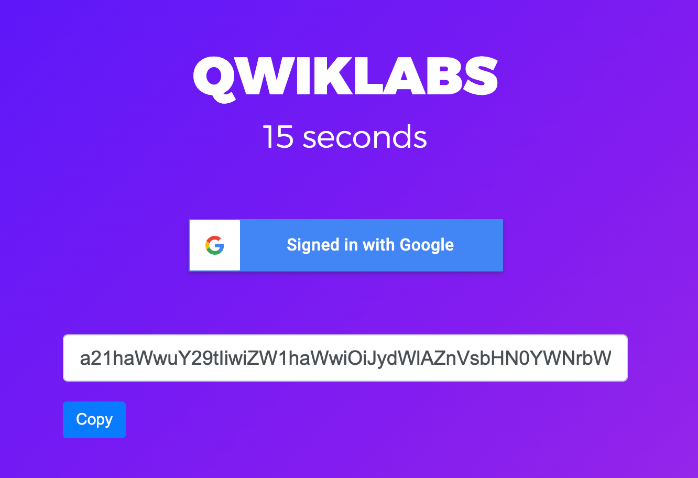
## Task 5: SML Bearer Token

### **Retrieve your Bearer Token**

To allow the pipeline to authenticate, and be authorized to call the endpoint to get the predictions, you will need to provide your Bearer Token.

Follow the instructions below to get your token. If you have issues getting the Bearer Token, this can be due to cookies in the incognito window. If this is happening to you, try this step in a non-incognito window.

1. Log in to <https://gsp-auth-kjyo252taq-uc.a.run.app/>
2. When logging in, use your student email address and password.
3. Click the **Copy** button. This will copy a very long token to your clipboard.



This token will only be available for about 60 seconds, so copy and and move on to the next steps.

If you have issues getting the Bearer Token, this can be due to cookies in the incognito window - try in a non-incognito window.

## Task 6: Get predictions

In this section, use the Shared Machine Learning (SML) service to work with an existing trained model.

|  |  |
| --- | --- |
| **ENVIRONMENT VARIABLE** | **VALUE** |
| AUTH\_TOKEN | Use the value from the previous section |
| ENDPOINT | https://sml-api-vertex-kjyo252taq-uc.a.run.app/vertex/predict/tabular\_classification |
| INPUT\_DATA\_FILE | INPUT-JSON |

To use the trained model, you will need to create some environment variables.

1. Open a Cloud Shell window.
2. Replace INSERT\_SML\_BEARER\_TOKEN with the bearer token value from the previous section:

AUTH\_TOKEN="INSERT\_SML\_BEARER\_TOKEN"

1. Download the lab assets:

gsutil cp gs://spls/cbl455/cbl455.tar.gz .

1. Extract the lab assets:

tar -xvf cbl455.tar.gz

1. Create an ENDPOINT environment variable:

ENDPOINT="https://sml-api-vertex-kjyo252taq-uc.a.run.app/vertex/predict/tabular\_classification"

1. Create a INPUT\_DATA\_FILE environment variable:

INPUT\_DATA\_FILE="INPUT-JSON"

After the lab assets are extracted, take a moment to review the contents. The **INPUT-JSON** file is used to provide Vertex AI with the model data required.

Alter this file to generate custom predictions. The smlproxy application is used to communicate with the backend.

The file INPUT-JSON is composed of the following values:

|  |  |  |  |
| --- | --- | --- | --- |
| **age** | **ClientID** | **income** | **loan** |
| 40.77 | 997 | 44964.01 | 3944.22 |

Test the SML Service by passing the parameters specified in the environment variables:

1. Perform a request to the SML service:

./smlproxy tabular \

-a $AUTH\_TOKEN \

-e $ENDPOINT \

-d $INPUT\_DATA\_FILE

This query should result in a response similar to this:

SML Tabular HTTP Response:

2022/01/10 15:04:45 {"model\_class":"0","model\_score":0.9999981}

1. Alter the INPUT-JSON file to test a new scenario:

|  |  |  |  |
| --- | --- | --- | --- |
| **age** | **ClientID** | **income** | **loan** |
| 30.00 | 998 | 50000.00 | 20000.00 |

Test the SML Service by passing the parameters specified in the environment variables:

1. Edit the file INPUT-JSON and replace the original values.
2. Perform a request to the SML service:

./smlproxy tabular \

-a $AUTH\_TOKEN \

-e $ENDPOINT \

-d $INPUT\_DATA\_FILE

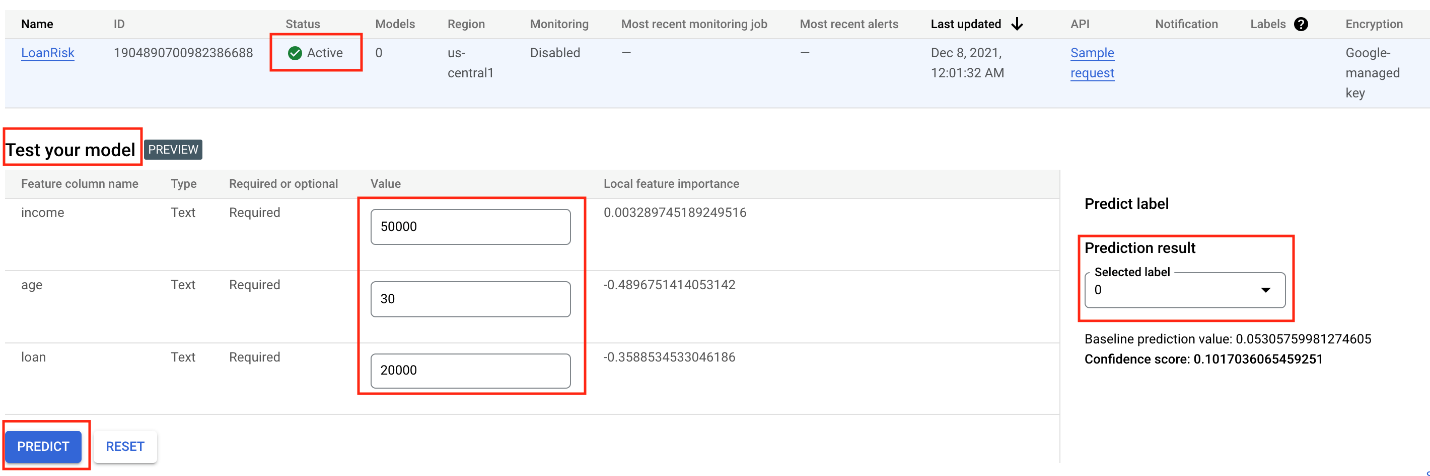
In this case, assuming that the person's income is 50,000, age 30, and loan 20,000, the model predicts that this person will repay the loan.

This query should result in a response similar to this::

SML Tabular HTTP Response:

2022/01/10 15:04:45 {"model\_class":"0","model\_score":0.9999981}

If you use the Google Cloud Console, the following image illustrates how the same action could be performed:



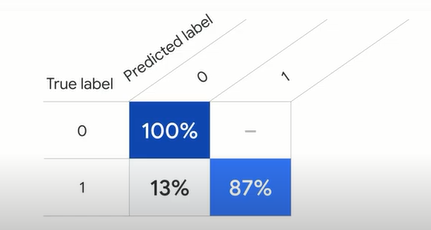
You can now use Vertex AI to:

* Upload a dataset.
* Train a model with AutoML.
* Evaluate the model performance.
* Deploy the trained AutoML model to an endpoint.
* Get predictions.

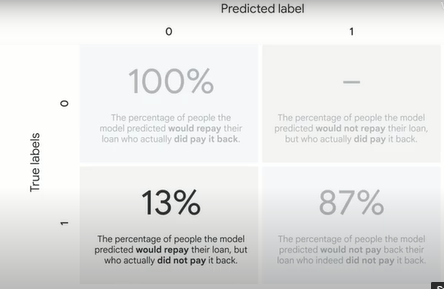
🎉 Congratulations! 🎉

**Lab recap: Predicting loan risk with AutoML**

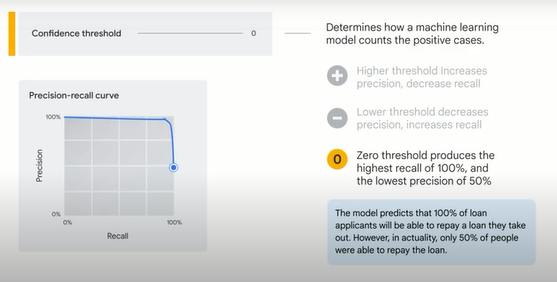
well done on completing the automl lab you've now had a chance to use vertex ai to build a machine learning model without writing a single line of code let's take a few moments to review the results of the lab starting with the confusion matrix but before that begins please pause to consider the matrix results for yourself



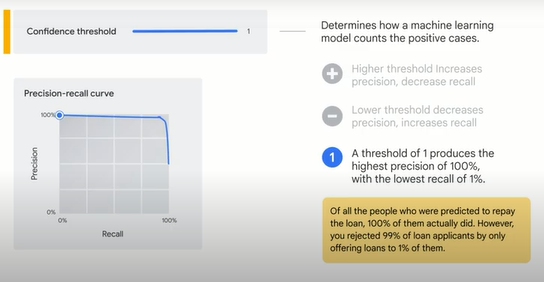
the true positives were 100 this represents the percentage of people the model predicted would repay their loan who actually did pay it back the true negatives were 87 percent this represents the percentage of people the model predicted would not repay their loan who indeed did not pay it back the false negatives were zero this represents the percentage of people the model predicted would not repay their loan but who actually did pay it back and finally the false positives were 13 percent this represents the percentage of people the model predicted would repay their loan but who actually did not pay it back as a general principle it's good to have high true positives and true negatives and low false positives and false negatives however how high or low they need to be really depends on the business goals you're looking to achieve there are different ways to improve the performance of a model which might include using a more accurate data source using a larger data set choosing a different type of ml model or tuning the hyper parameters



let's also review the precision recall curve from the automl lab the confidence threshold determines how a machine learning model counts the positive cases a higher threshold increases the precision but decreases recall a lower threshold decreases the precision but increases recall moving the threshold to zero produces the highest recall of one hundred percent and the lowest precision of fifty percent so what does that mean that means the model predicts that one hundred percent of loan applicants will be able to repay a loan they take out however in actuality only 50 percent of people were able to repay the loan using this threshold to identify the default cases in this example can be risky because it means that you're only likely to get half of the loan investment back

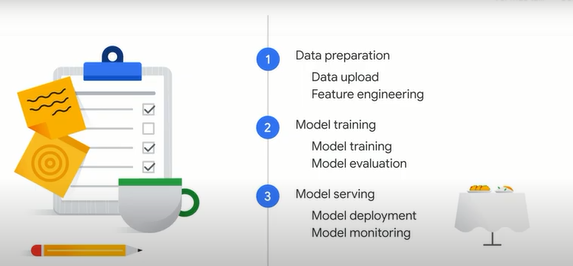


now let's move to the other extreme by moving the threshold to one this would produce the highest precision of one hundred percent with the lowest recall of one percent so what does that mean it means that of all the people who are predicted to repay the loan a hundred percent of them actually did however you rejected 99 of loan applicants by only offering loans to one percent of them that's a pretty big loss of business for your company these are both extreme examples but it's important that you always try to set an appropriate threshold for your model.



**Summary.**

before we finish this final section of the course let's quickly review the three stages of the machine learning workflow with the help of our restaurant analogy in stage one data preparation we uploaded data and applied feature engineering this translated to gathering our ingredients and then chopping and prepping them in the kitchen in stage 2 model training the model was trained and evaluated this is where we experimented with our recipes and tasted the meal to make sure it turned out as expected and in the final stage model serving the model was deployed and monitored this translates to serving the meal to the hungry customers and adjusting the menu as more people tried the dish**.**



**Quiz**

1.A hospital uses Google’s machine learning technology to help pre-diagnose cancer by feeding historical patient medical data to the model. The goal is to identify as many potential cases as possible. Which metric should the model focus on?

Feature importance

**Recall**

Precision

Confusion matrix

2. Which Vertex AI tool automates, monitors, and governs machine learning systems by orchestrating the workflow in a serverless manner?

**Vertex AI Pipelines**

Vertex AI console

Vertex AI Workbench

Vertex AI Feature Store

3. Which stage of the machine learning workflow includes model evaluation?

Model serving

**Model training**

Data preparation

4.Which stage of the machine learning workflow includes feature engineering?

Model serving

Model training

**Data preparation**

5. A farm uses Google’s machine learning technology to detect defective apples in their crop, such as those that are irregular in size or have scratches. The goal is to identify only the apples that are actually bad so that no good apples are wasted. Which metric should the model focus on? (*Una granja utiliza la tecnología de aprendizaje automático de Google para detectar manzanas defectuosas en su cultivo, como las que tienen un tamaño irregular o tienen arañazos. El objetivo es identificar solo las manzanas que son realmente malas para que no se desperdicien manzanas buenas. ¿En qué métrica debe centrarse el modelo*?)

Feature importance

Recall

**Precision**

Confusion matrix

6. Select the correct machine learning workflow.

Model serving, data preparation, model training

Data preparation, model serving, model training

Model training, data preparation, model serving

**Data preparation, model training, model serving**

**Course Summary**

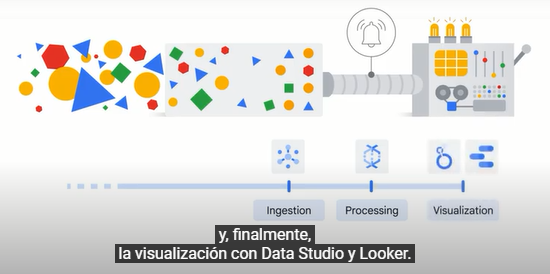
congratulations on completing the big data and machine learning fundamentals course we hope that you learned some valuable information from this course that will help advance your career

throughout the course we introduced a number of products and technologies to support google's data to ai lifecycle let's do a final review of the main concepts presented

**in the first section** of the course you are introduced to the google cloud infrastructure and google's big data and machine learning products of the three layers of the google cloud infrastructure you explored the middle and top layers on the middle layer sit compute and storage google cloud decouples computer storage so they can scale independently based on need and on the top layer sit the big data and machine learning products which enable you to perform tasks to ingest store process and deliver business insights data pipelines and machine learning models



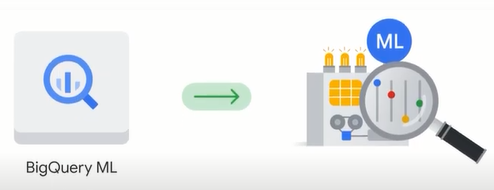
**in the second section** of the course you explored data engineering for streaming data this included how to build a streaming data pipeline from ingestion with pub sub to processing with data flow and finally to visualization using data studio and looker



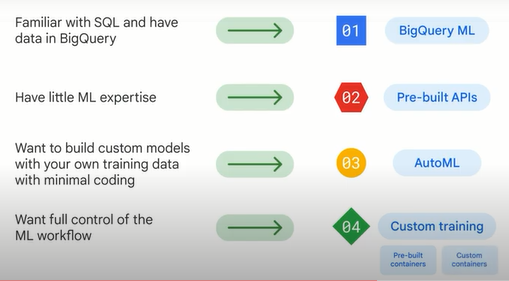
after that **in the third section** of the course you are introduced to bigquery which is google's fully managed data warehouse bigquery provides two services in one storage plus analytics



you also learned about bigquery ml the machine learning tool used for developing machine learning models directly in bigquery



**in the fourth section** of the course you explored the options available to build and deploy machine learning models with google cloud if you're familiar with sql and already have data in bigquery you can use bigquery ml to develop machine learning models if you have little ml experience using pre-built apis is likely the best choice pre-built apis address common perceptual tasks such as vision video and natural language they are ready to use without any ml expertise or model development effort if you want to build custom models with your own training data while spending minimal time coding then automl is a great choice automl provides a codeless solution to enable you to focus on business problems instead of the underlying model architecture and ml provisioning and if you want full control of the machine learning workflow vertex ai custom training lets you train and serve custom models with code on vertex workbench using pre-built containers you can leverage popular ml libraries such as tensorflow and pytorch alternatively you can build a custom container from scratch



**in the final section** of the course you learned about the machine learning workflow using vertex ai a unified platform that brings all the components of the machine learning ecosystem and workflow together the machine learning workflow comprises three stages in stage one data preparation data is uploaded and feature engineering is applied in stage 2 model training the model is trained and evaluated and in stage 3 model serving the model is deployed and monitored we hope that this course is just the beginning of your big data and machine learning journey for more training and hands-on practice with machine learning and ai please explore the options available at cloud.google.com forward slash training forward slash machine learning dash ai and if you're interested in validating your expertise and showcasing your ability to transform businesses with google cloud technology you might consider working toward a google cloud certification you can learn more about google cloud certification offerings at cloud.google.com forward slash certifications thanks for completing this course we'll see you next time