# Machine Learning in Enterprise

# Data Preprocessing with BigQuery

**BigQuery**

Use BigQuery to process tabular data

If you're using tabular data, use BigQuery for data processing and transformation steps.

When you're working with ML, use BigQuery ML in BigQuery. Perform the transformation as a normal BigQuery query, then save the results to a permanent table.

# Transforming unstructured data with Dataflow

**Dataflow**

Use Dataflow to process large volumes of unstructured data

Use Dataflow to convert the unstructured data into binary data formats like TFRecord, which can improve performance of data ingestion during training.

If you need to perform transformations that are not expressible in Cloud SQL or are for streaming, you can use a combination of Dataflow and the pandas library.

# Data Preprocessing with DataProc

DataProc

Existing Hadoop with Spark

Dataproc is recommended for customers with existing implementations using Hadoop with Spark to perform ETL, or who want to leverage their experience with Hadoop on-premises to create a cloud-based solution.

Autoscaling is supported

# TensorFlow Extended

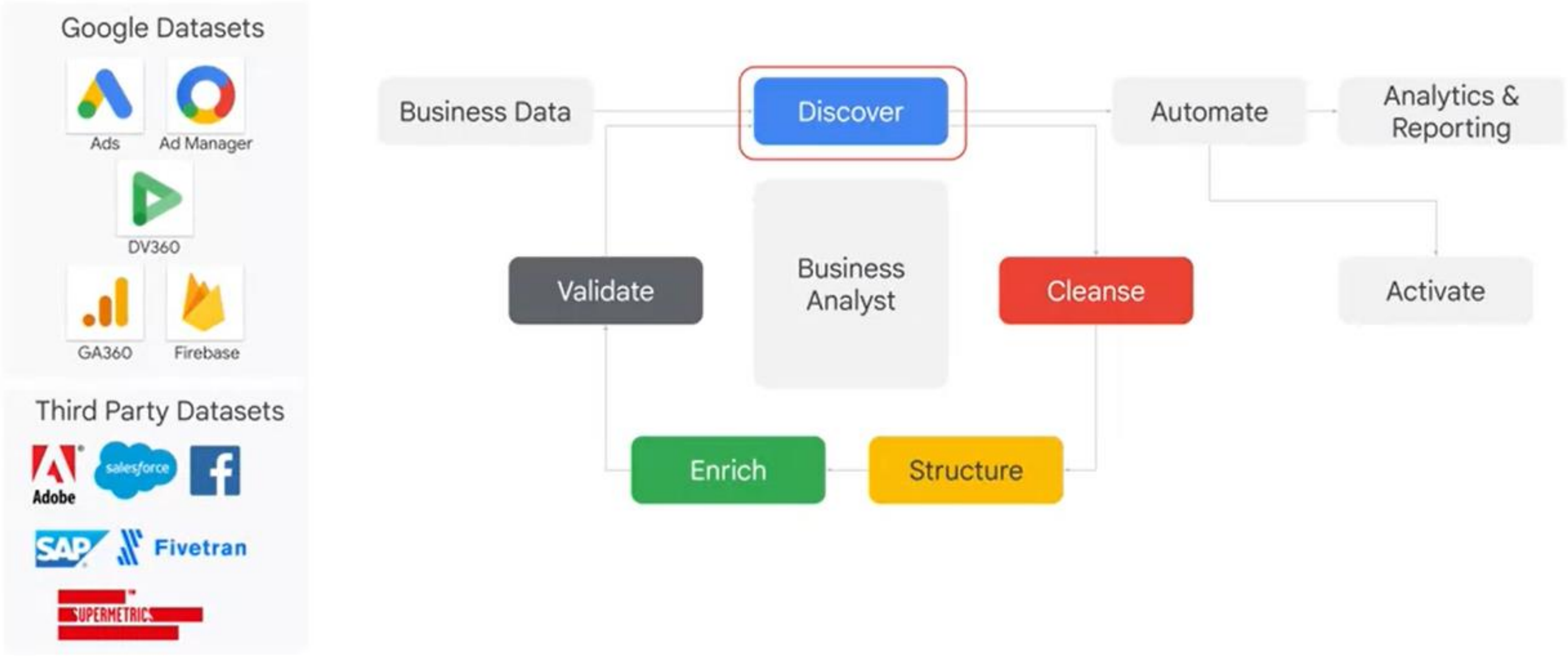
**TensorFlow Extended**

Use TensorFlow Extended when leveraging TensorFlow ecosystem.

If you're using TensorFlow for model development, use TensorFlow Extended to prepare your data for training.

TensorFlow Transform is the TensorFlow component that enables defining and executing a preprocessing function to transform your data.

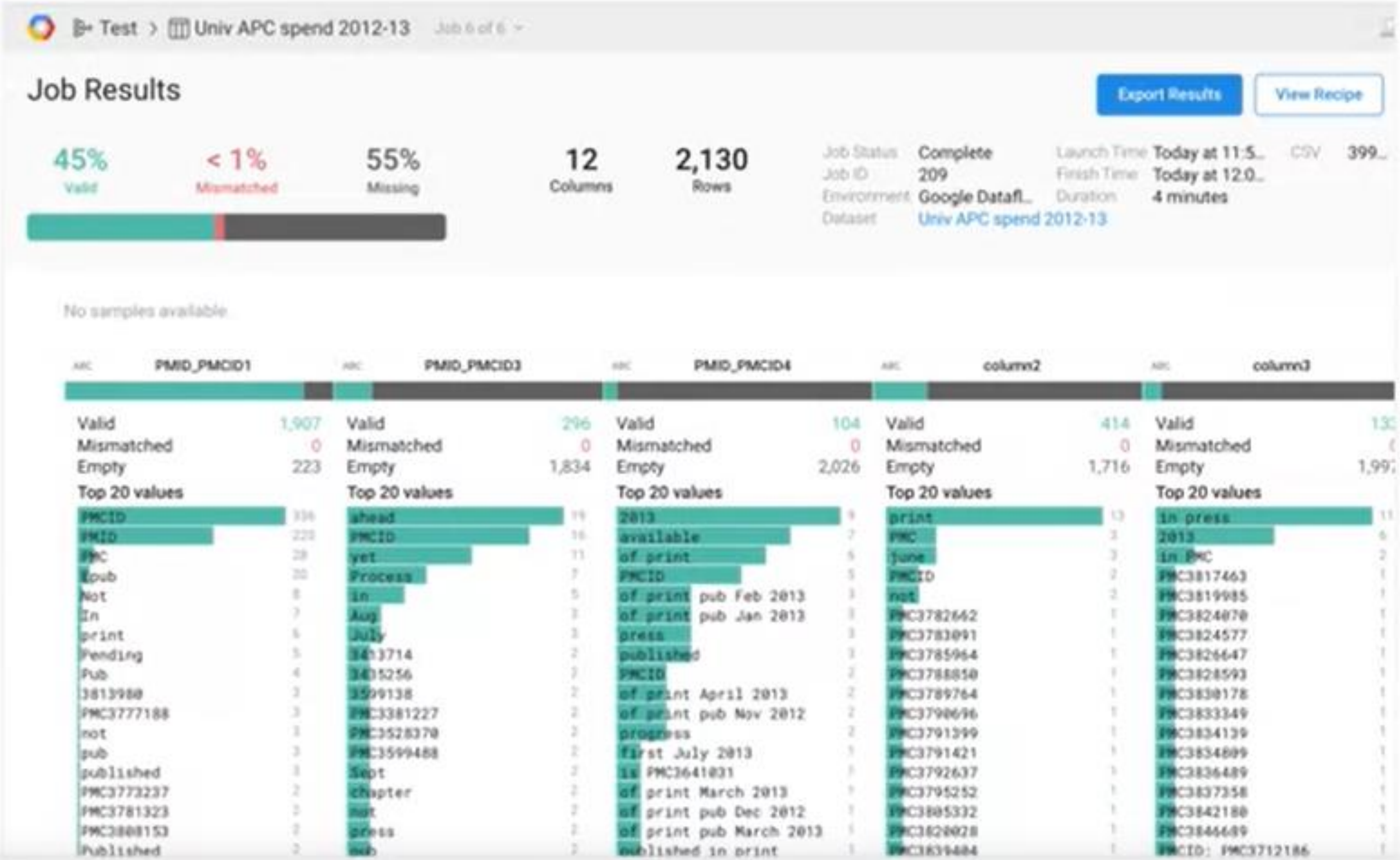
# Data lifecycle with Dataprep



# How does Dataprep fit into Google Cloud?

# Fast exploration and anomaly detection

- Visually explore and interact with data
- Instantly understand data distribution and patterns



# Learning rate controls the size of the step in weight space

If too small, training will take a long time

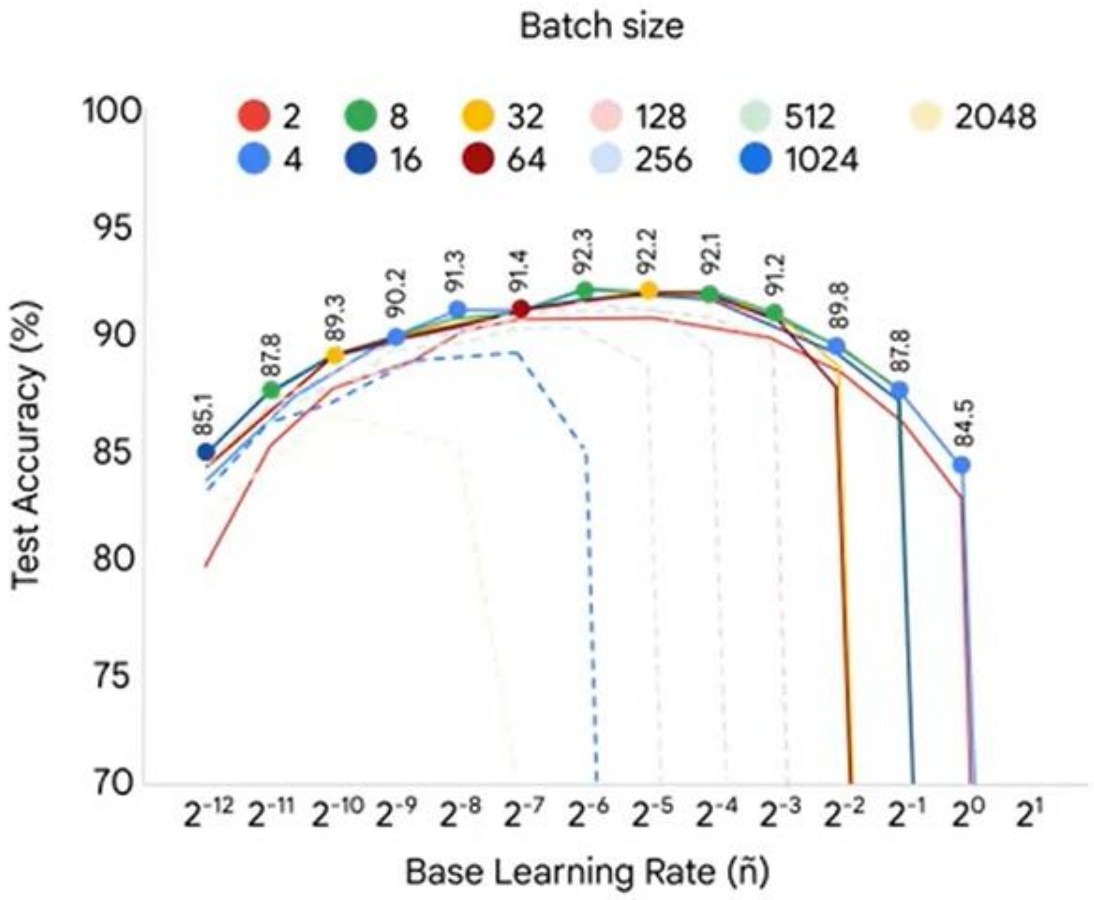
If too large, training will bounce around

# The batch size controls the number of samples that gradient is calculated on.

If too small, training will bounce around

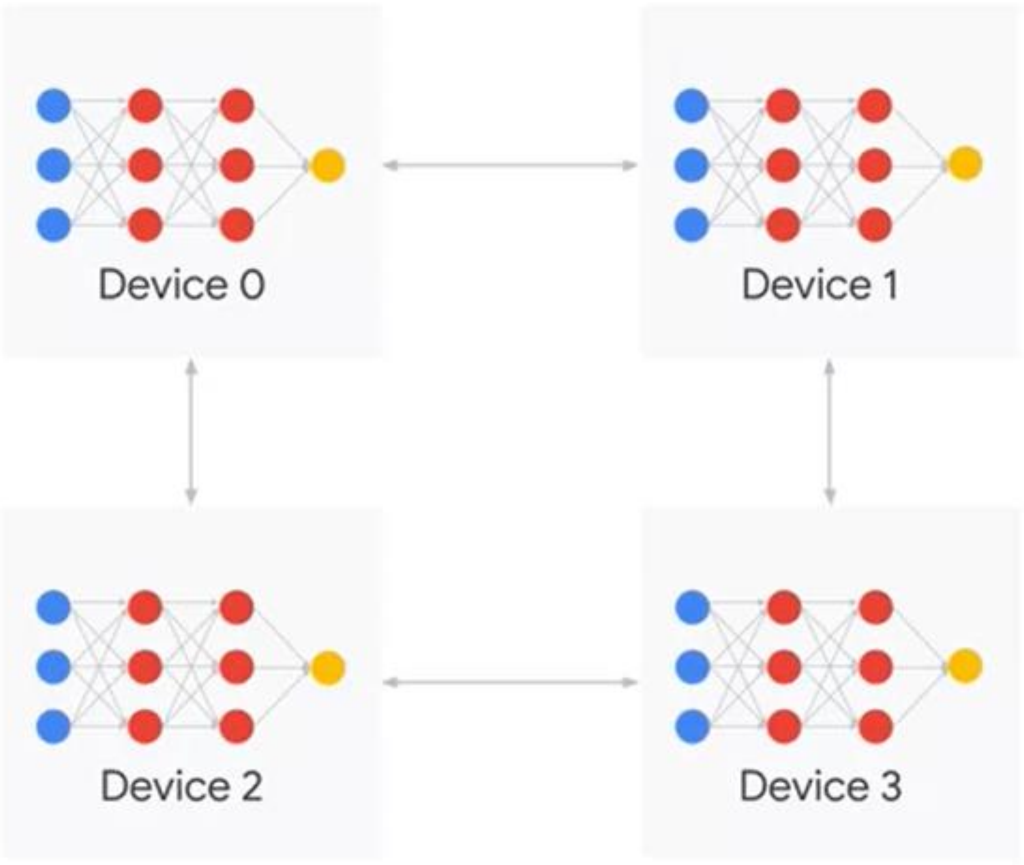
If too large, training will take a very long time

Larger batch sizes require smaller learning rates

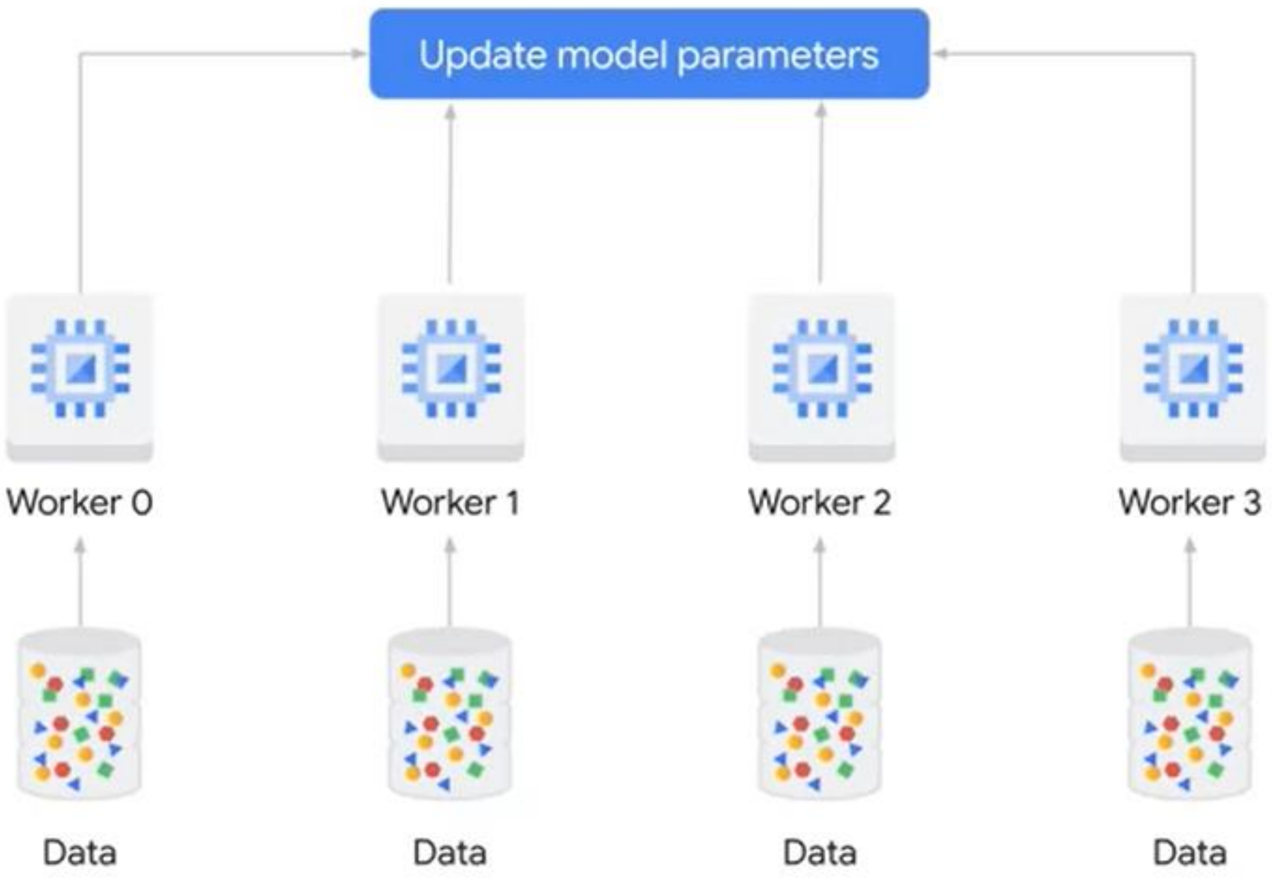



Revisiting Small Batch Training for Deep Neural Networks, Masters and Luschi, 2018

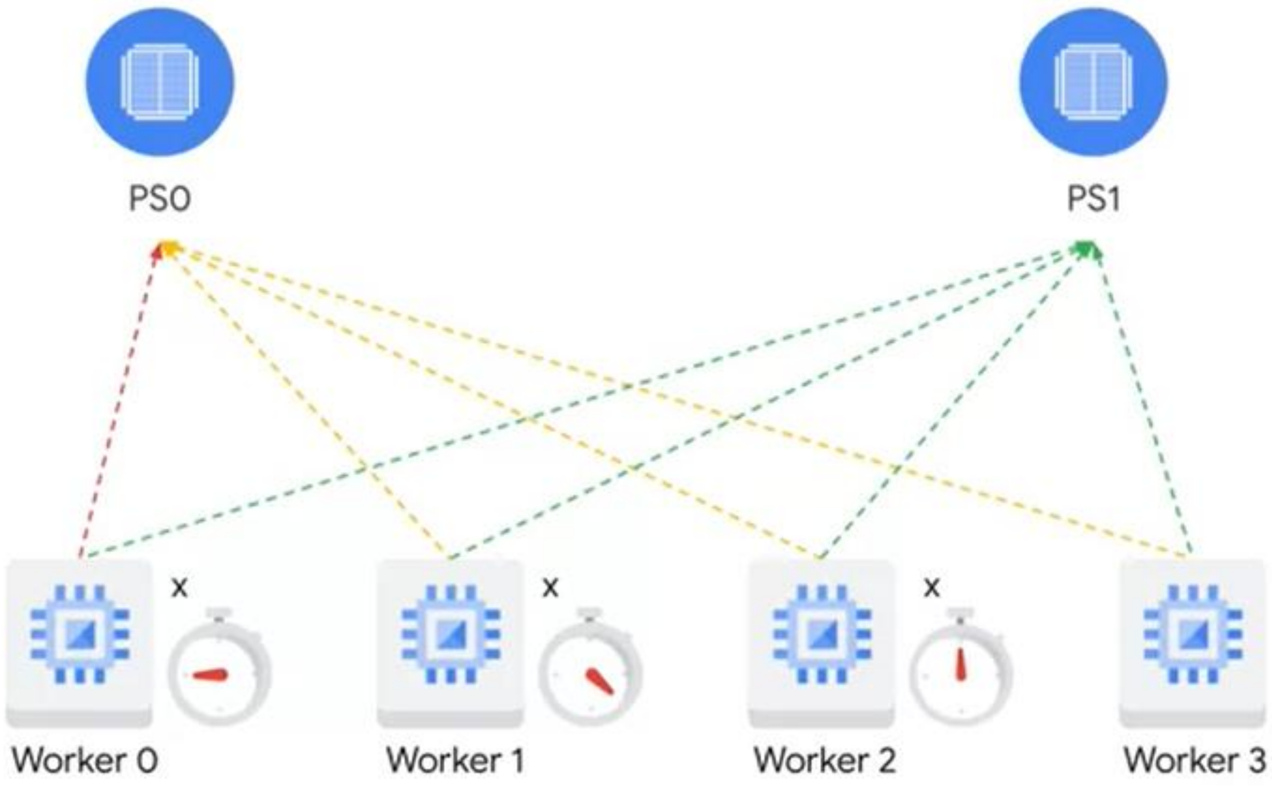
# How can you make model training faster?

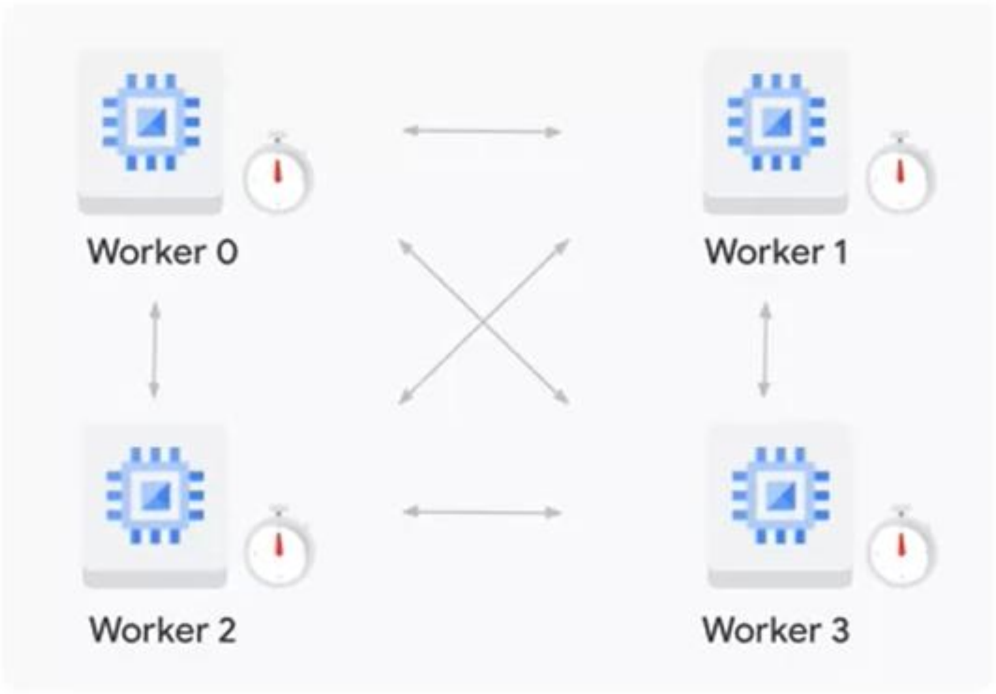
# Data parallelism



# Async parameter server

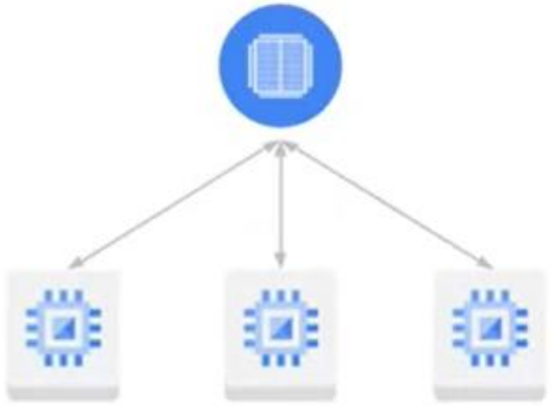


# Sync allreduce architecture



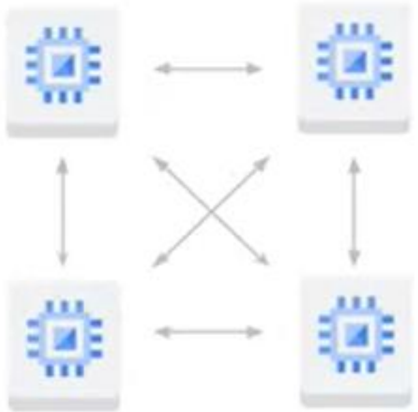
# There isn’t one right answer, but here are some considerations

Consider async parameter server if...



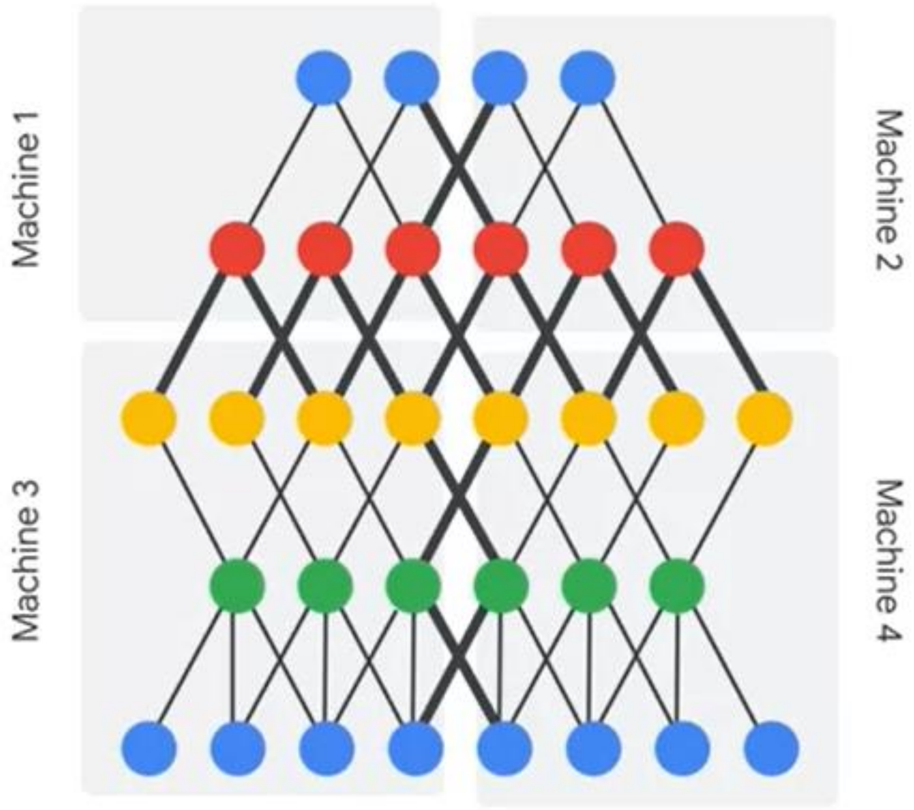
- Many low-power or unreliable workers.
- More mature approach.
- Constrained by I/O.

Consider allreduce parameter server if...



- Multiple devices on one host.
- Fast devices with strong links (e.g. TPUs).
- Better for multiple GPUs.
- Constrained by compute power.

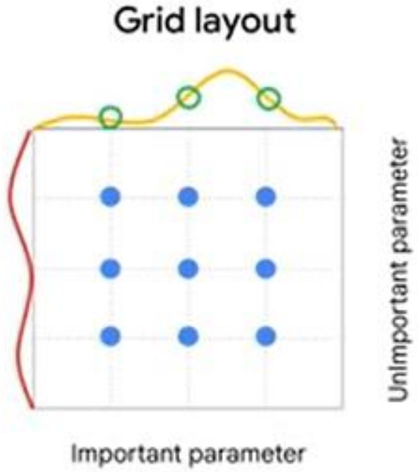
# Model parallelism

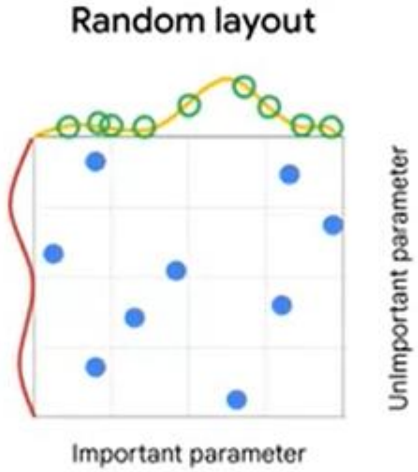


# Vertex Vizier

hyperparameter tuning

# Grid and Random Search




**Grid search**

- Sets up a grid of specific model hyperparameters
- Train/Test model on every combination
- Not suitable for large parameter spaces

**Random search**

- Sets up a grid of specific model hyperparameters
- Randomly selects the combination of hyperparameter values
- Faster than Grid Search but not as effective

# Bayesian optimization

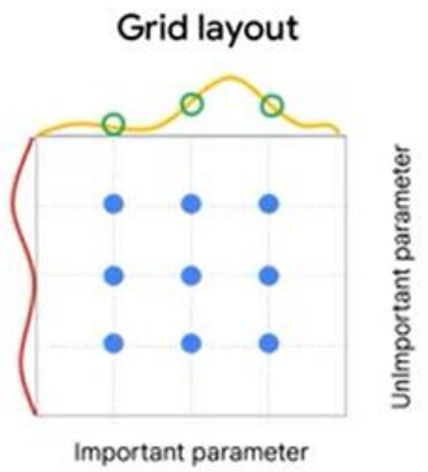
## Advantages

- Past evaluations when choosing the hyperparameter are set
- Typically requires less iterations to get to the optimal set of hyperparameter value
- Limits the number of times a model needs to be trained for validation

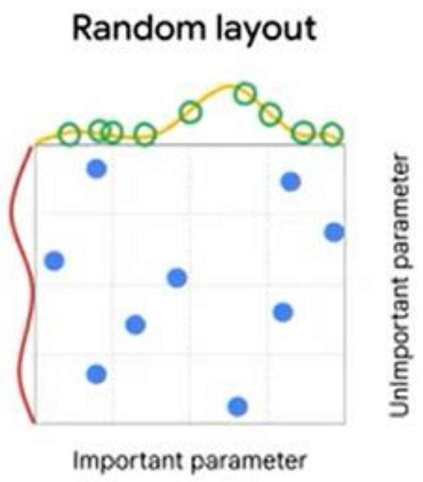
## Process

- Build a model
- Select hyperparameters
- Train and evaluate
- Update the model
- Repeat (or iterate) until max iterations are reached

# Vertex Vizier




GRID_SEARCH




RANDOM_SEARCH

BAYESIAN OPTIMIZATION

# Deploy your model and make online predictions

When you're satisfied with your model's performance, it's time to use your model.

Depending on your use case, you can use your model in different ways.

**Batch prediction**

- Useful for making many prediction requests at once
- Asynchronous

**Online prediction**

- Useful if your model is part of an application and you need quick prediction turnaround
- Used with a model made available using a REST API
- Synchronous

# Use pre-built and custom containers to make predictions

## Pre-built containers

- Provided as Docker container images
- Organized by machine learning (ML) framework and framework version
- Can be used to serve predictions with minimal configuration

## Custom containers

- Require a Docker container image that runs an HTTP server
- Must listen and respond to liveness checks, health checks, and prediction requests

# Batch predictions requirements

## BigQuery table requirements

- BigQuery data source tables must be no larger than 100 GB.
- You must use a multi-regional BigQuery dataset in the US or EU locations.
- If the table is in a different project, you must provide the BigQuery Data Editor role to the Vertex AI service account in that project.

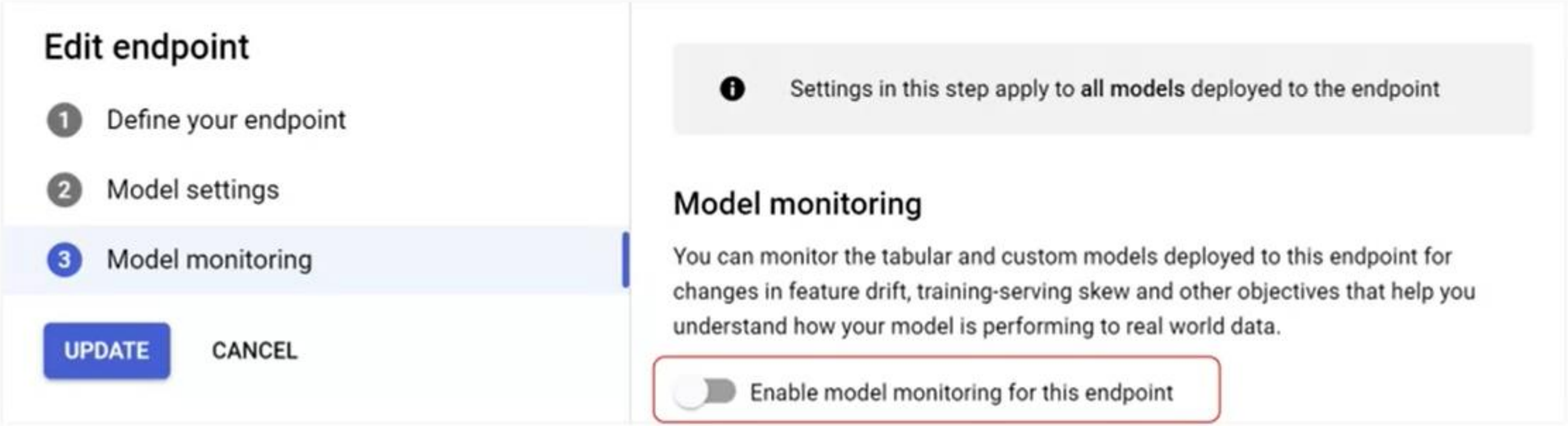
## CSV file requirements

- The first line of the data source must contain the name of the columns.
- Each data source file must not be larger than 10 GB. You can include multiple files, up to a maximum amount of 100 GB.
- If the Cloud Storage bucket is in a different project than where you use Vertex AI, you must provide the Storage Object Creator role to the Vertex AI service account in that project.

# Vertex AI Model Monitoring

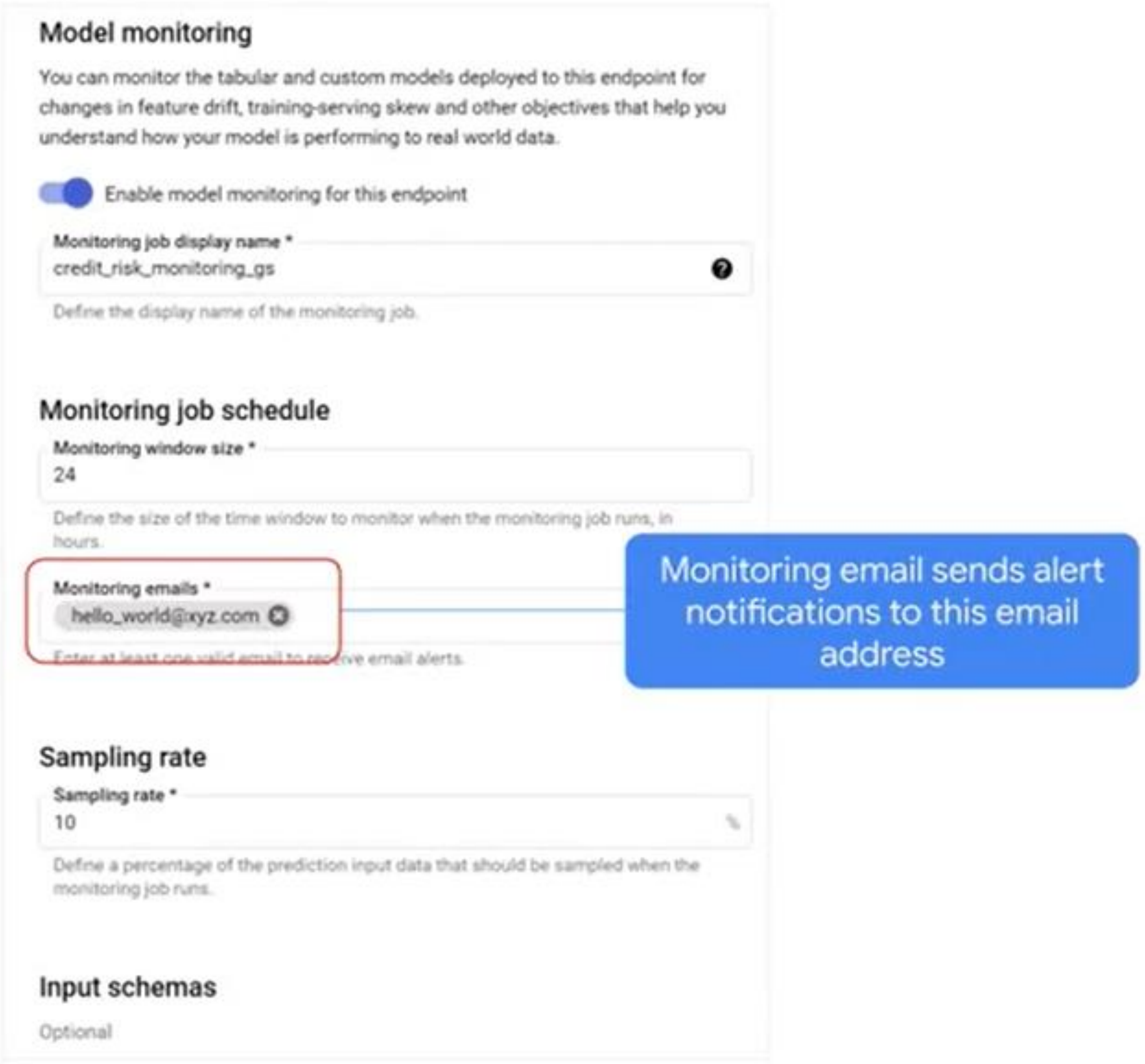
- Drift in data quality
- Skew in training vs. serving data
- Feature attribution
- UI visualizations

# Toggle the switch to enable model monitoring

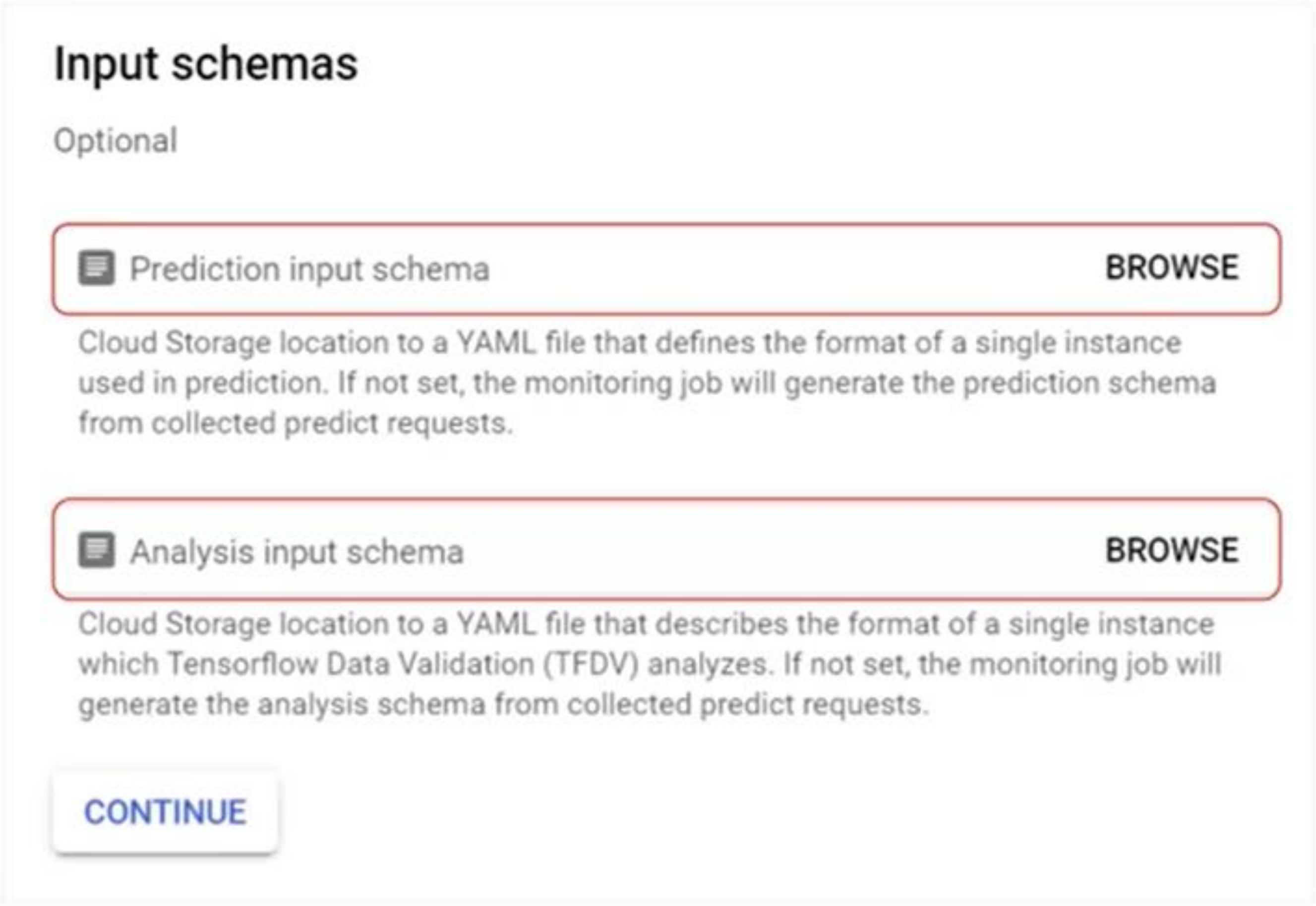


# Monitoring job schedule

- The time at which the monitoring job ran
- The name of the feature that has skew or drift
- The alerting threshold as well as the recorded statistical distance measure



# Input schema (optional)



# Calculate training-serving skew and prediction drift

Model Monitoring computes the statistical distribution of the latest feature values seen in production.

## Baselines for skew and drift

Model Monitoring uses different baselines for skew detection and drift detection:

- For skew detection, the baseline is the statistical distribution of the feature's values in the training data.
- For drift detection, the baseline is the statistical distribution of the feature's values seen in production in the recent past.

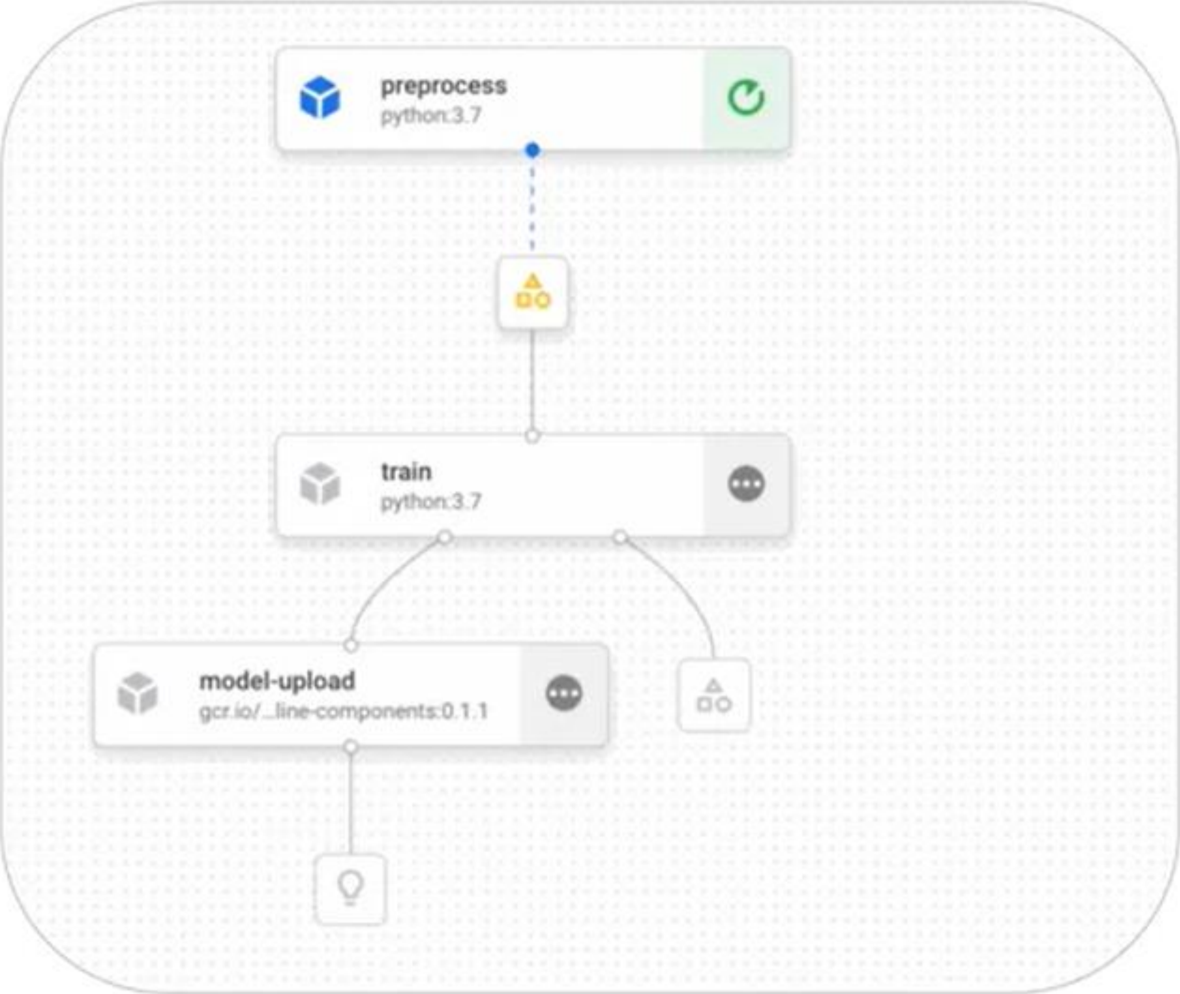
# VERTEX AI PIPELINE

# Pipeline

A pipeline is composed of **modular** pieces , components

Offers **automation** and **orchestration**

**Components are chained** with dsl to form a pipeline



# Building a Pipeline

## Describe workflow as a pipeline

- Before Vertex AI Pipelines can orchestrate your ML workflow, you must describe your workflow as a pipeline.
- ML pipelines are portable and scalable ML workflows that are based on containers and Google Cloud services.

## Which pipeline SDK?

- If you use TensorFlow in an ML workflow that processes terabytes of structured data or text data, we recommend that you build your pipeline using TFX.
- For other use cases, build your pipeline using the Kubeflow Pipelines SDK. Implement your workflow by building custom components or reusing prebuilt components, such as the Google Cloud Pipeline Components.