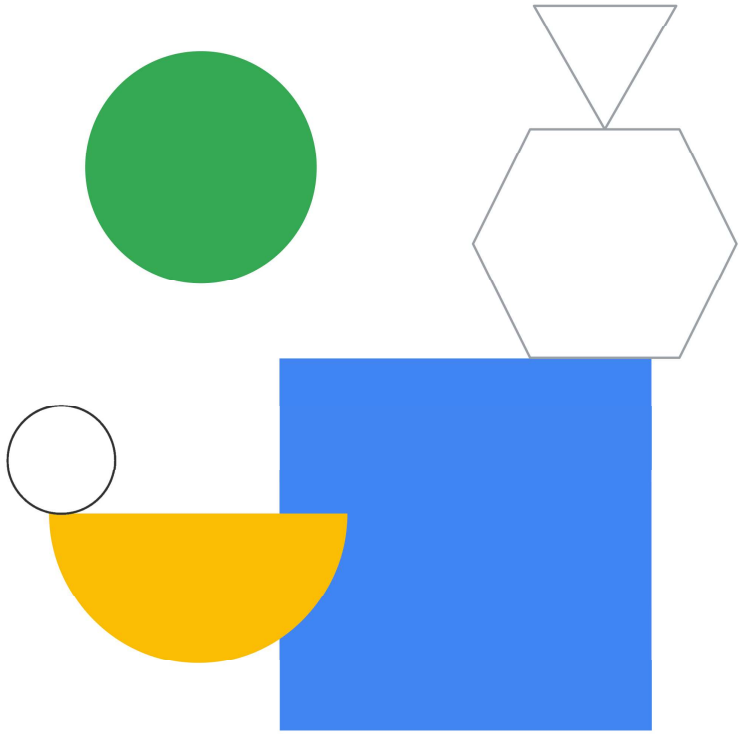
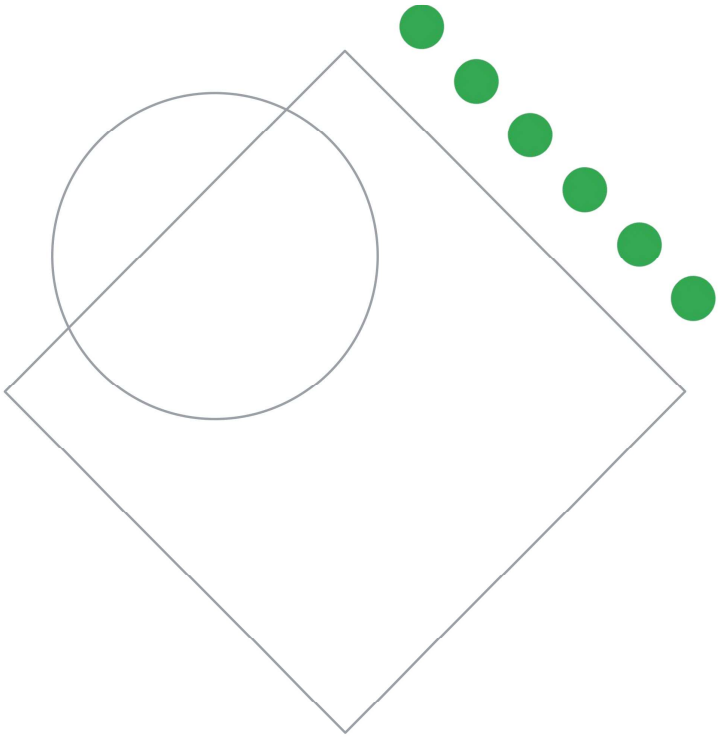
Google Cloud

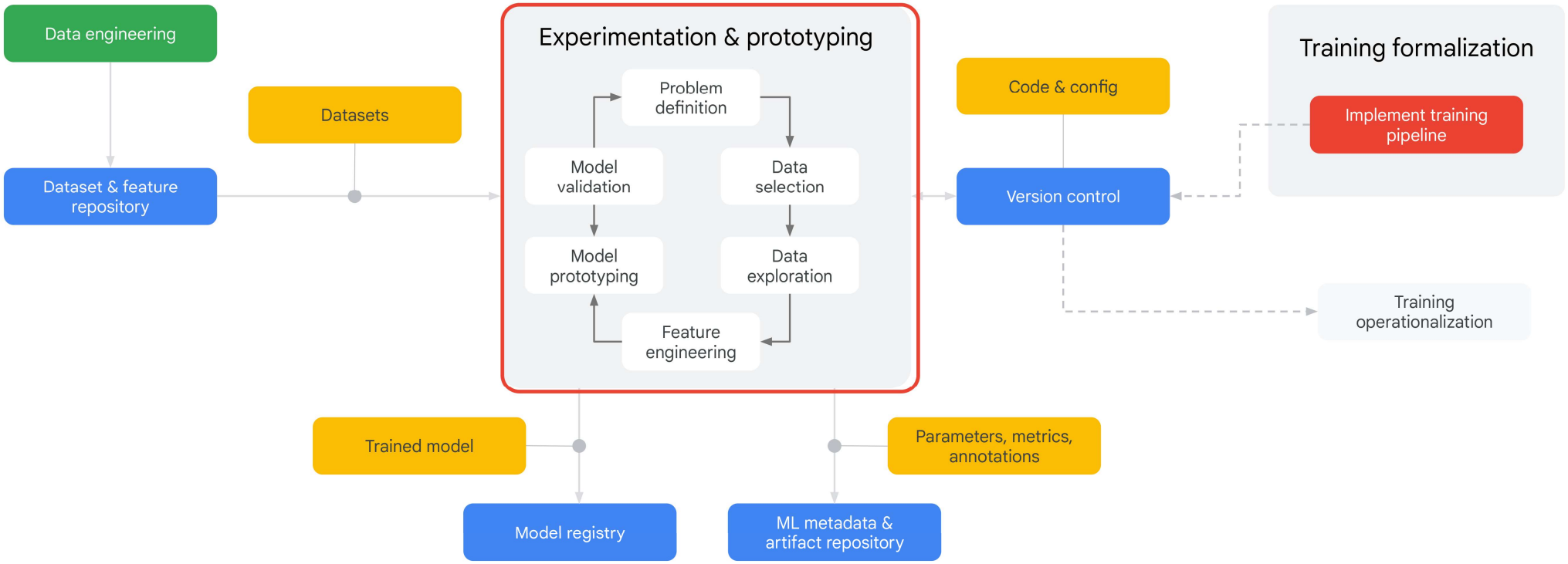
Machine Learning in the Enterprise



Understanding the ML Enterprise Workflow

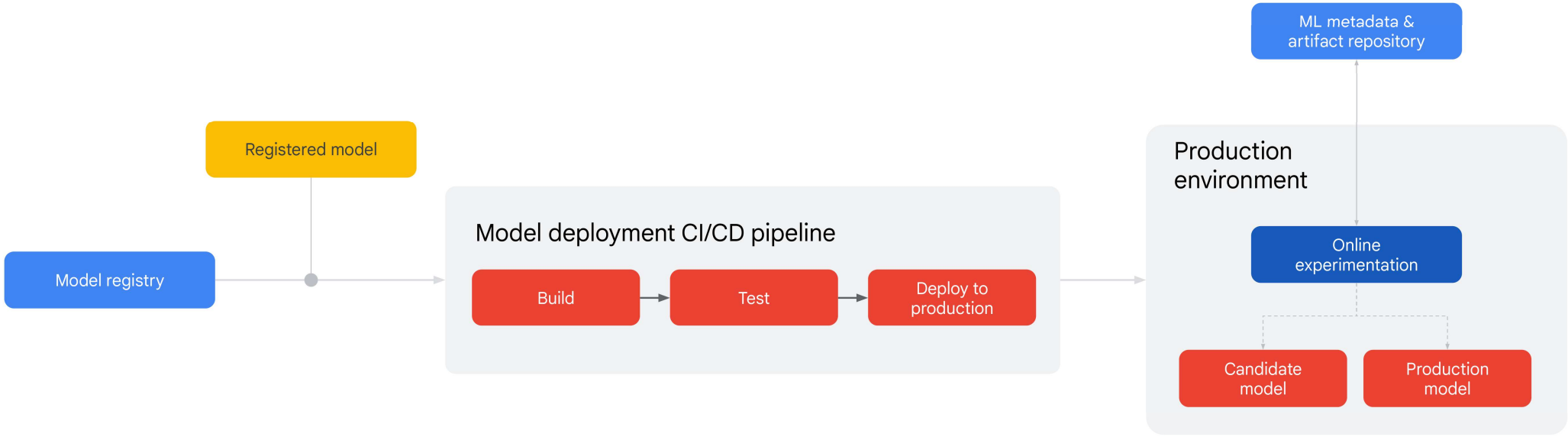


ML development

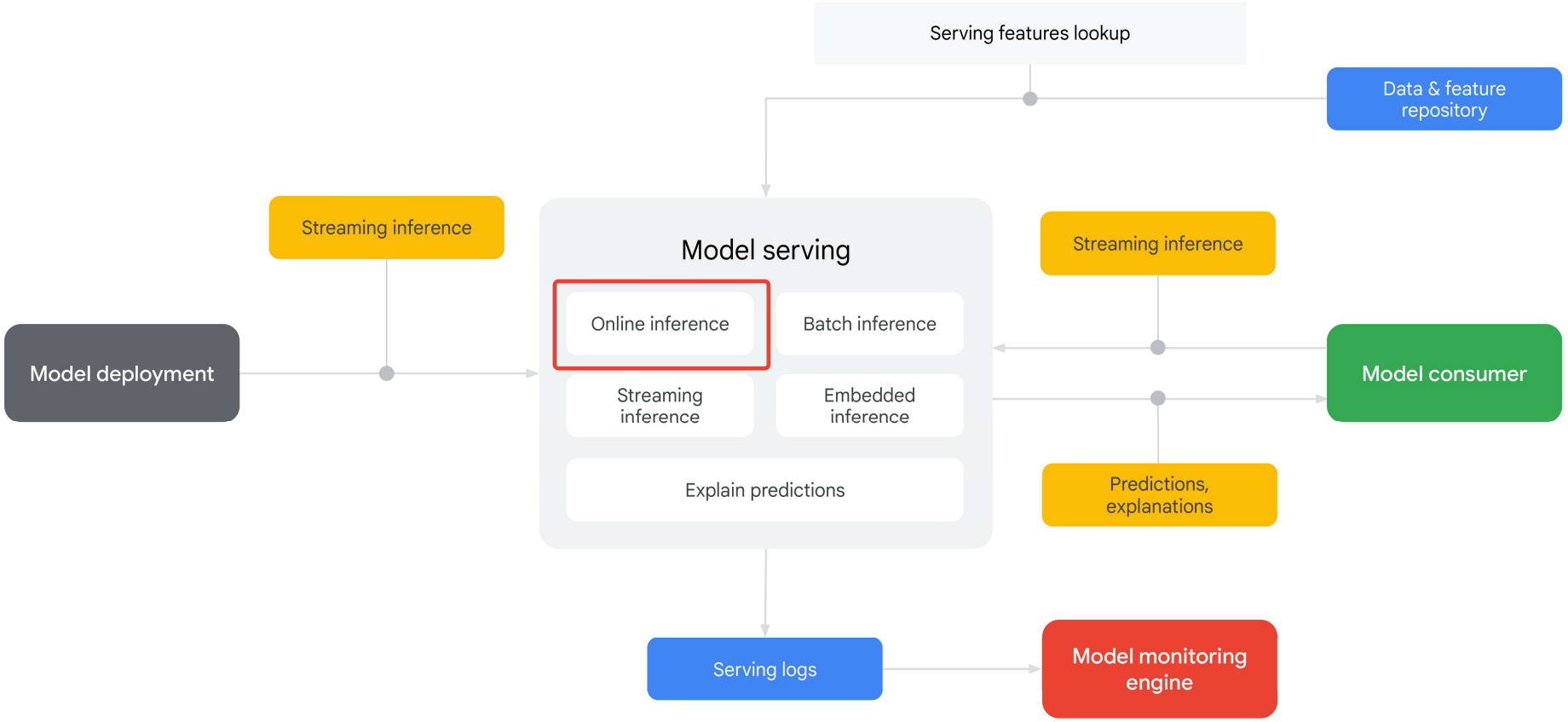


Model deployment

Progressive delivery workflow

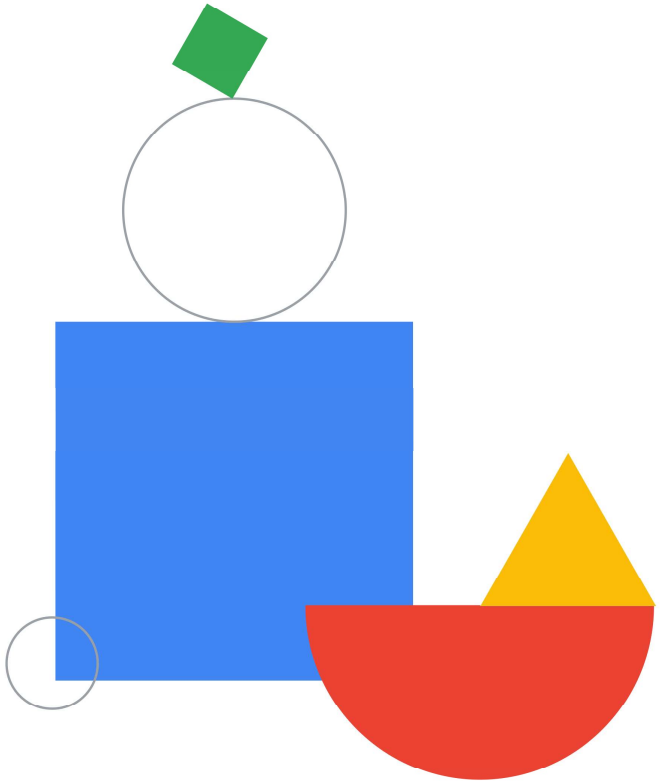


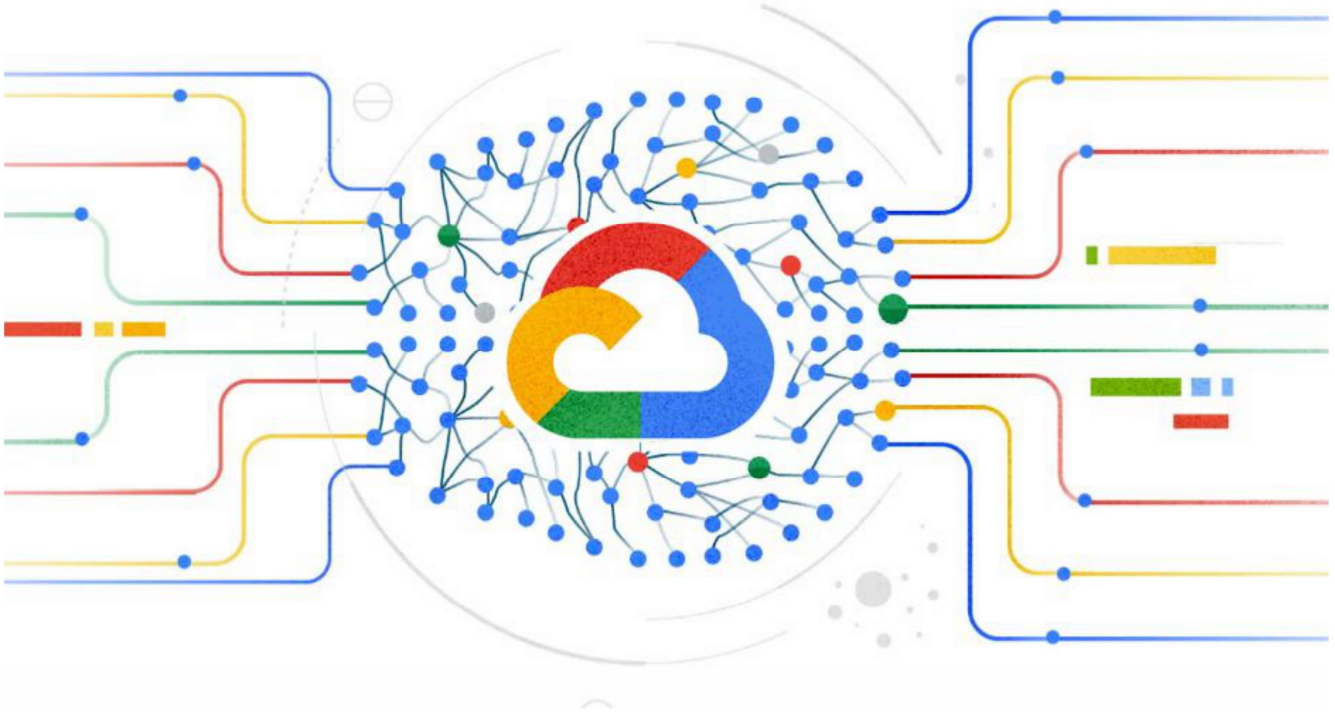
Prediction serving



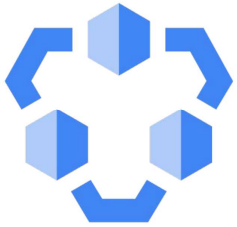
Google Cloud

Data in the Enterprise





Feature Store



Data Catalog

Why do you need a Data Catalog?

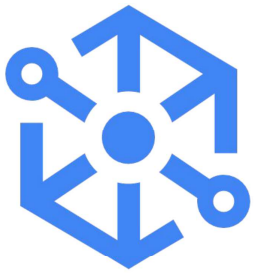
Data stakeholders (consumers, producers, and administrators) within an organization face a number of challenges:

Searching for insightful data:

- Data consumers don't know what data is where. They have to navigate data "swamps" they stumble into.
- Data consumers don't know what data to use to get insights because most data is not well documented and, even if documented, is not well maintained.
- Data can't be found and is often lost when it resides only in people's minds.

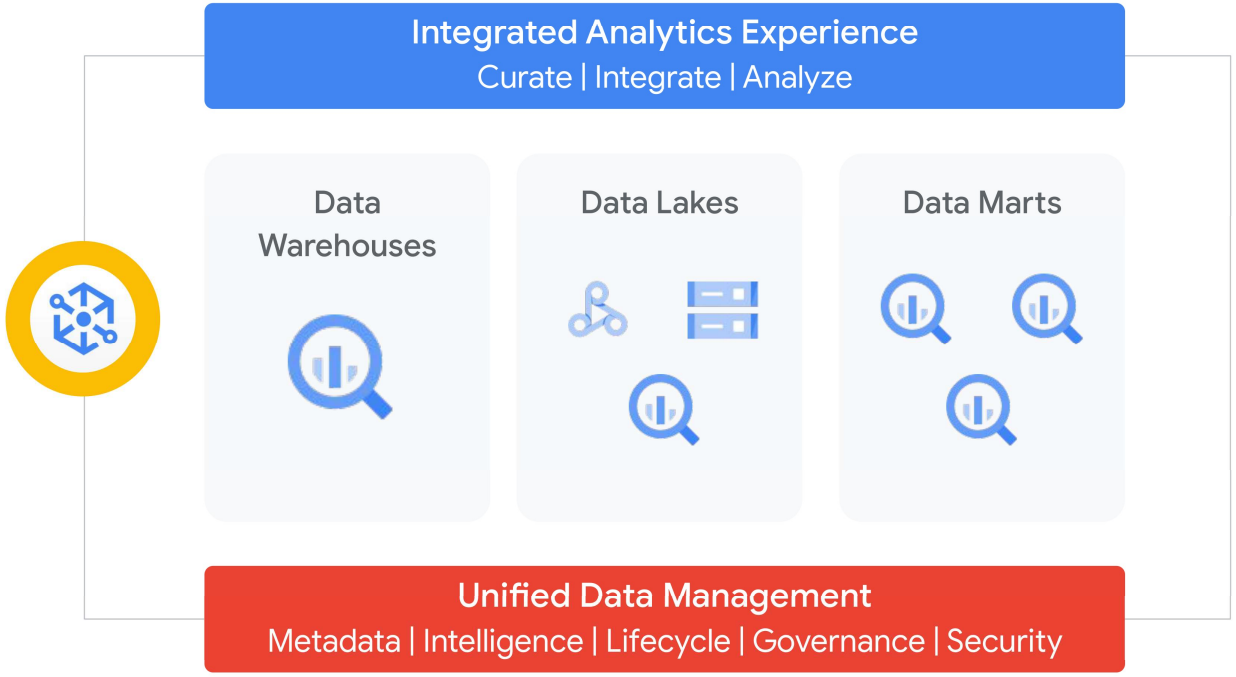
Understanding data:

- Is the data fresh, clean, validated, approved for use in production?
- Which data set out of several duplicate sets is relevant and up-to-date?
- How does one data set relate to another?
- Who is using the data and who is the owner?
- Who and what processes are transforming the data?



Dataplex

Dataplex


Integrated Analytics Experience
Curate | Integrate | Analyze
Data Warehouses
Data Lakes
Data Marts
Unified Data Management
Metadata | Intelligence | Lifecycle | Governance | Security

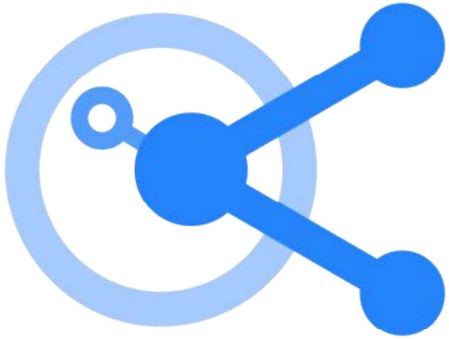
Dataplex enables you to:

- Achieve **freedom** of choice.
- Store data wherever you want.
- Choose the best analytics tools for the job.
- Enforce **consistent controls**.

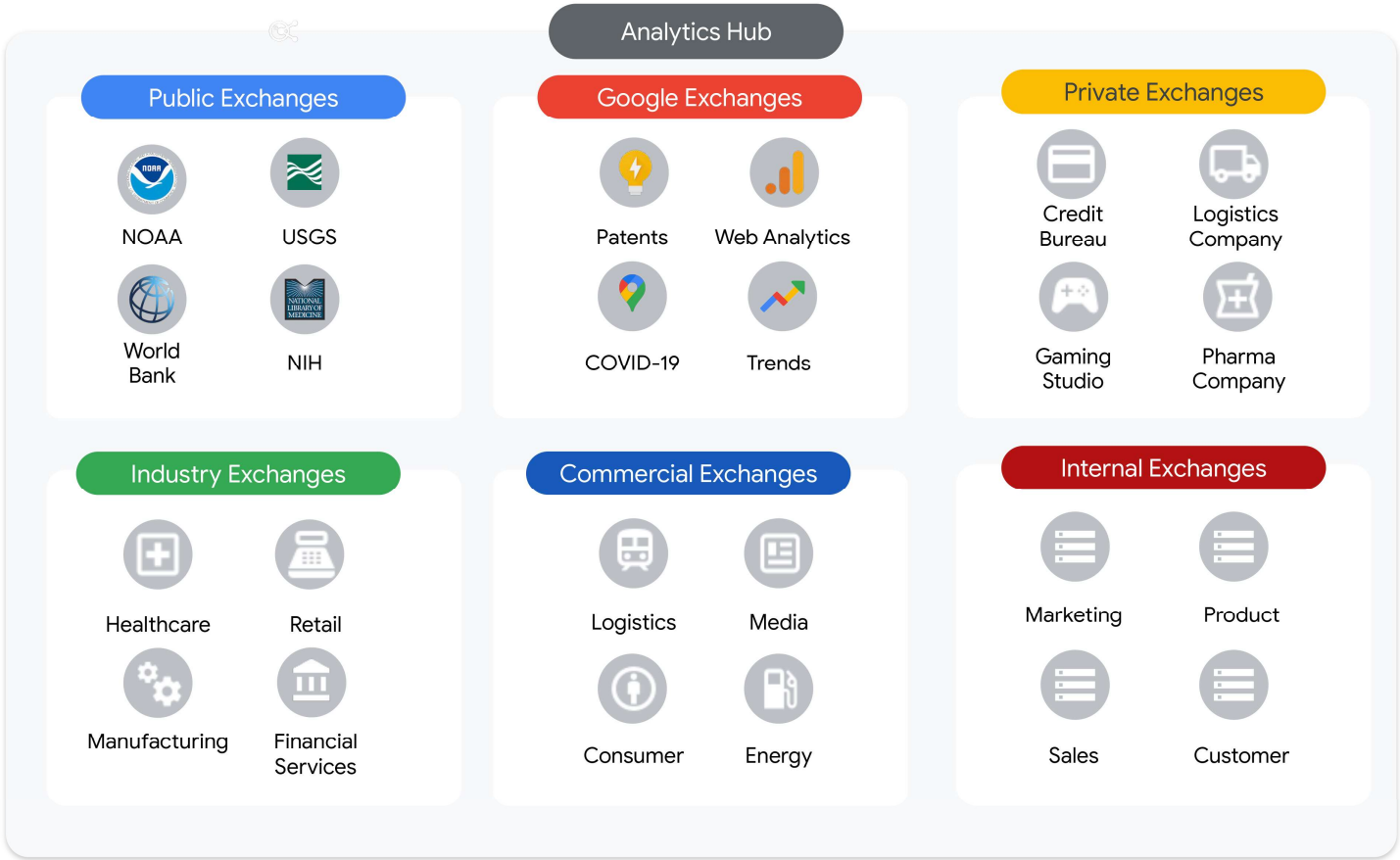
- Use **built-in data intelligence**.
- Automate data management.
- Get access to higher quality data.

Analytics Hub

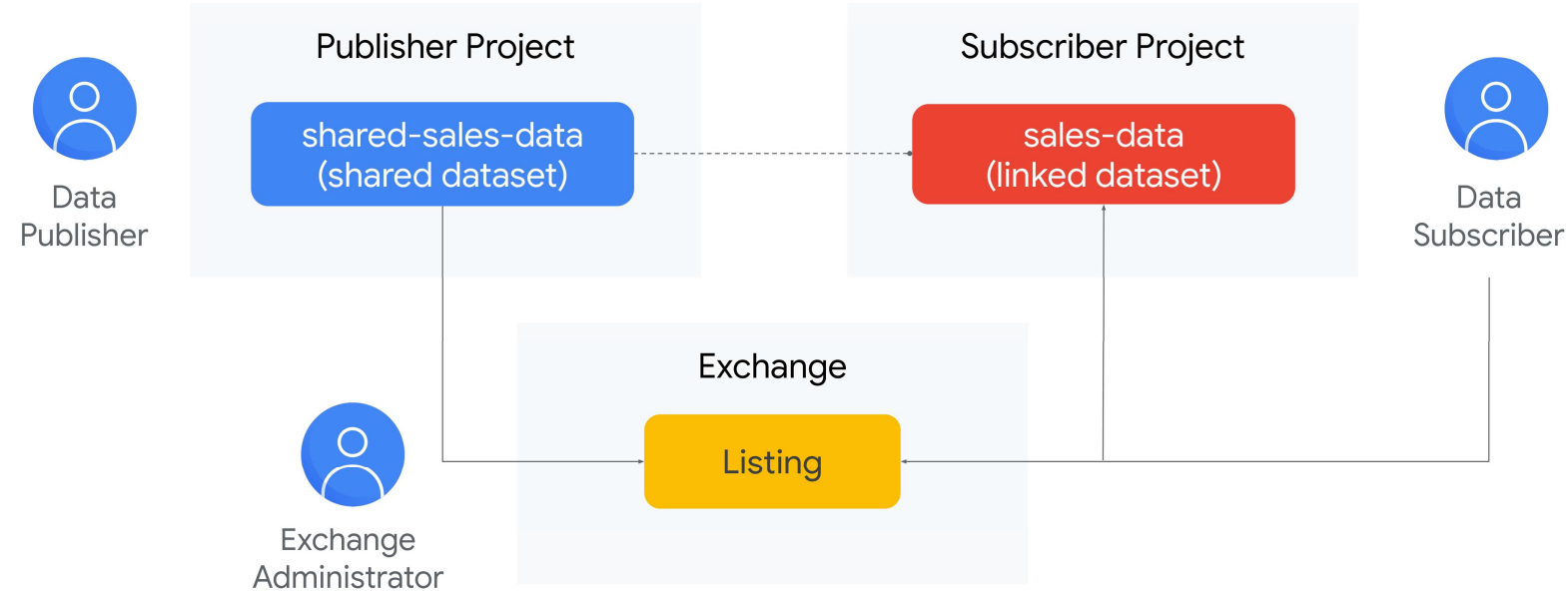
- Drive innovation with datasets from Google, commercial data providers, or your partners
- Exchange data, ML models, or other analytics assets to increase the ROI of data initiatives
- Easily publish or subscribe to shared datasets in an open, secure, and privacy-safe environment



Building a data ecosystem with Analytics Hub



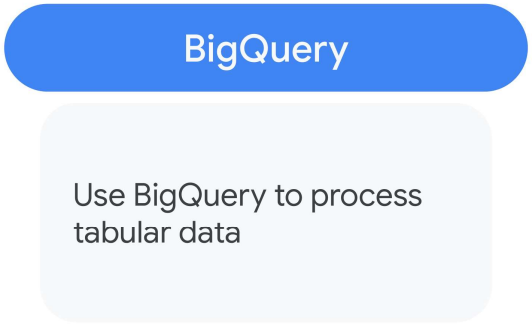
Analytics Hub Walkthrough



For cross-dataset joins, publisher and subscriber datasets must be in same cloud region but can be in different organizations

Data preprocessing options

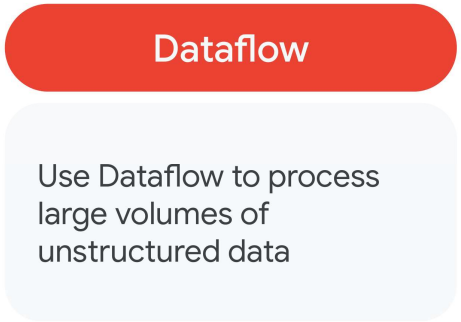
Data Preprocessing with BigQuery



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.

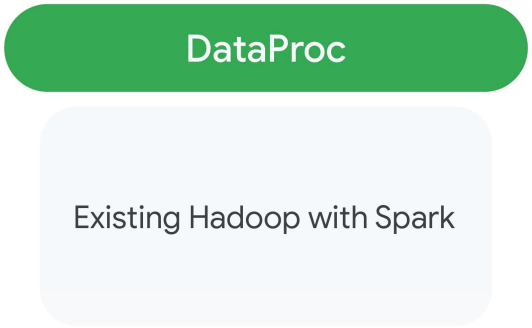
Transforming unstructured data with Dataflow



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.

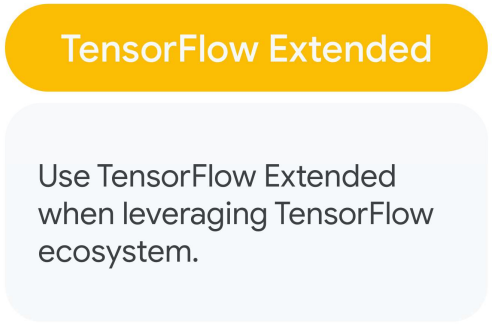
Data Preprocessing with DataProc



Dataproc 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.

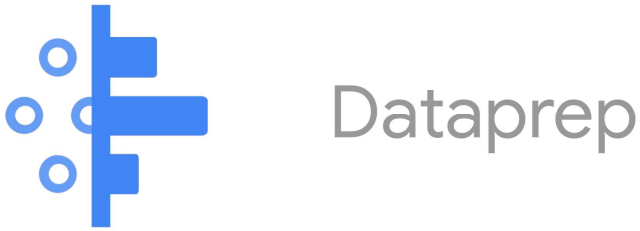
Autoscaling is supported

TensorFlow Extended



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

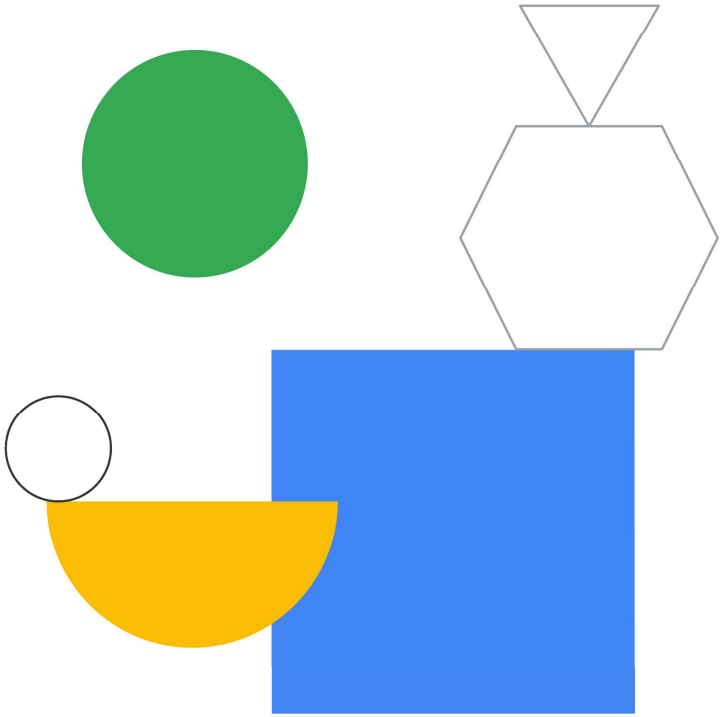
TensorFlow Transform is the TensorFlow component that enables defining and executing a preprocessing function to transform your data.


Dataprep

How does Dataprep fit into Google Cloud?

Google Cloud

Science of Machine Learning and Custom Training



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

Default learning rate in Estimator's LinearRegressor
is smaller of 0.2 or 1/sqrt(num_features) --
this assumes that your feature and label values are small numbers

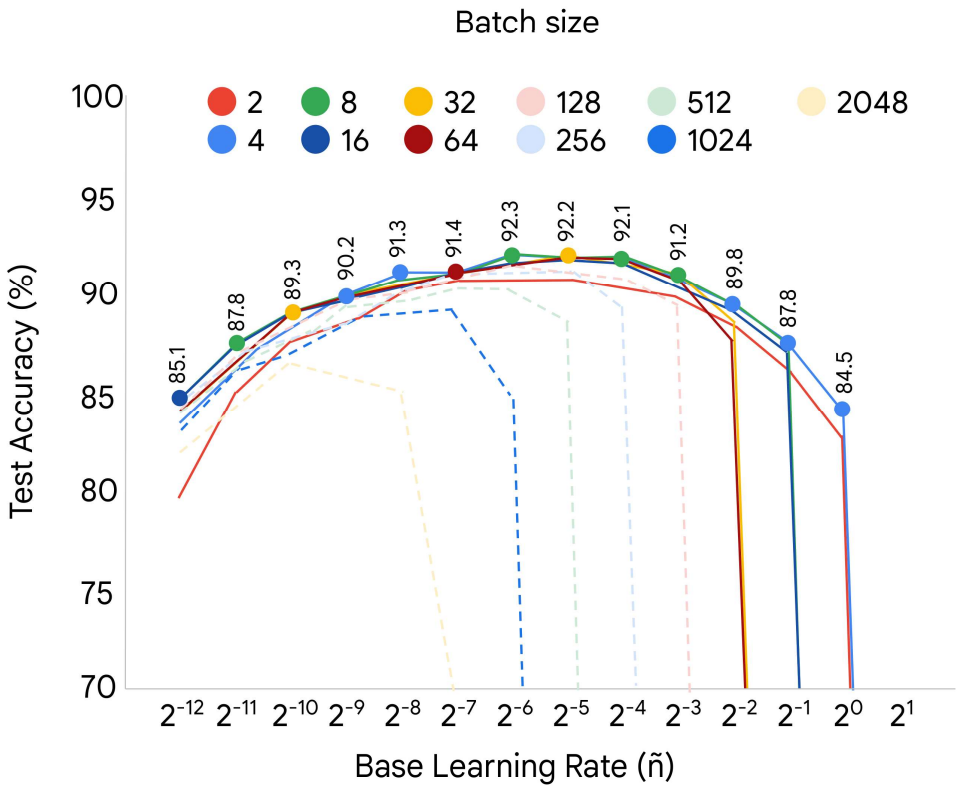
The batch size controls the number of samples that gradient is calculated on.

If too small, training will bounce around

If too large, training will take a very long time

40-100 tends to be a good range for batch size
Can go up to as high as 500

Larger batch sizes require smaller learning rates



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

Compiling a Keras model

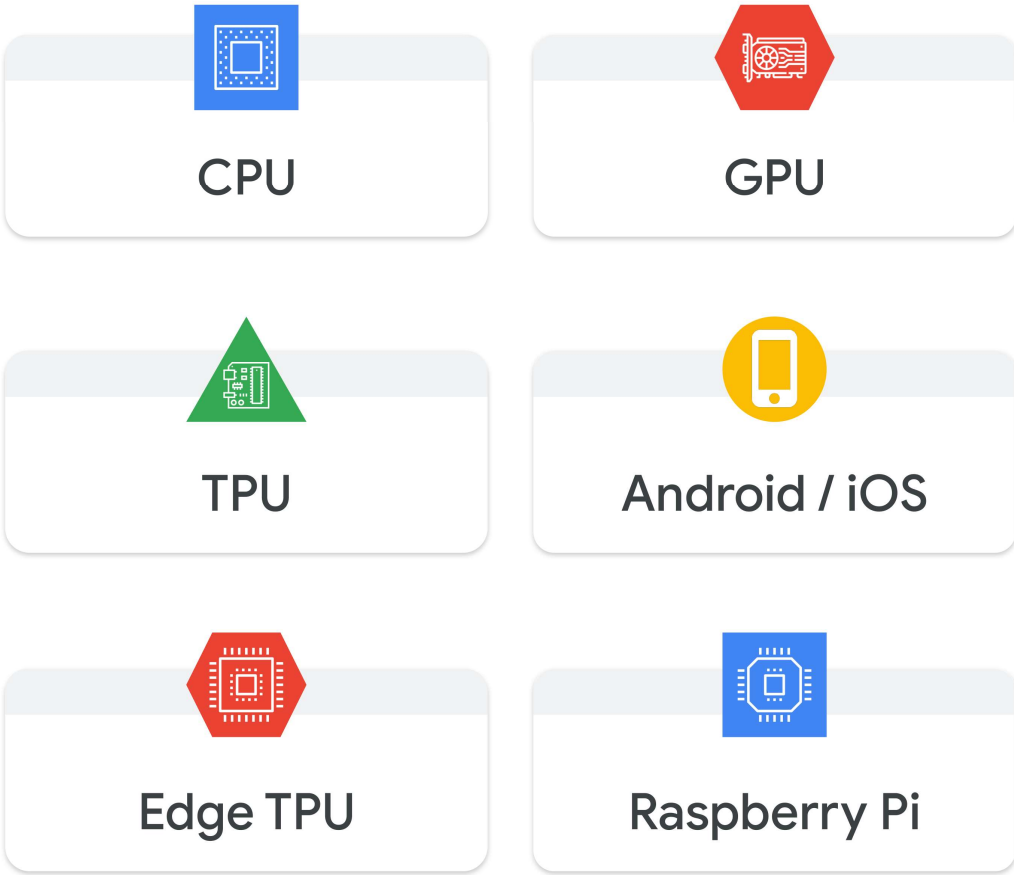
```
def rmse(y_true, y_pred):
    return tf.sqrt(tf.reduce_mean(tf.square(y_pred - y_true)))


model.compile(optimizer="adam", loss="mse", metrics=[rmse, "mse"])
```

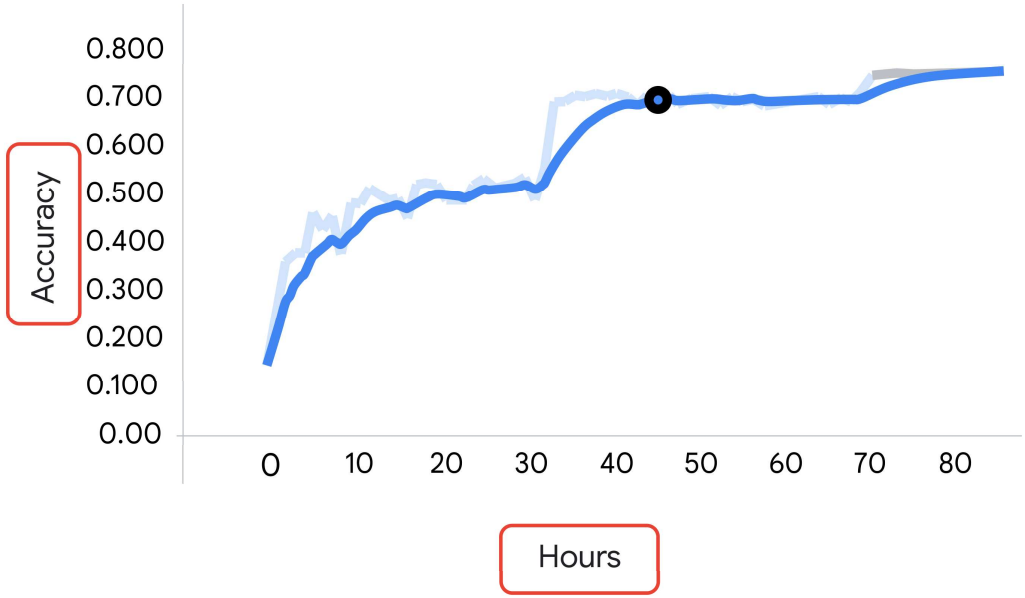
Custom Metric

Compilation adds optimizer, loss, and metrics to a model.

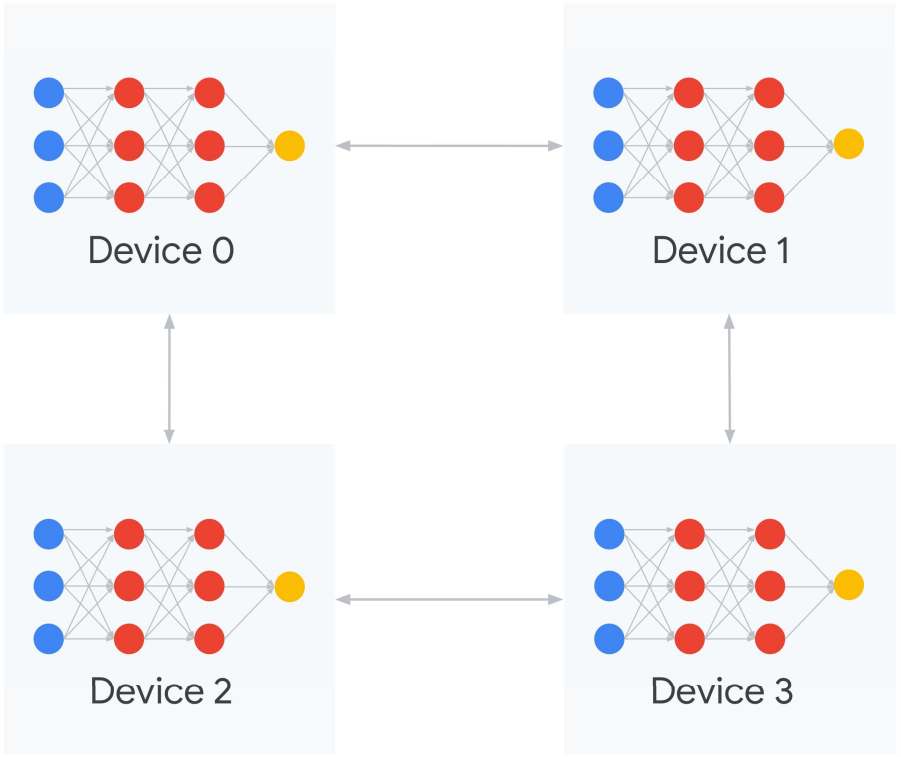
Heterogeneous systems require our code to work anywhere



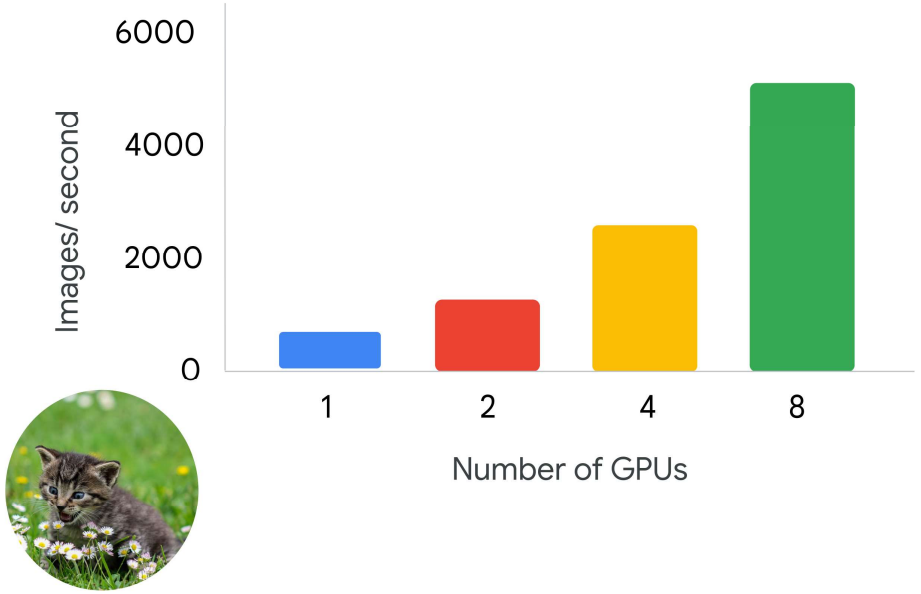
Training can take a long time



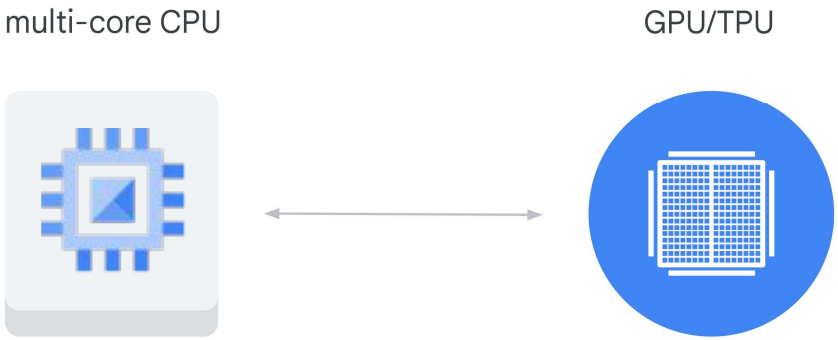
How can you make model training faster?



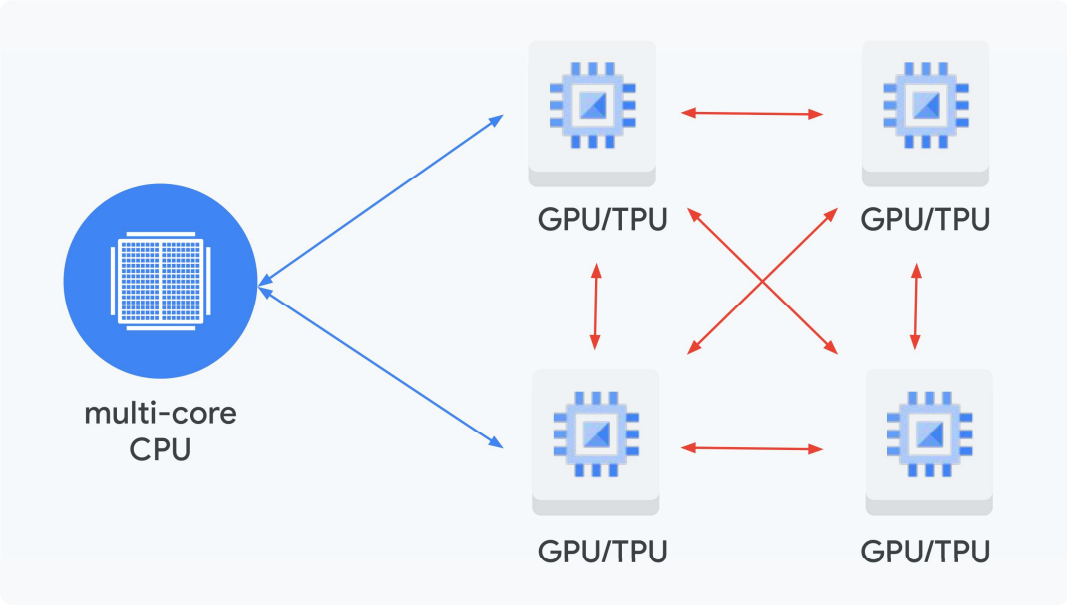
Scaling with distributed training



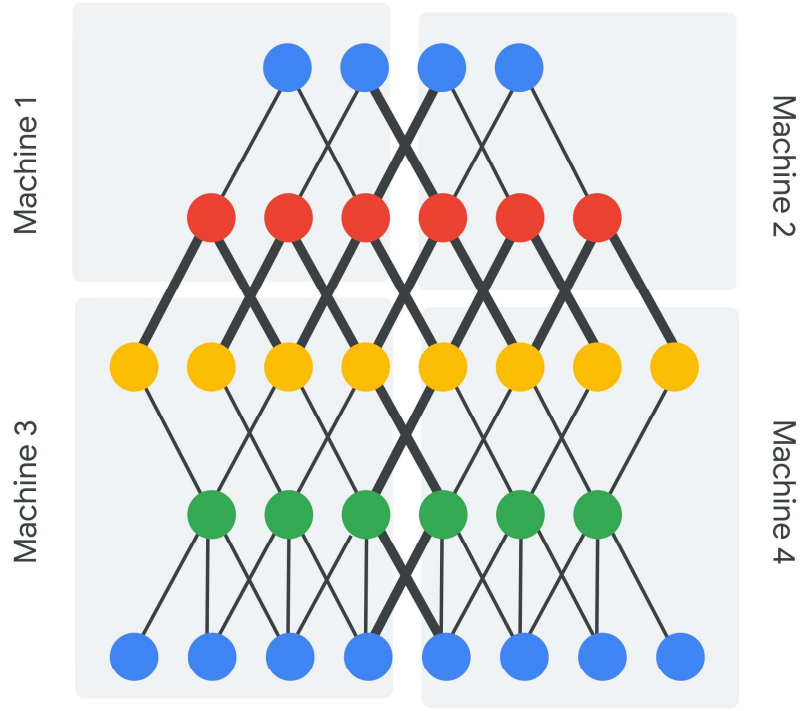
Start training on a machine with a multi-core CPU , then add a single accelerator



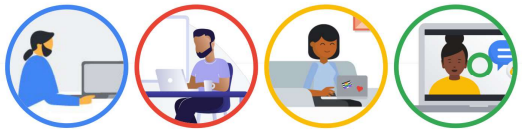
Adding many accelerators to a single device



Model parallelism



Training custom ML models



AutoML

- App developers
- DB admins
- Analysts
- ML/data scientists
- Data engineers

BigQuery ML

- DB admins
- ML/data scientists
- Data engineers

Custom

- ML/data scientists

Benefits of using the Vertex AI custom training service

Local training first

Instead of training your model directly within your notebook instance, you can submit a training job from your notebook.

Modularize architecture

Put your training code into a container to operate as a portable unit.

Cloud logging

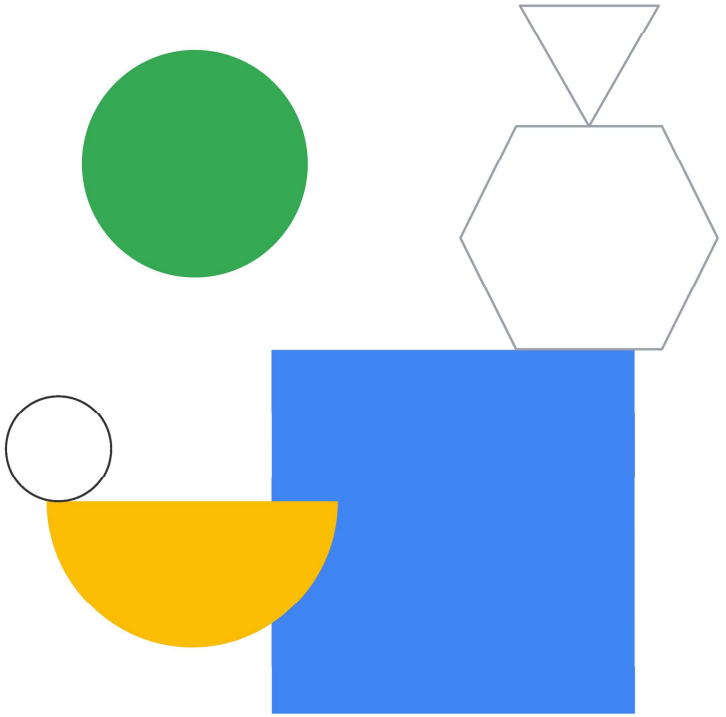
Each training job is tracked with inputs, outputs, and the container image used.

Distributed training

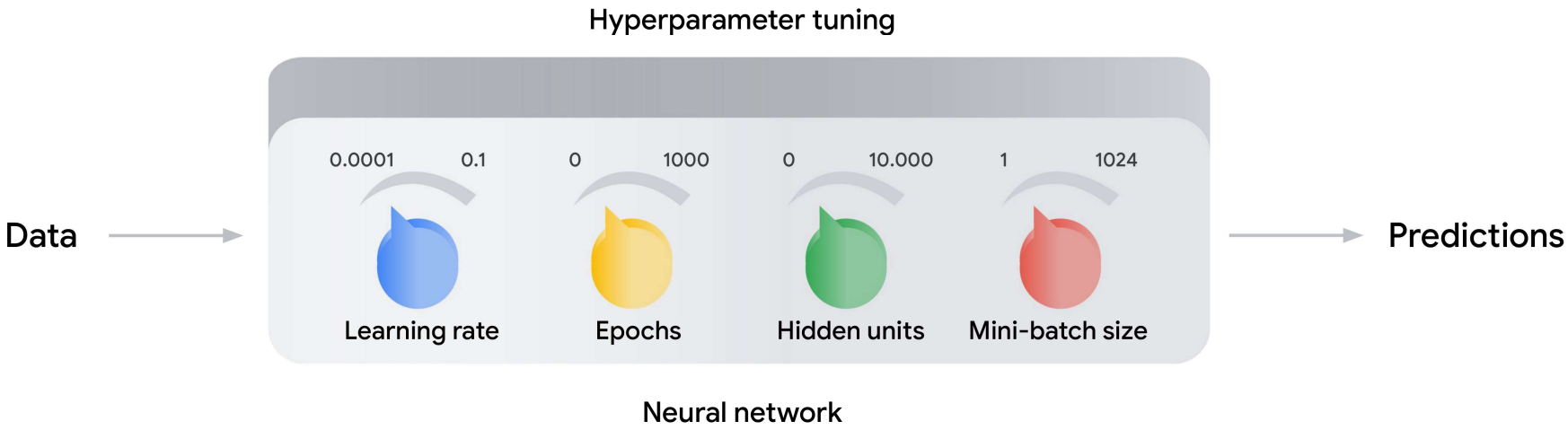
You can train models across multiple nodes in parallel.

Google Cloud

Vertex Vizier Hyperparameter Tuning



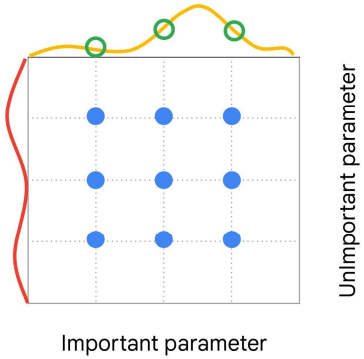
Hyperparameter tuning



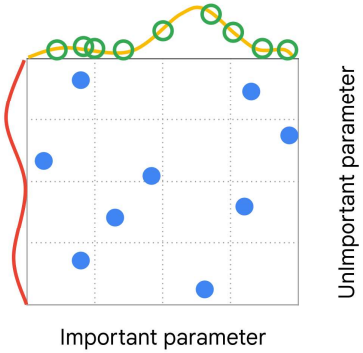
The hyperparameters are knobs that act as the network-human interface.

Grid and Random Search

Grid layout



Random layout



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

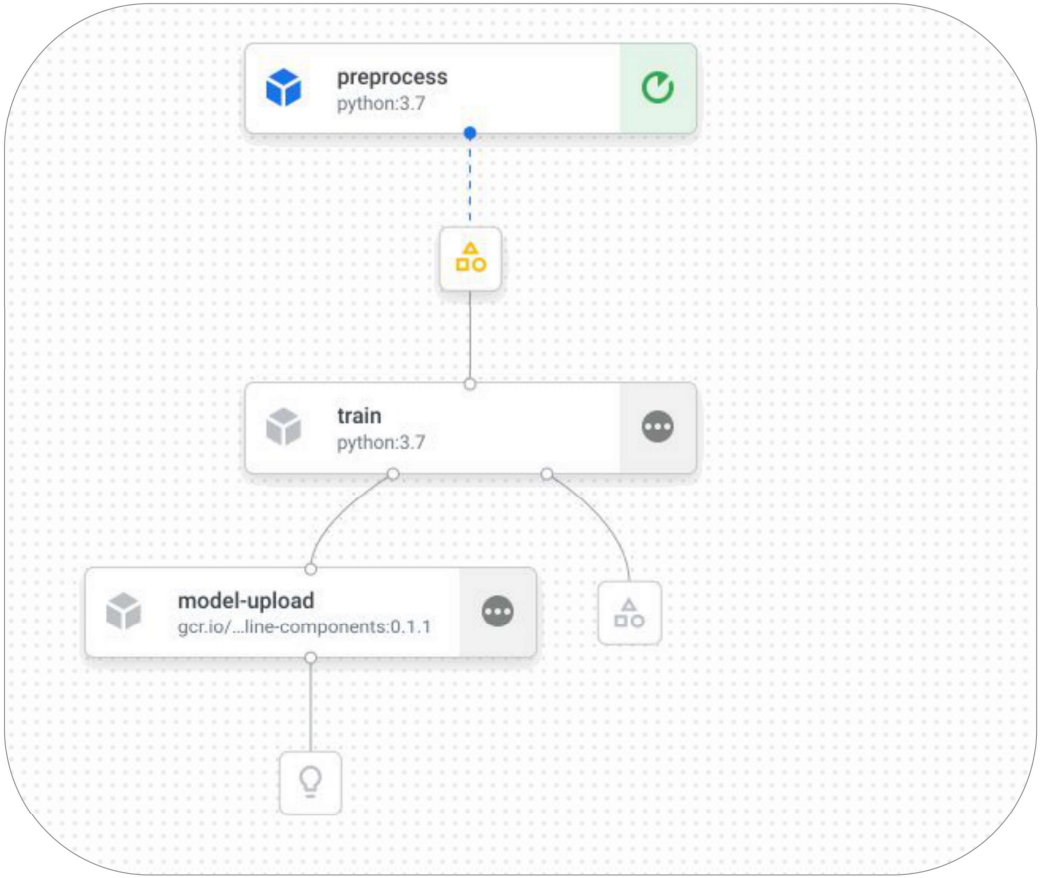
Vertex AI pipeline

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

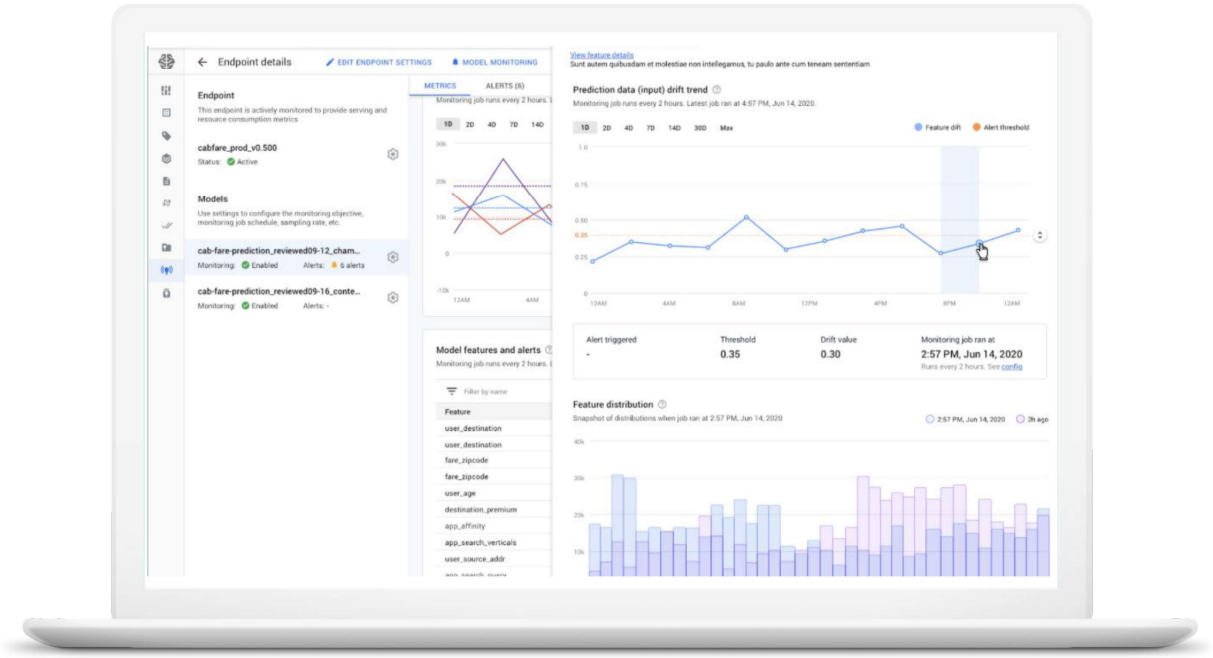
Describe workflow as a pipeline

- 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.

Best practices:
Model deployment and serving



Machine type

Specify the number and type of machines you need.

Model inputs

Plan inputs to the model.

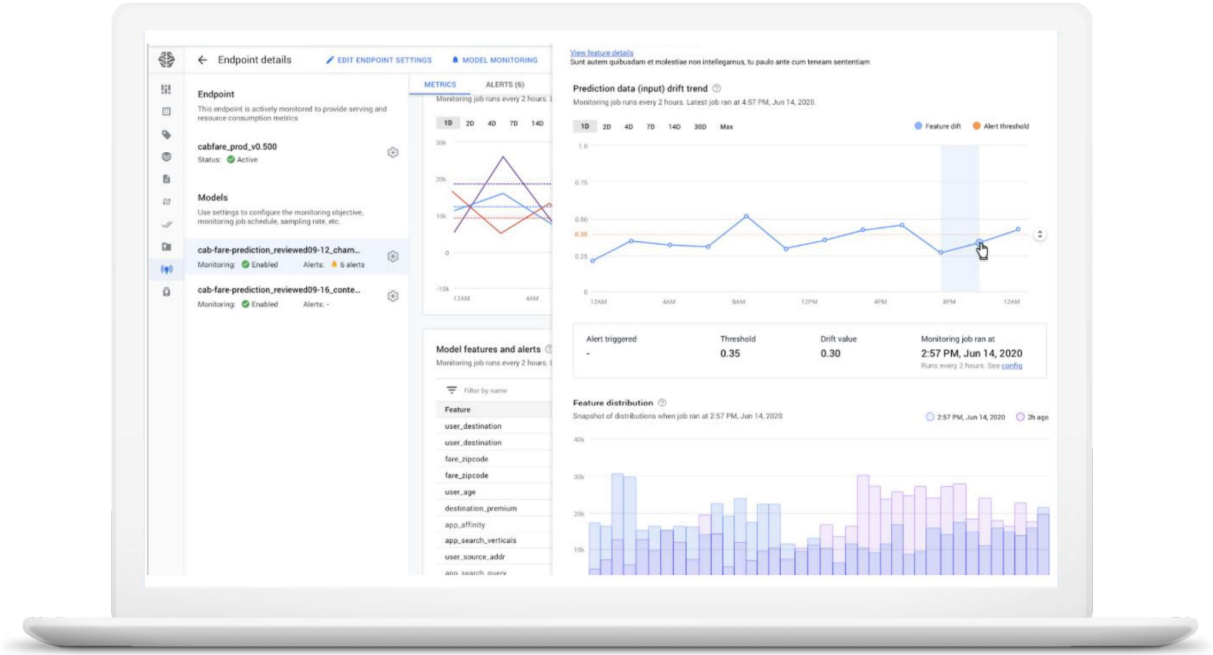
Automatic scaling

Turn on automatic scaling.

Specify performance requirements

Define what is good and bad performance.

Best practices: Model monitoring



Skew

Use skew detection.

Data drift

Use feature attributions to detect data drift.

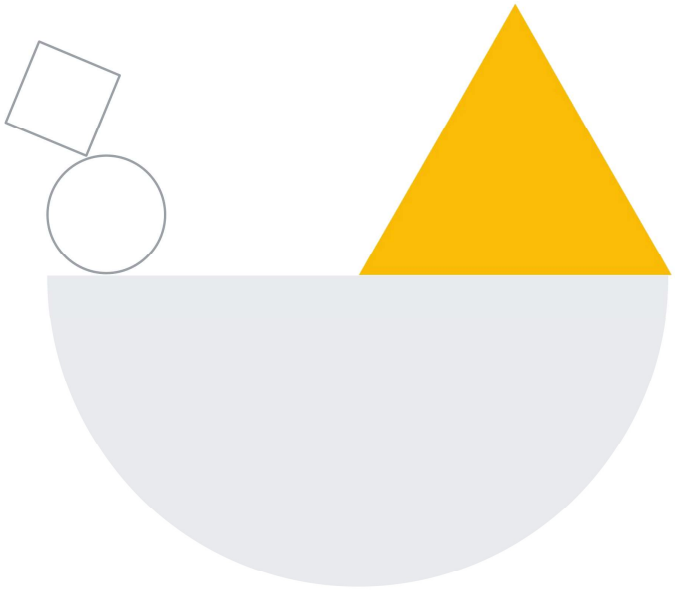
Alert thresholds

Fine tune alert thresholds.

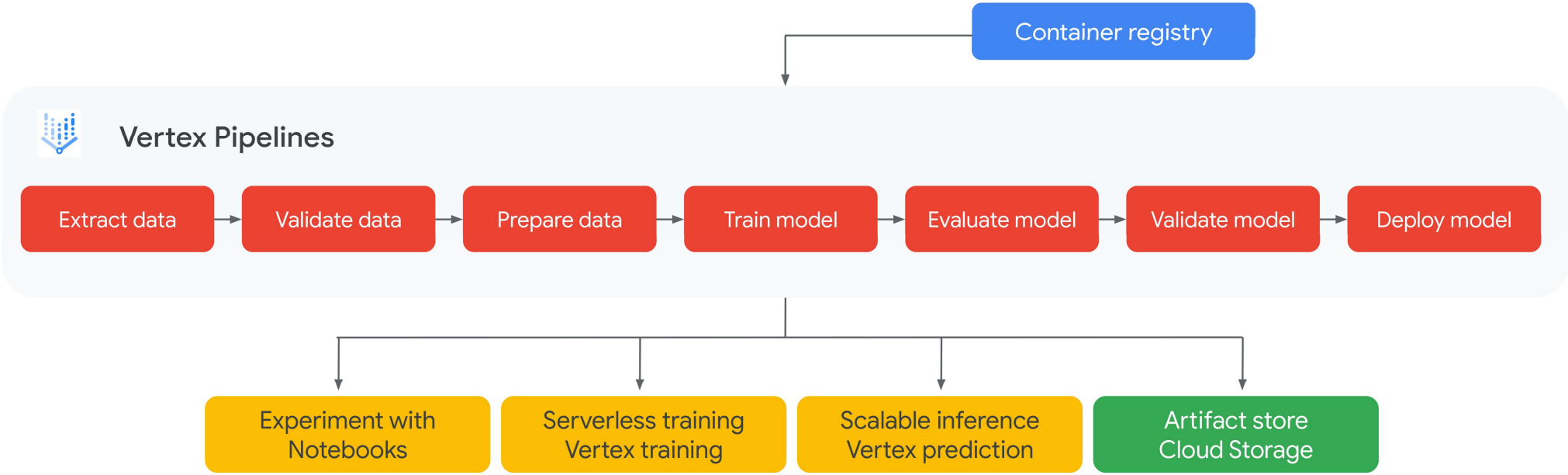
Model inputs

Track model inputs.

Vertex AI Pipeline best practices

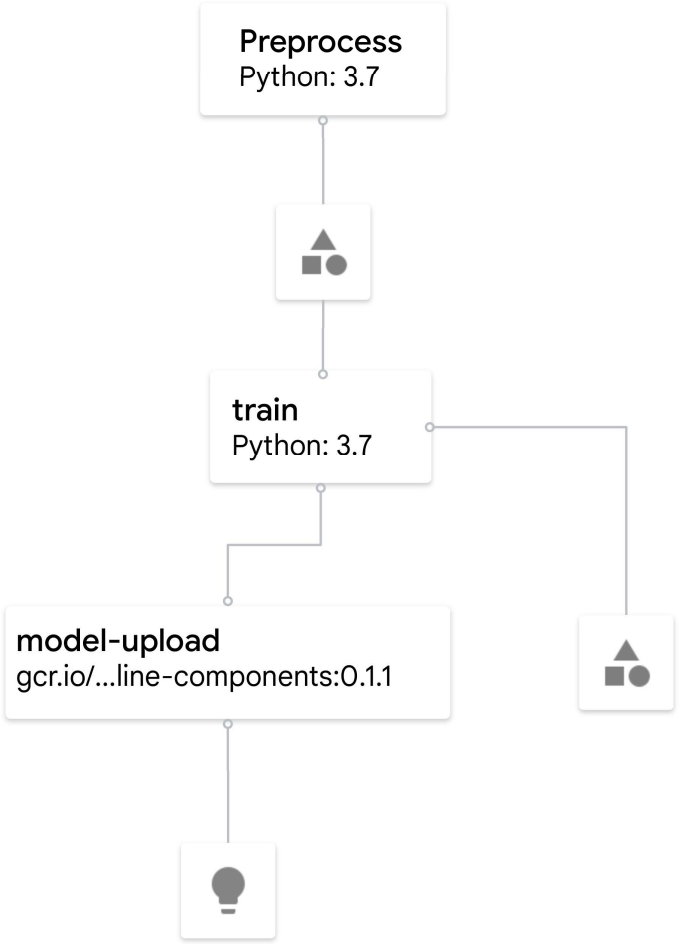


Pipelines automate the training and deployment of models

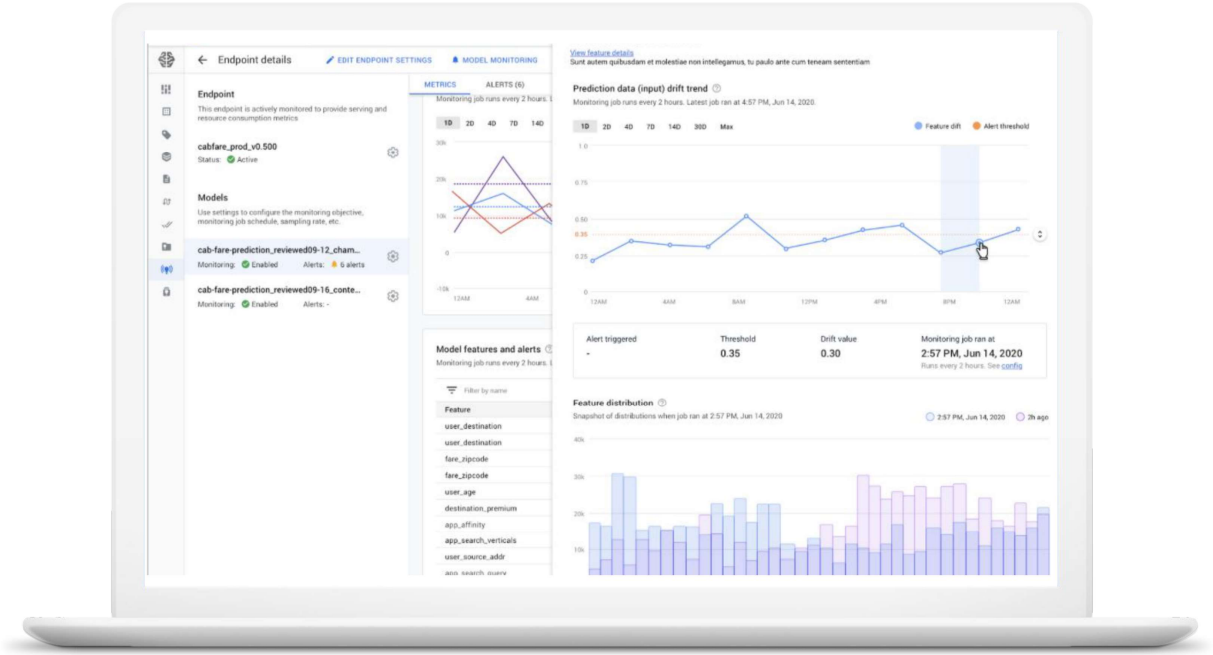


Pipeline

- A pipeline is composed of modular components
- A pipeline offers automation and orchestration
- Components are chained with dsl to form a pipeline



Best practices: Vertex AI Pipelines



Assess perfection

Why did a pipeline produce an especially accurate model?

Compare pipelines

Which pipeline run produced the most accurate model and parameters used?

System governance

Which version of your model is in production at a given time?

Pipeline SDK

Kubeflow SDK
TensorFlow Extended

Best practices: Artifact organization

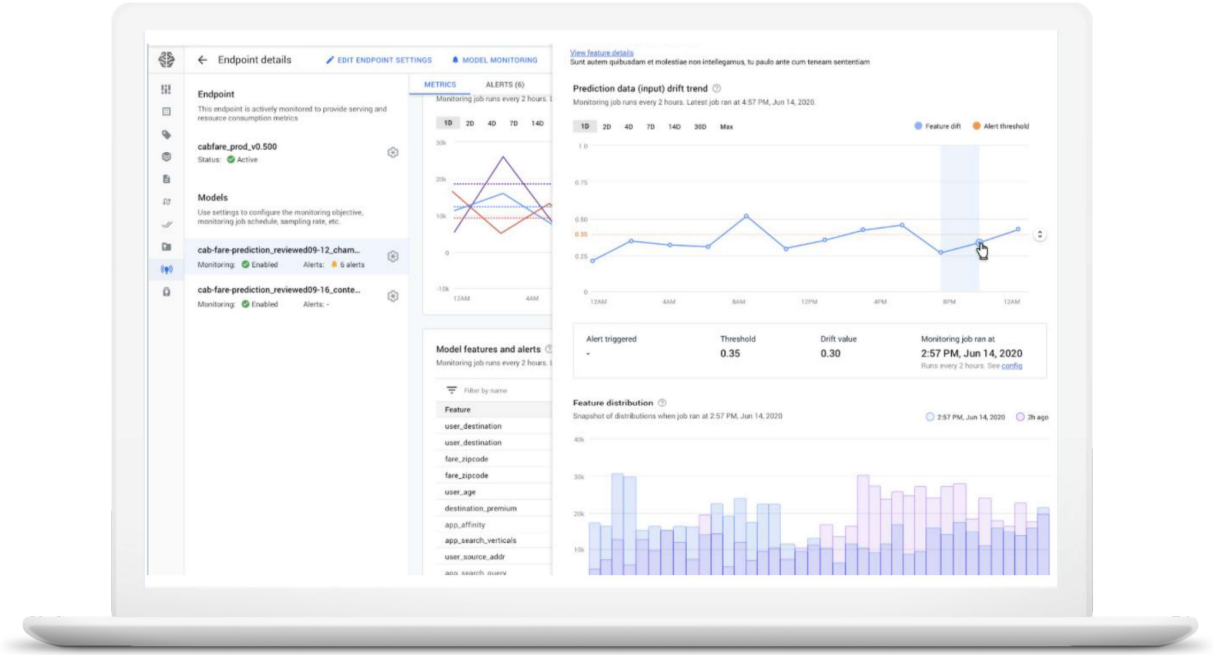
Artifact lineage describes all the factors that resulted in an artifact.

You can understand differences in performance or accuracy over several pipeline runs.

A model's lineage could include the following:

- The training, test, and evaluation data used to create the model.
- The hyperparameters used during model training.
- The code that was used to train the model.
- Metadata recorded from the training and evaluation process.
- Artifacts that descend from this model.

Best practices: Artifact organization



Artifacts are outputs resulting from each step in the ML workflow.

Artifacts

Organize your
ML model artifacts.

Git repo

Use a Git repo for pipeline
definitions and training code.

Thanks