Google Cloud

How Google Does Machine Learning

In this course, you learn to ...

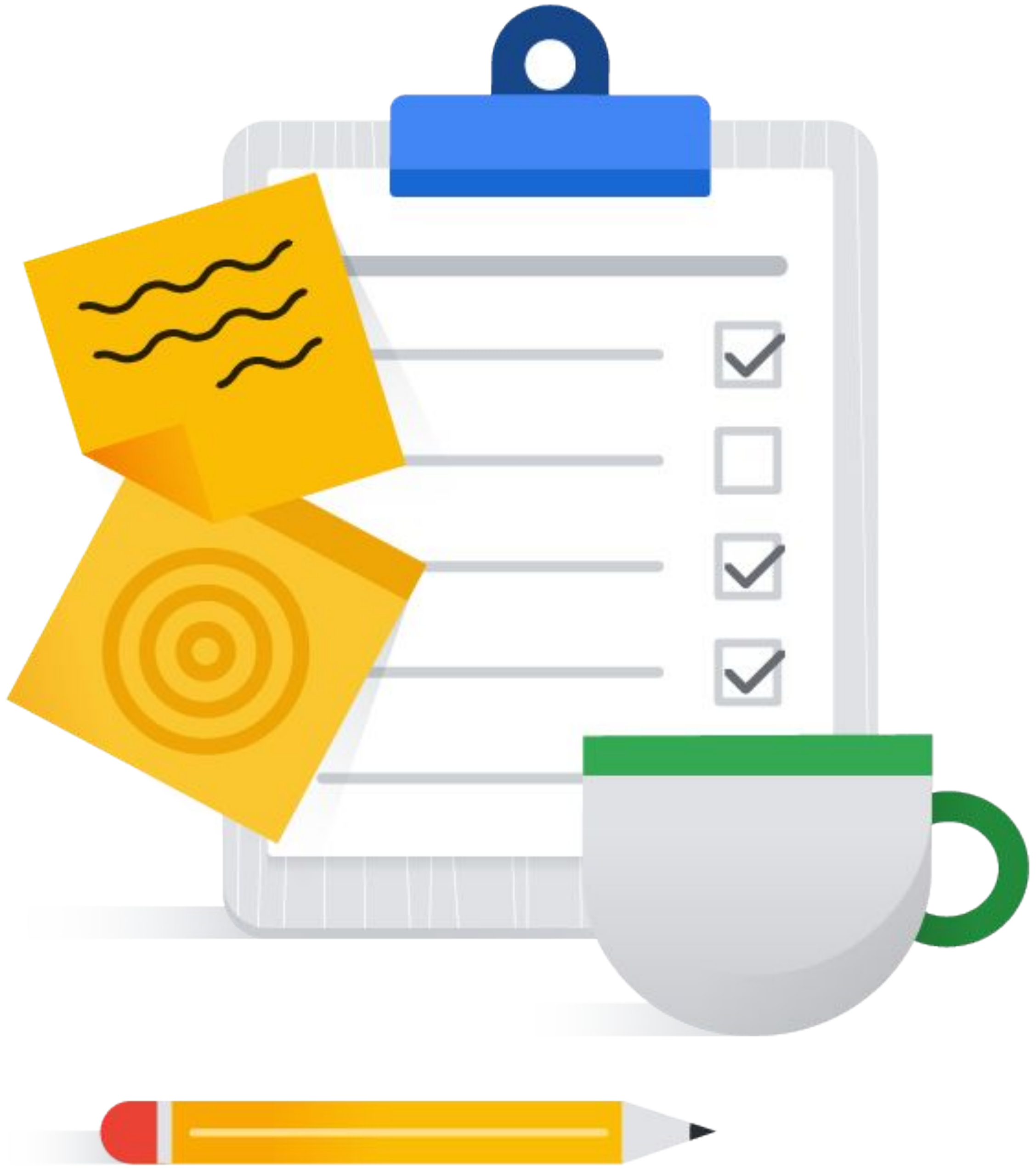
01 Understand what it means to be AI first and how Google does Machine Learning.

02 Leverage Vertex AI to do machine learning - from AutoML to Custom Training.

03 Describe Best Practices for Machine Learning and Responsible AI development

04 Gain a broad perspective on machine learning and where it can be used.

05 Frame a business use case as a machine learning problem.



Practical, real-world introduction to ML

Data Analysts
Citizen Data Scientists

Build without writing a single piece of code

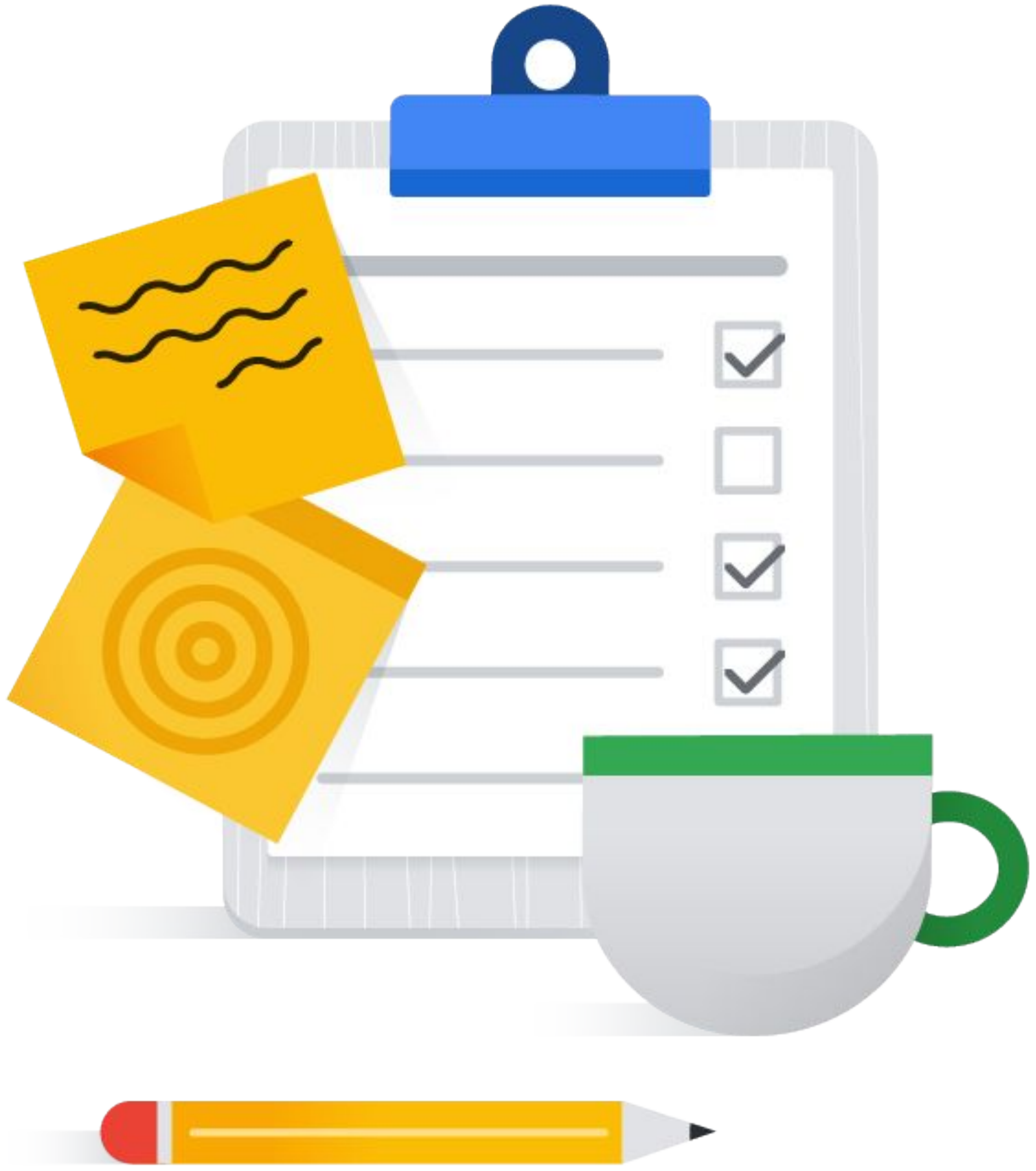
ML Engineers
ML Scientists

Go "code deep" with custom training

What It Means to be AI-First

In this module, you learn to ...

01	Build a data strategy around ML
02	Identify and solve ML problems
03	Infuse your apps with ML

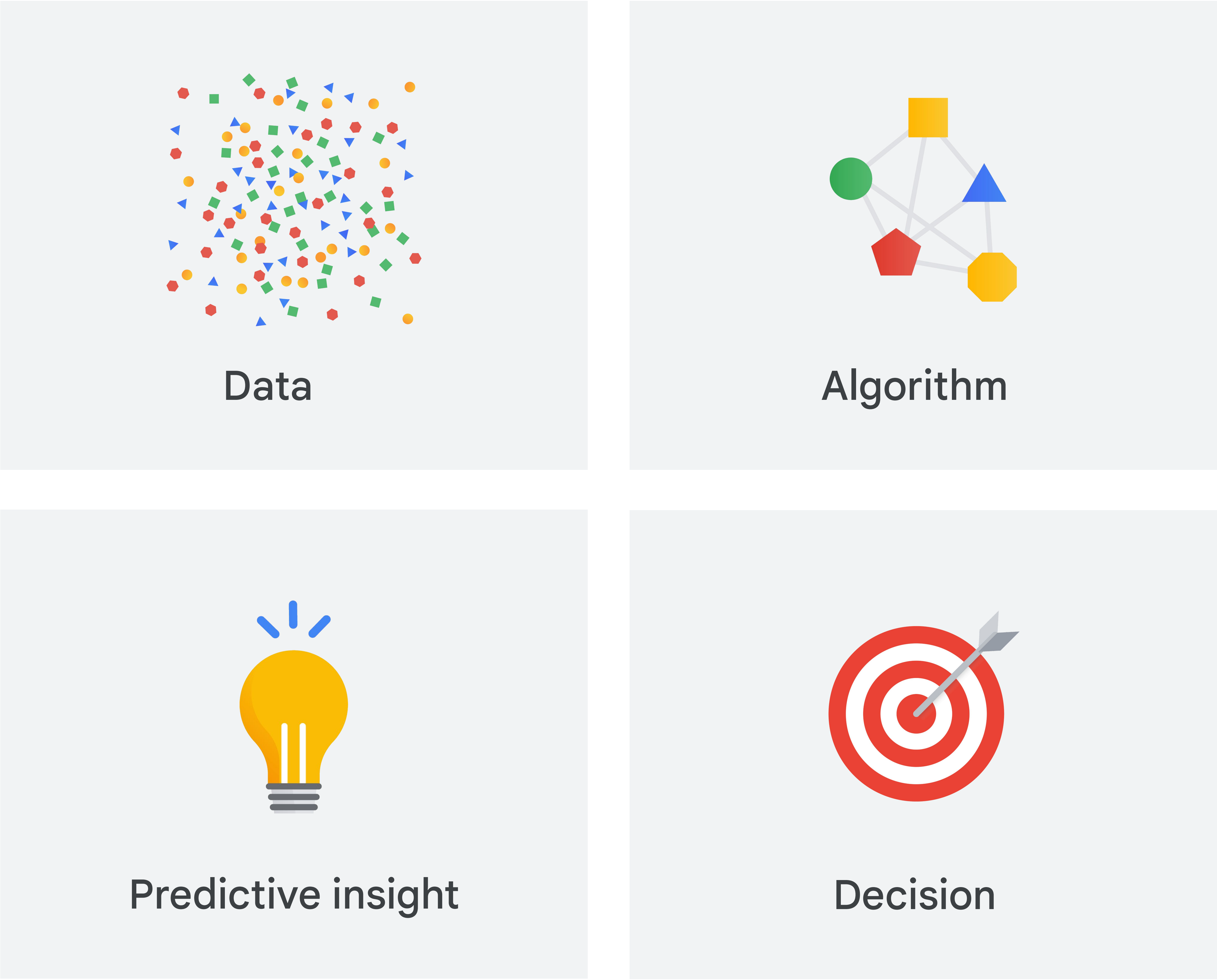


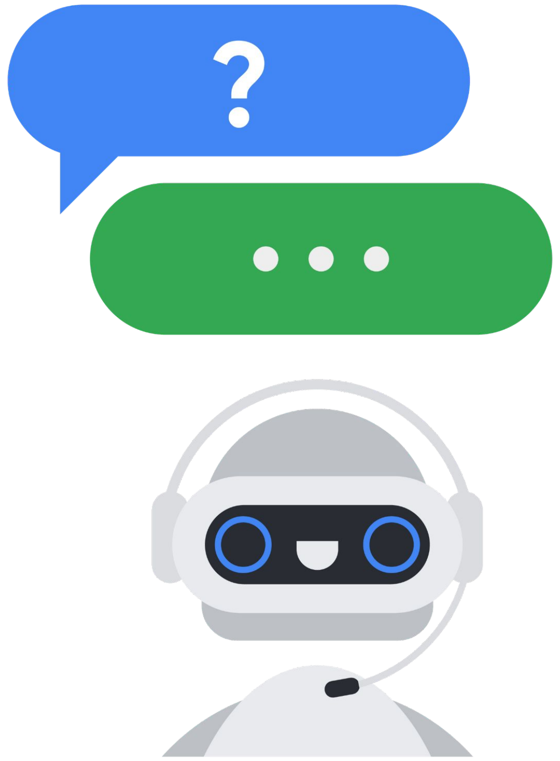
What is machine learning?

What does it mean to be AI-first?

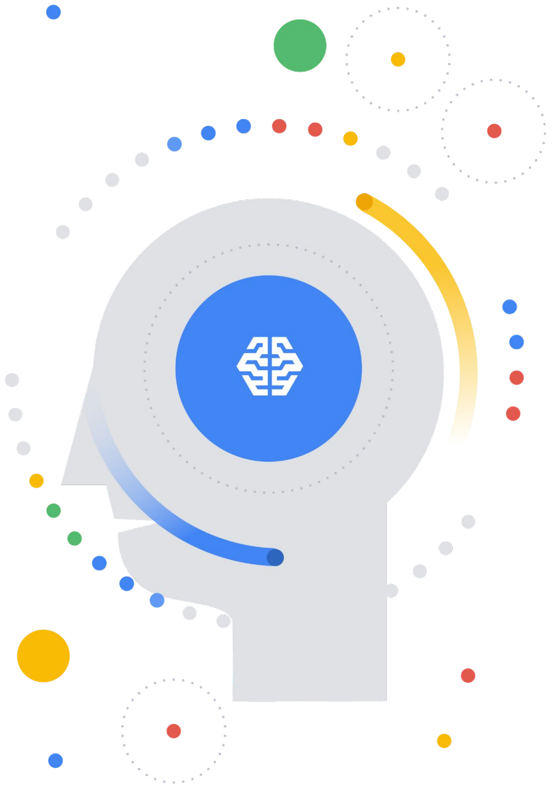
What kinds of problems can ML solve?

Machine learning is a way to use standard algorithms to derive predictive insights from data and make repeated decisions.

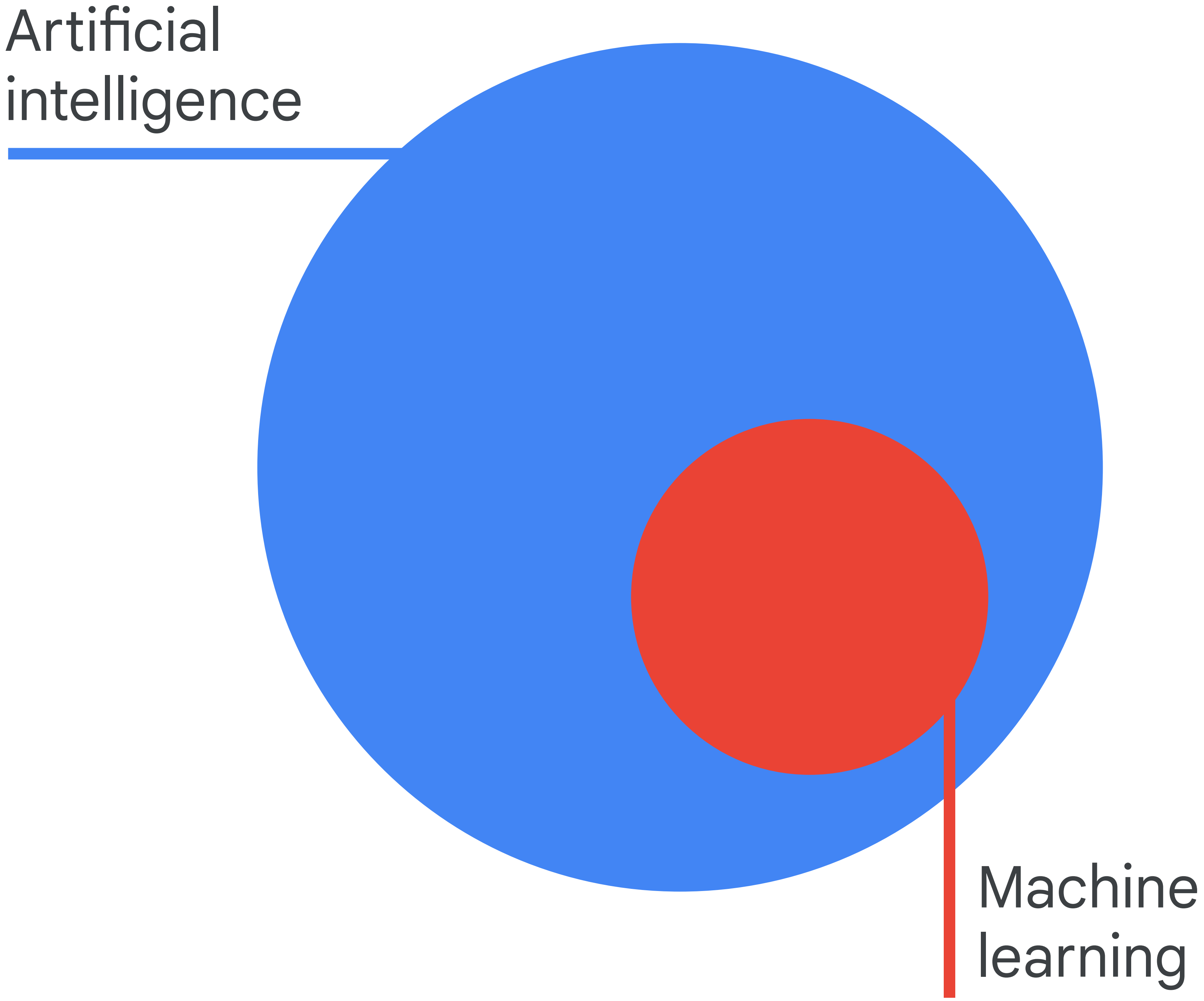




versus



Artificial Intelligence is a discipline; machine learning is a specific way of solving AI problems.



Stage 1: Train an ML model with examples

“cat” 

“dog” 

“car” 

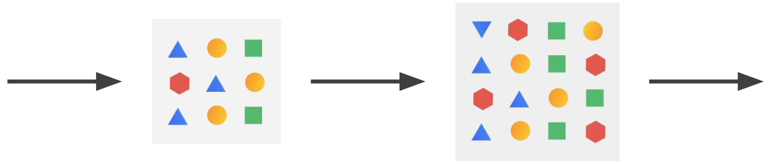
“apple” 

label, input

Stage 1: Train an ML model with examples

“cat”

“dog”

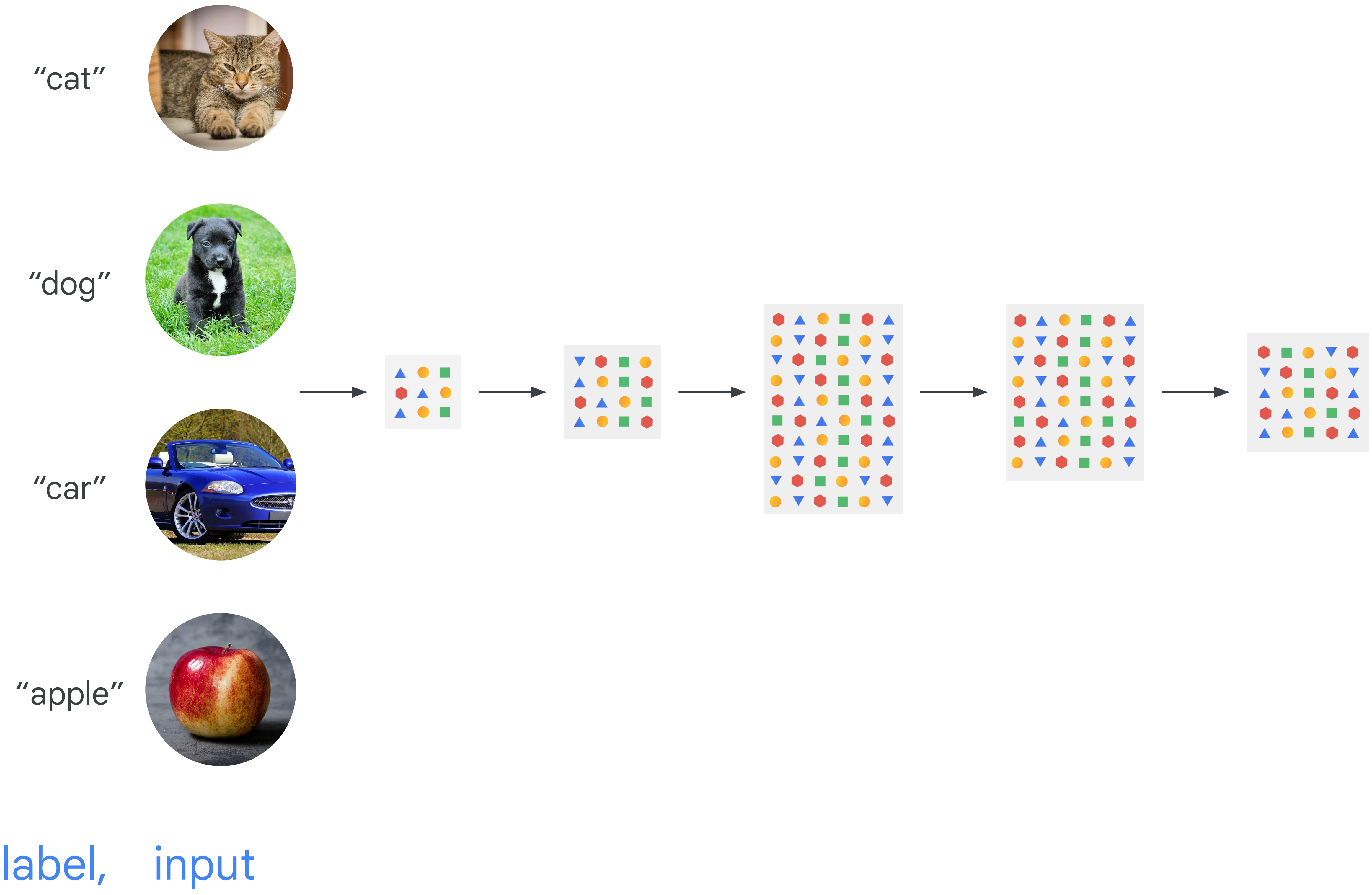


“car”

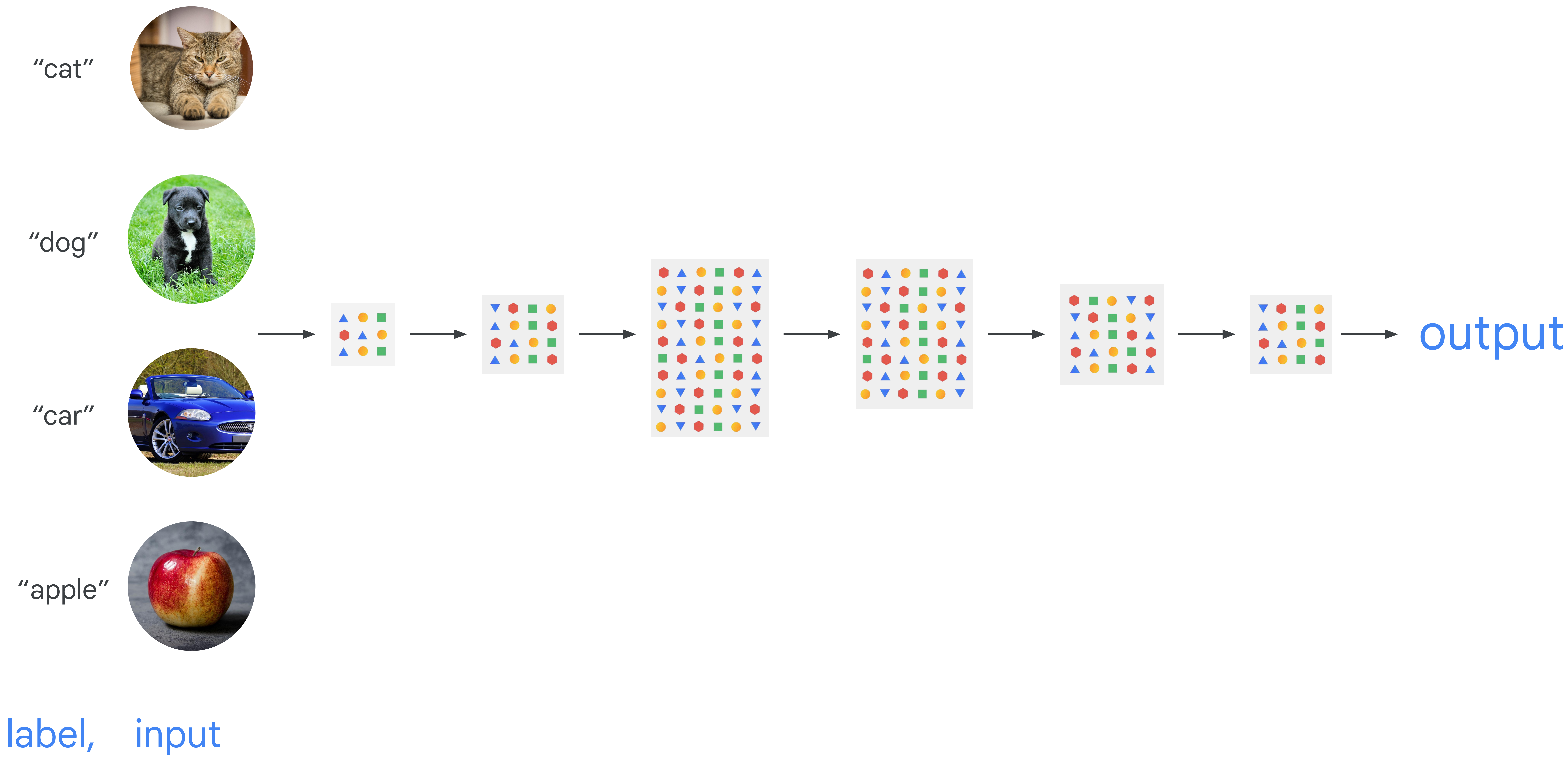
“apple”

label, input

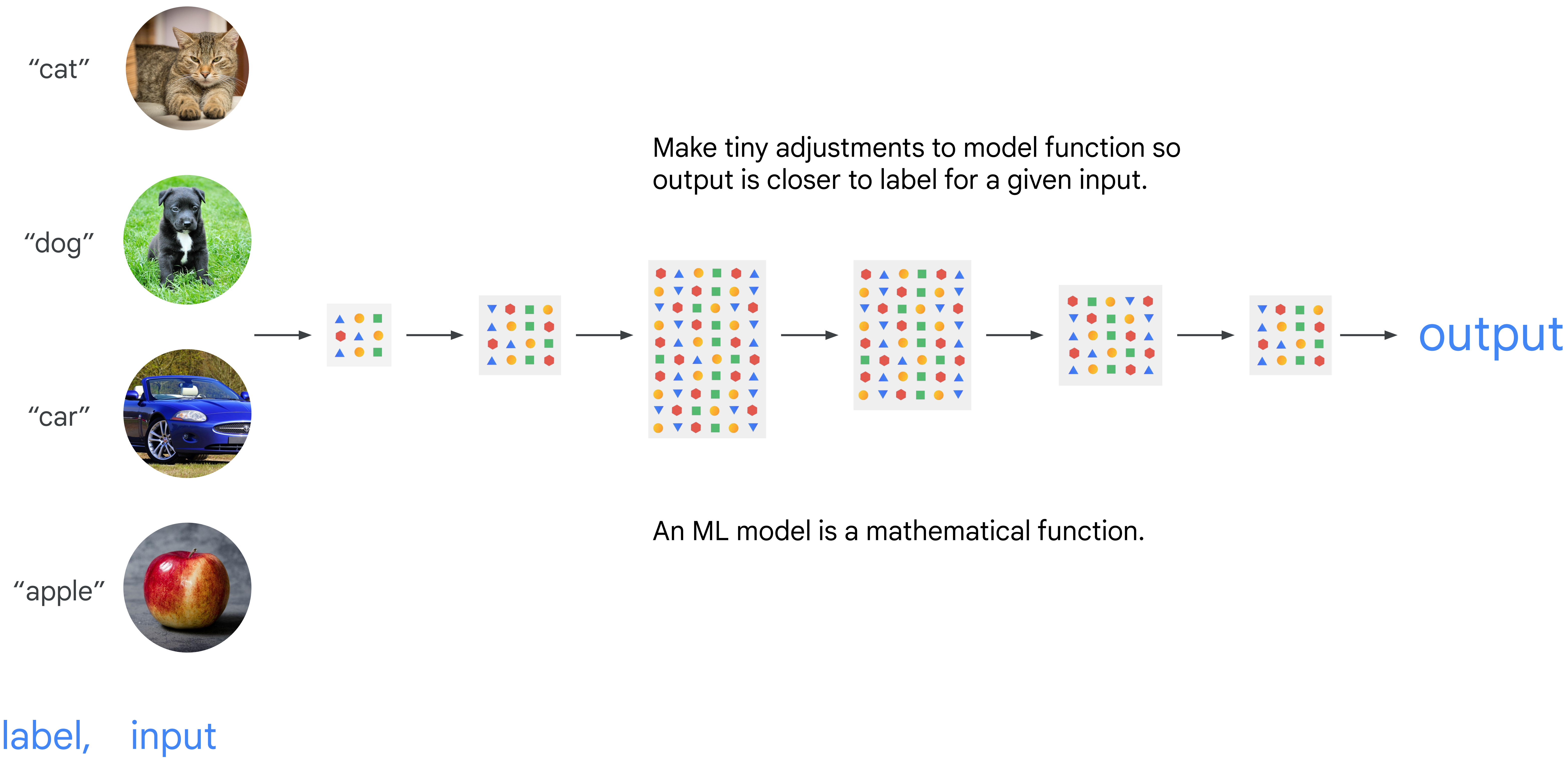
Stage 1: Train an ML model with examples



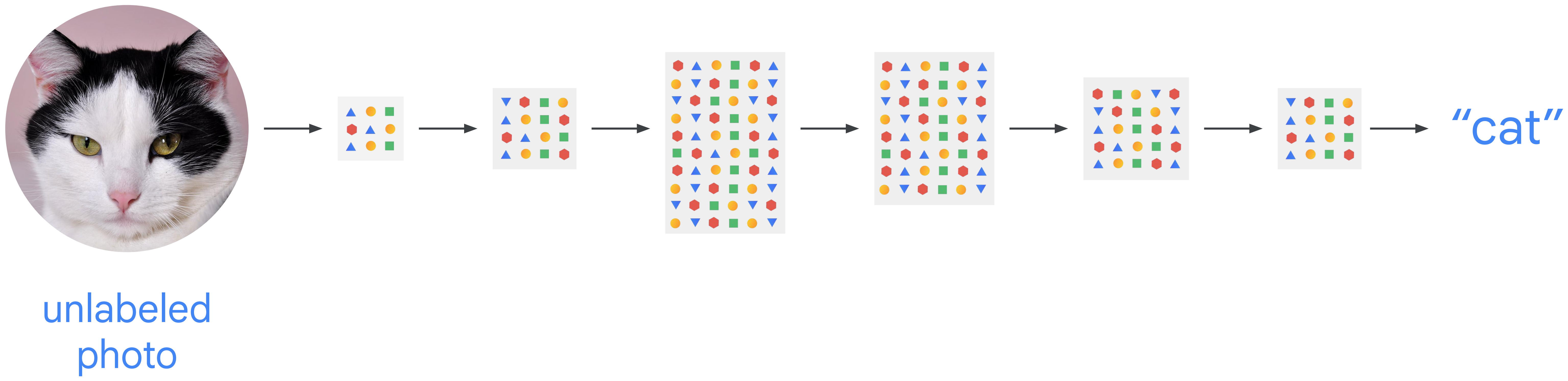
Stage 1: Train an ML model with examples

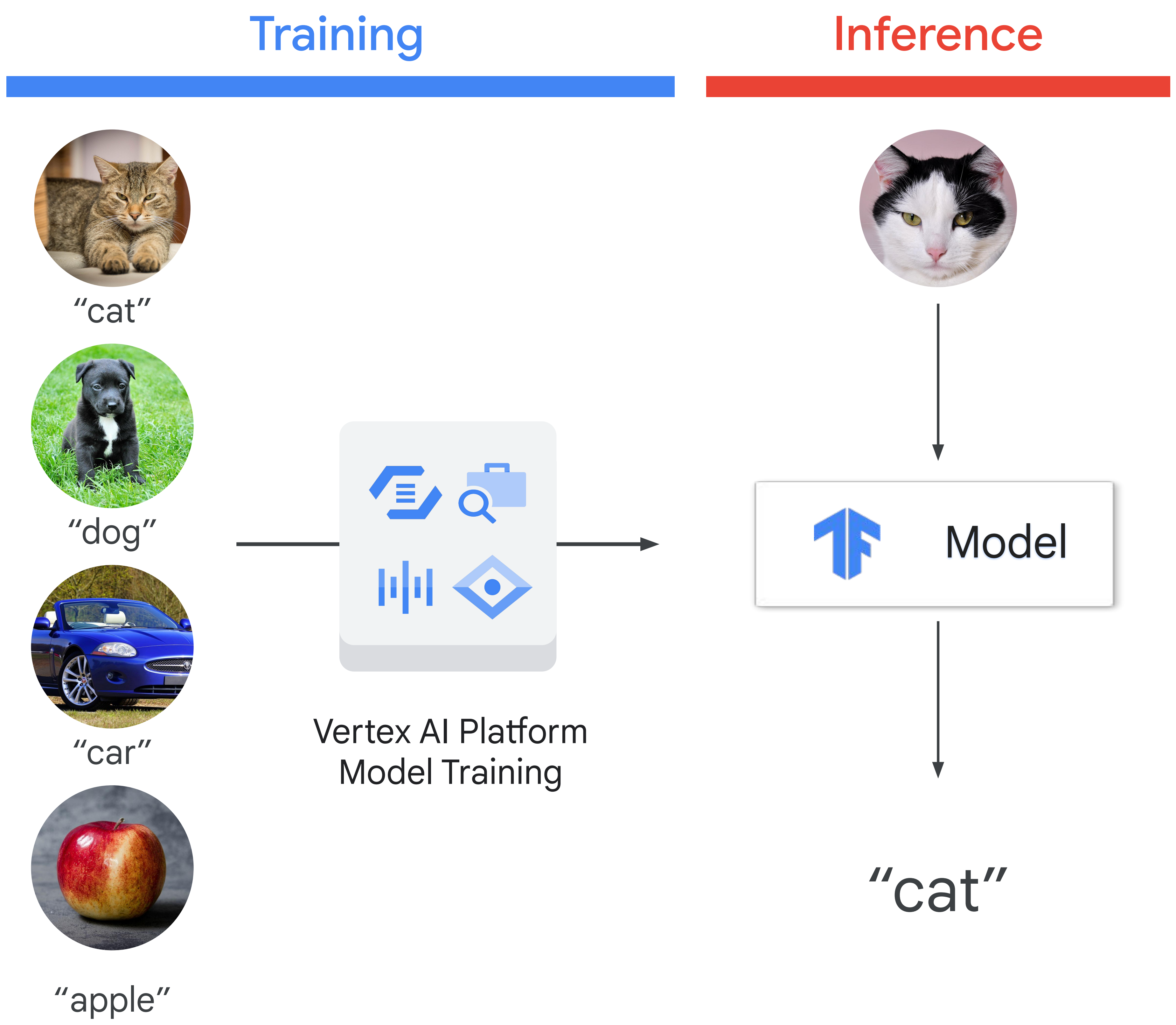


Stage 1: Train an ML model with examples



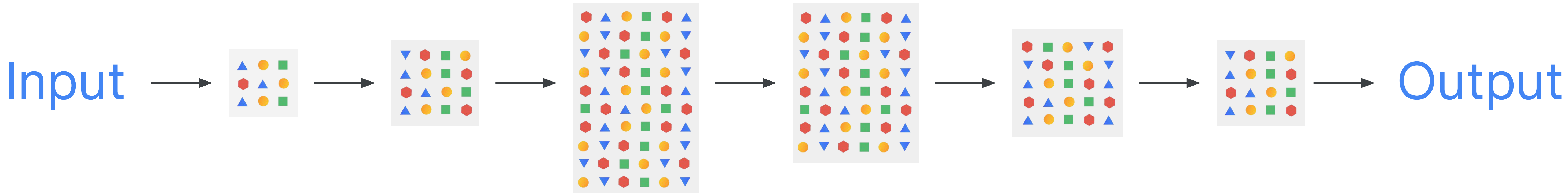
Stage 2: Predict with a trained model





Focus on both the training and inference stages of ML

Neural networks are one important technology we use



Google has more than
10,000 deep learning models

Google infuses Machine Learning into almost all its products.

Deep learning has come a long way in just the past few years

Google Photos

illustrates how far ML has come.

Google Translate

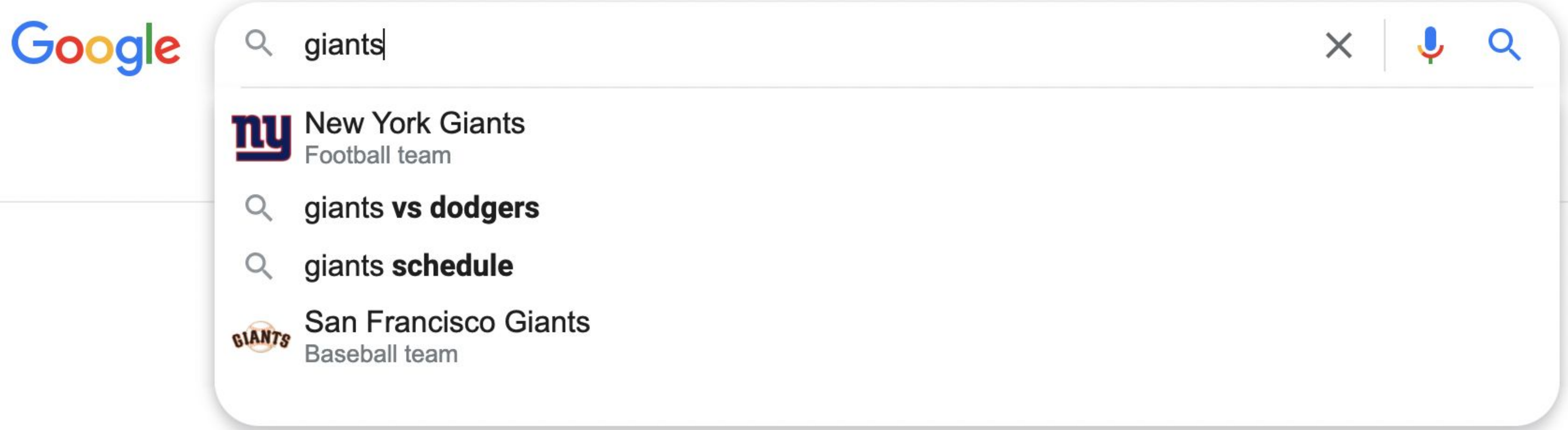
is a combination of several models.

Gmail

Smart Reply Inbox
20% of all responses sent on mobile.

1. ML is a way to derive repeated predictive insights from data
2. Stages of ML: Training and Prediction
3. Google products with ML: Photos, Translate, Smart Reply, etc.

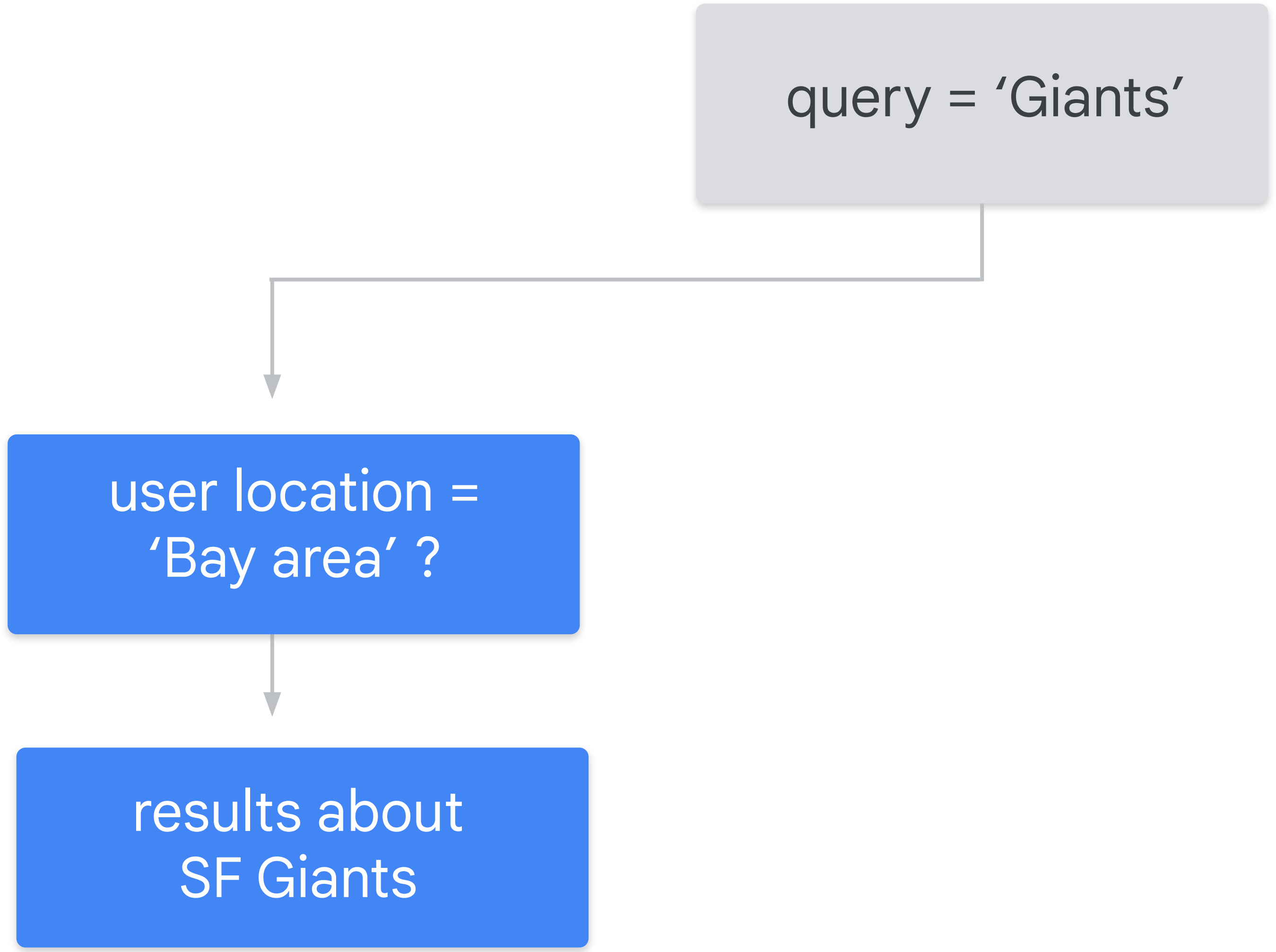
Google Search,
our flagship application



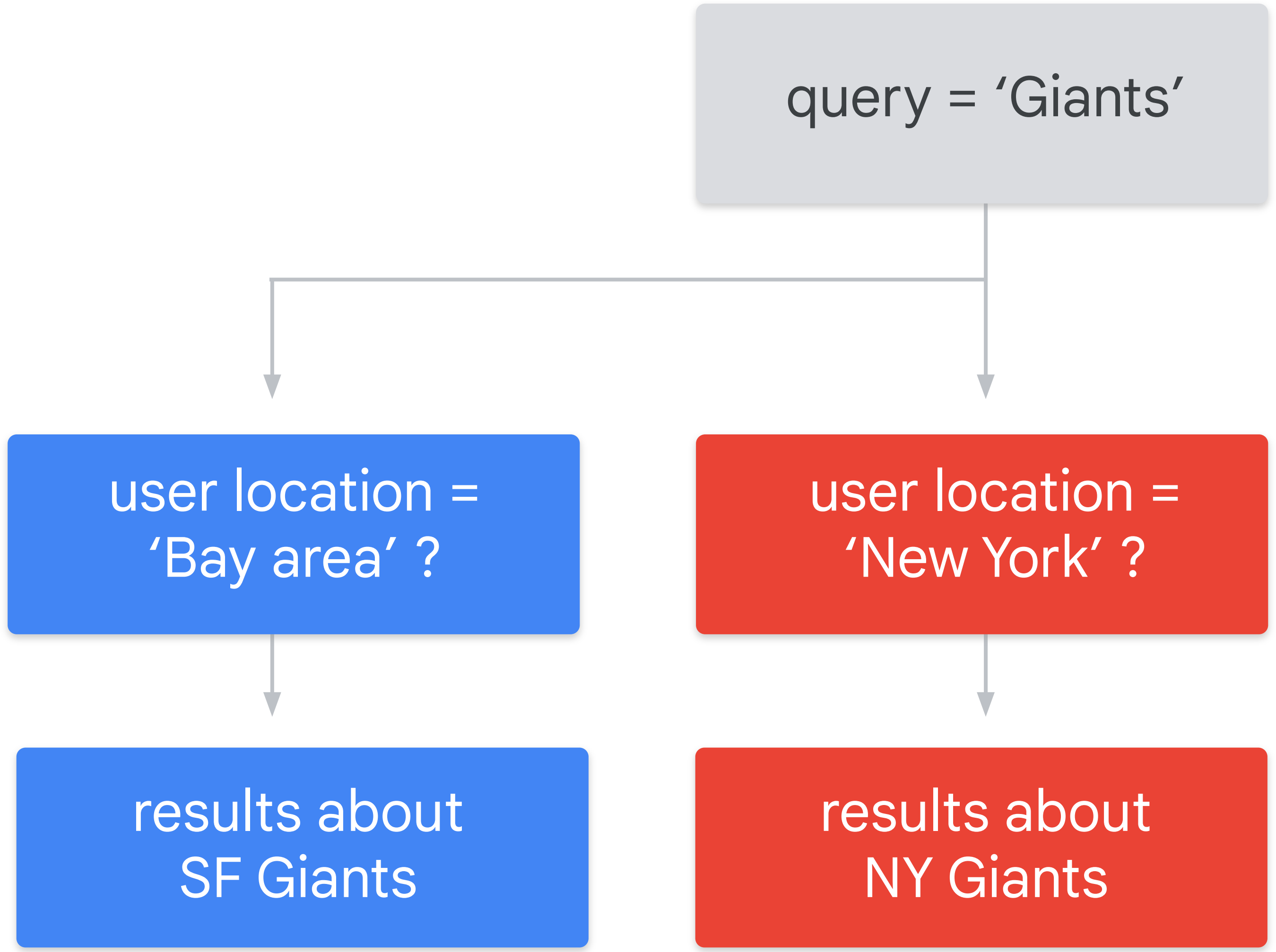
Machine learning scales better than hand-coded rules

```
query = ‘Giants’
```

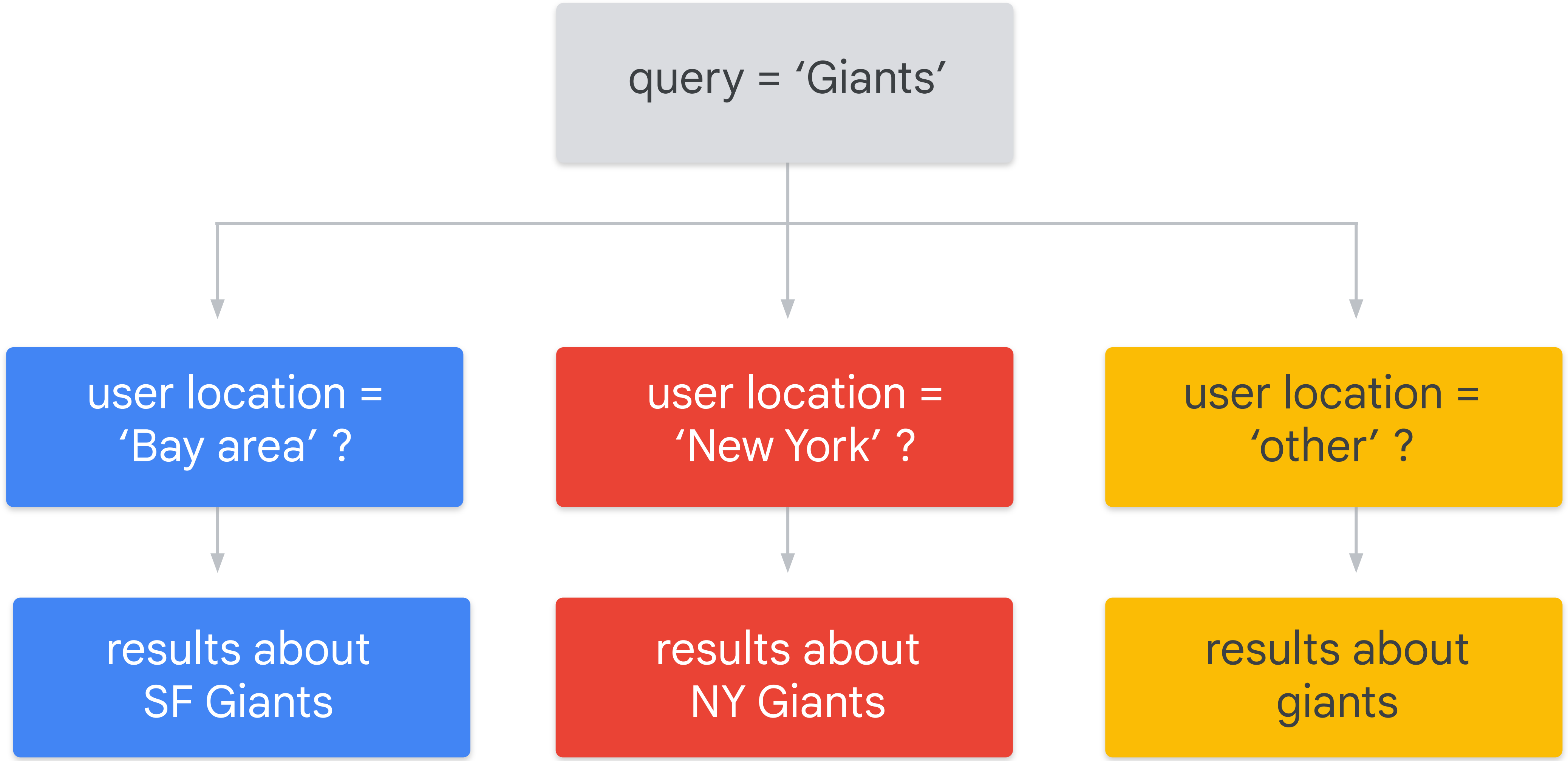
Machine learning scales better than hand-coded rules



Machine learning scales better than hand-coded rules



Machine learning scales better than hand-coded rules



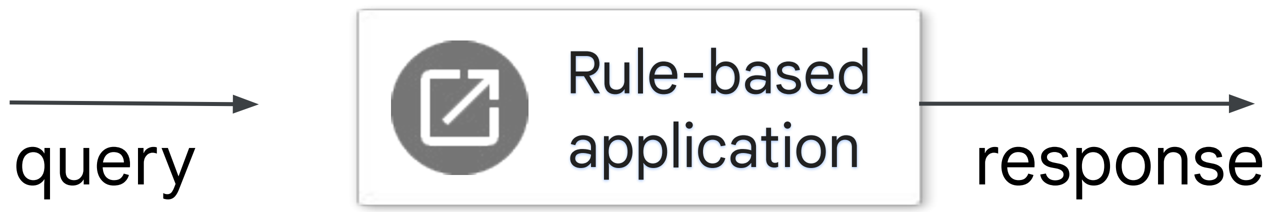
RankBrain (a deep neural network for search ranking) improved performance significantly

#3 Signal for search ranking, out of hundreds

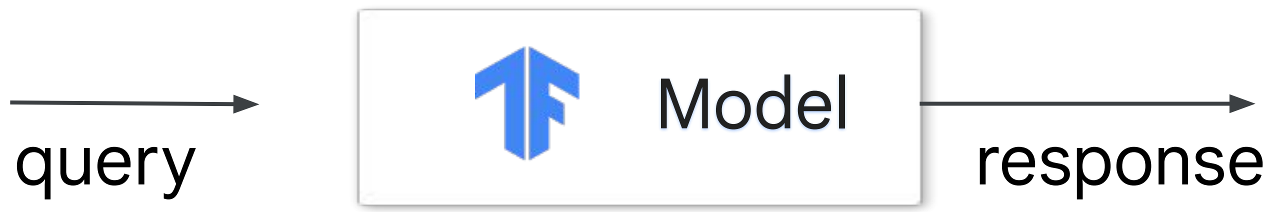
#1 Improvement to ranking quality in 2+ years

Google thinks of ML as the way
to scale, to automate, to personalize

ML can be used to solve many problems for which you are writing rules today



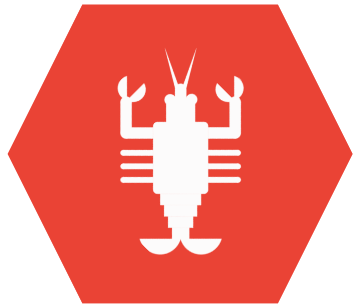
- Codeup rules based on human expertise
- Apply rules program to make decisions
- Add new rules in response to bug reports



- Train model based on data
- Deploy model at scale to make predictions
- Continuously train model on data

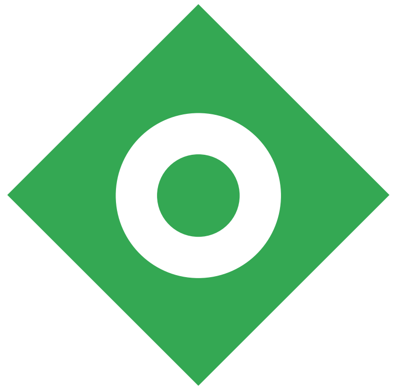
What do these search queries have in common?

Japanese toys in San Francisco



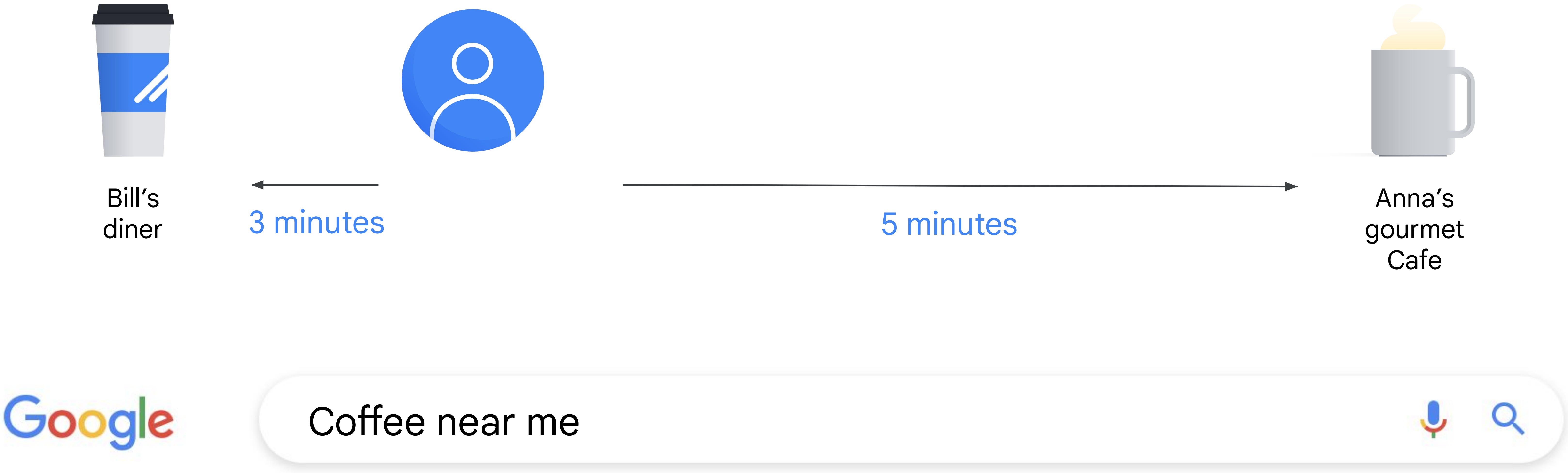
Buy live lobster in Kissimmee FL

Bee hive removal Pasadena MD

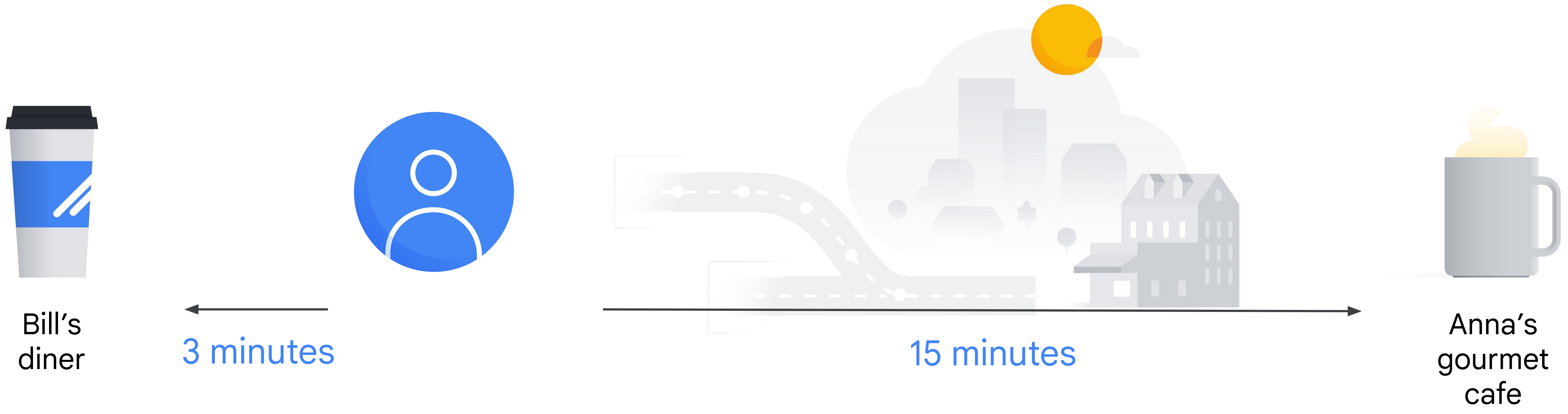


Vegan donuts near me

ML converts examples into knowledge



ML converts examples into knowledge



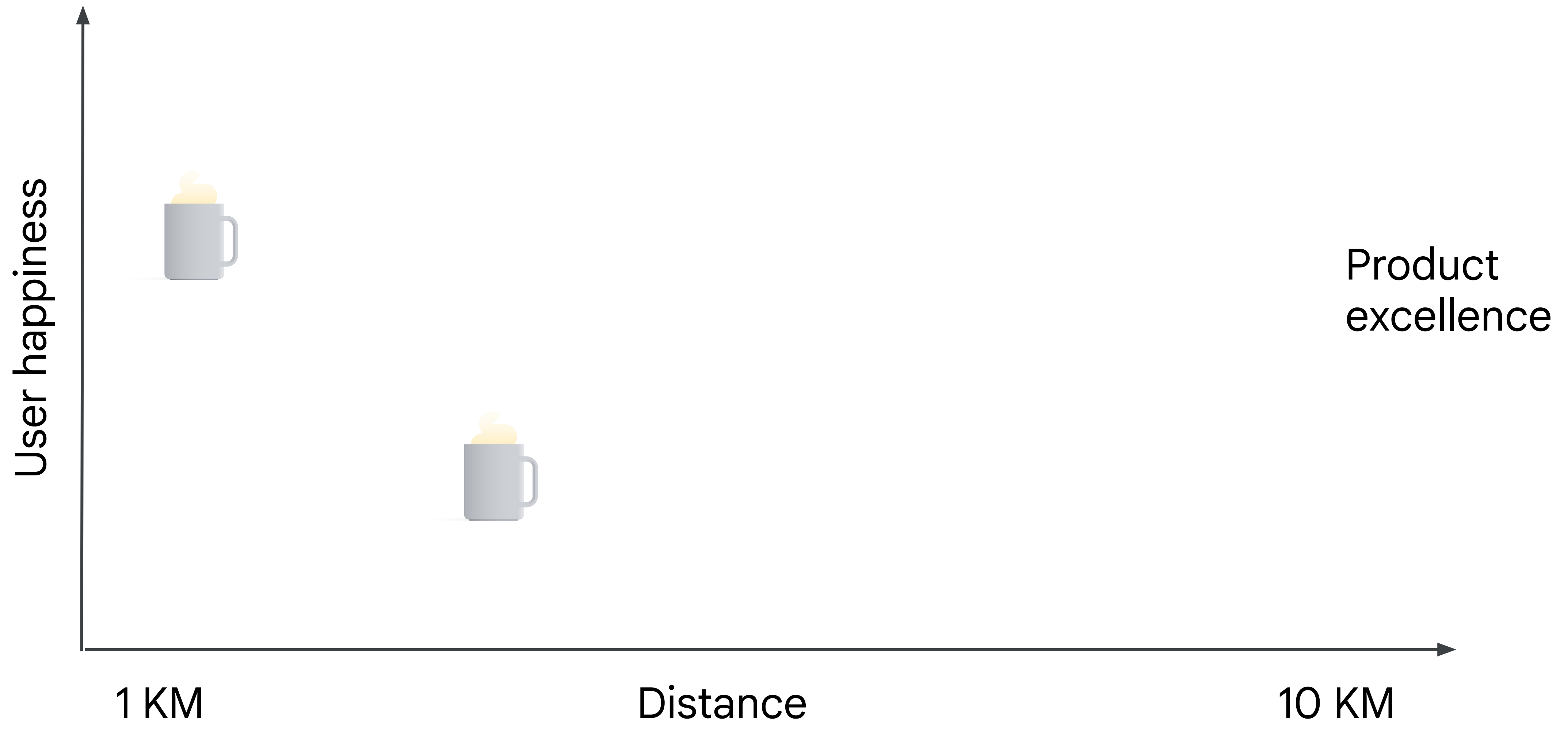
Google

Coffee near me

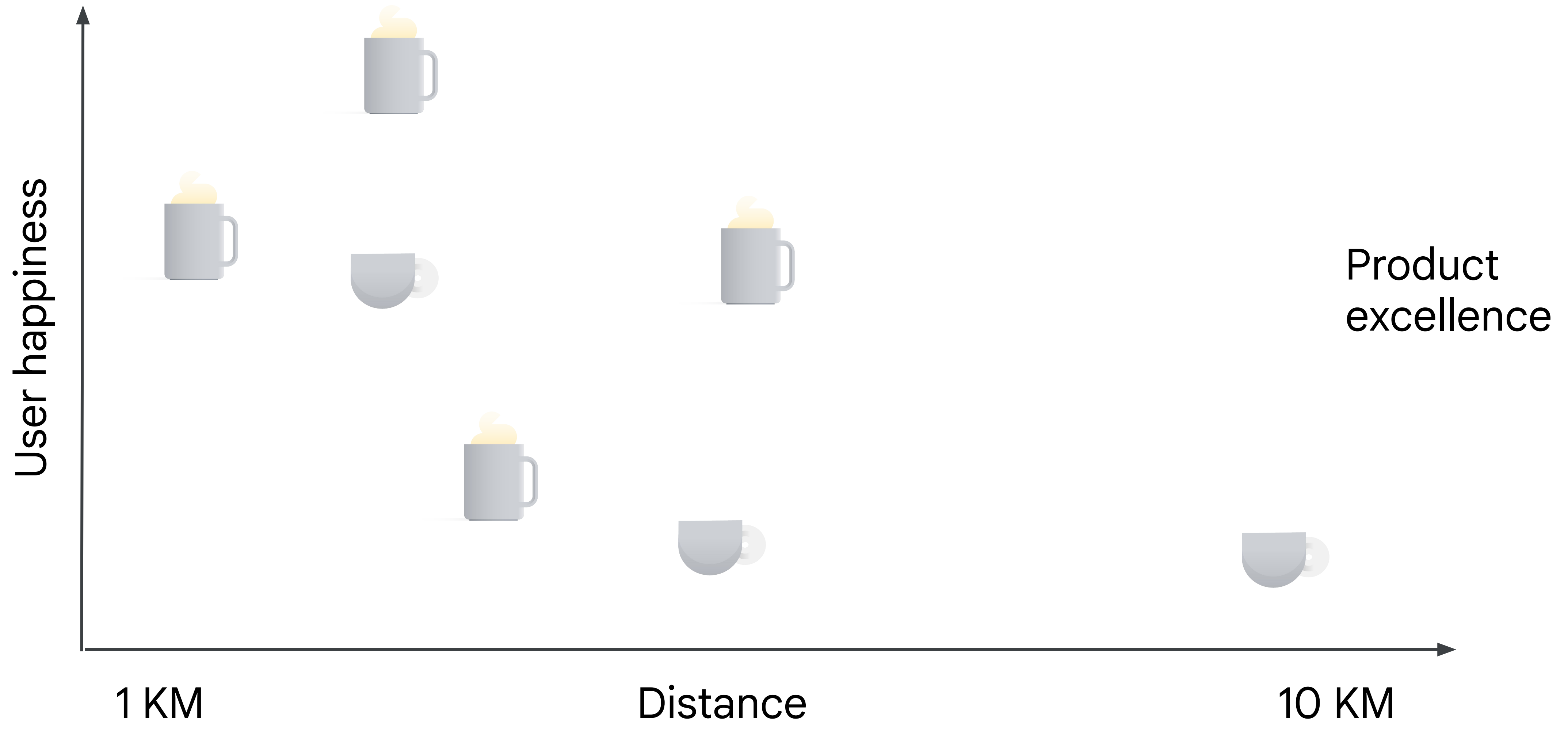
Good learning involves blending all the users’ preferences

Good learning involves blending all the users' preferences

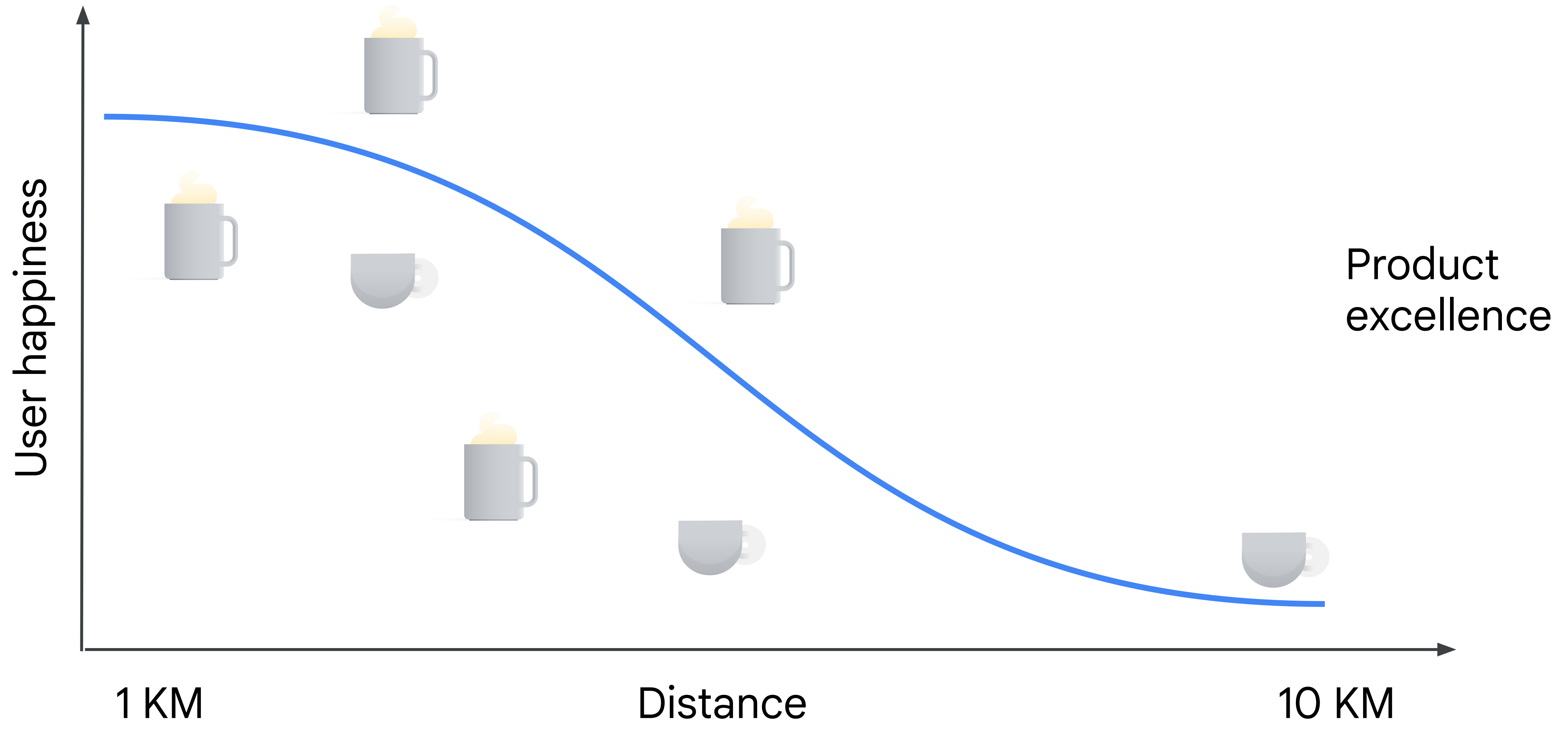
Good learning involves blending all the users' preferences



Good learning involves blending all the users' preferences



Good learning involves blending all the users' preferences



Lab intro

Framing a machine learning problem

Cloud machine learning use cases

Manufacturing

- Predictive maintenance or condition monitoring
- Warranty reserve estimation
- Propensity to buy
- Demand forecasting
- Process optimization
- Telematics

Retail

- Predictive inventory planning
- Recommendation engines
- Upsell and cross-channel marketing
- Market segmentation and targeting
- Customer ROI and lifetime value

Healthcare and Life Sciences

- Alerts and diagnostics from real-time patient data
- Disease identification and risk satisfaction
- Patient triage optimization
- Proactive health management
- Healthcare provider sentiment analysis

Cloud machine learning use cases (Continued)

Travel and Hospitality

- Aircraft scheduling
- Dynamic pricing
- Social media – consumer feedback and interaction analysis
- Customer complaint resolution
- Traffic patterns and congestion management

Financial Services

- Risk analytics and regulation
- Customer segmentation
- Cross-selling and up-selling
- Sales and marketing campaign management
- Credit worthiness evaluation

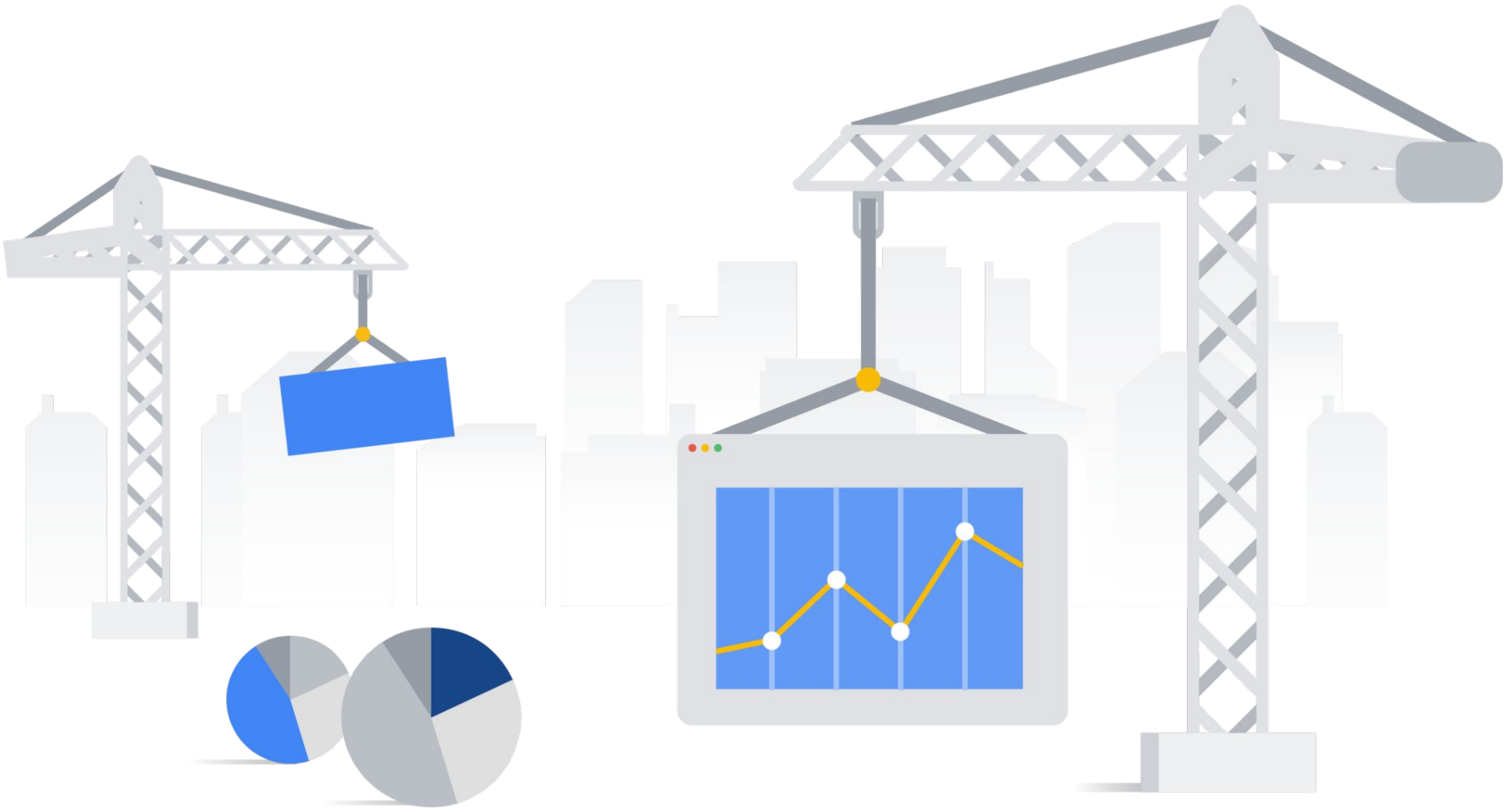
Energy, Feedstock and Utilities

- Power usage analytics
- Seismic data processing
- Carbon emissions and trading
- Customer-specific pricing
- Smart grid management
- Energy demand and supply optimization

Lab solutions

Framing a machine learning problem

Example solution: Demand forecasting in manufacturing



ML problem:

What is being predicted?

How many units of widgets X should you manufacture this month?

What data is needed?

Historical data on # of units sold, price it was sold at, # of units returned, price of competitor product, # of units of all items that use widget X that were sold (e.g. if widget is a phone display panel, how many smartphones were sold, regardless of which display panel they carried?), economic figures (e.g. customer confidence, interest rate), this-month-last-year

Example solution:
As a software problem

predictDemand(widgetID, month=CurrentTime.month)

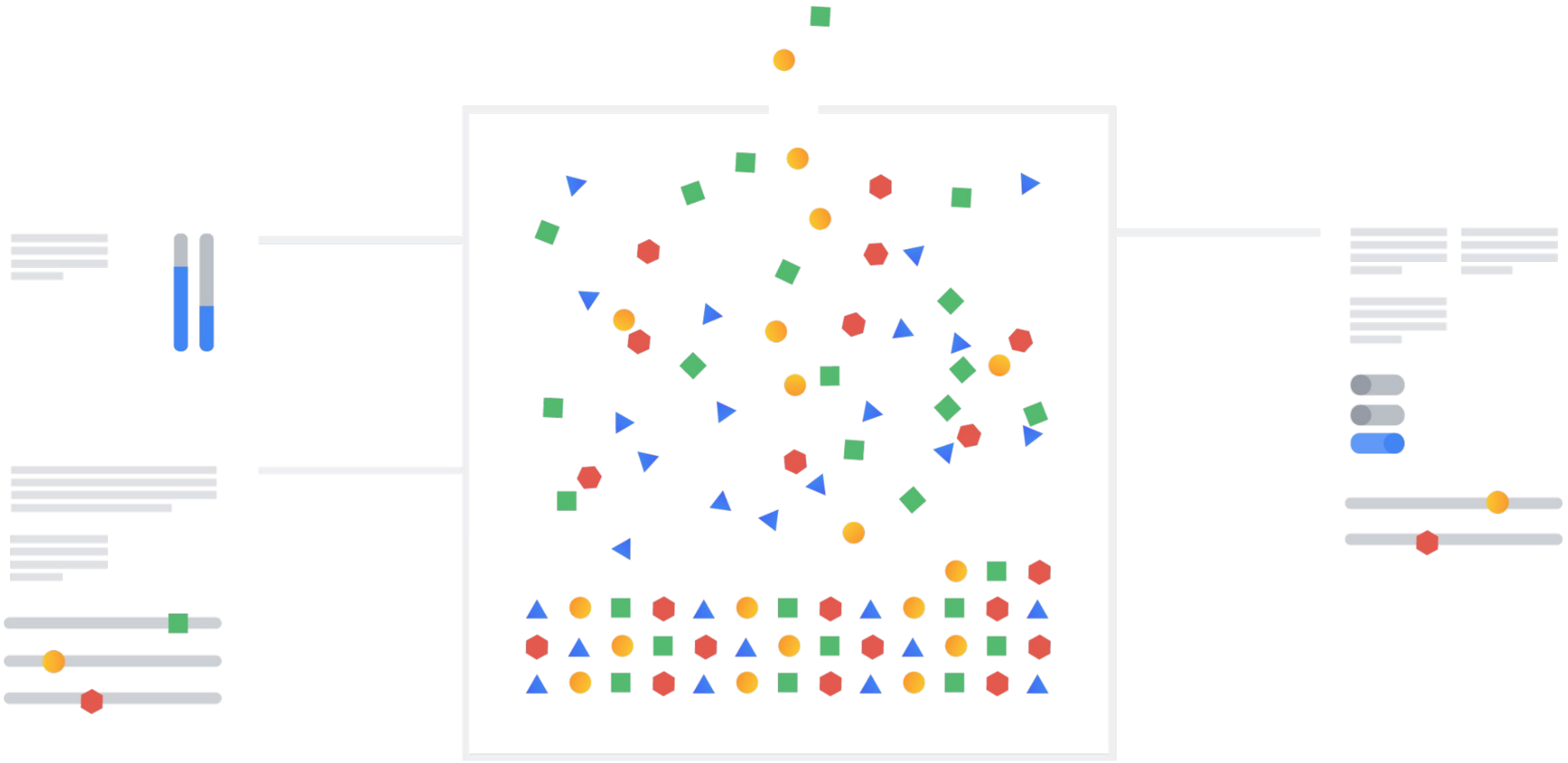
Who will use this service?

Product managers, logistics managers

How are they doing it today?

They examine trends of e.g. phone sales, overall economy, trade publications and make a decision

Example solution:
As a data problem



Data problem:

Collect: economic data, competitor data, industry data, our figures

Analyze: craft features that our experts are looking at today from this data and use as inputs to model

React: automatic?

Pre-trained models

How much is this car worth?

AUCNET

1st Guess

89%

TOYOTA Land Cruiser PRADO

CBA-TRJ150W

Price range (USD)

39,390~43,320

There are pre-trained machine learning services available on Google Cloud

Custom ML models

Vertex AI

TensorFlow

There are pre-trained machine learning services available on Google Cloud

Custom ML models

Vertex AI

TensorFlow

Pre-trained ML Models

Vision API

Speech API

Jobs API

Translation API

Natural Language API

Video Intelligence API

Ocado routes emails based on NLP

Improves natural language processing of customer service claims

"Hi Ocado,
I love your website. I have children so it's easier for me to do the shopping online. Many thanks for saving my time!
Regards"

Feedback

Customer is happy

"Thanks to the Google Cloud Platform, Ocado was able to use the power of cloud computing and train our models in parallel."

Let your users

talk to you

50% of enterprises will be spending more per annum on bots and chatbot creation than traditional mobile app development by 2021

Gartner

The ML marketplace is moving towards increasing levels of ML abstraction

AUCNET

Create a custom image model to price cars.

ocado

Build off NLP API to route customer emails.

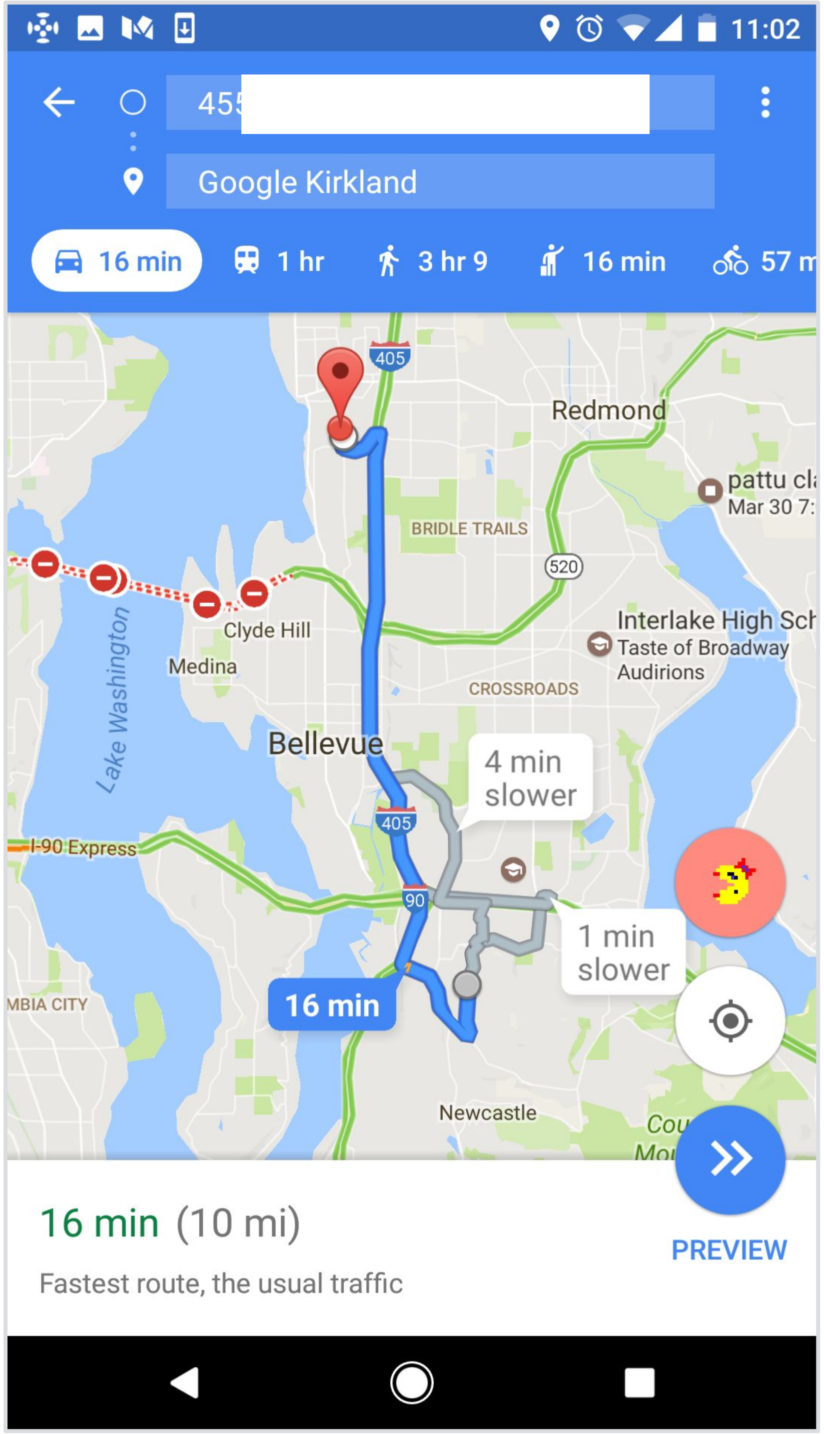
GIPHY

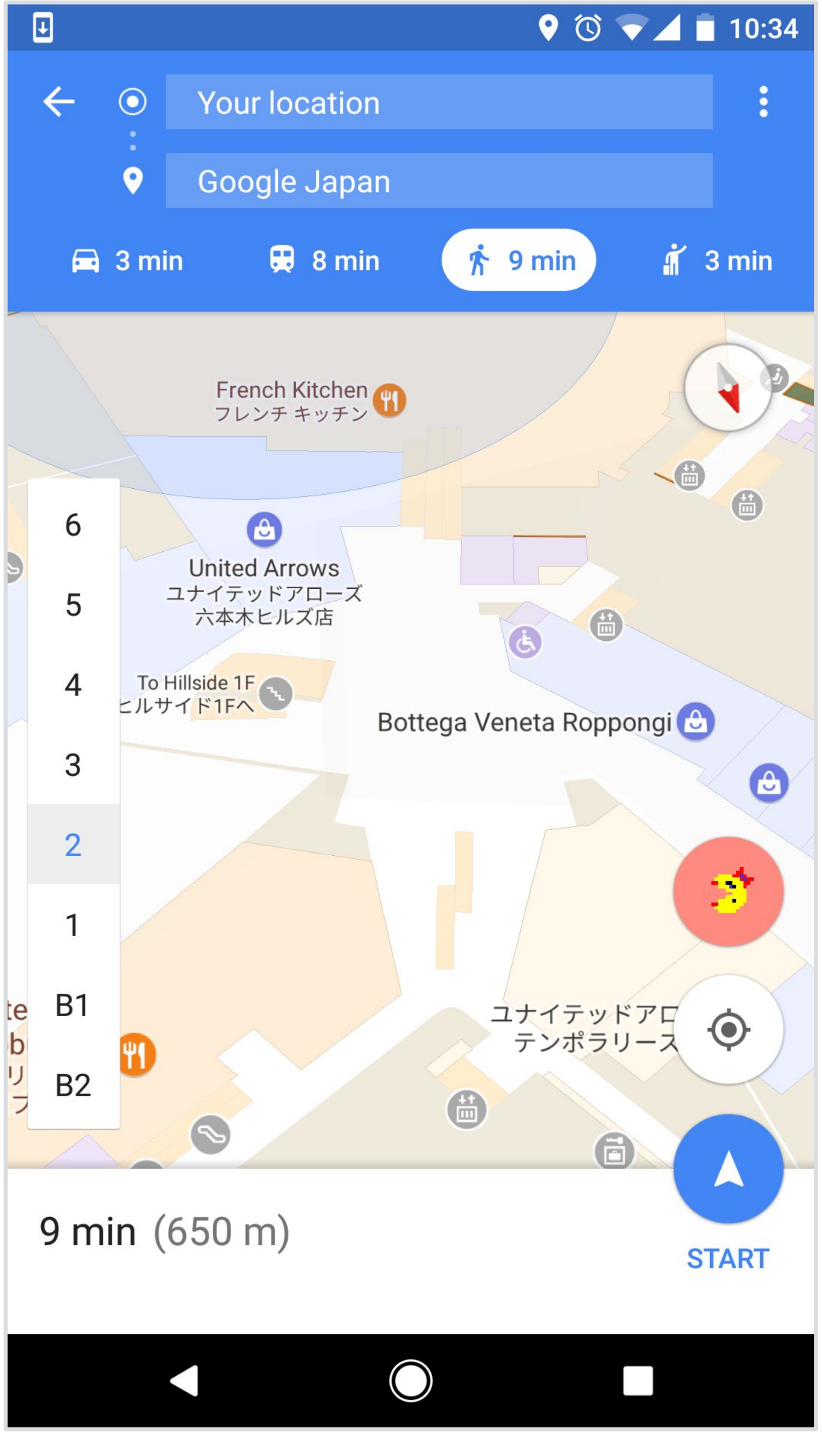
Use Vision API as-is to find text in memes.

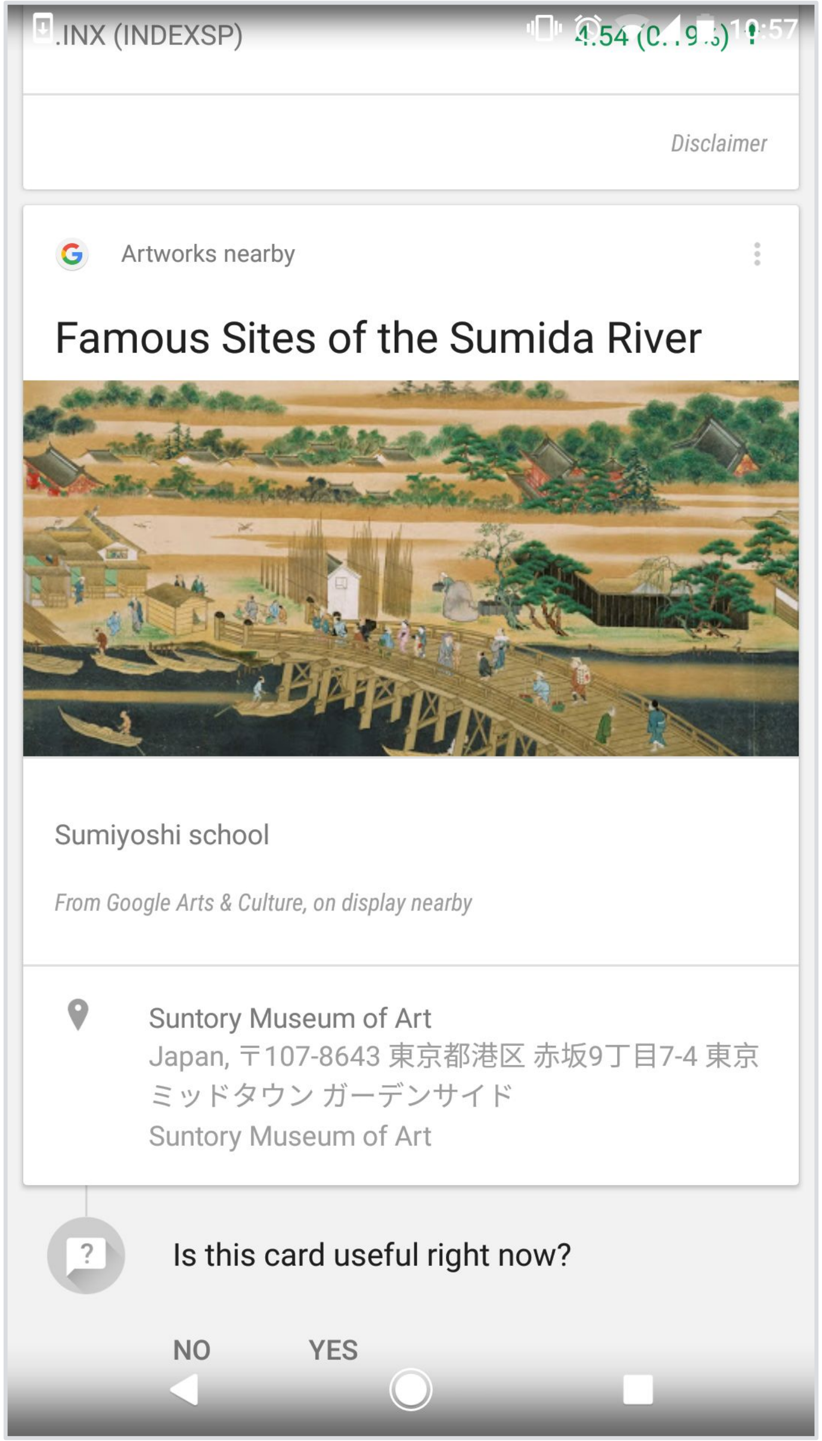
UNIQLO

Use Dialogflow to create a new shopping experience.

Is this machine learning? What’s needed for ML?





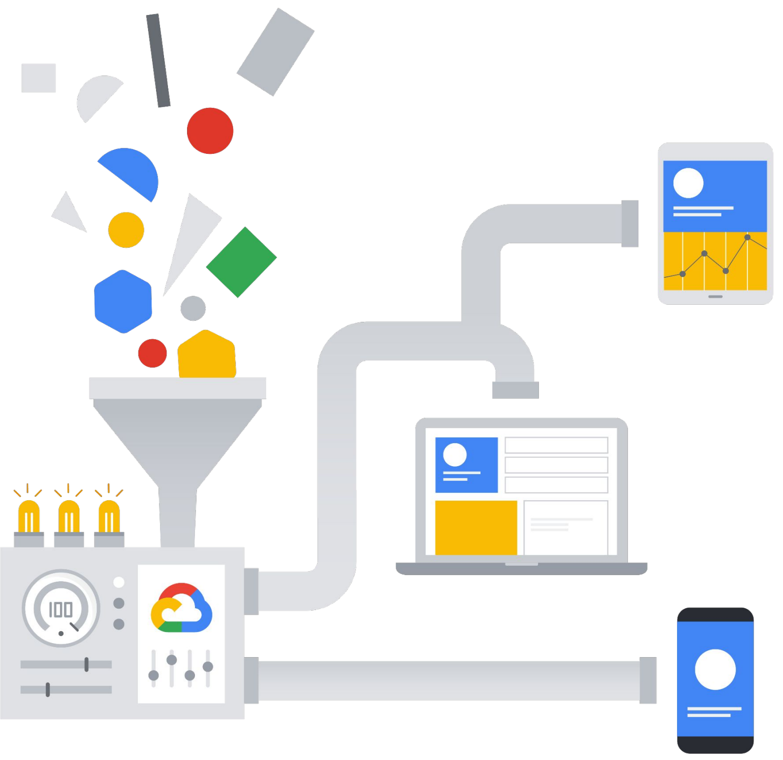


If ML is a rocket engine,
data is the fuel.

Simple ML and more data >
Fancy ML and small data

Typical customer journey involves going from manual data analysis to ML

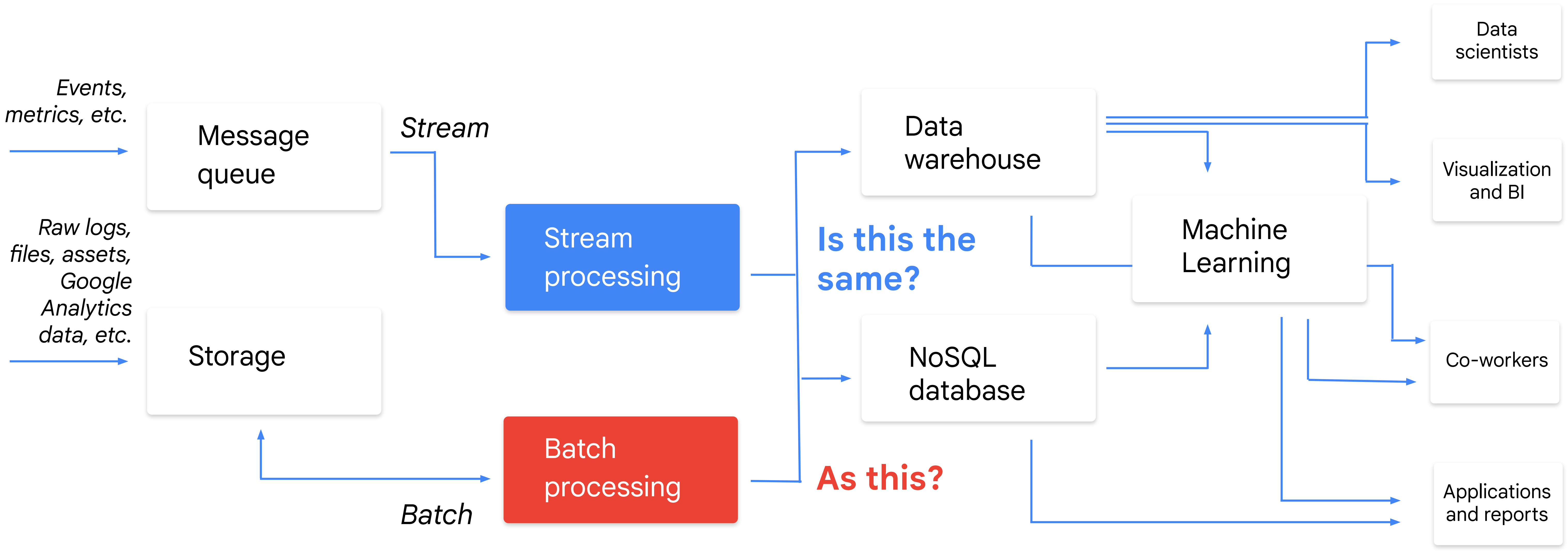
Enables automation of previously manual global fishing data analyses



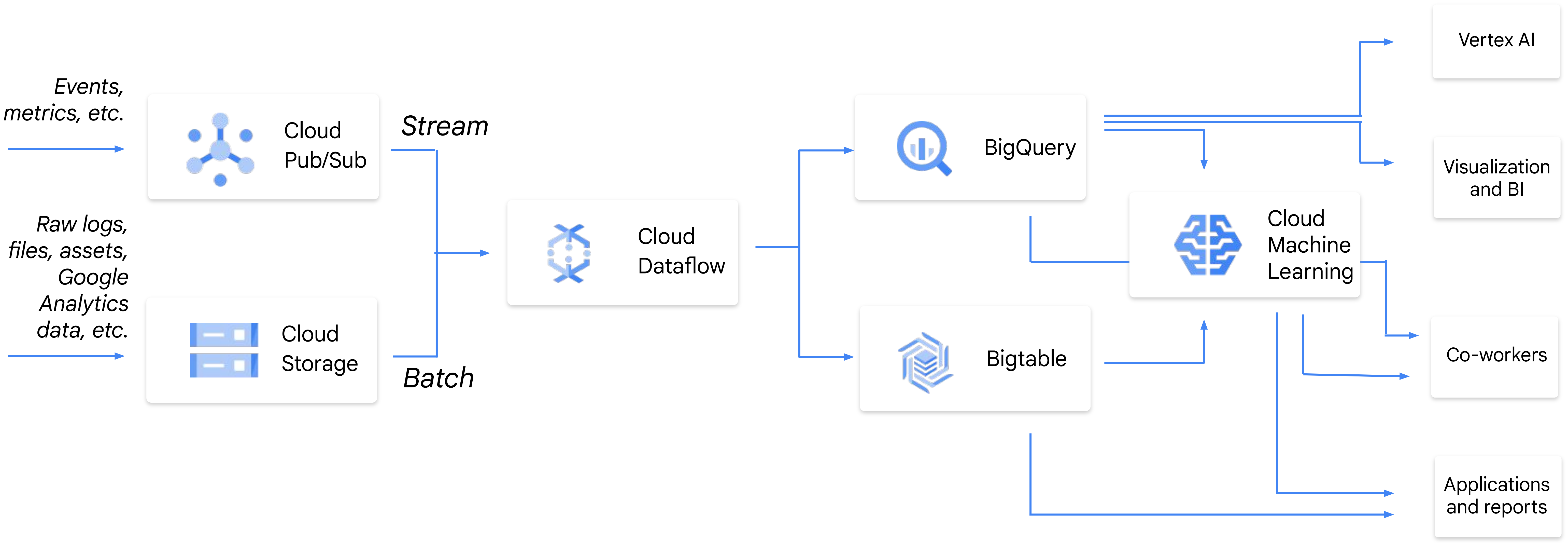
Processes 22 million fishing data points daily

1. Collecting data is often the longest and hardest part of an ML project, and the one most likely to fail
2. Manual analysis helps you fail fast and try new ideas in a more agile way
3. To build a good ML model, you have to know your data
4. ML is a journey towards automation and scale

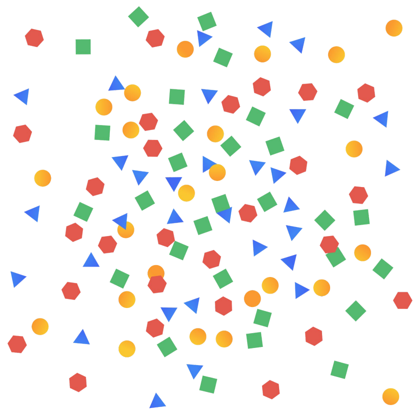
For machine learning, you need to build a streaming pipeline in addition to a batch pipeline



Sophistication around real-time data is key



Performance metrics for training are different than for predictions

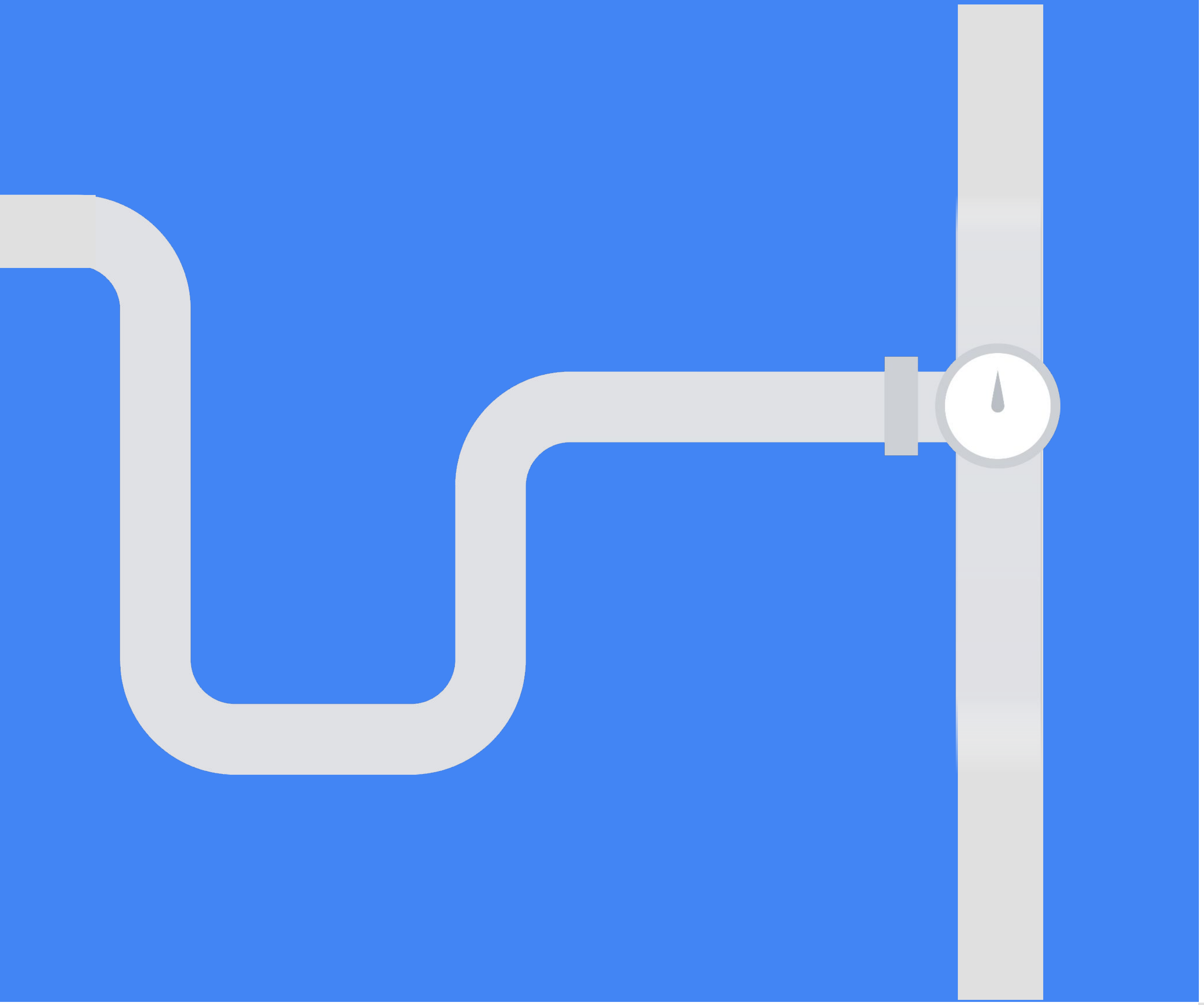


Training should scale to handle a lot of data.

Vertex AI

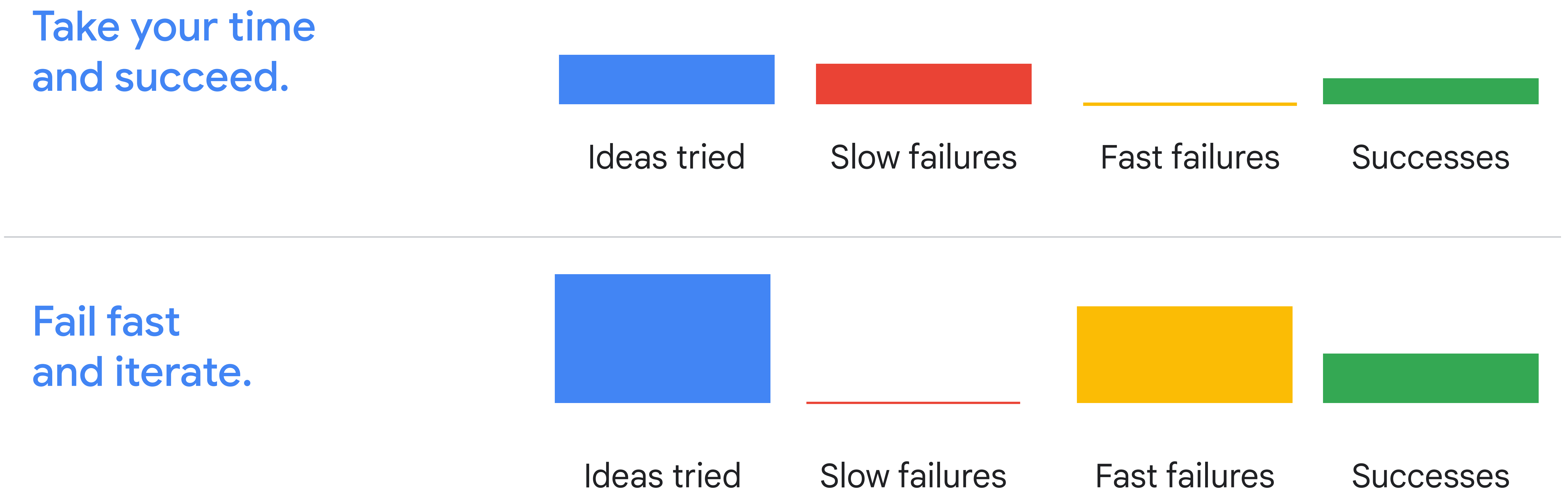
TensorFlow

Predictions should scale to handle large number of queries per second.

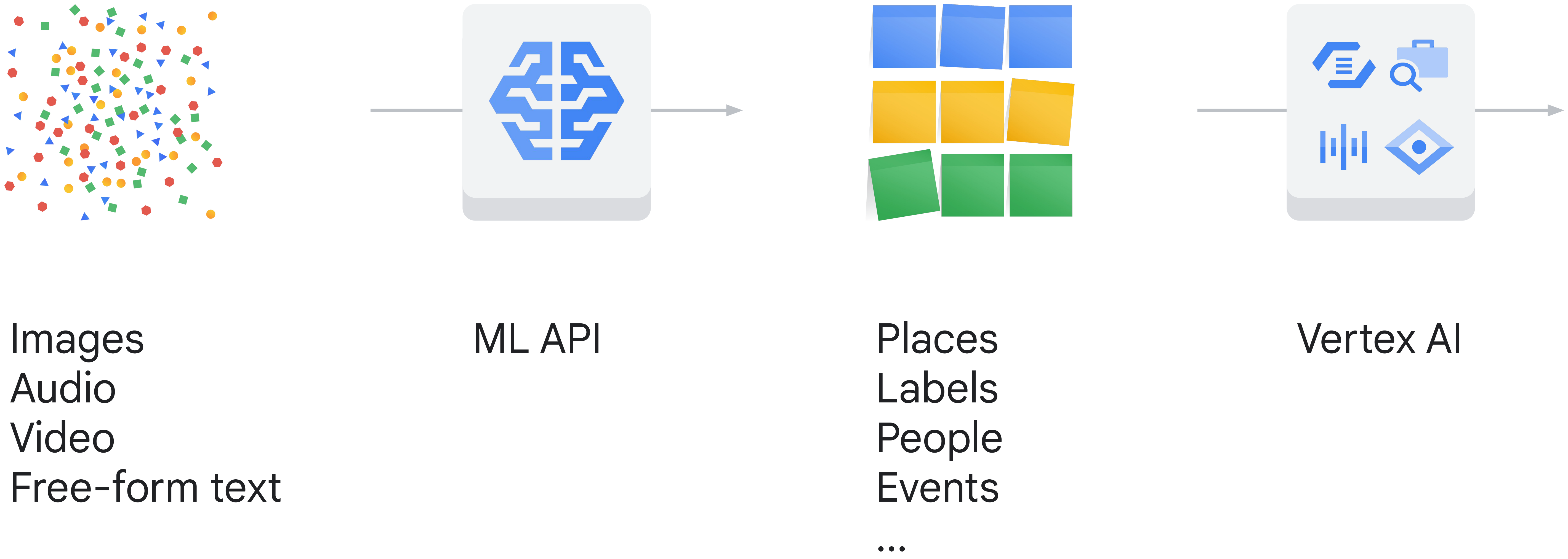


Connect simple ML models into a pipeline.

Freedom to experiment (and fail) is important



Build on top of Google



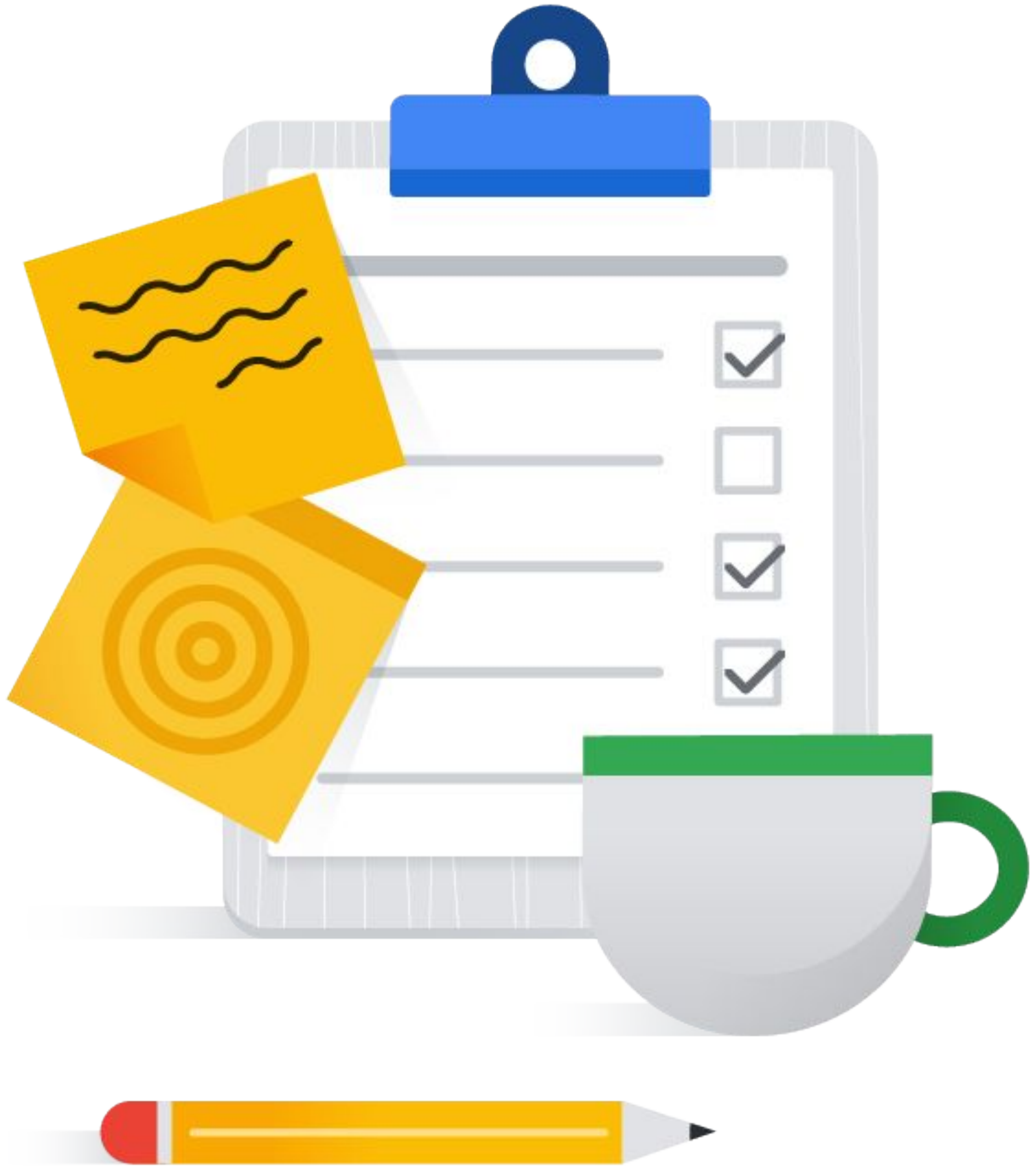
Google Cloud

How Google Does Machine Learning

December 2021

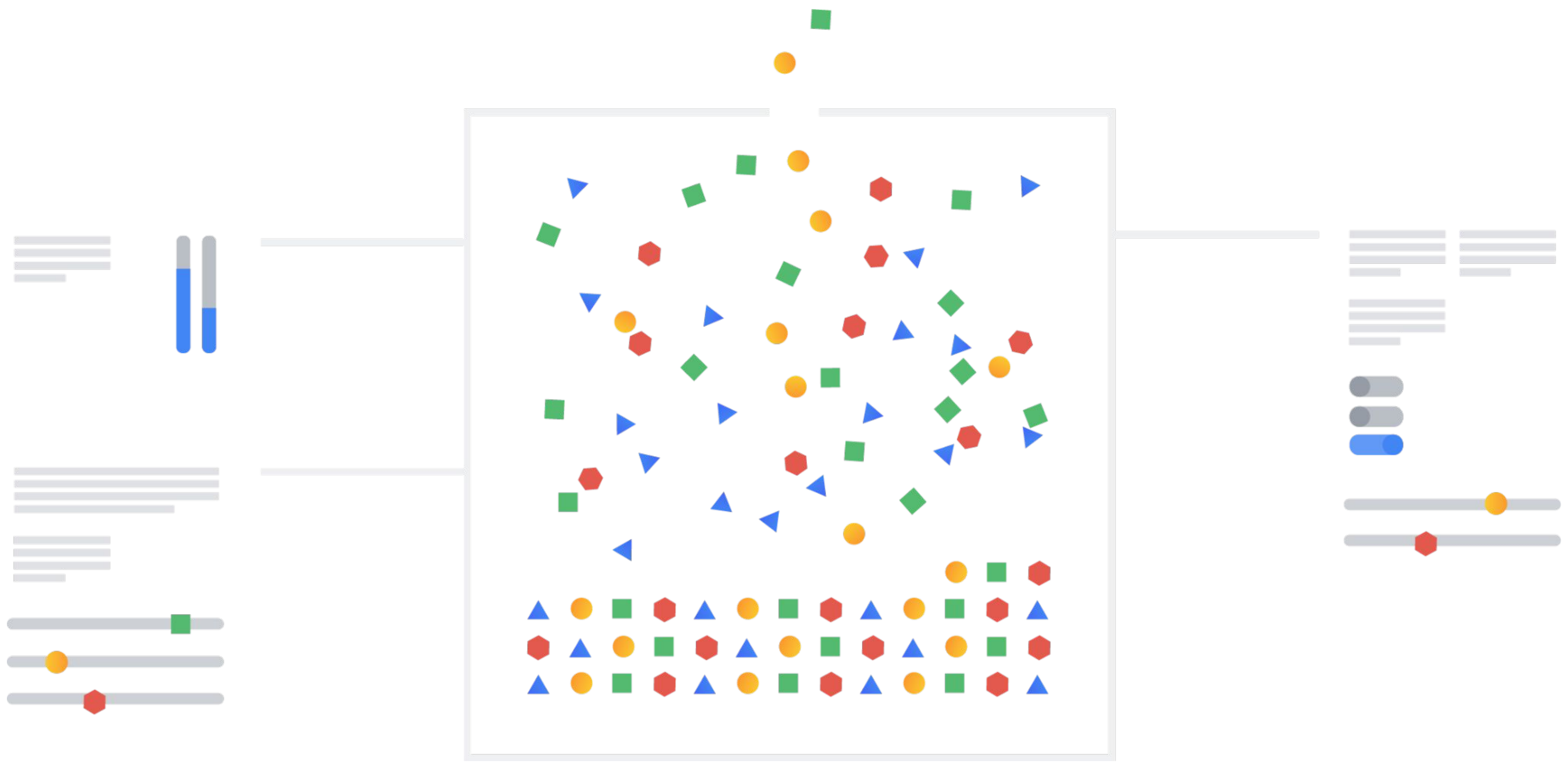
In this module, you learn to ...

01	Acquire the organizational know-how to implement machine learning
02	Leverage the experience of Google to avoid common pitfalls

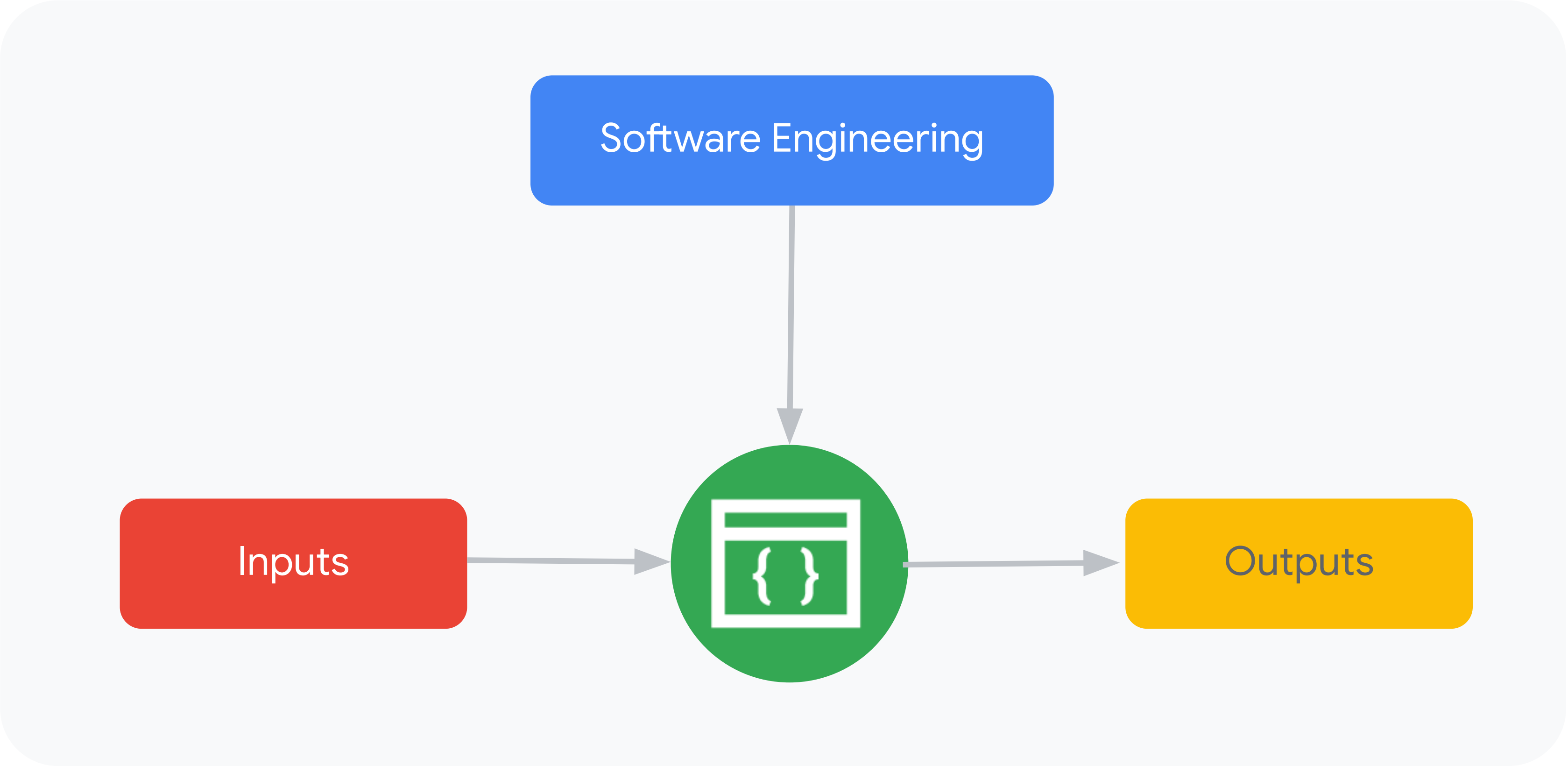


What is ML?

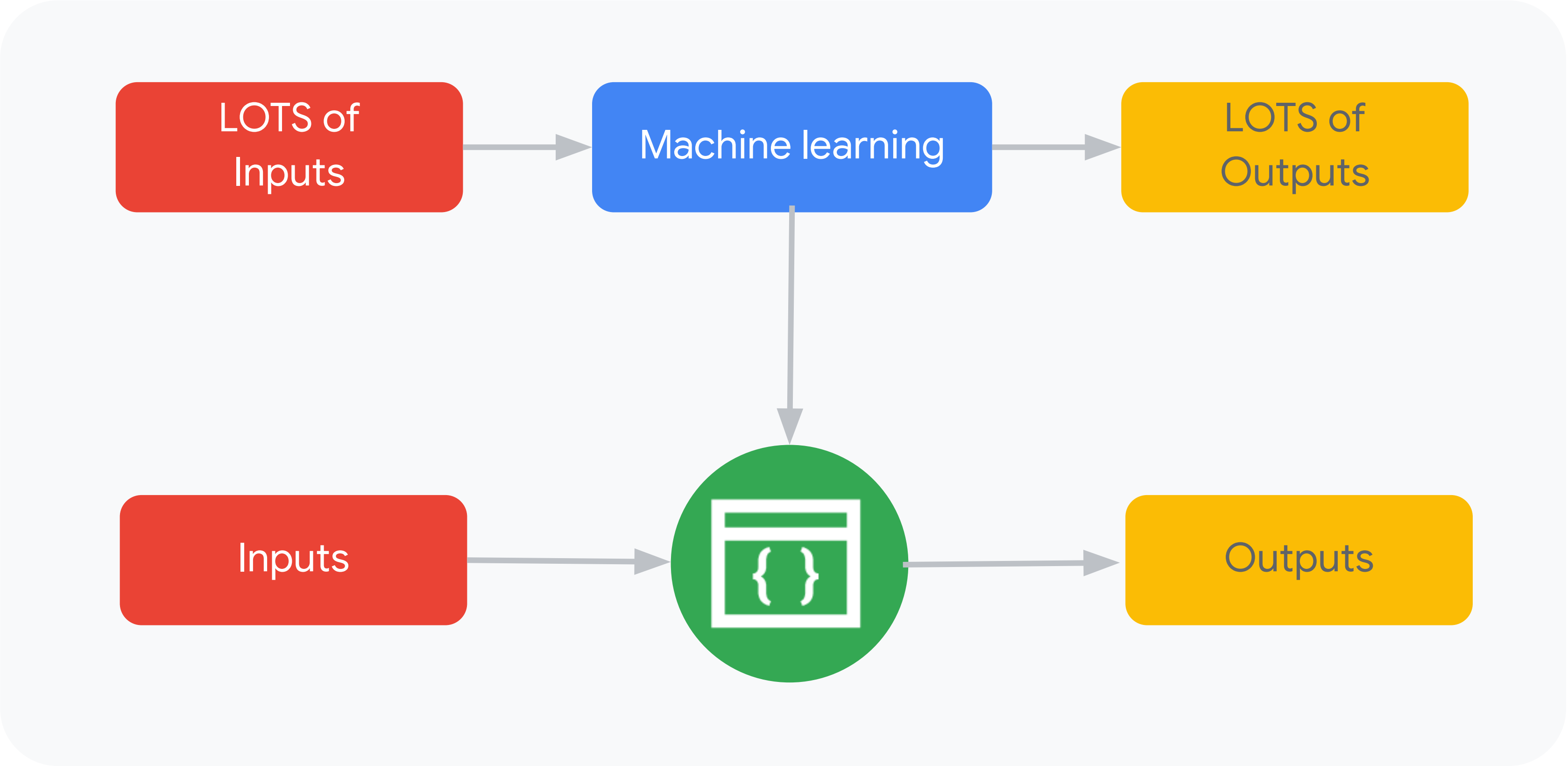
Machine learning (ML) is the process of a computer writing a computer program to accomplish a task, and figures out the best program by looking at a set of example.



Software engineers write program rules



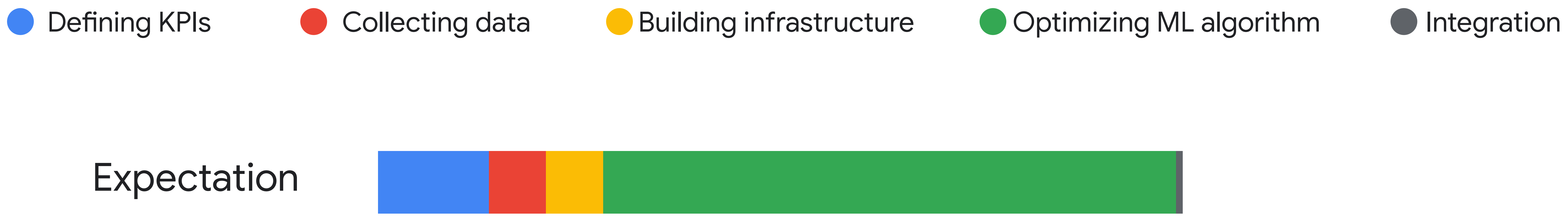
Machine learning figures out program rules



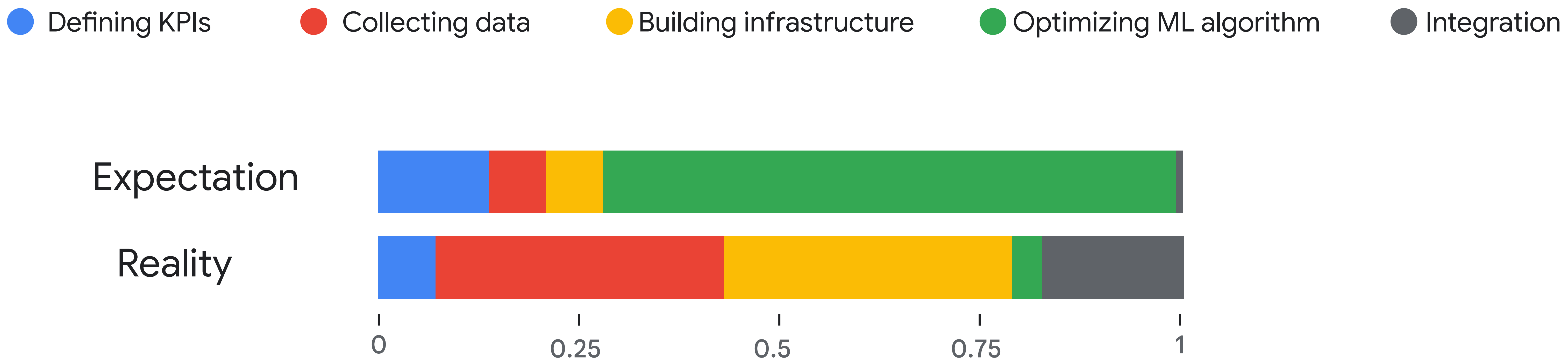
ML effort allocation

- Defining KPIs
- Collecting data
- Building infrastructure
- Optimizing ML algorithm
- Integration

ML effort allocation



ML effort allocation



Then why are we learning about ML?

Google is going to share
the secret sauce

Large-Scale Deep Learning with TensorFlow

Jeff Dean
Google Brain team
g.co/brain
In collaboration with many other people at Google

<hello world>

$y = mX + b$

Get your hands dirty by practicing with technical skills

Avoid these top 10 ML pitfalls

- Defining KPIs
- Collecting data
- Integration
- Infrastructure
- Optimizing ML

Avoid these top 10 ML pitfalls

- Defining KPIs
- Collecting data
- Integration
- Infrastructure
- Optimizing ML

01 ML requires just as much software infrastructure

Avoid these top 10 ML pitfalls

- Defining KPIs
- Collecting data
- Integration
- Infrastructure
- Optimizing ML

01 ML requires just as much software infrastructure

02 No data collected yet

Avoid these top 10 ML pitfalls

Defining KPIs   Collecting data   Integration   Infrastructure   Optimizing ML

01 ML requires just as much software infrastructure

02 No data collected yet

03 Assume the data is ready for use

Avoid these top 10 ML pitfalls

Defining KPIs | Collecting data | Integration | Infrastructure | Optimizing ML

01 ML requires just as much software infrastructure
02 No data collected yet
03 Assume the data is ready for use
04 Keep humans in the loop

Avoid these top 10 ML pitfalls

Defining KPIs · Collecting data · Integration · Infrastructure · Optimizing ML

01 ML requires just as much software infrastructure
02 No data collected yet
03 Assume the data is ready for use
04 Keep humans in the loop
05 Product launch focused on the ML algorithm

Avoid these top 10 ML pitfalls

Defining KPIs | Collecting data | Integration | Infrastructure | Optimizing ML

01 ML requires just as much software infrastructure
02 No data collected yet
03 Assume the data is ready for use
04 Keep humans in the loop
05 Product launch focused on the ML algorithm
06 ML optimizing for the wrong thing

Avoid these top 10 ML pitfalls

Defining KPIs Collecting data Integration Infrastructure Optimizing ML

01 ML requires just as much software infrastructure
02 No data collected yet
03 Assume the data is ready for use
04 Keep humans in the loop
05 Product launch focused on the ML algorithm
06 ML optimizing for the wrong thing
07 Is your ML improving things in the real world

Avoid these top 10 ML pitfalls

Defining KPIs   Collecting data   Integration   Infrastructure   Optimizing ML

01 ML requires just as much software infrastructure
02 No data collected yet
03 Assume the data is ready for use
04 Keep humans in the loop
05 Product launch focused on the ML algorithm
06 ML optimizing for the wrong thing
07 Is your ML improving things in the real world
08 Using a pre-trained ML algorithm vs building your own

Avoid these top 10 ML pitfalls

Defining KPIs · Collecting data · Integration · Infrastructure · Optimizing ML

01 ML requires just as much software infrastructure
02 No data collected yet
03 Assume the data is ready for use
04 Keep humans in the loop
05 Product launch focused on the ML algorithm
06 ML optimizing for the wrong thing
07 Is your ML improving things in the real world
08 Using a pre-trained ML algorithm vs building your own
09 ML algorithms are trained more than once

Avoid these top 10 ML pitfalls

Defining KPIs · Collecting data · Integration · Infrastructure · Optimizing ML

- 01 ML requires just as much software infrastructure
- 02 No data collected yet
- 03 Assume the data is ready for use
- 04 Keep humans in the loop
- 05 Product launch focused on the ML algorithm
- 06 ML optimizing for the wrong thing
- 07 Is your ML improving things in the real world
- 08 Using a pre-trained ML algorithm vs building your own
- 09 ML algorithms are trained more than once
- 10 Trying to design your own perception or NLP algorithm

Ugh, so that’s the bad news, what's the good news?

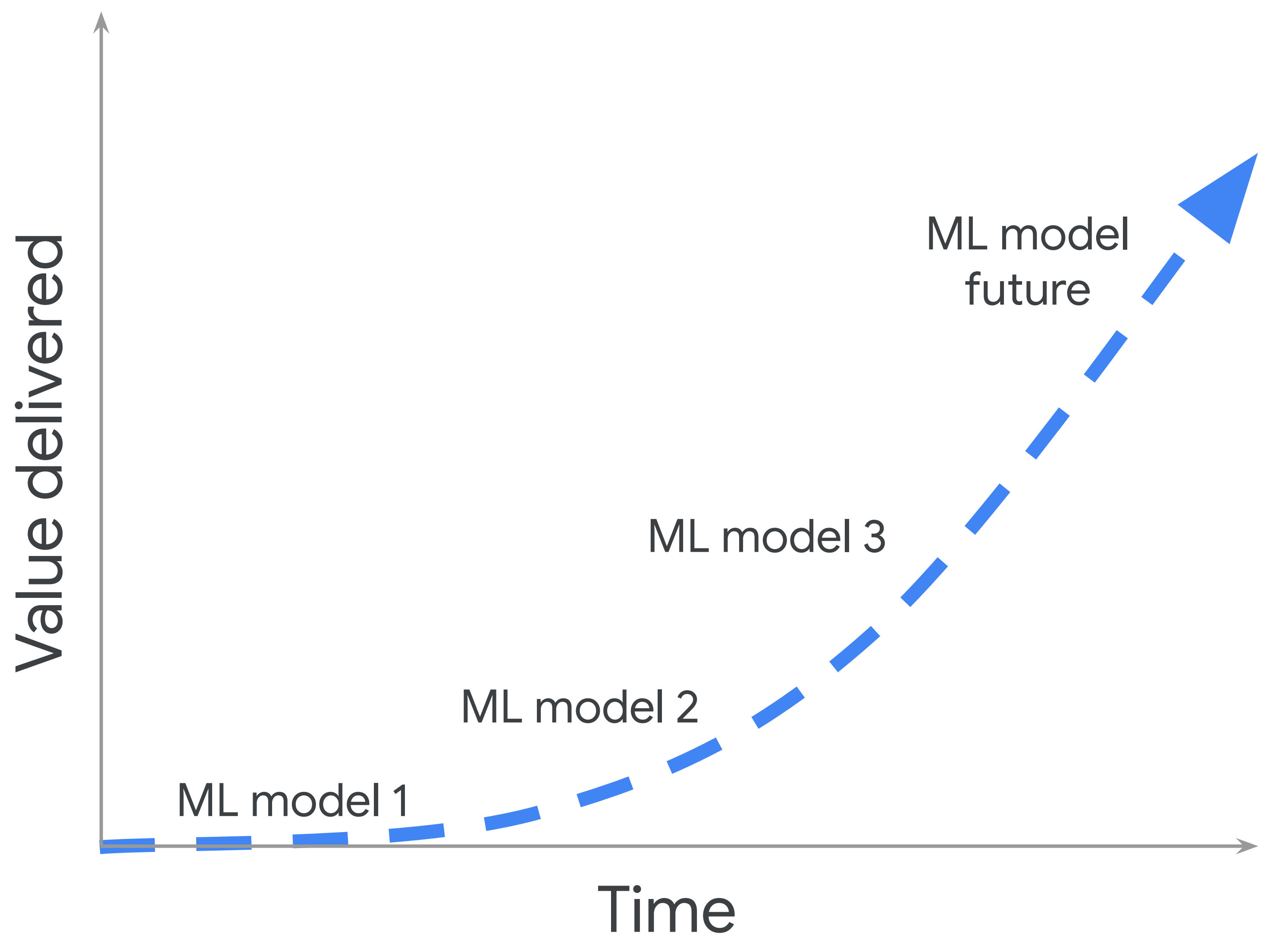
Most ML value comes along the way

ML improves almost everything it touches

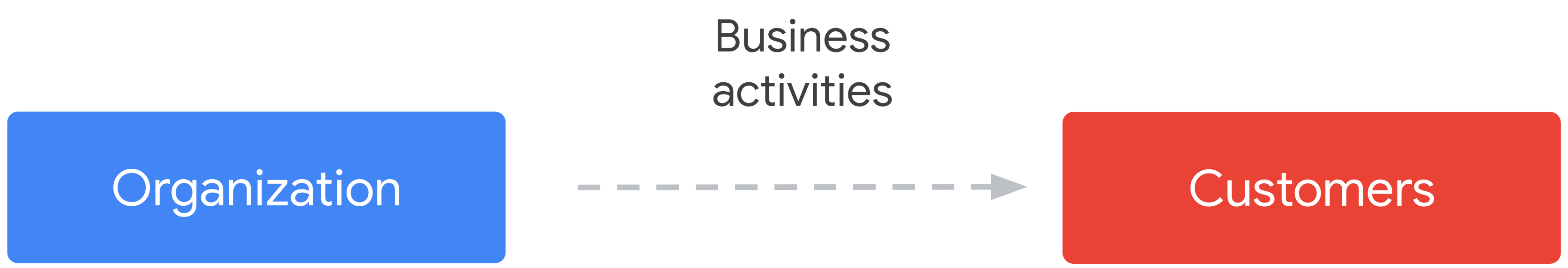
If ML is hard, it’s hard for your competitors too

ML is a great differentiator

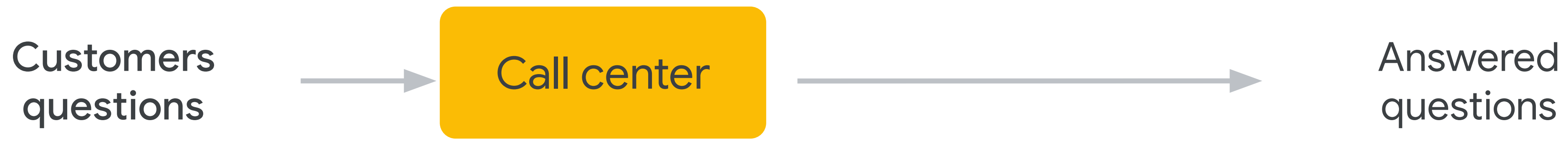
Value comes along the way



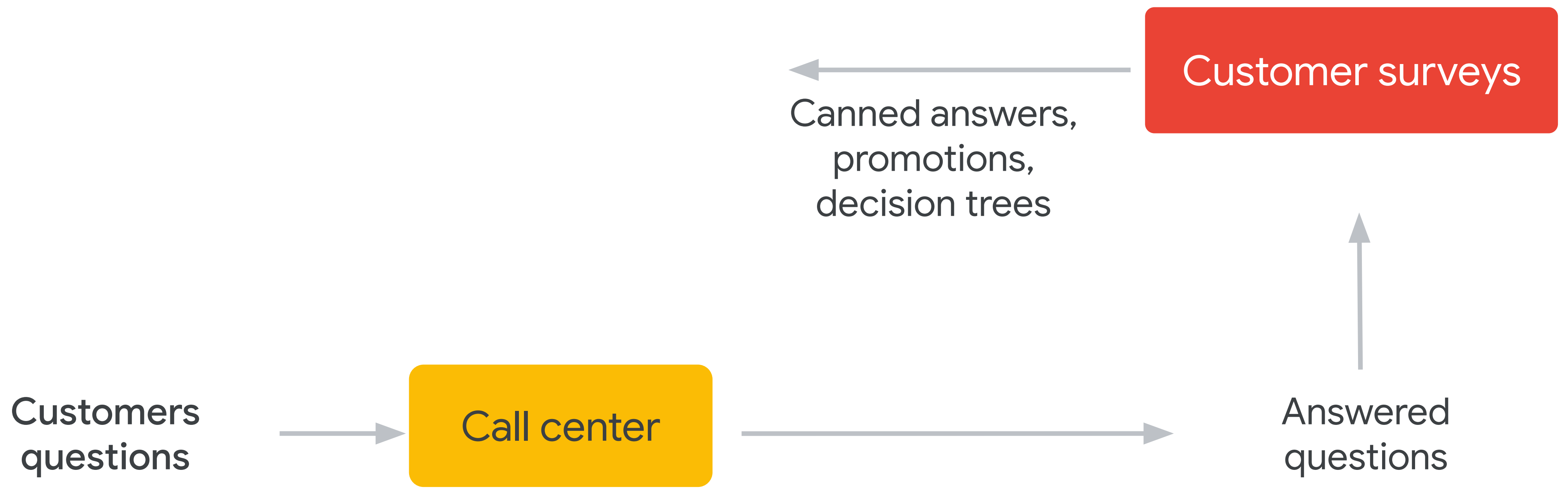
Evolution of a business process



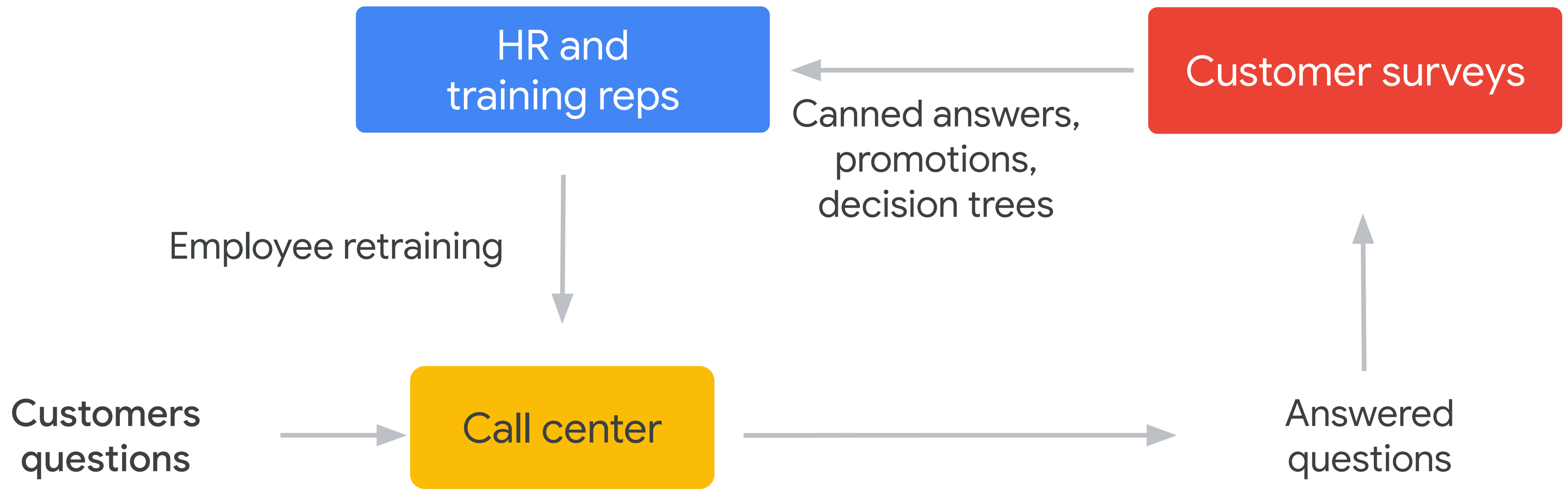
Example: Call center feedback loop



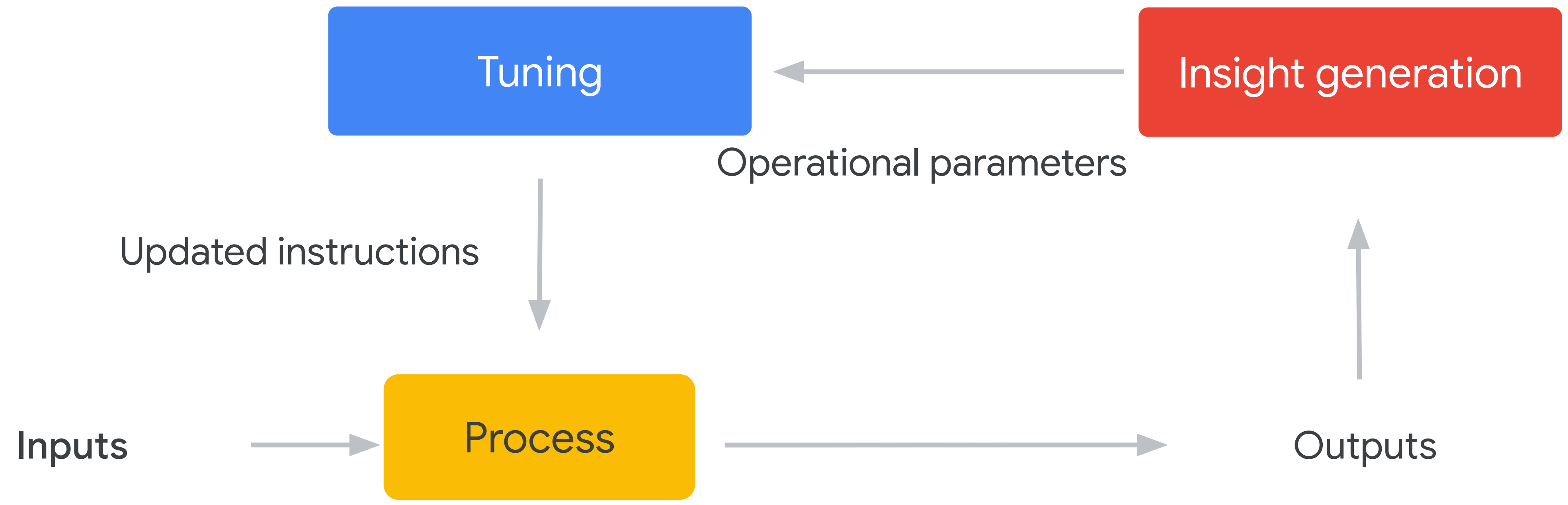
Example: Call center feedback loop



Example: Call center feedback loop



General feedback loop

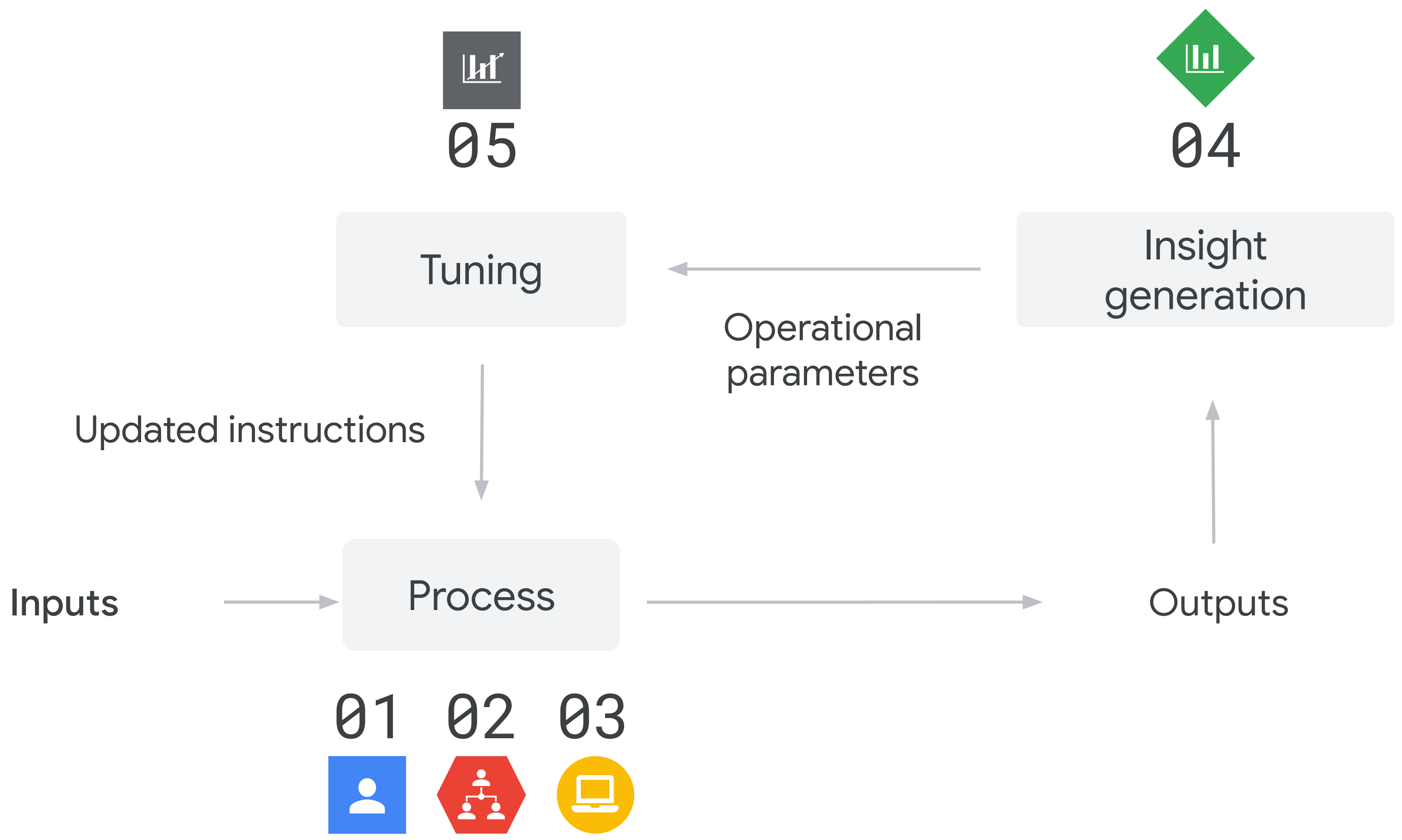


Path to ML: The 5 phases

How change happens in phases:

- 01 Individual contributor
- 02 Delegation
- 03 Digitization
- 04 Big data and analytics
- 05 Machine learning

General feedback loop



Path to ML: The 5 phases

Google reception in San Francisco, CA

1 2 3 4 5

Individual contributor

Path to ML: The 5 phases

Store checker

1 2 3 4 5

Delegation

Path to ML: The 5 phases

ATM

1 — 2 — 3 — 4 — 5

Digitization

Path to ML: The 5 phases

Efficient car manufacturing

1 2 3 4 5

Big data and analytics

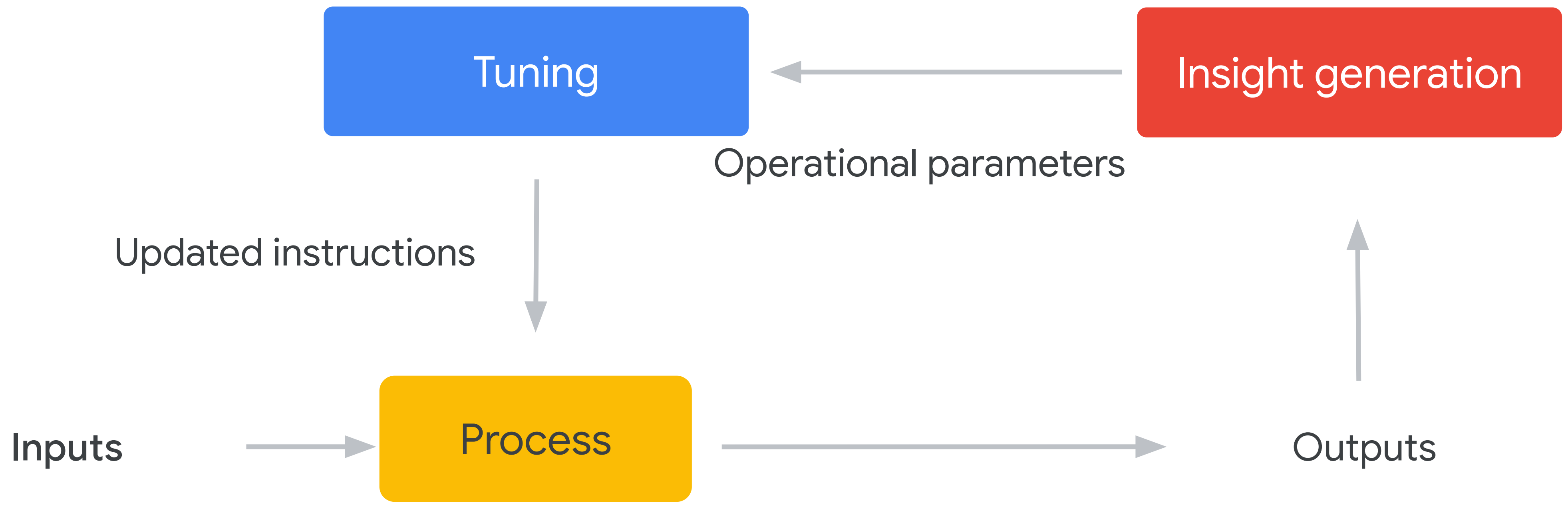
Path to ML: The 5 phases

YouTube recommendation engine

1 2 3 4 5

Machine learning

Path to ML: Your turn



The path to ML

1. Individual contributor
2. Delegation
3. Digitization
4. Big data and analytics
5. Machine learning

Prototype and try out ideas

1

Individual contributor

Dangers of skipping this step:

- Inability to scale.
- Product heads make big, incorrect assumptions that are hard to change later.

Dangers of lingering too long here:

- One person gets skilled and then leaves.
- Fail to scale up the process to meet demand in time.

Gently ramp up to include more people

2

Delegation

Dangers of skipping this step:

- Not forced to formalize the process.
- Inherent diversity in human responses become a testbed--great product learning opportunity.
- Great ML systems will need humans in the loop.

Dangers of lingering too long here:

- Paying a high marginal cost to serve each user.
- More voices will say automation isn't possible.
- Organizational lock-in.

Automate mundane parts of the process

3

Digitization

Dangers of skipping this step:

- You will always need infrastructure.
- IT project and ML success tied and the whole project will fail if either does.

Dangers of lingering too long here:

- Your competitors are collecting data and tuning their offers from these new insights.

Measure and achieve data-driven success

4

Big data and analytics

Dangers of skipping this step:

- Unclean data means no ML training.
- You can’t measure success.

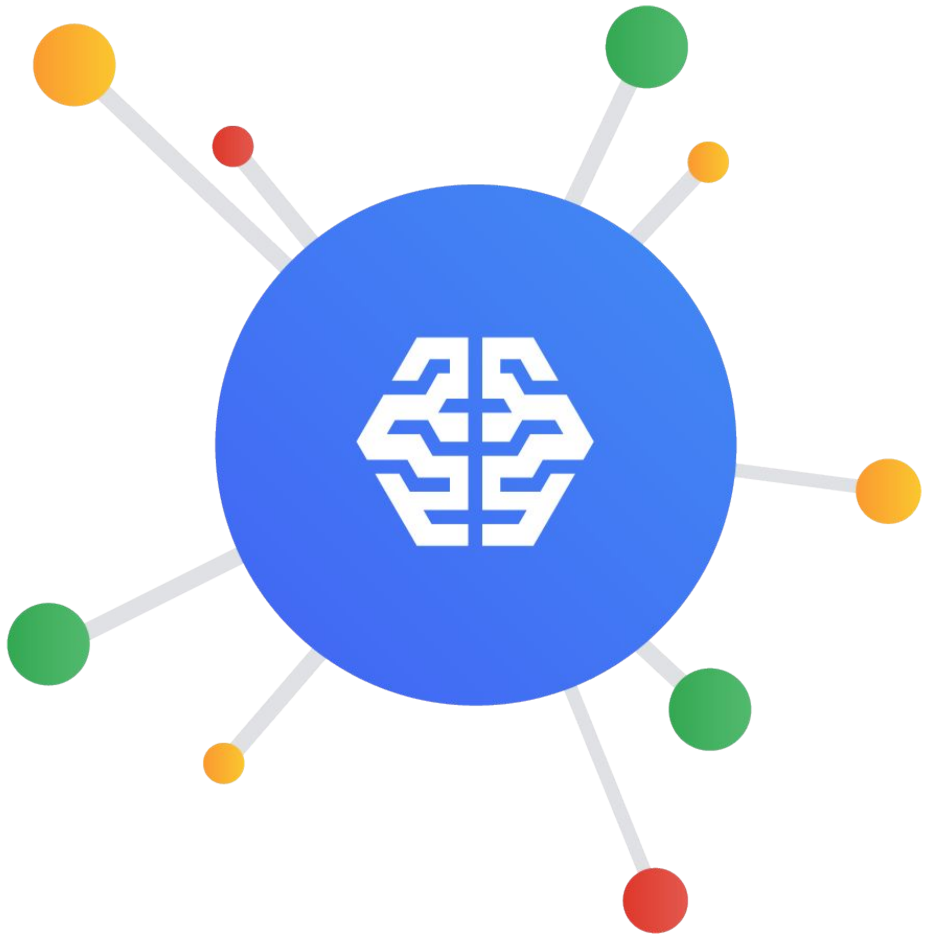
Dangers of lingering too long here:

- Limit the complexity of problems you can solve.

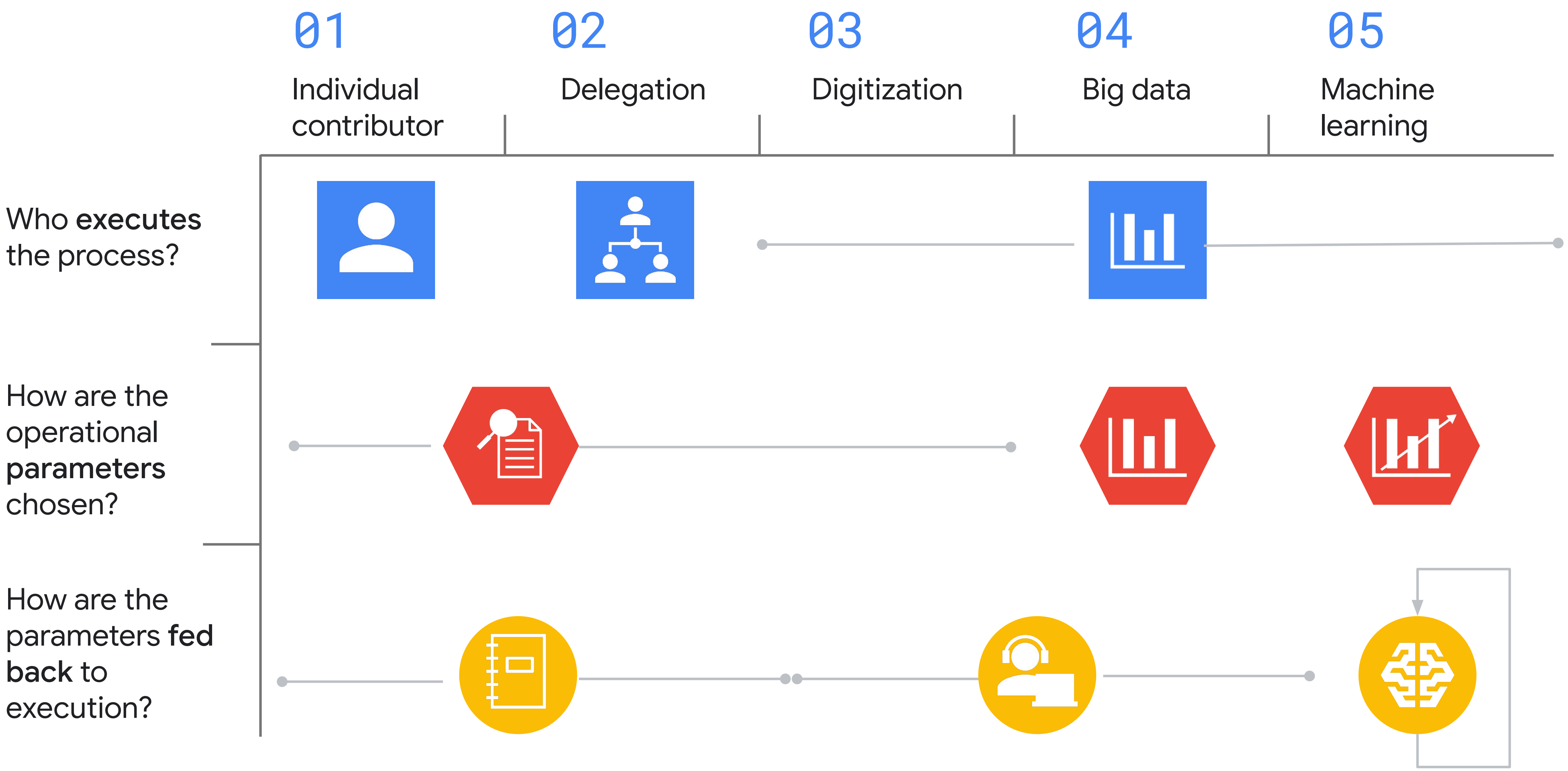
Automated feedback loop that can outpace human scale

5

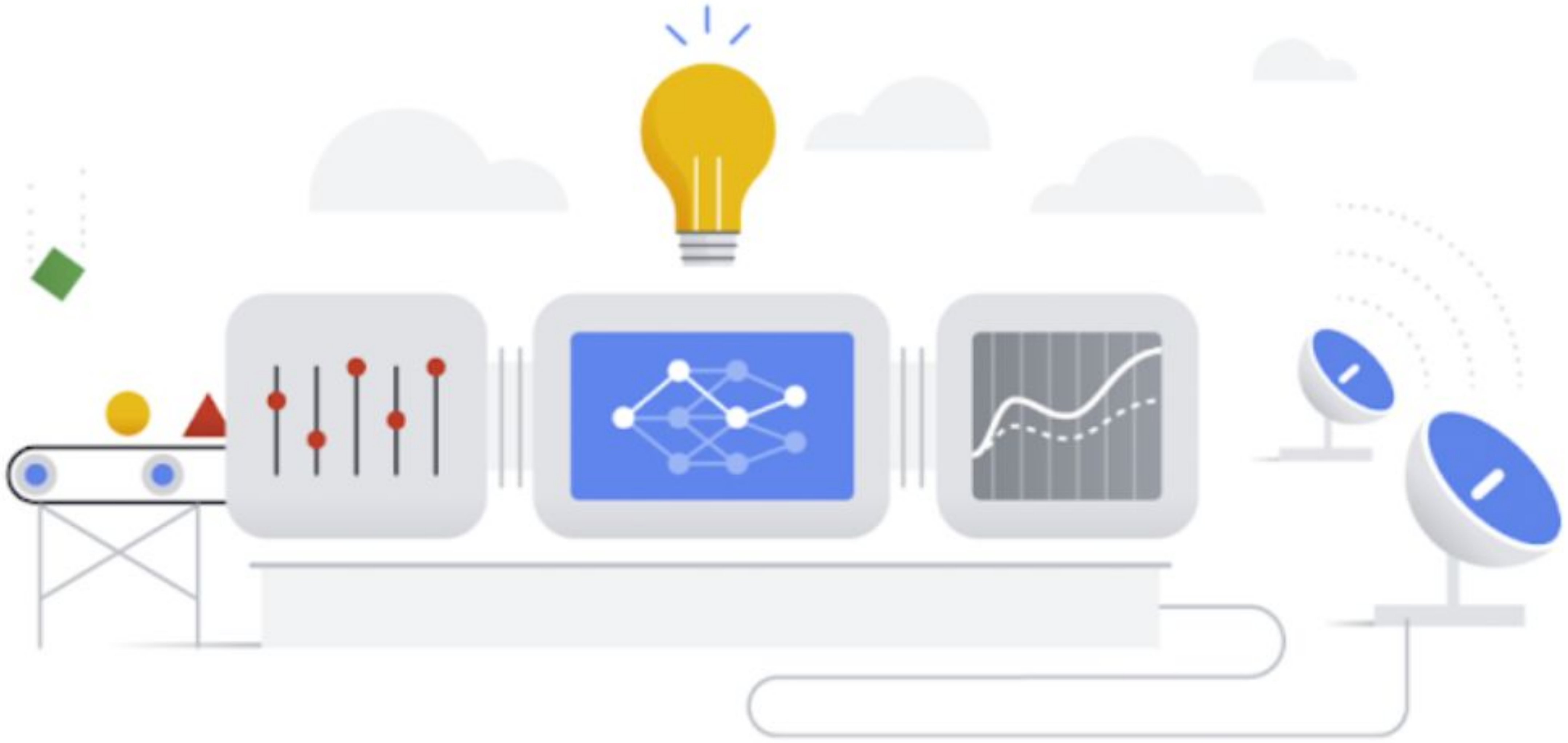
Machine learning



Reviewing the path to ML: 5 phases



Machine Learning Development with Vertex AI



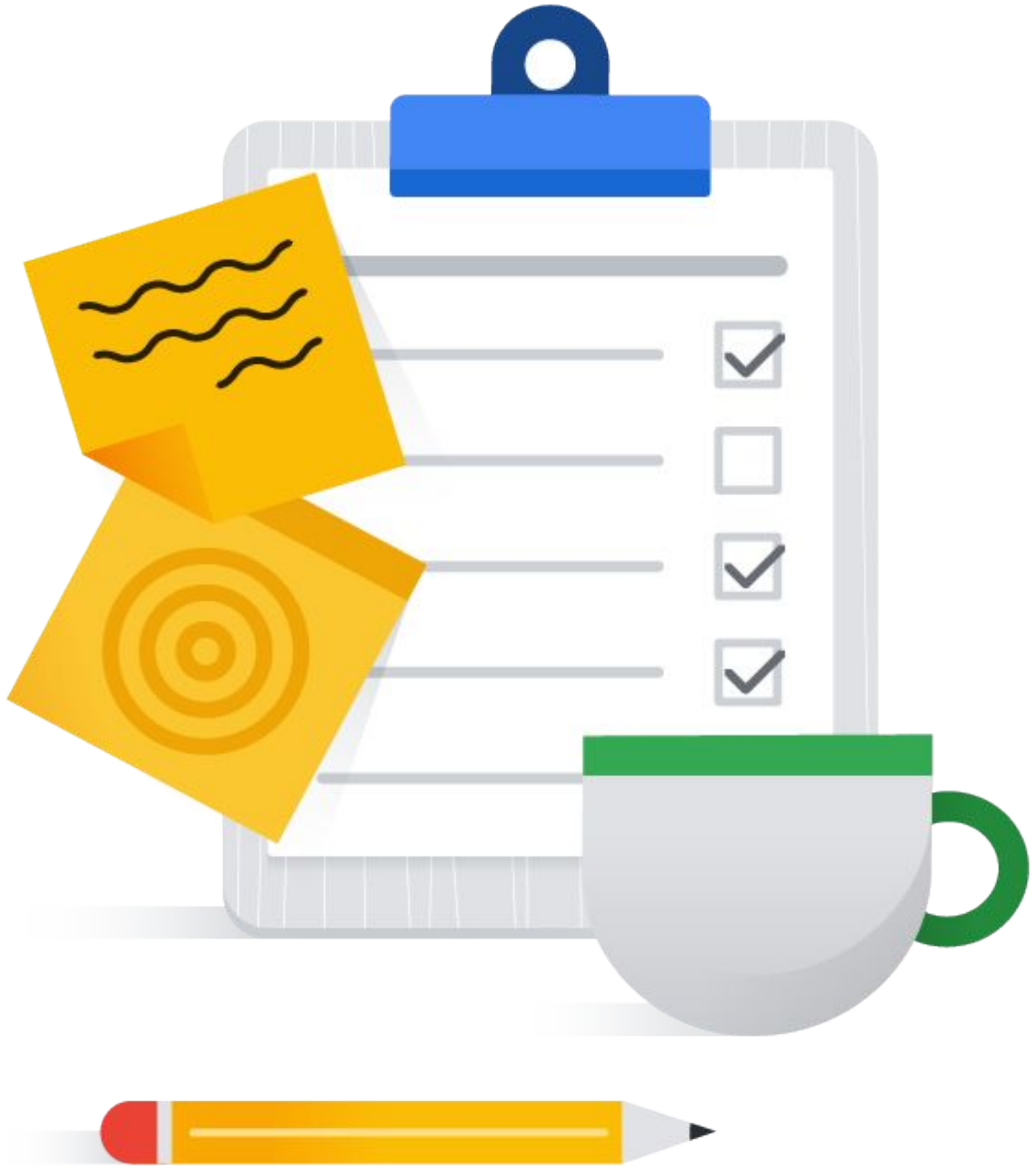
All machine learning starts with a business requirement or goal you are trying to solve.

To build a machine learning model for production

1. Identify goal
2. Acquire + explore data + prepare data
3. Build a model
4. Train the model
5. Evaluate the model
6. Deploy + monitor + maintain

In this module, you learn to ...

01	Move from experimentation to production
02	Describe the components of Vertex AI
03	Describe tools to interact with Vertex AI



Moving from experimentation to production

Typical ML development during experimentation

Framing the problem | Prepare training data | Experimenting | Evaluating the model

Typical ML development during experimentation

Framing the problem | **Prepare training data** | Experimenting | Evaluating the model

Typical ML development during experimentation

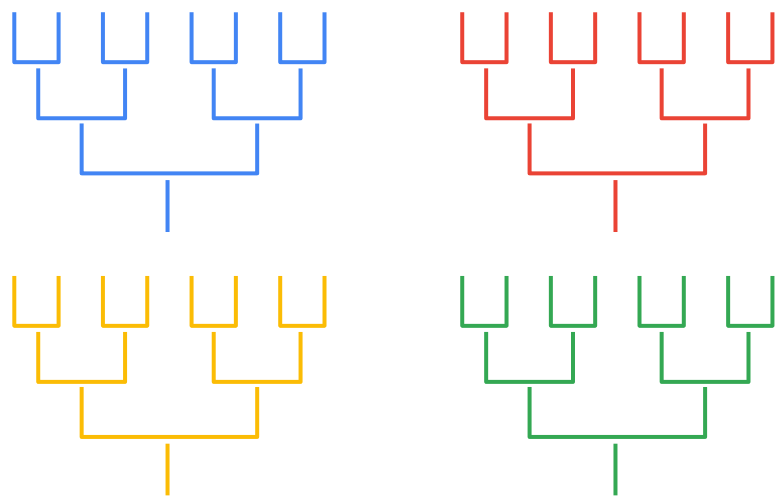
Framing the problem | Prepare training data | **Experimenting** | Evaluating the model

- Model A
- Model B
- Model C

Experiment

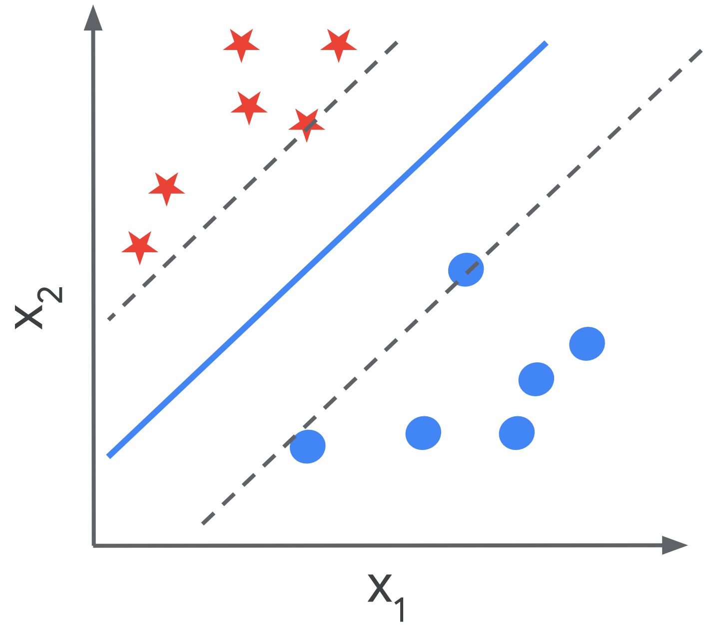
Model A

Random forests



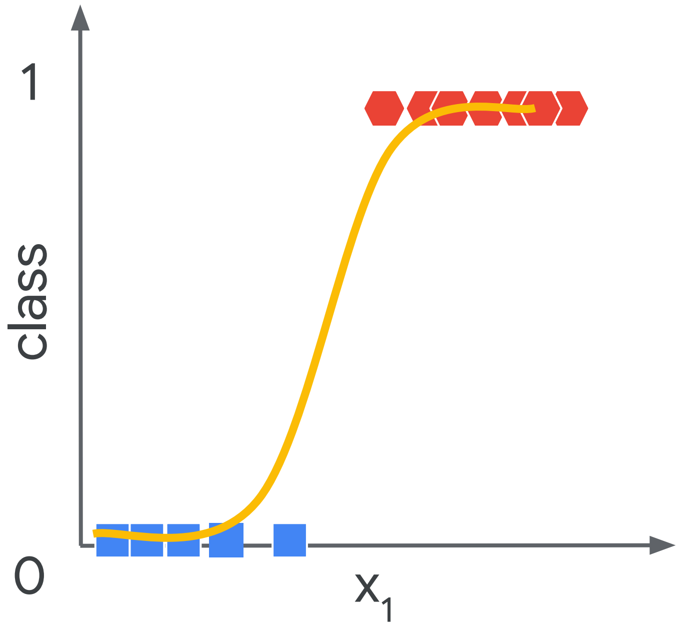
Model B

Support vector machines



Model C

Logistic regression



Typical ML Development during Experimentation

Framing the problem

Prepare training data

Experimenting

Evaluating the model

- Recall
- F1 Score
- Precision
- Cross Entropy

Typical ML development during experimentation

Different architectures

Different input data sets

Different hyperparameters

Different hardware

Typical ML development during experimentation

Different architectures	Different input data sets	Different hyperparameters	Different hardware
• CNN • RNN • Sorting/Clustering • GANs			

Typical ML development during experimentation

Different architectures	Different input data sets	Different hyperparameters	Different hardware
	• Numerical data sets • Bivariate data sets • Multivariate data sets • Categorical data sets • Correlation data sets		

Typical ML development during experimentation

Different architectures

Different input data sets

Different hyperparameters

- Learning rate
- Number of layers
- Num_estimators
- Max_depth

Different hardware

Typical ML development during experimentation

Different architectures

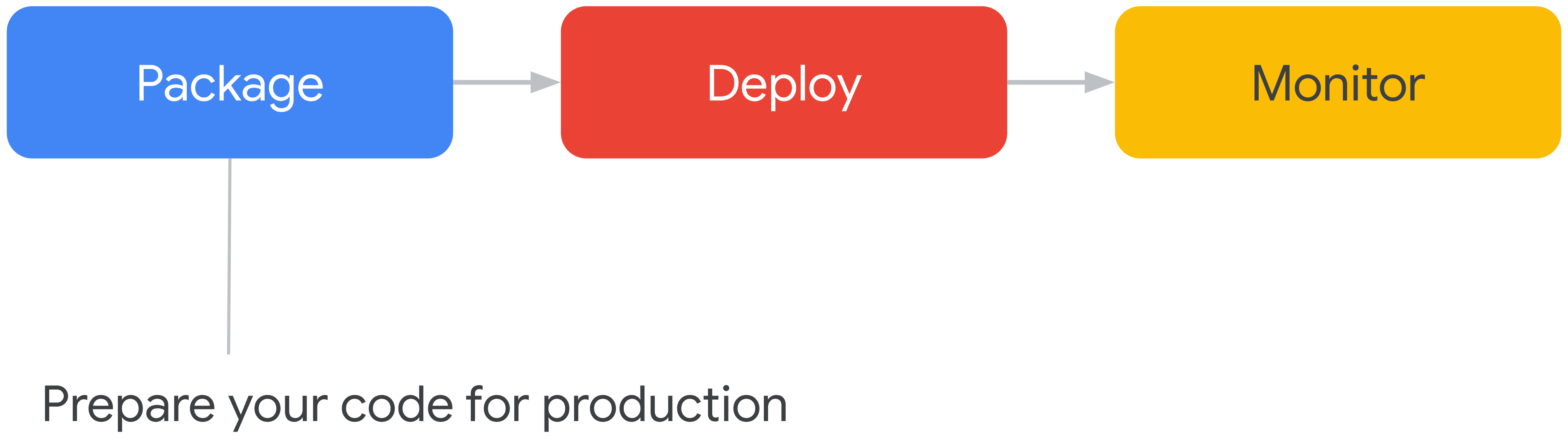
Different input data sets

Different hyperparameters

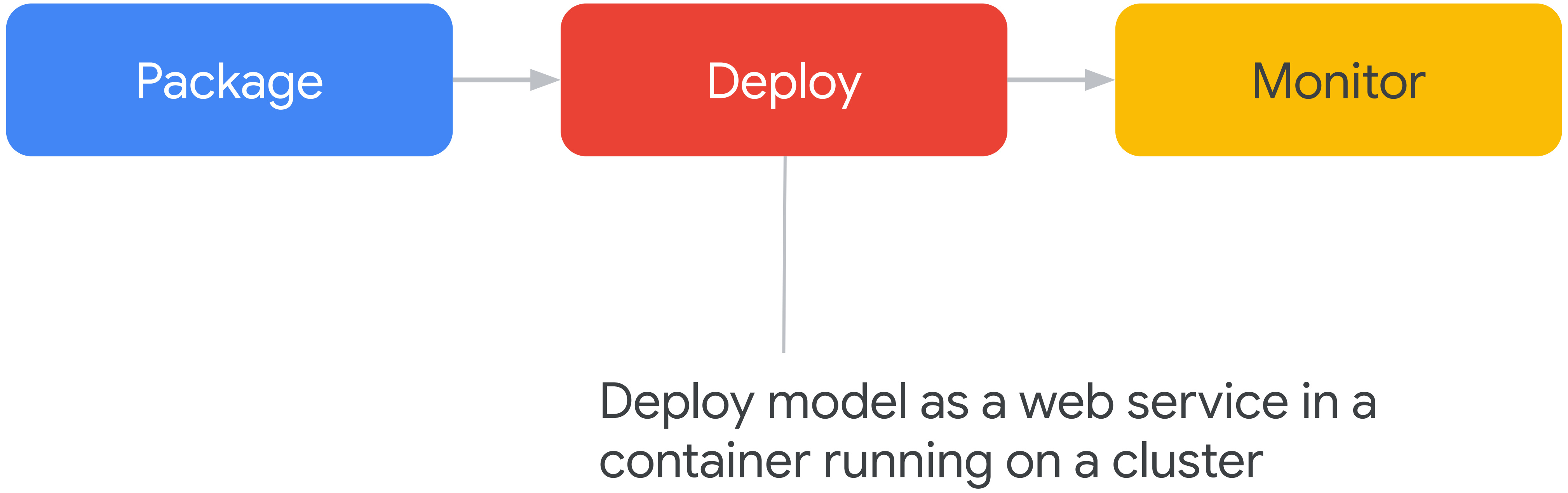
Different hardware

- CPUs
- GPUs
- TPUs

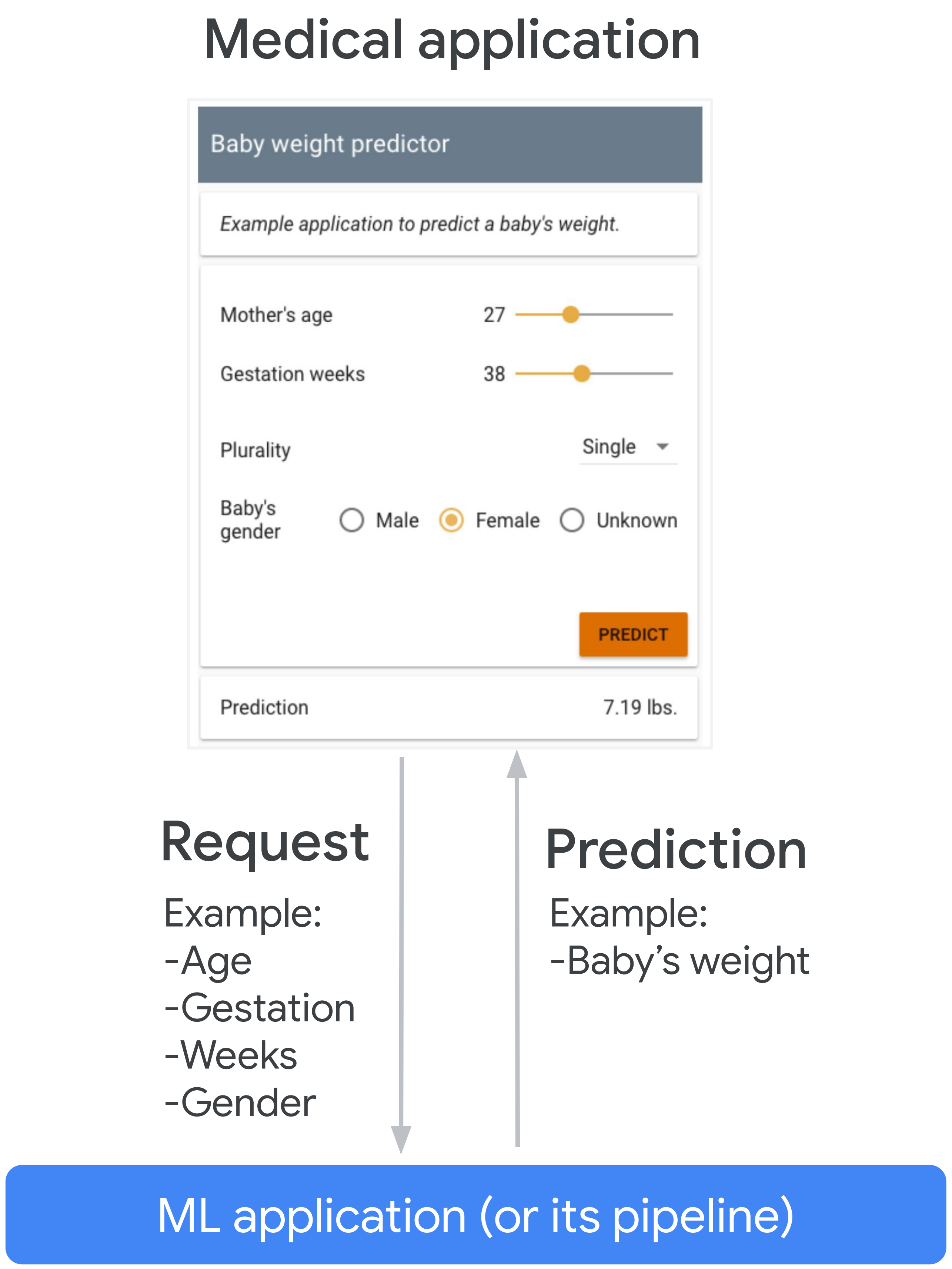
Moving from experimentation to production requires packaging, deploying, and monitoring your model



Moving from experimentation to production requires packaging, deploying, and monitoring your model



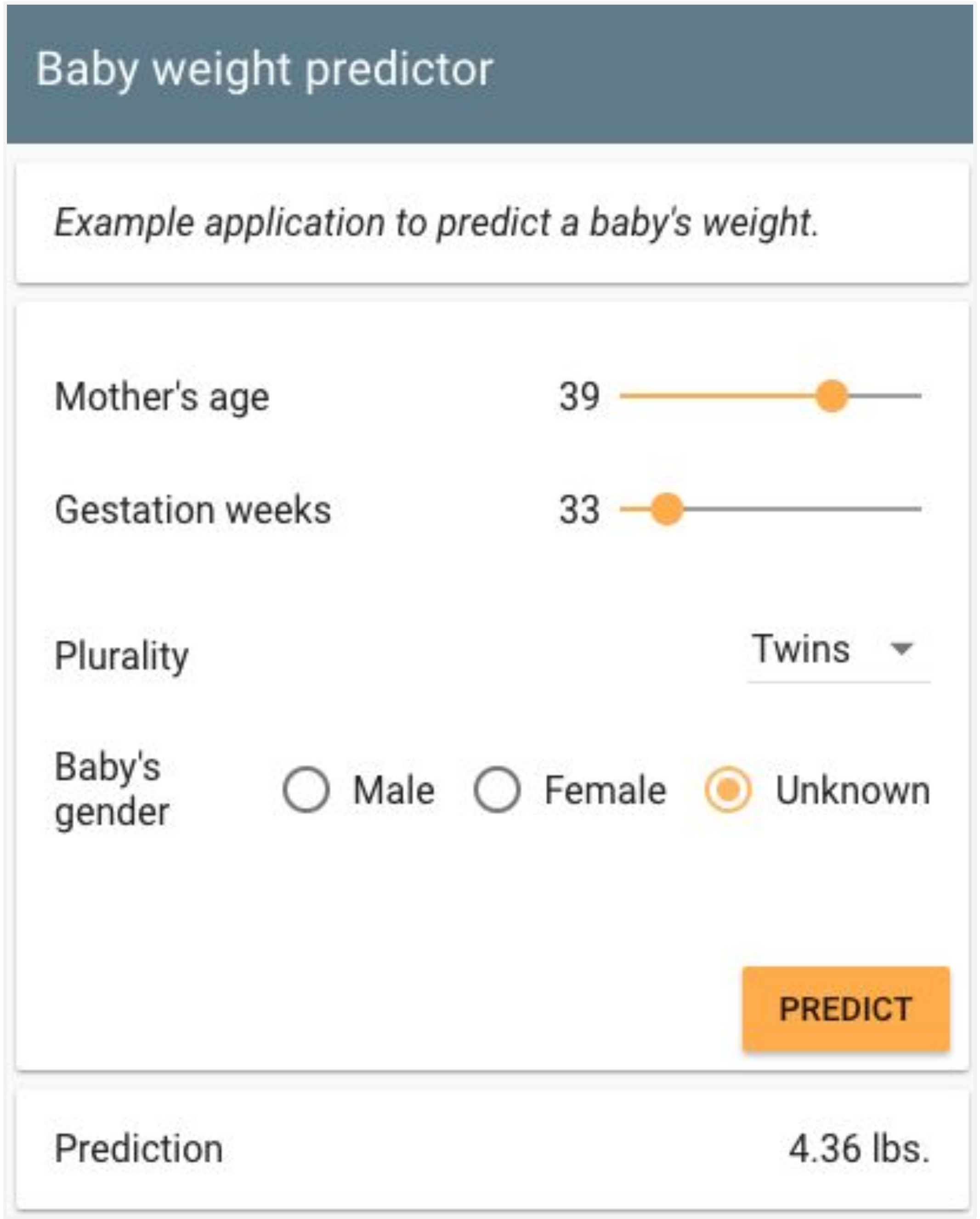
ML application generates a REST service for use by a medical application



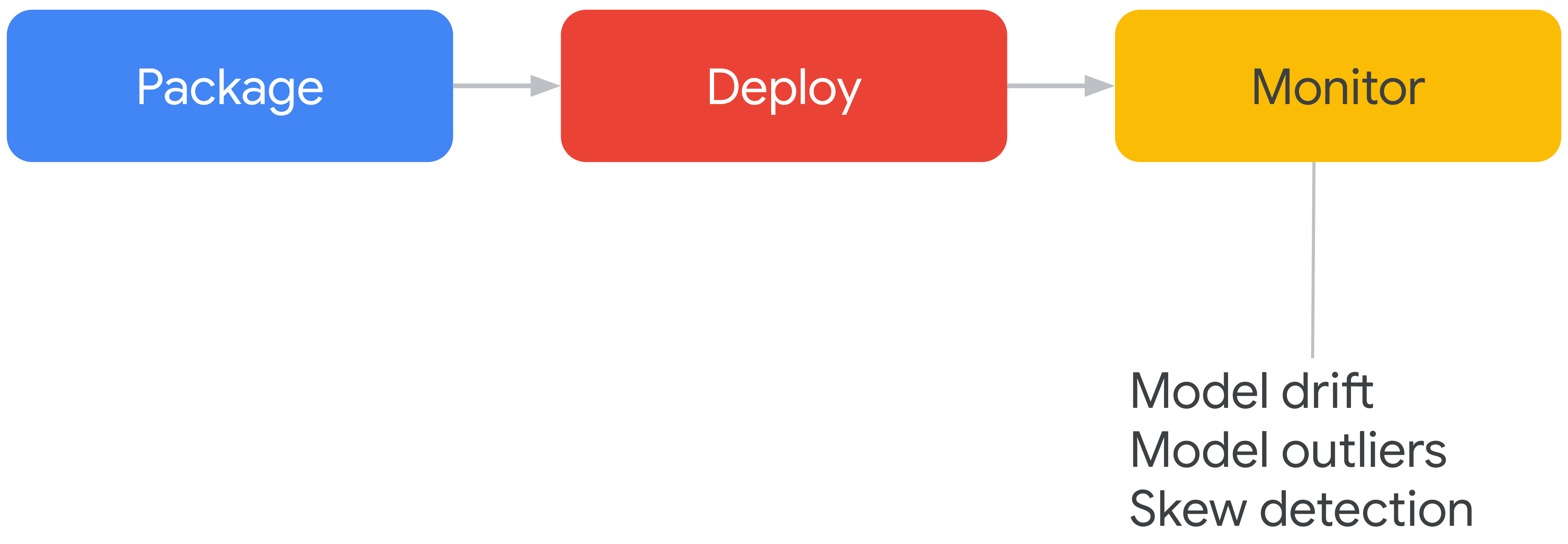
Why model monitoring

- Stale model - The underlying data distribution has shifted over time.

- Misconfigured model in production deployment.



Moving from experimentation to production requires packaging, deploying, and monitoring your model



To build a machine learning model for production

1. Identify goal
2. Acquire + explore data + prepare data
3. Build a model
4. Train the model
5. Evaluate the model
6. Deploy + monitor + maintain

ML product knowledge required

Start here

Build model/ MVP style app

ML product knowledge required

Start here

- Build model/ MVP style app
- Upload model to Google Storage
- Host model on AI Platform

ML product knowledge required

Start here

- Build model/ MVP style app
- Upload model to Google Storage
- Host model on AI Platform
- Create Service Account to access model
- Wrap app in Docker container
- Push Docker container to GCR

ML product knowledge required

Start here

- Build model/ MVP style app
- Upload model to Google Storage
- Host model on AI Platform
- Create Service Account to access model
- Wrap app in Docker container
- Push Docker container to GCR
- Deploy GCR container to App Engine
- Monitor app with Monitoring

What is there to unify?

Dataset is

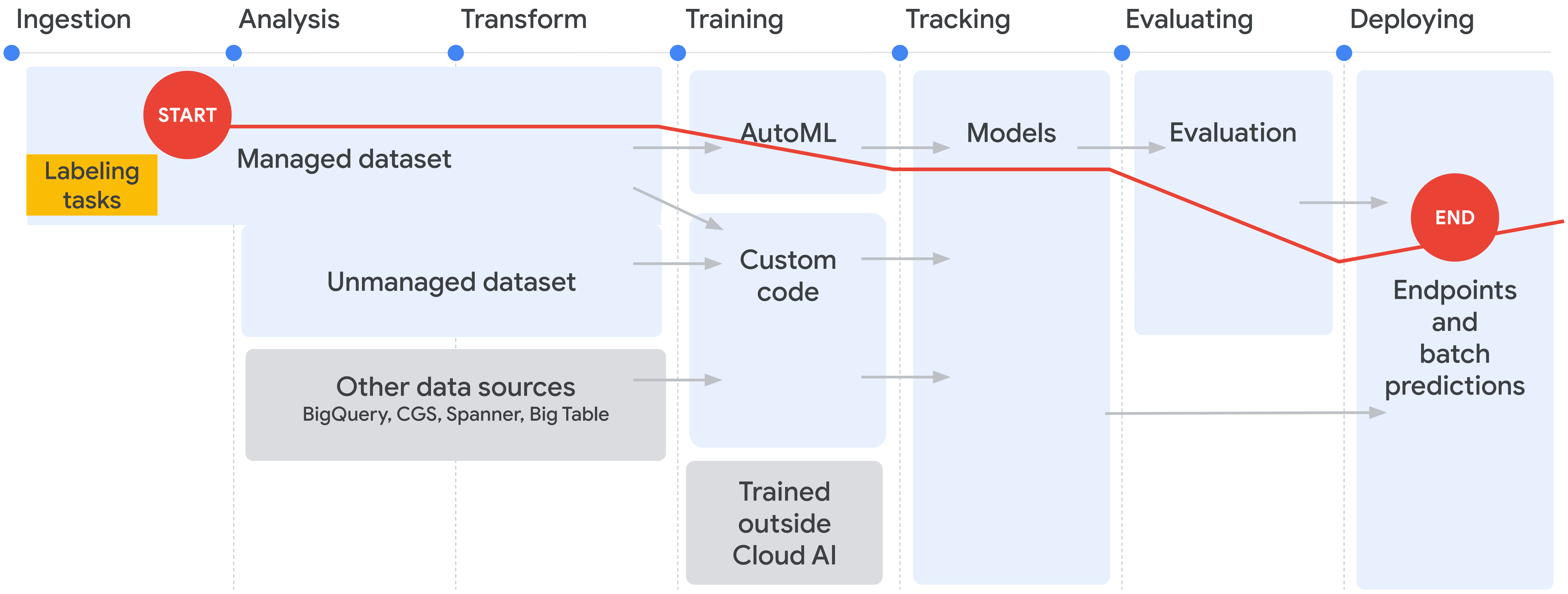
- Created
- Ingested
- Analyzed
- Cleaned (ETL or ELT)

Model is

- **Trained,** which includes experimentation and hypothesis. testing, and hyperparameter tuning.
- **Versioned** and rebuilt when there is new data, on a schedule, or when the code changes (ML Ops).
- **Evaluated** and compared to existing model versions.
- **Deployed** and used for online and batch predictions.

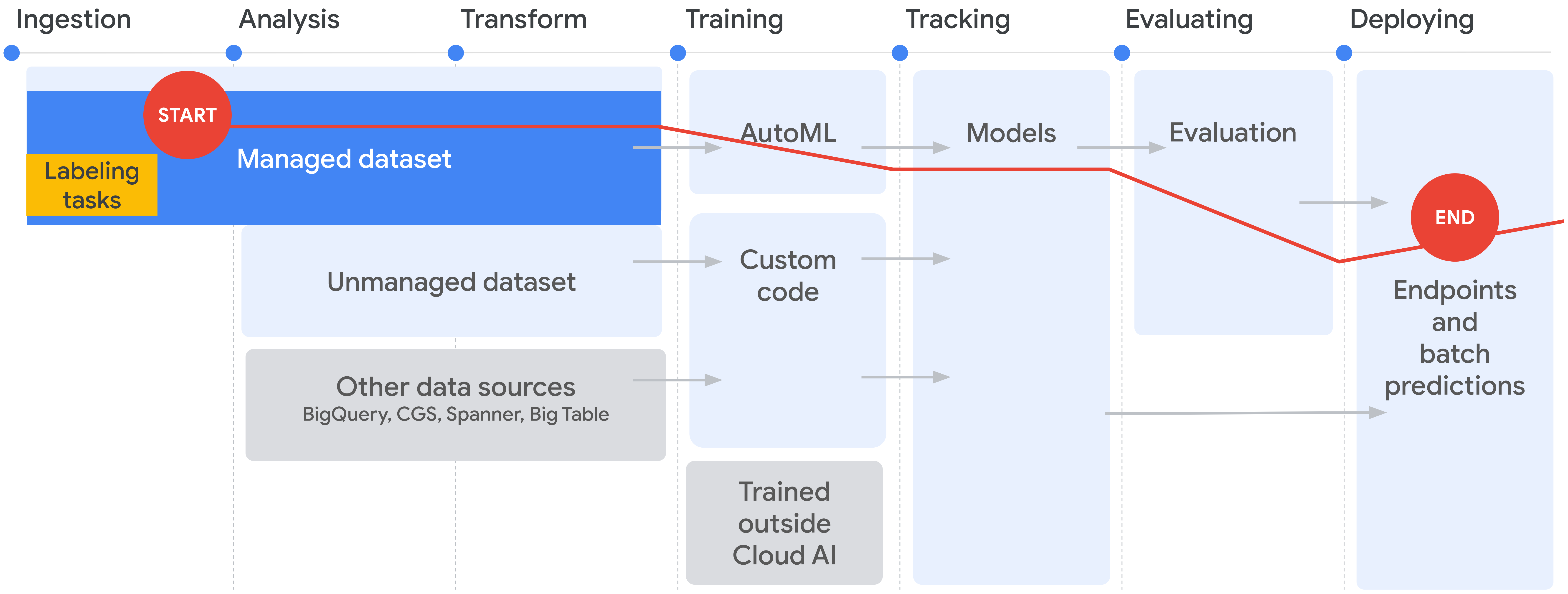
Source: Giving Vertex AI, the New Unified ML Platform on Google Cloud, a Spin, by Lak Lakshmanan

Vertex AI



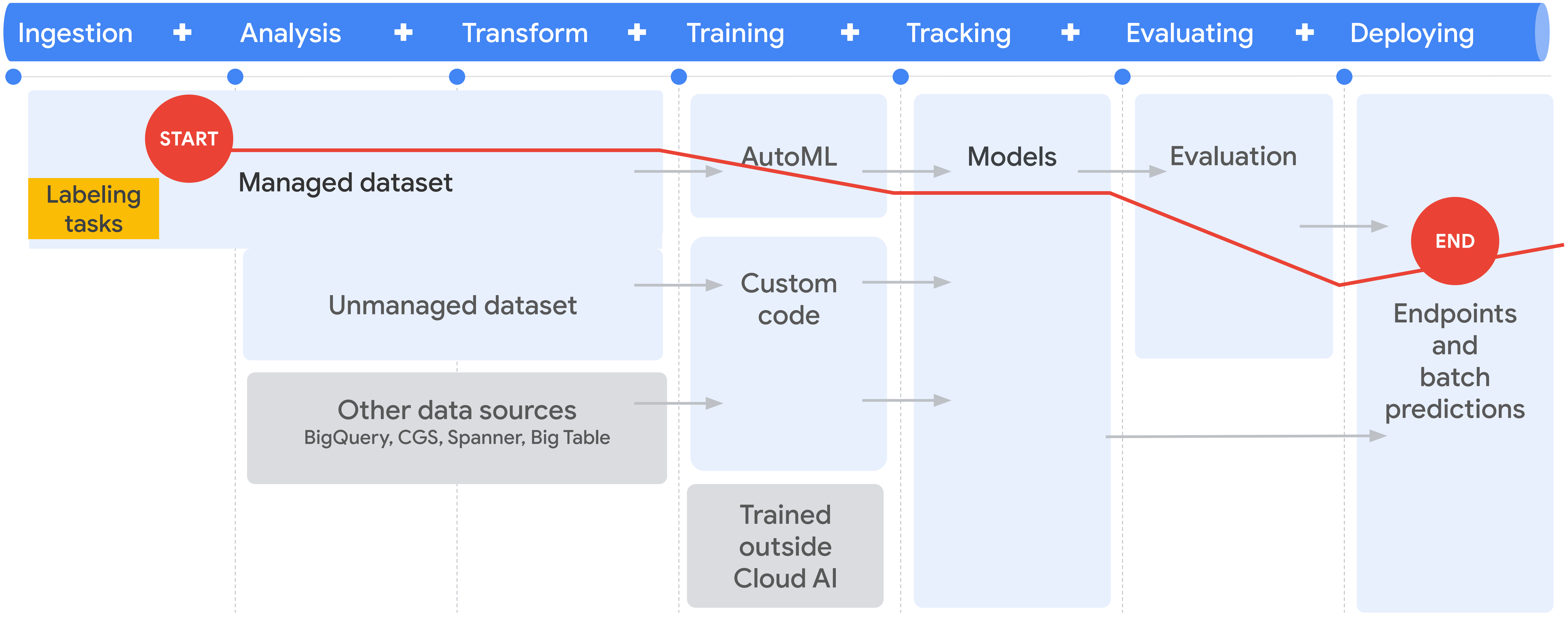
Vertex AI provides a unified set of APIs for the ML lifecycle. Diagram courtesy Henry Tappen and Brian Kobashikawa

Vertex AI



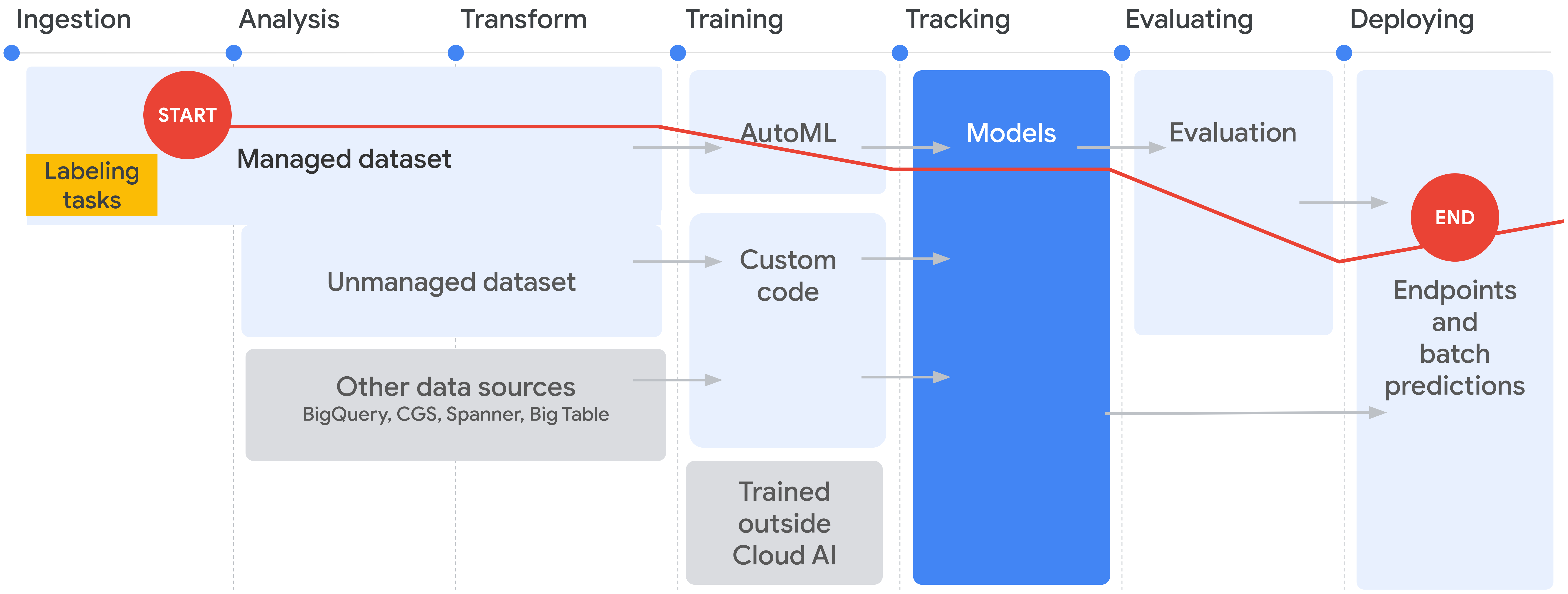
Vertex AI provides a unified set of APIs for the ML lifecycle. Diagram courtesy Henry Tappen and Brian Kobashikawa

Vertex AI



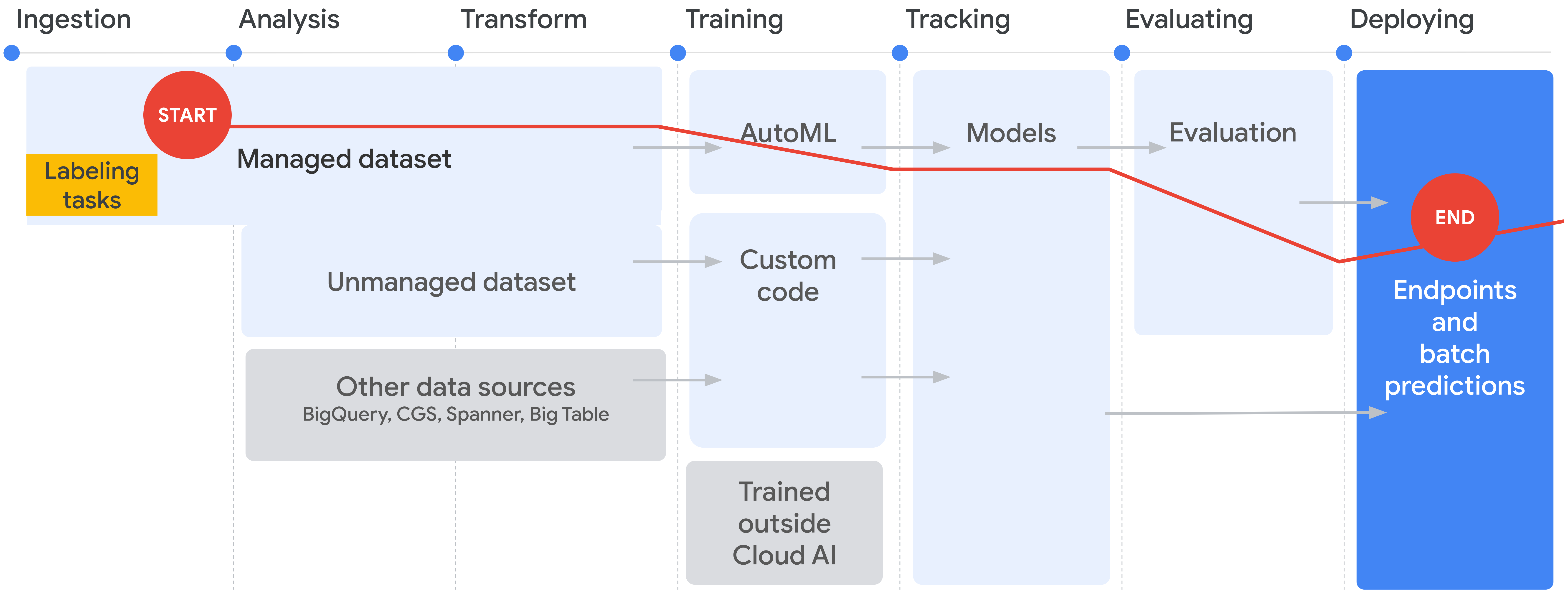
Vertex AI provides a unified set of APIs for the ML lifecycle. Diagram courtesy Henry Tappen and Brian Kobashikawa

Vertex AI



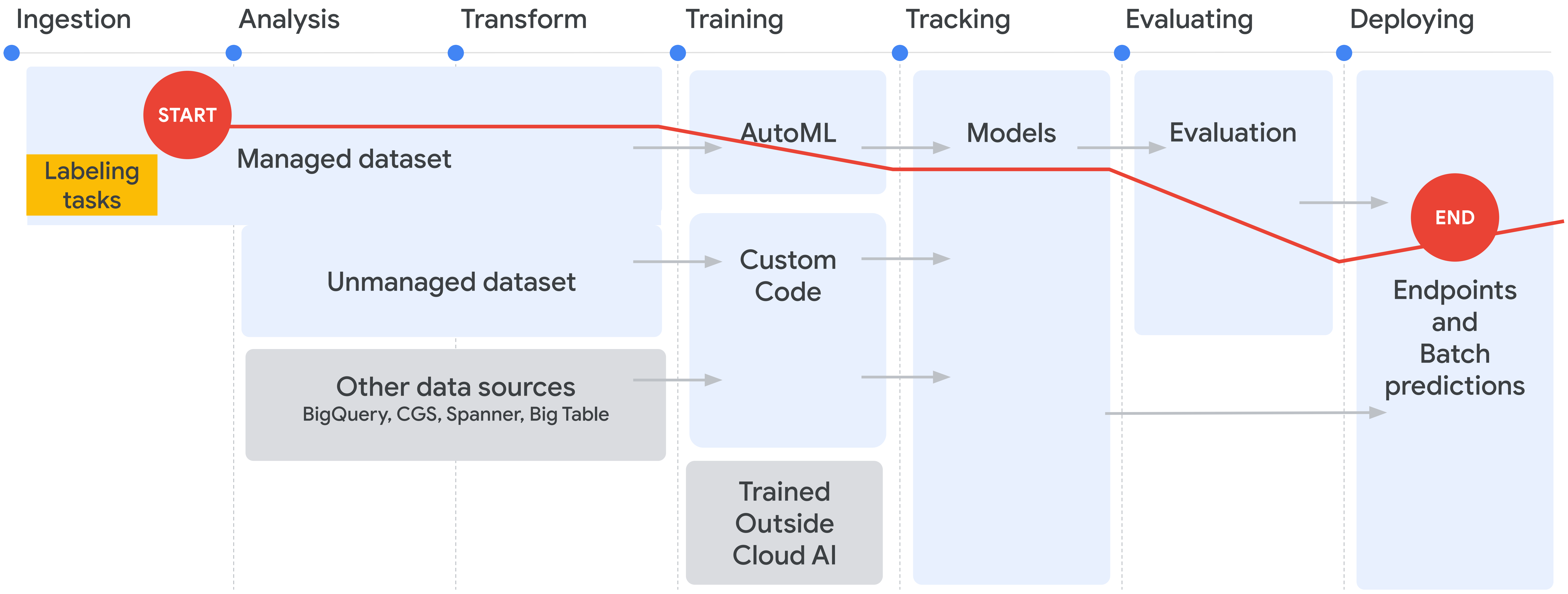
Vertex AI provides a unified set of APIs for the ML lifecycle. Diagram courtesy Henry Tappen and Brian Kobashikawa

Vertex AI



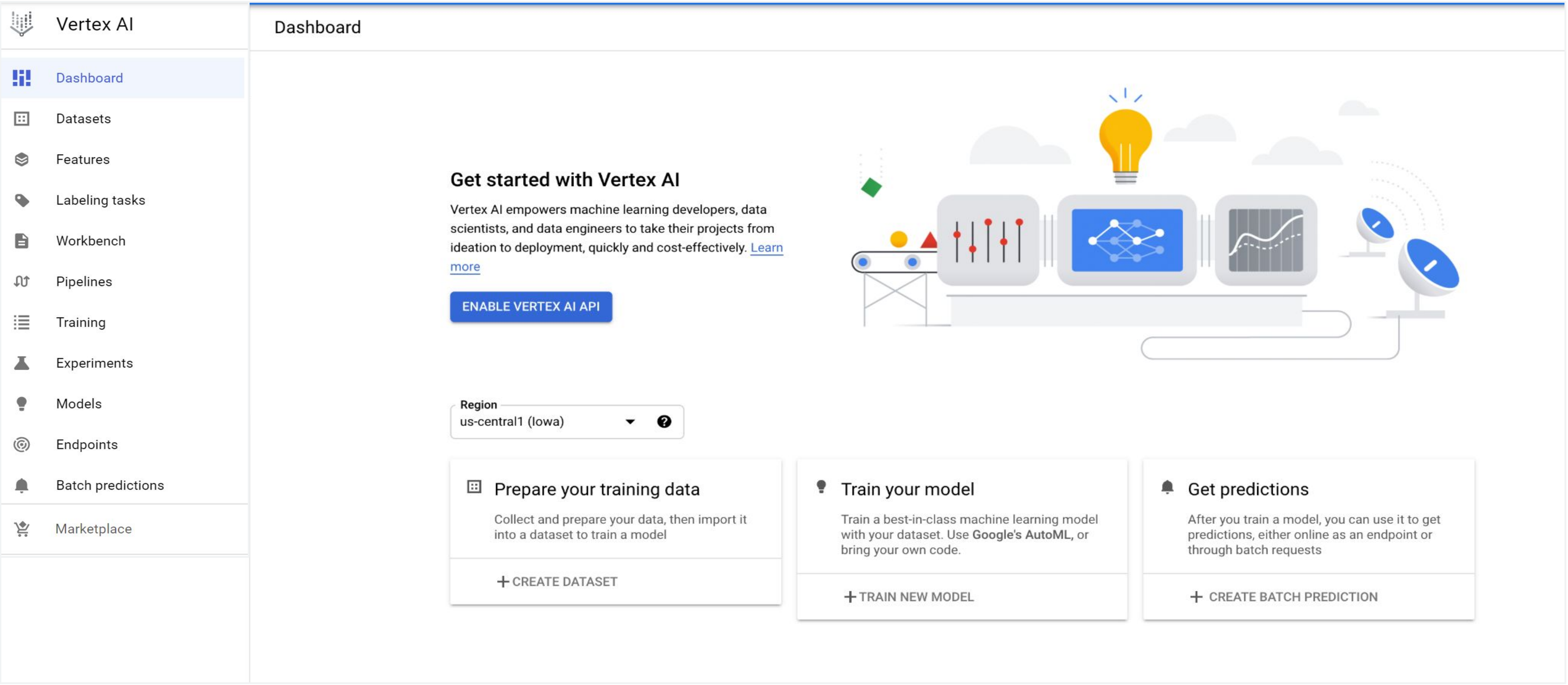
Vertex AI provides a unified set of APIs for the ML lifecycle. Diagram courtesy Henry Tappen and Brian Kobashikawa

Vertex AI



Vertex AI provides a unified set of APIs for the ML lifecycle. Diagram courtesy Henry Tappen and Brian Kobashikawa

Vertex AI > Dashboard



Choose a training method

AutoML

- Create and train a model with minimal technical effort.
- Quickly prototype models or explore datasets before developing in a custom training application.

Custom training

- Create a training application optimized for your targeted outcome.
- Maintain complete control over training application functionality.
  - Target any objective, use any algorithms, develop your own loss functions or metrics, or make other customizations.

Vertex AI

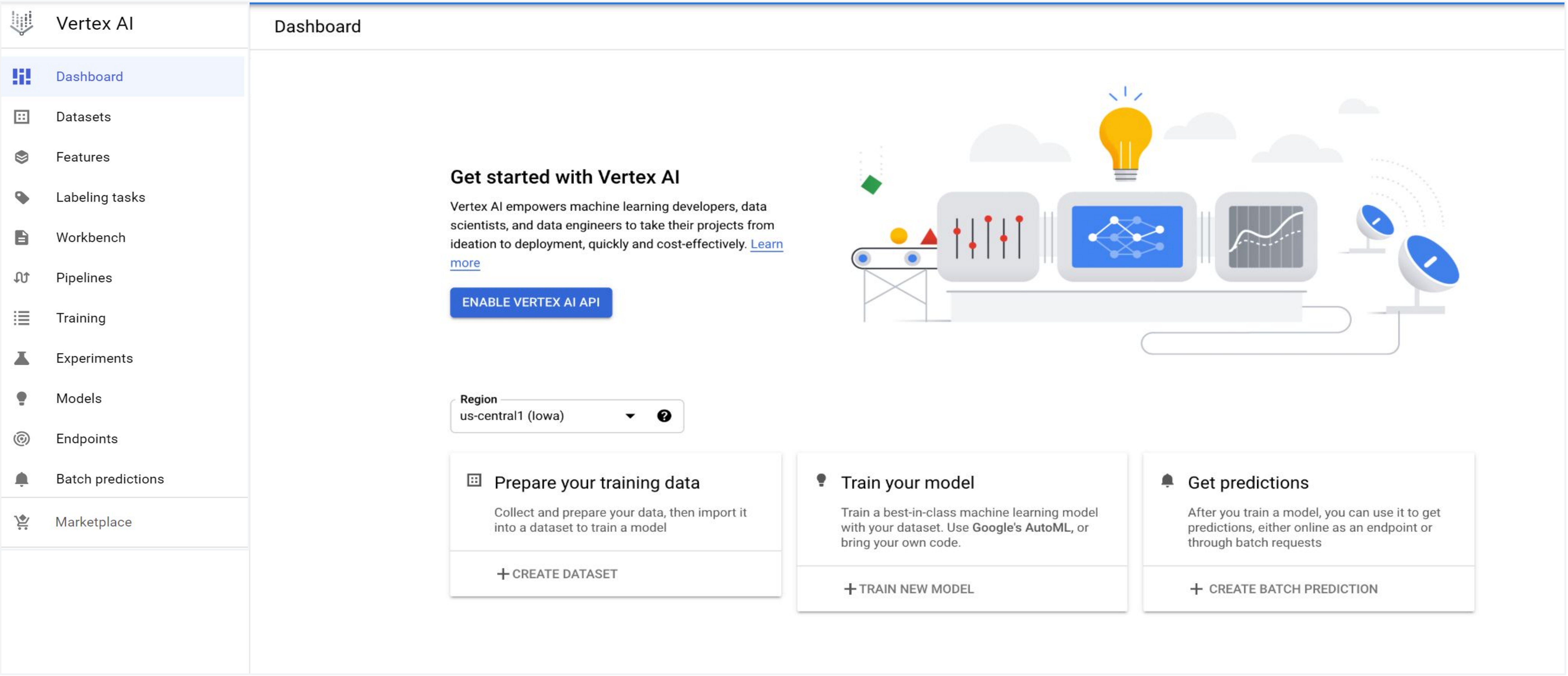
Fast experimentation

Accelerated deployment

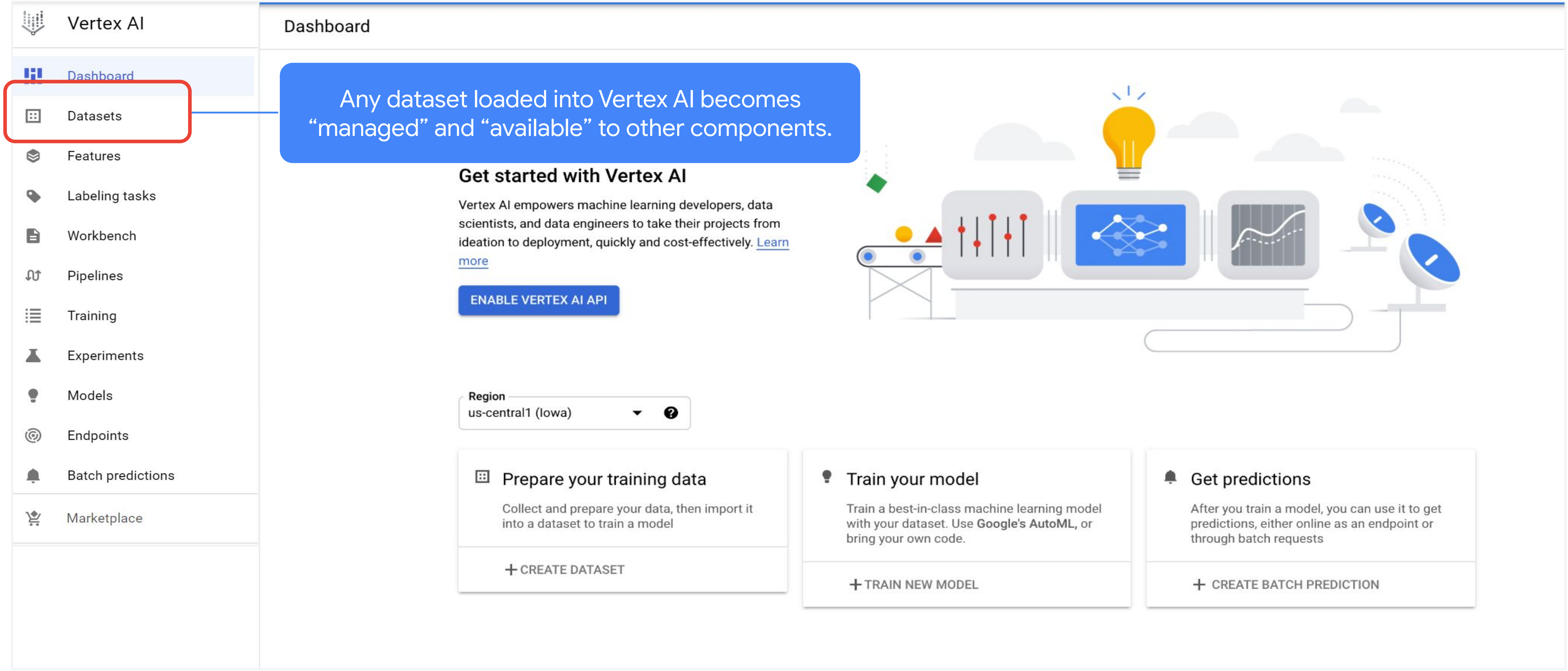
Simplified model management

Vertex AI Components

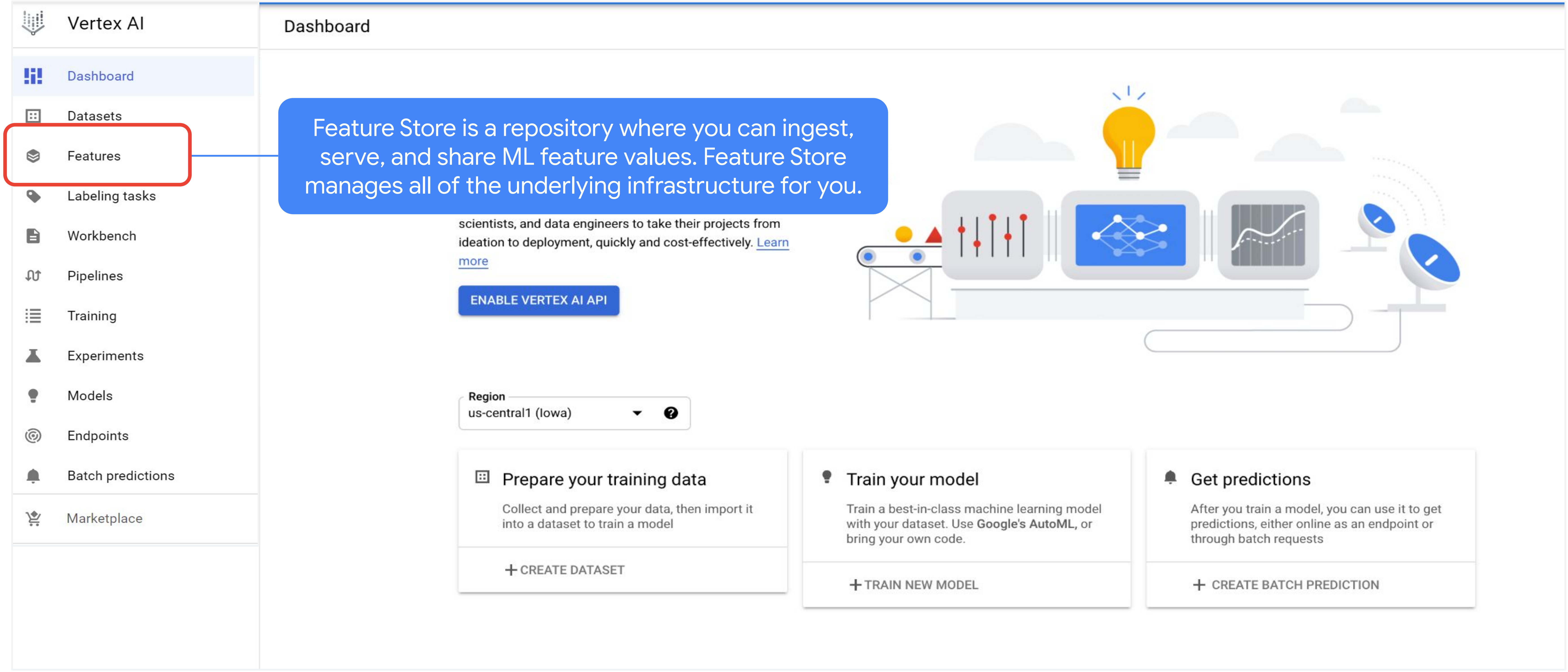
Vertex AI > Dashboard



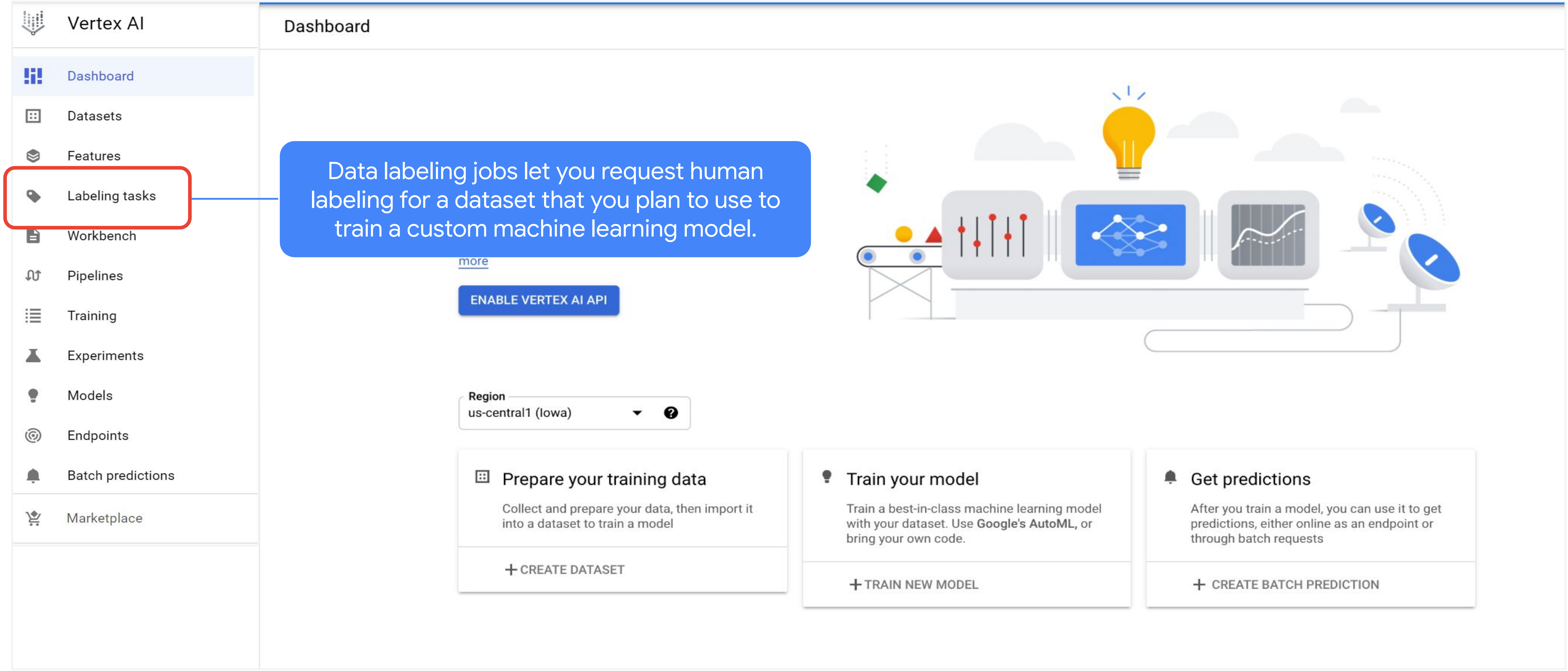
Vertex AI > Datasets



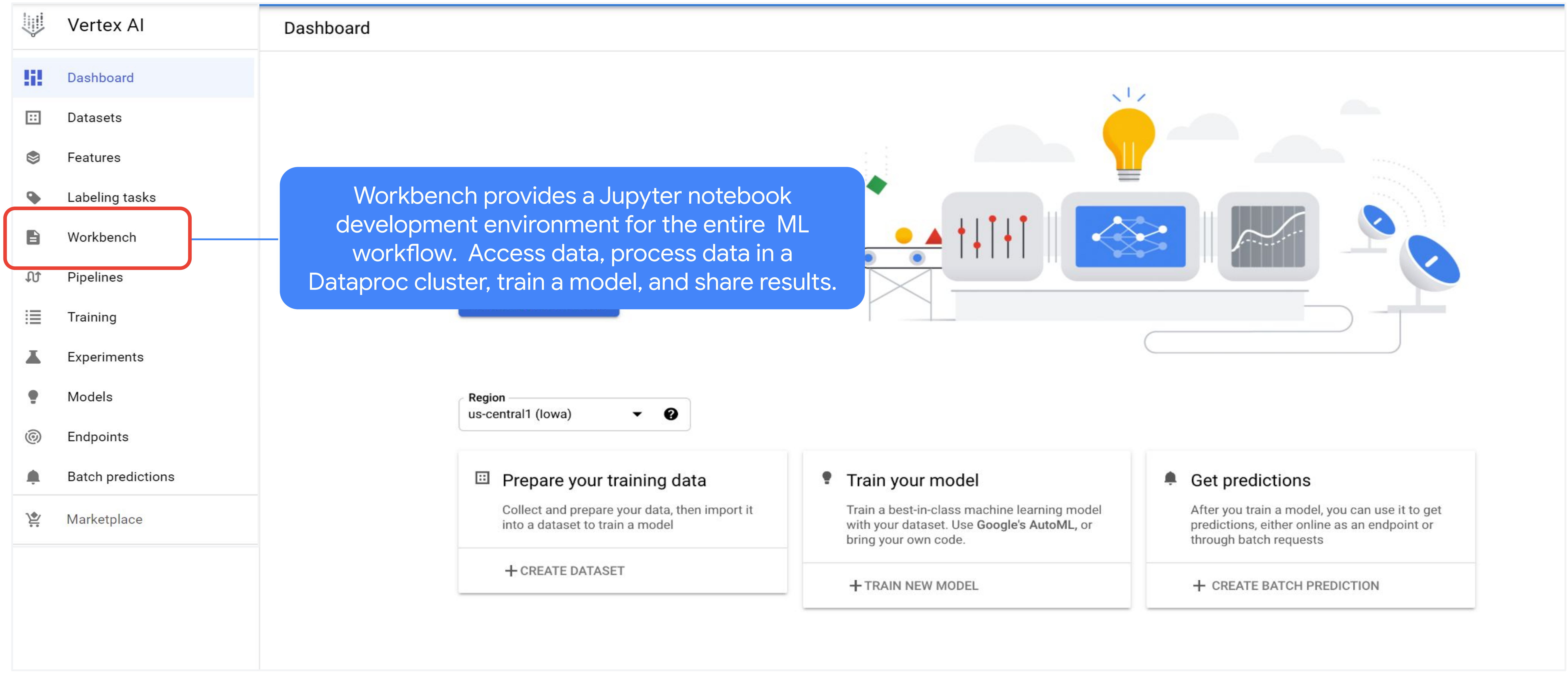
Vertex AI > Features



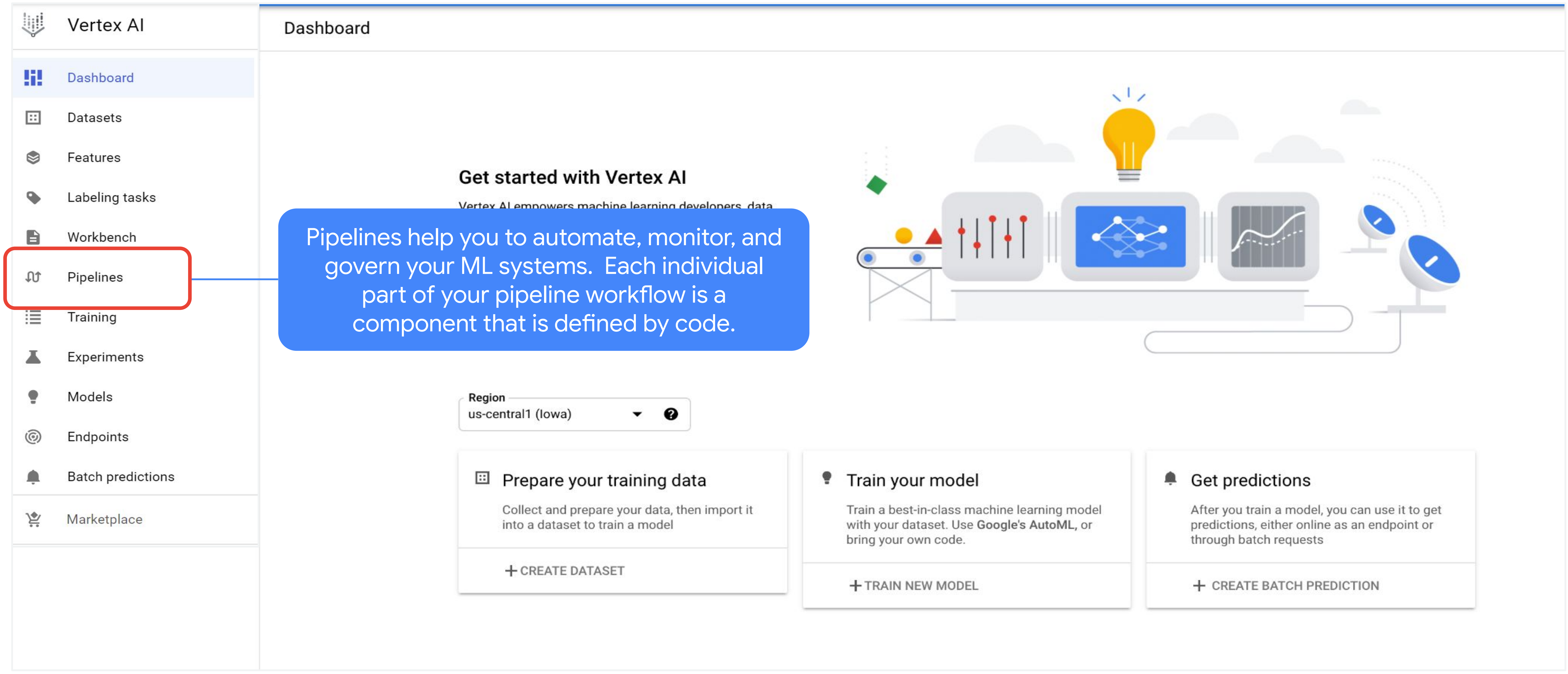
Vertex AI > Labeling tasks



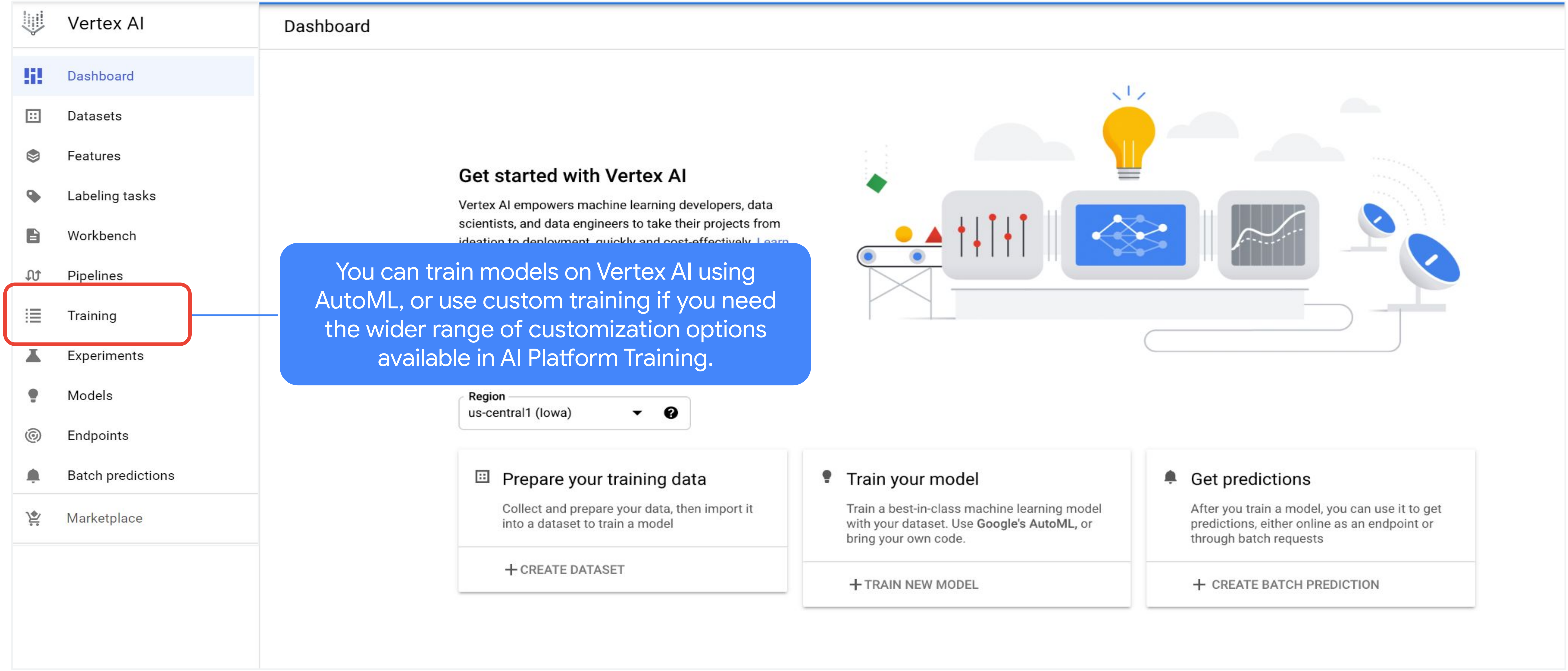
Vertex AI > Workbench



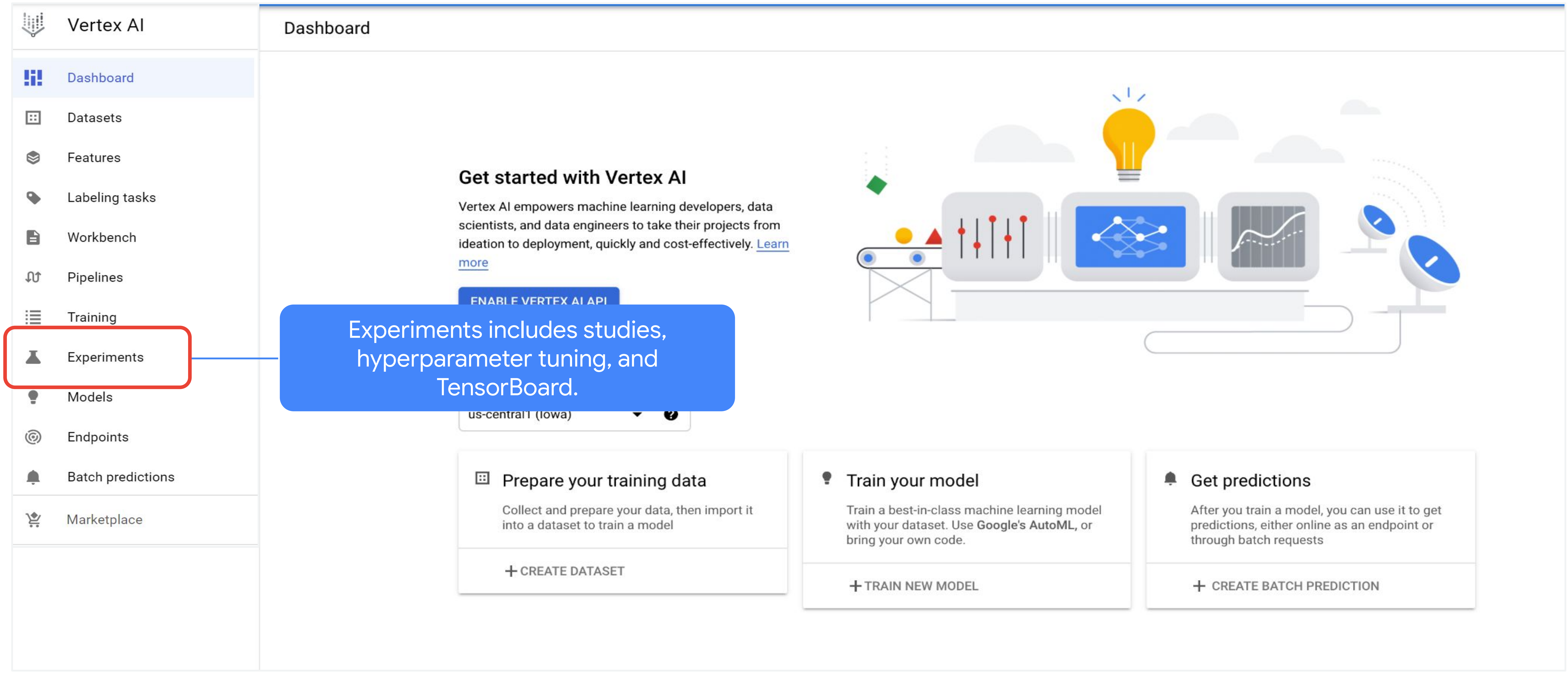
Vertex AI > Pipelines



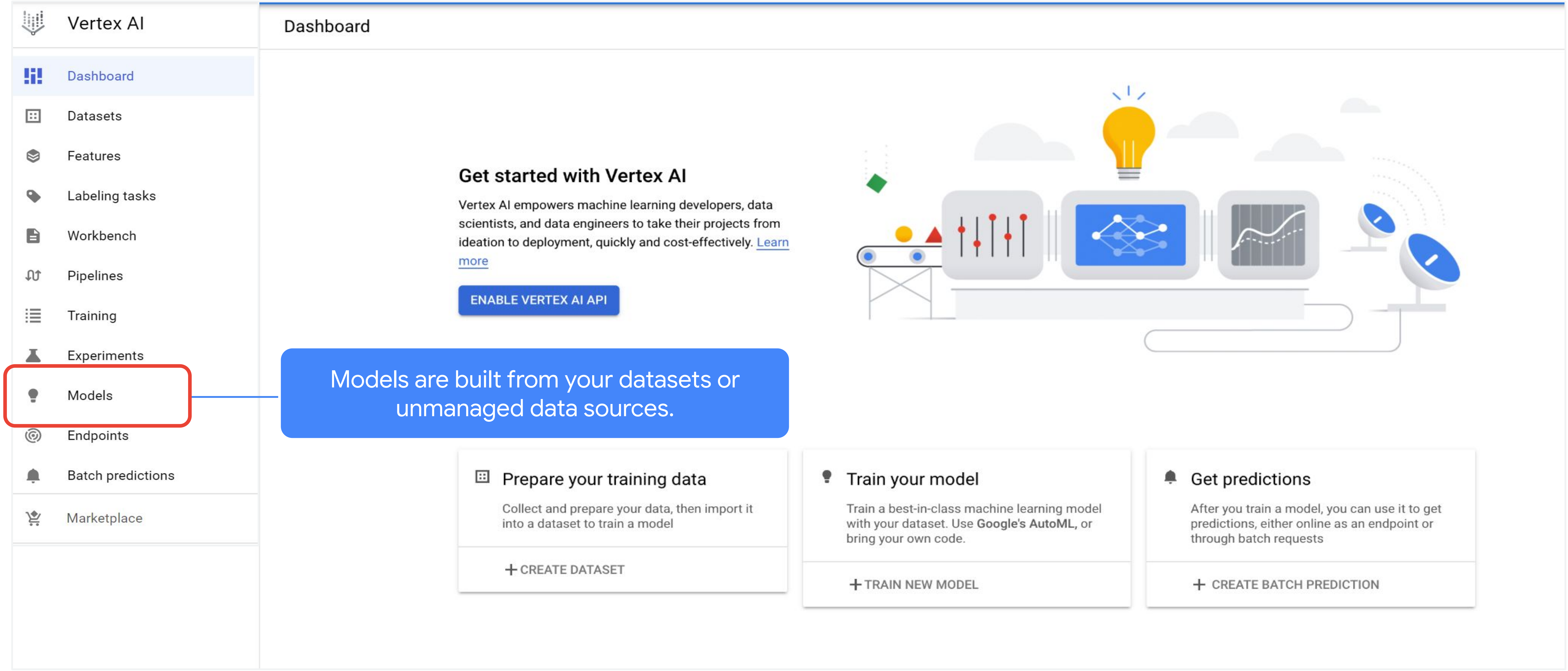
Vertex AI > Training



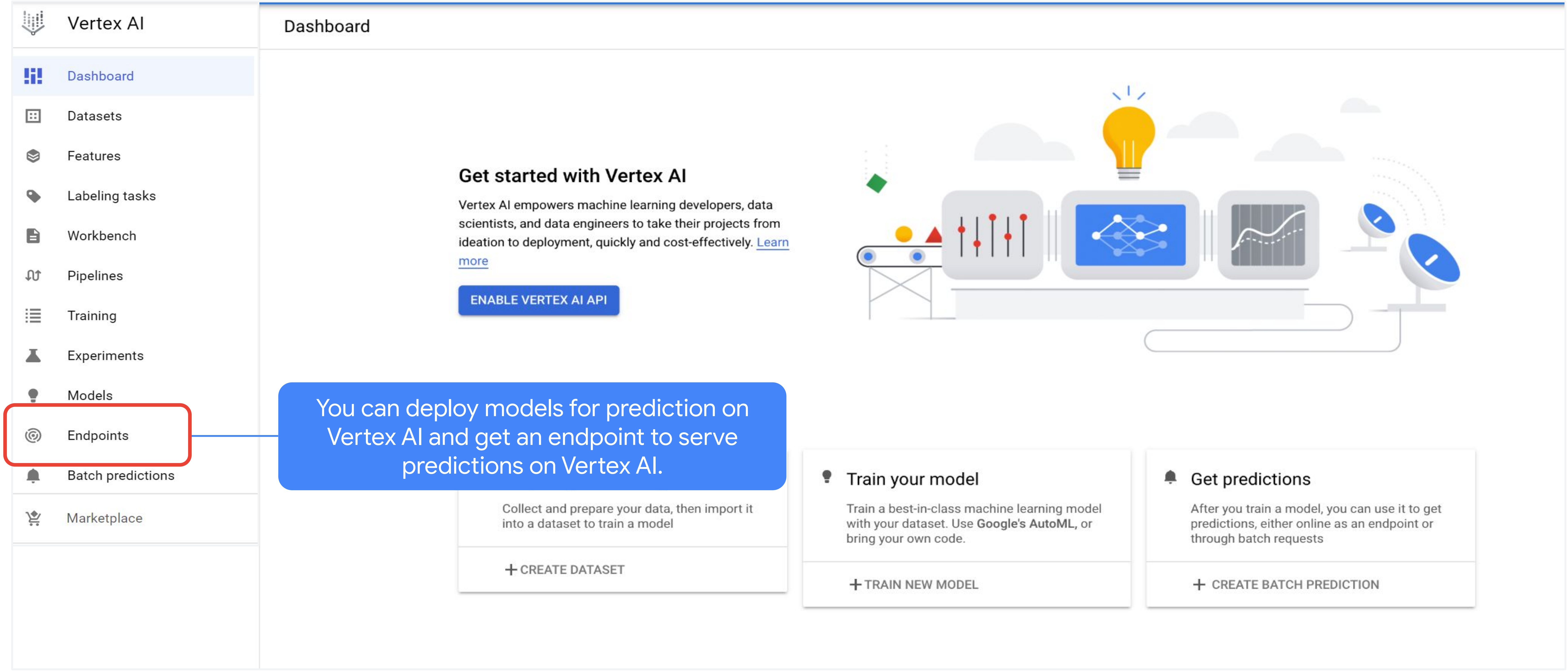
Vertex AI > Experiments



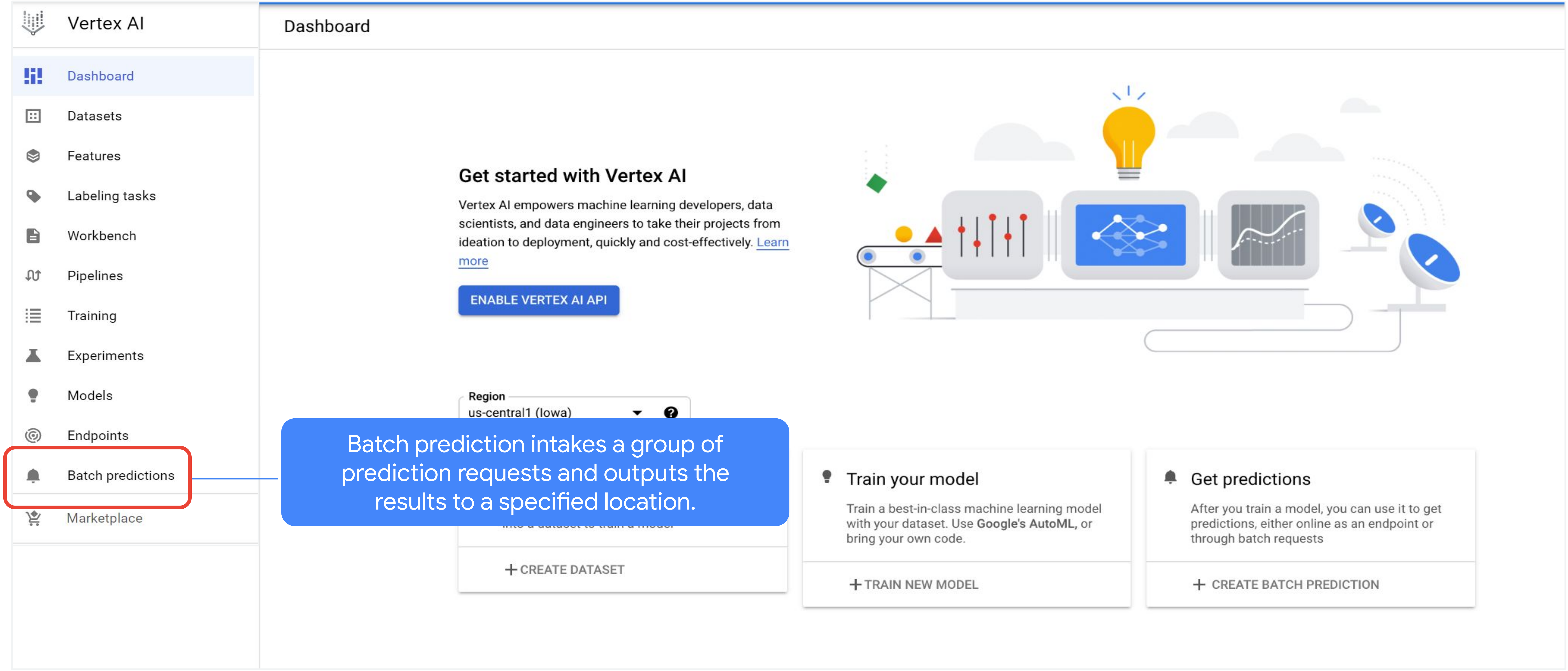
Vertex AI > Models



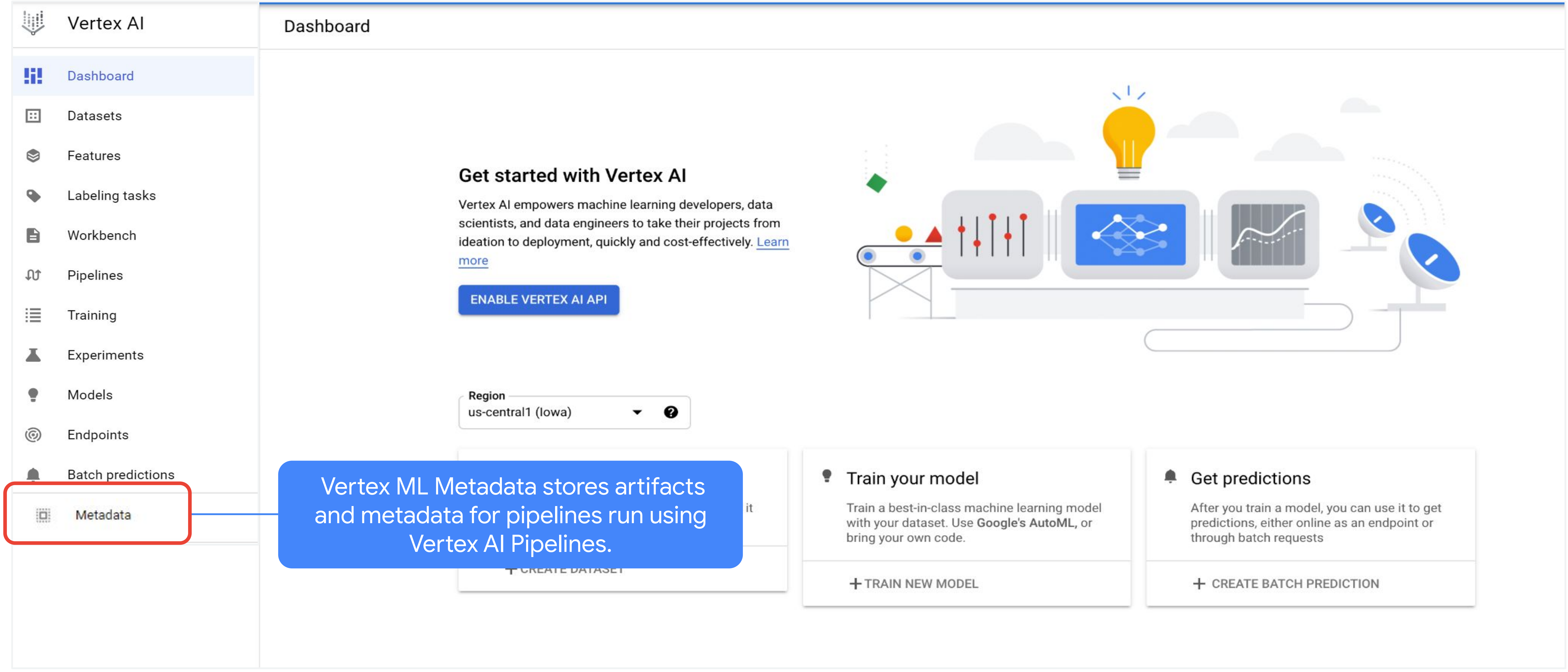
Vertex AI > Endpoints



Vertex AI > Batch predictions



Vertex AI > Metadata



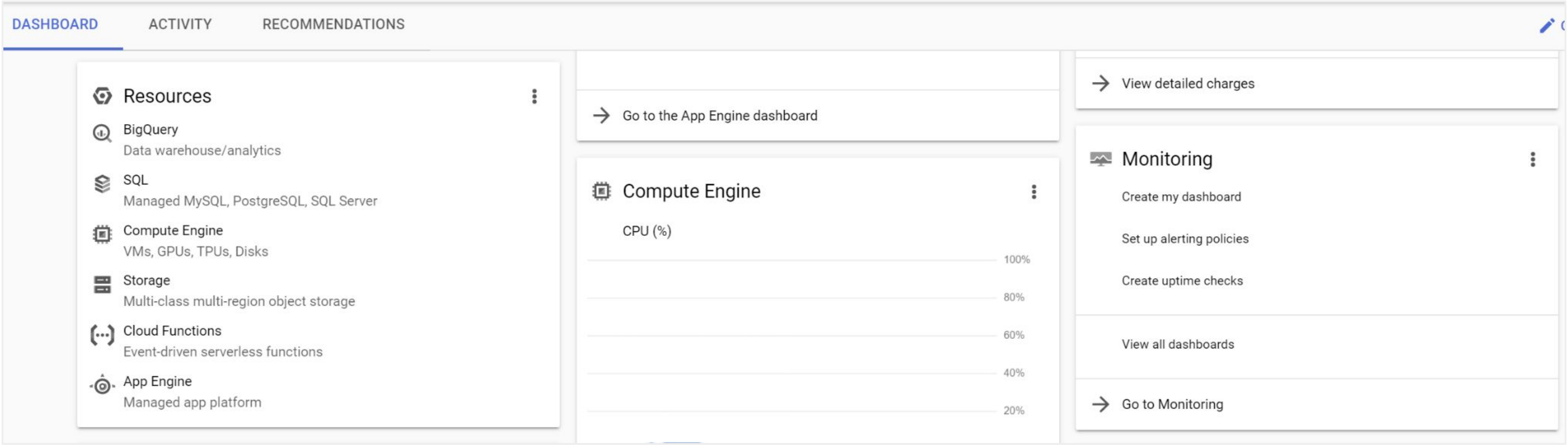
Vertex AI components:

For more information see the following courses:

- Launching into Machine Learning
- Feature Engineering
- Machine Learning in the Enterprise

Tools to interact with Vertex AI

Google Cloud Console



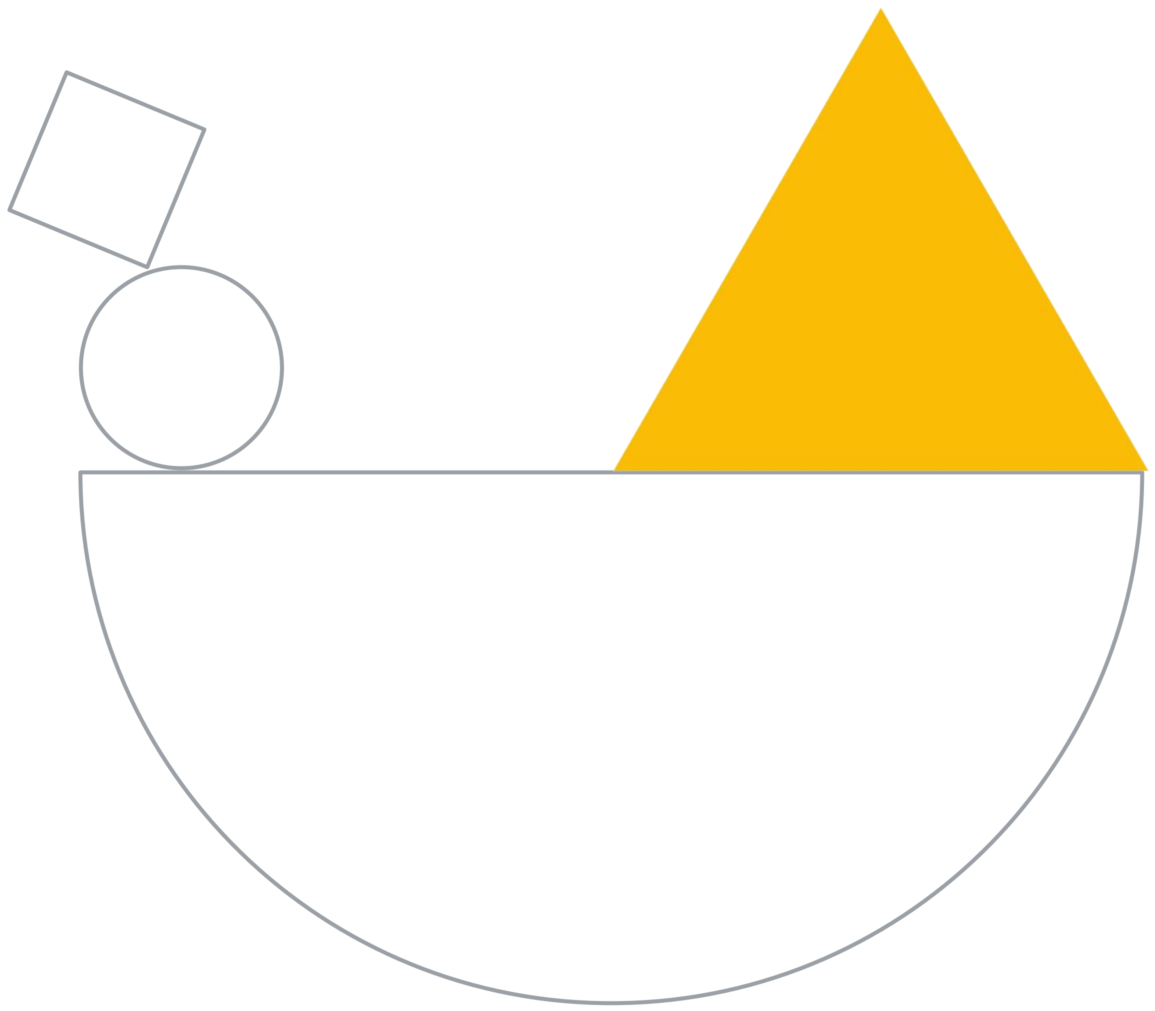
Vertex AI provides client libraries for some languages to help you make calls to the Vertex AI API.

Alternatively, you can use the Google API Client Libraries to access the Vertex AI API by using other languages.

Tools to interact with Vertex AI

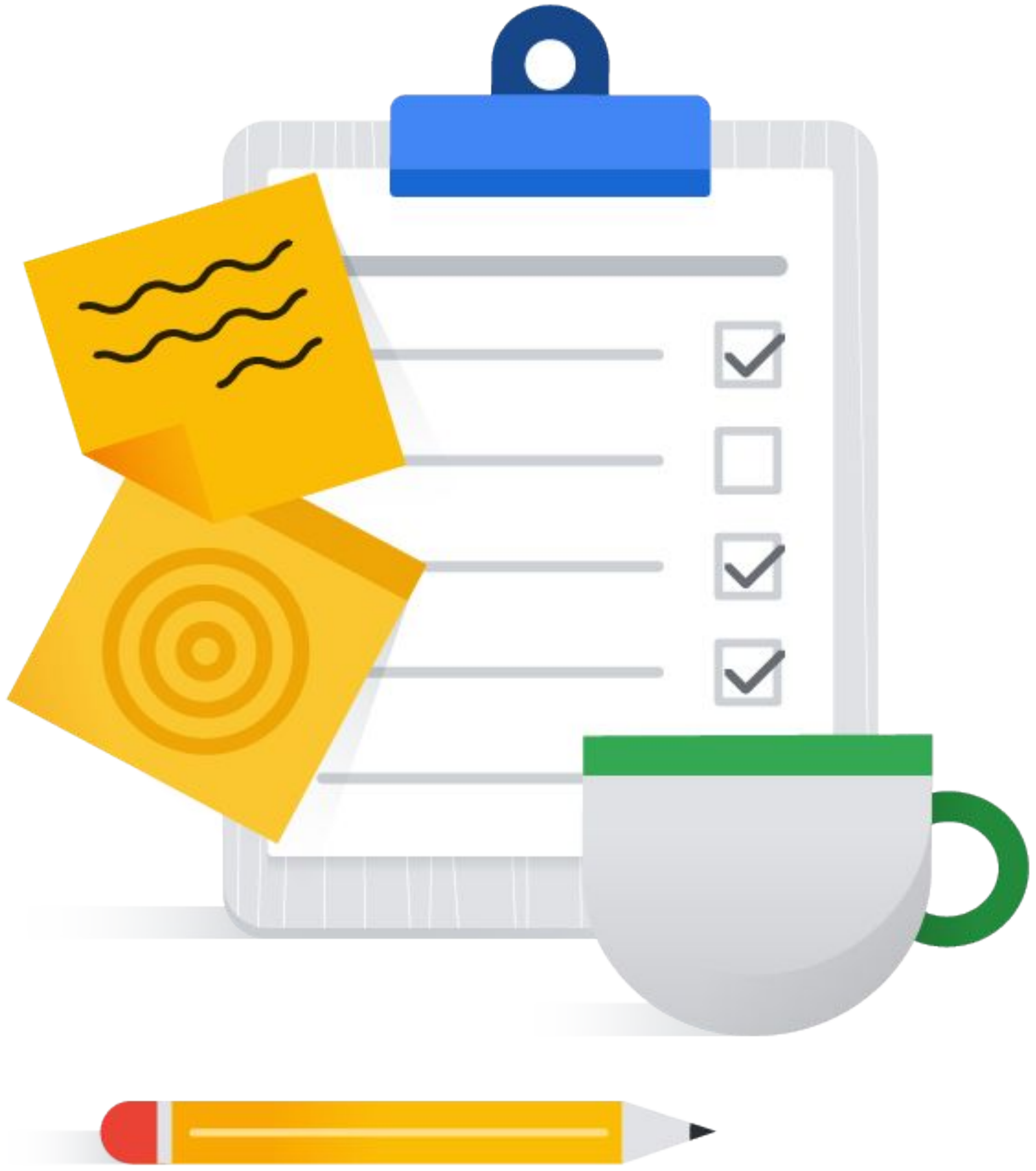
- Client Libraries
- VM Images
- REST API
- Containers

Machine Learning Development with Vertex Notebooks



In this module, you learn to ...

01	Describe Workbench Notebook options
02	Create a managed notebook
03	Create a user-managed notebook



Machine learning development with Vertex Notebooks

Vertex AI Workbench provides two Jupyter notebook-based options for your data science workflow

Managed notebooks

User-managed notebooks

Vertex AI Workbench provides **two Jupyter notebook-based options** for your data science workflow

Managed notebooks

User-managed notebooks

Google-managed environments

Vertex AI Workbench provides **two Jupyter notebook-based options** for your data science workflow

Managed notebooks

User-managed notebooks

Deep Learning VM Images

Both options are pre-packaged with JupyterLab with support for TensorFlow and Pytorch frameworks

Managed notebooks

Google-managed environments
End-to-end notebook-based production environment.

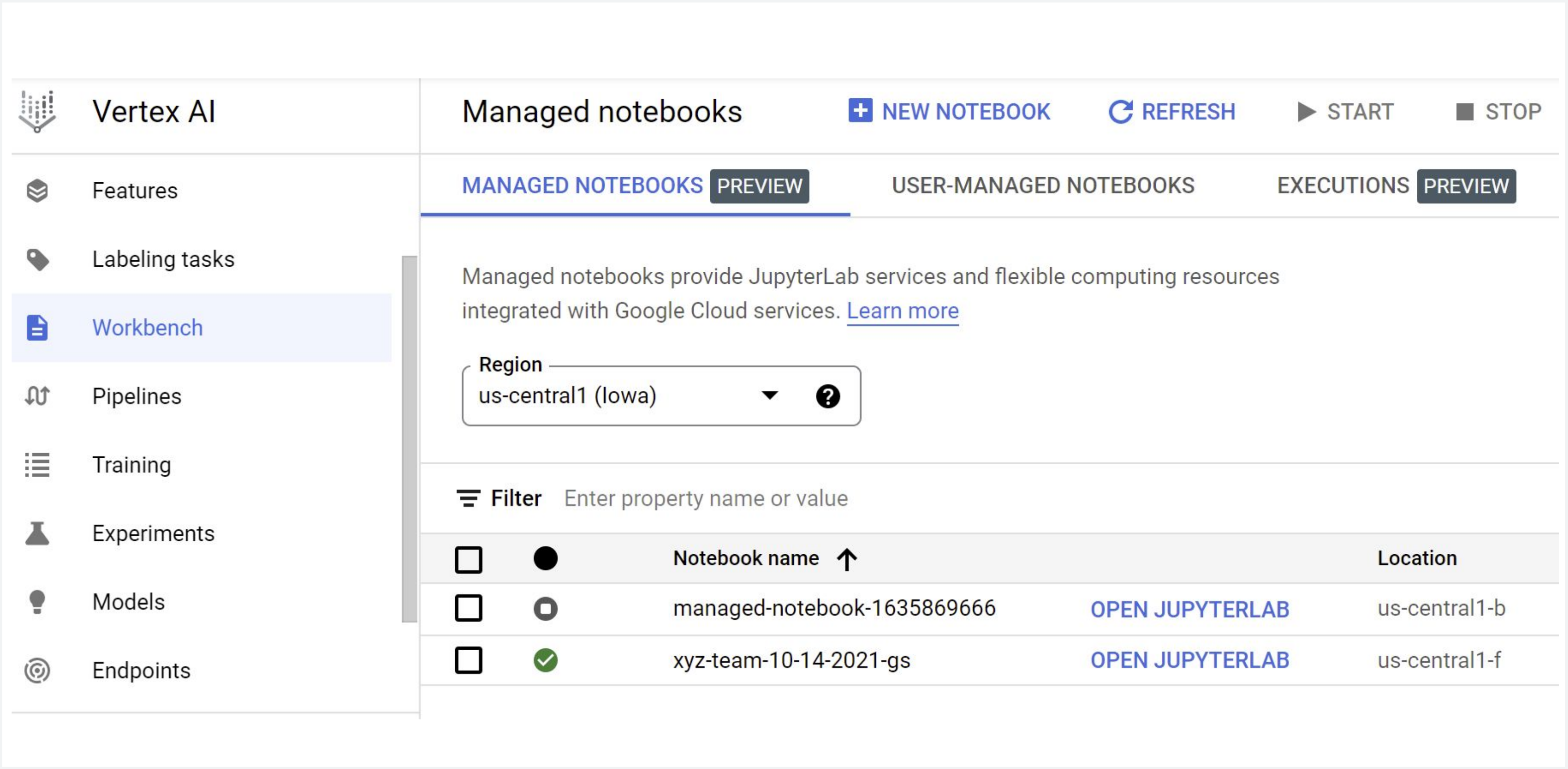
User-managed notebooks

[Deep Learning VM Images](#) customizable environment

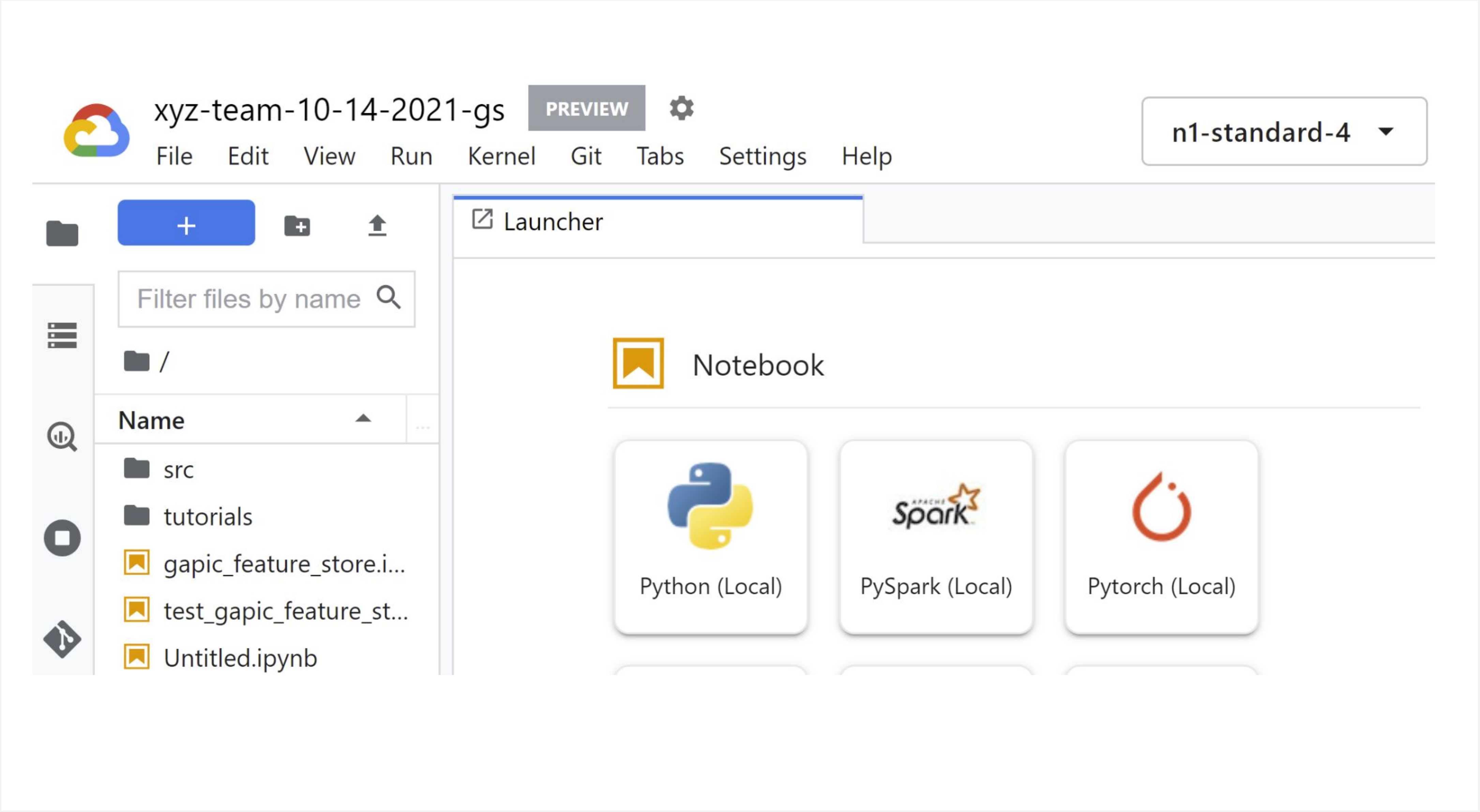
Good choice for data exploration, analysis, modeling, or as part of an end-to-end data science workflow.

Perform workflow-oriented tasks without leaving the JupyterLab interface.

Managed notebooks



Managed notebooks



Control hardware

Determine compute resources (GPUs, RAM).

Managed notebooks

Training with custom containers on Vertex Training

Dockerfile	Cloud Build	Container Registry	Vertex Training
Image with custom training code (e.g., custom dependencies, ML frameworks not supported)	Build a Docker container image that meets the container image requirements	Push container image to an Artifact Registry repository or a Container Registry repository	Configure and run custom job on Vertex Training with custom container

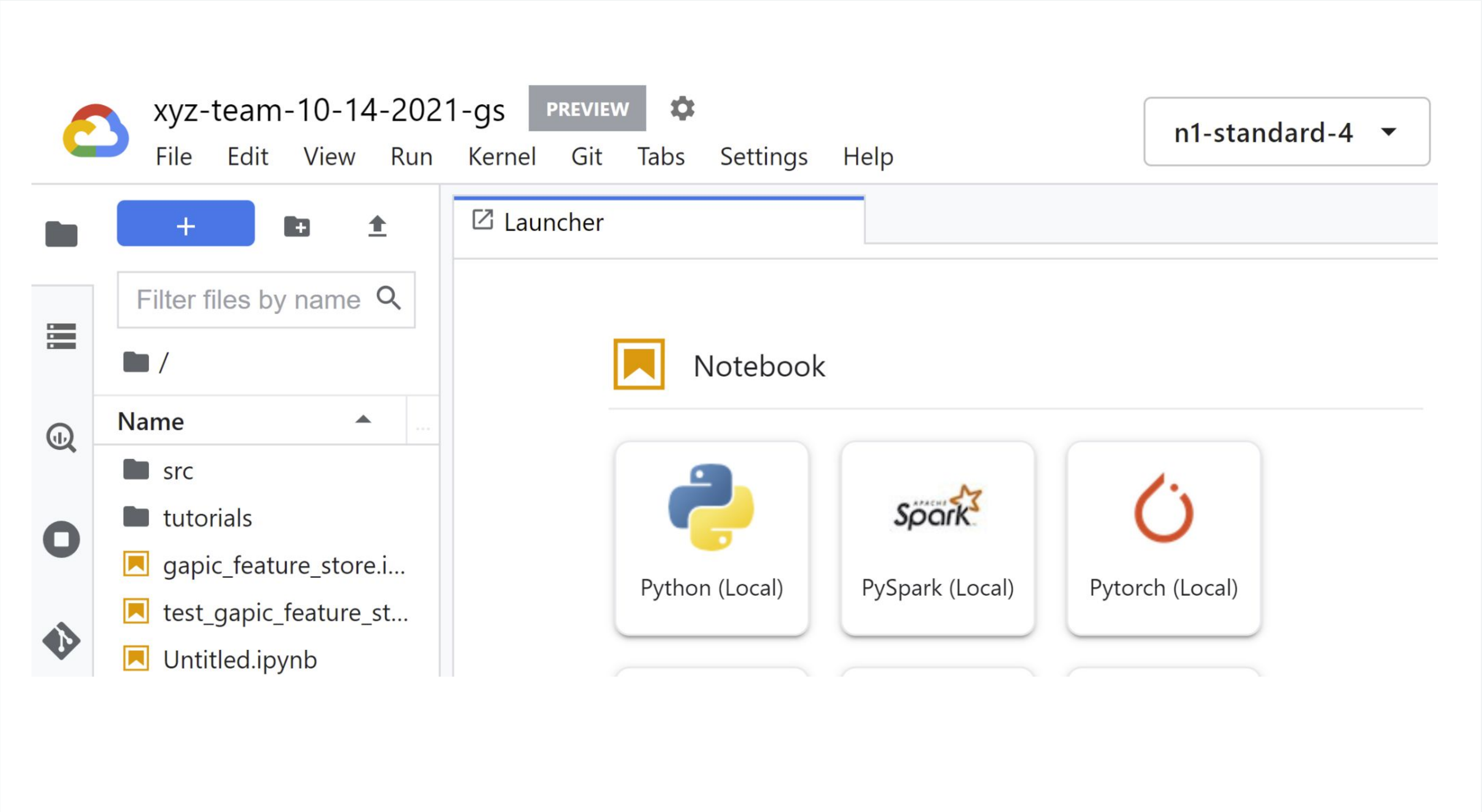
Control hardware

Determine compute resources (GPUs, RAM).

Custom containers

Use TensorFlow, Pytorch, PySpark, R. Add custom Docker container images.

Managed notebooks



Control hardware

Determine compute resources (GPUs, RAM).

Custom containers

Use TensorFlow, Pytorch, PySpark, R. Add custom Docker container images.

Access to data

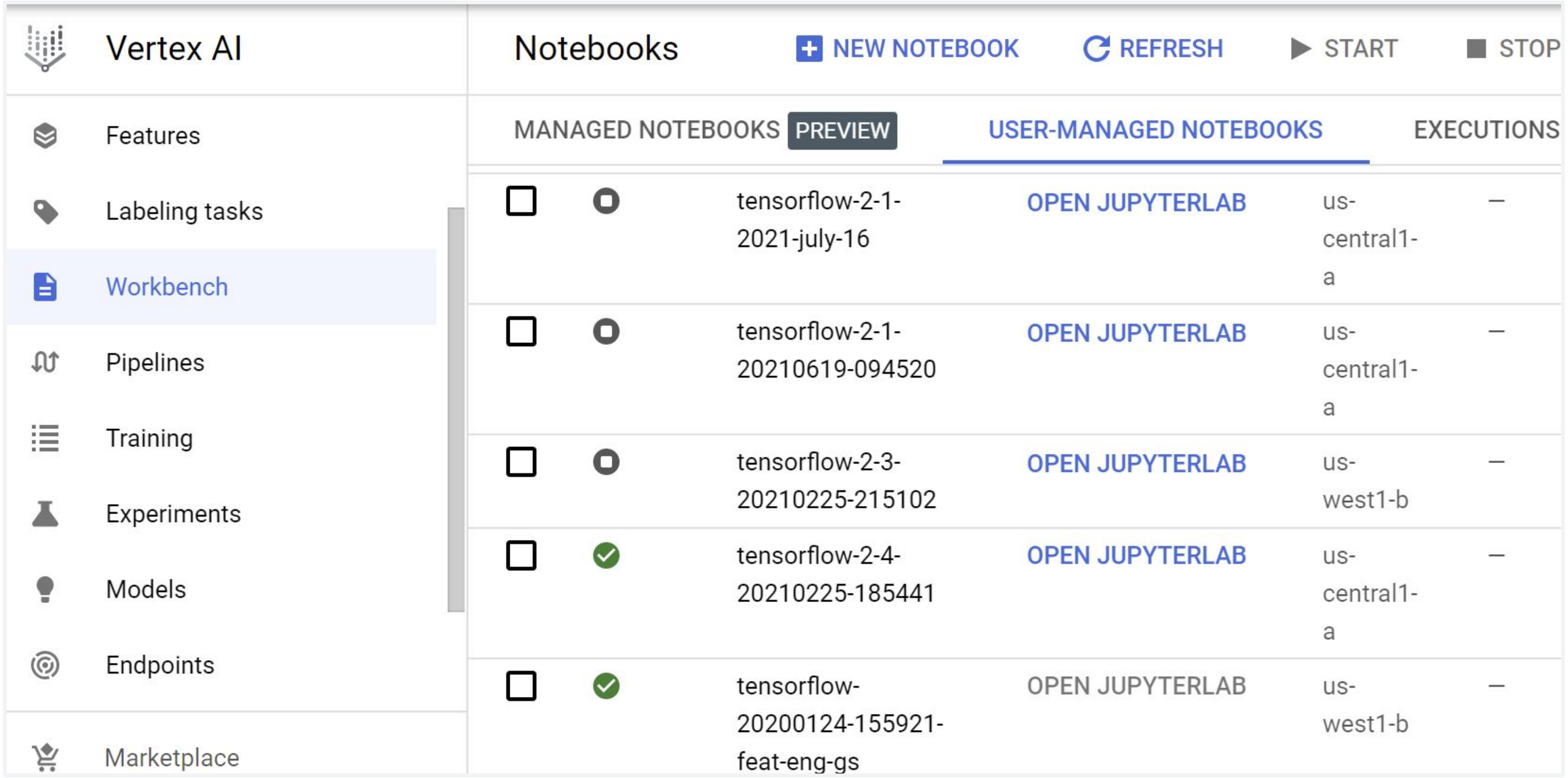
Use Cloud Storage and BigQuery extension to browse data.

Dataproc integration

Process data quickly by running a notebook on a Dataproc cluster.

User-managed notebooks can be a good choice for users who require extensive customization or who need a lot of control over their environment.

User-managed notebooks



User-managed notebooks

Deep Learning VM

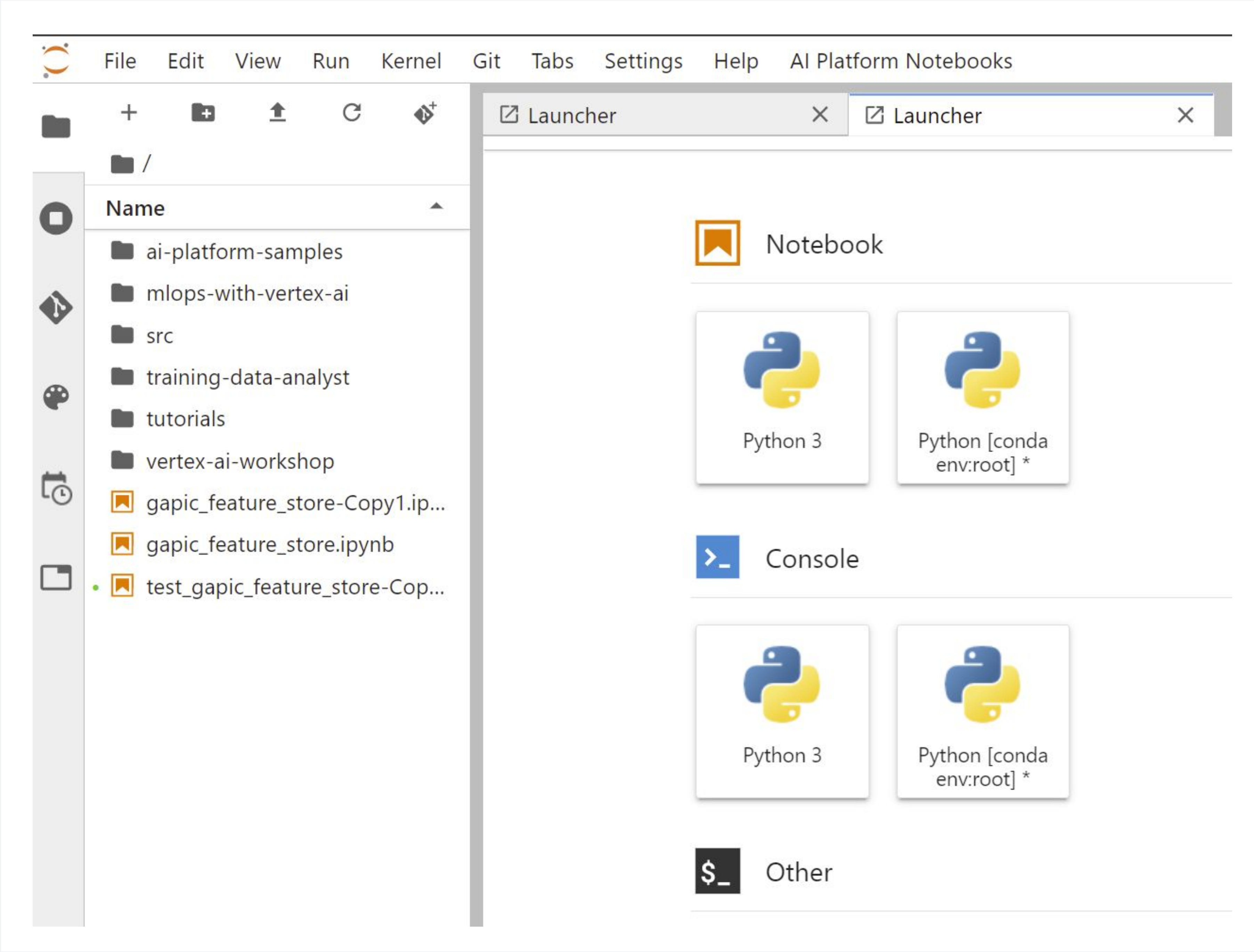
Choose VM configuration and customize Compute Engine instances.

Health status monitoring

Report system health and custom metrics to Cloud Monitoring. Use the diagnostic tool to verify the status of core services (Jupyter, Docker), check disk space, and collect instance logs on network information.

Networking and security

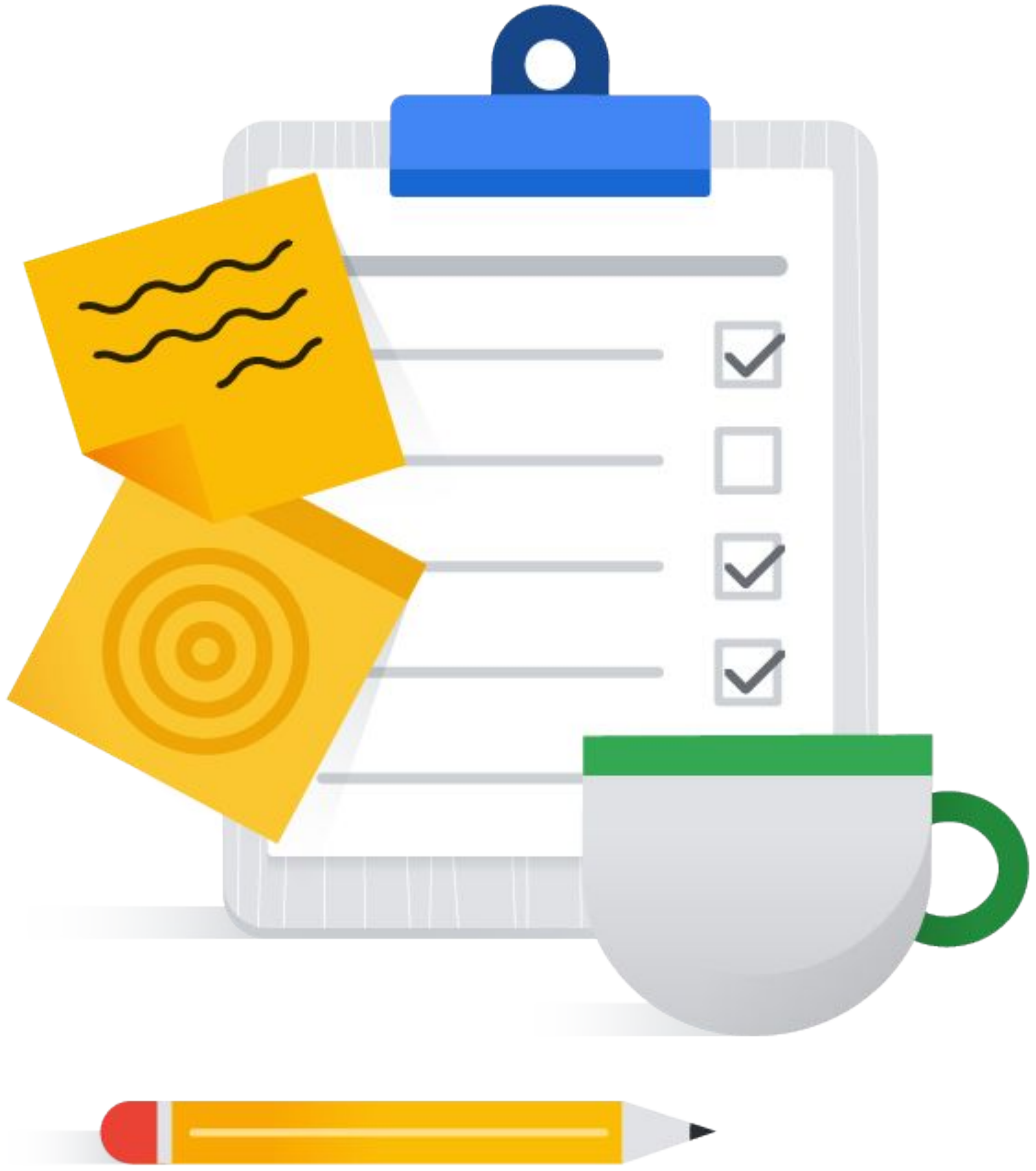
Choose VPC Service Controls.



Best Practices for Implementing Machine Learning on Vertex AI

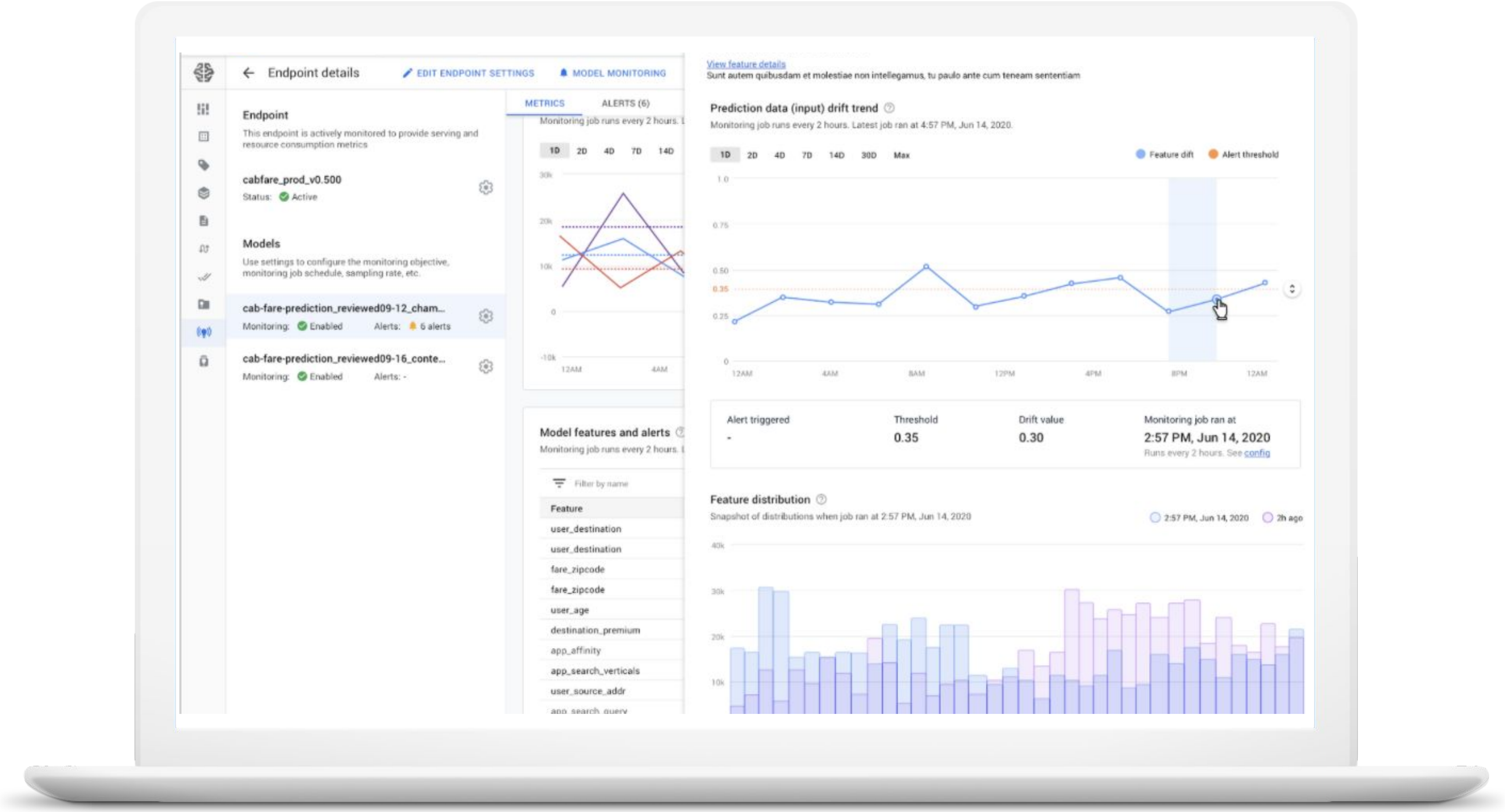
In this module, you learn ...

01	Best practices for machine learning development
02	Best practices for data preprocessing
03	Best practices for machine learning environment setup



Best practices for machine learning development

Best practices



Data

How it is prepared and stored

Workbench Notebooks

Using Notebooks to evaluate and understand your models

Model

Tips for training, maximizing predictive accuracy, and feature attributions for insights

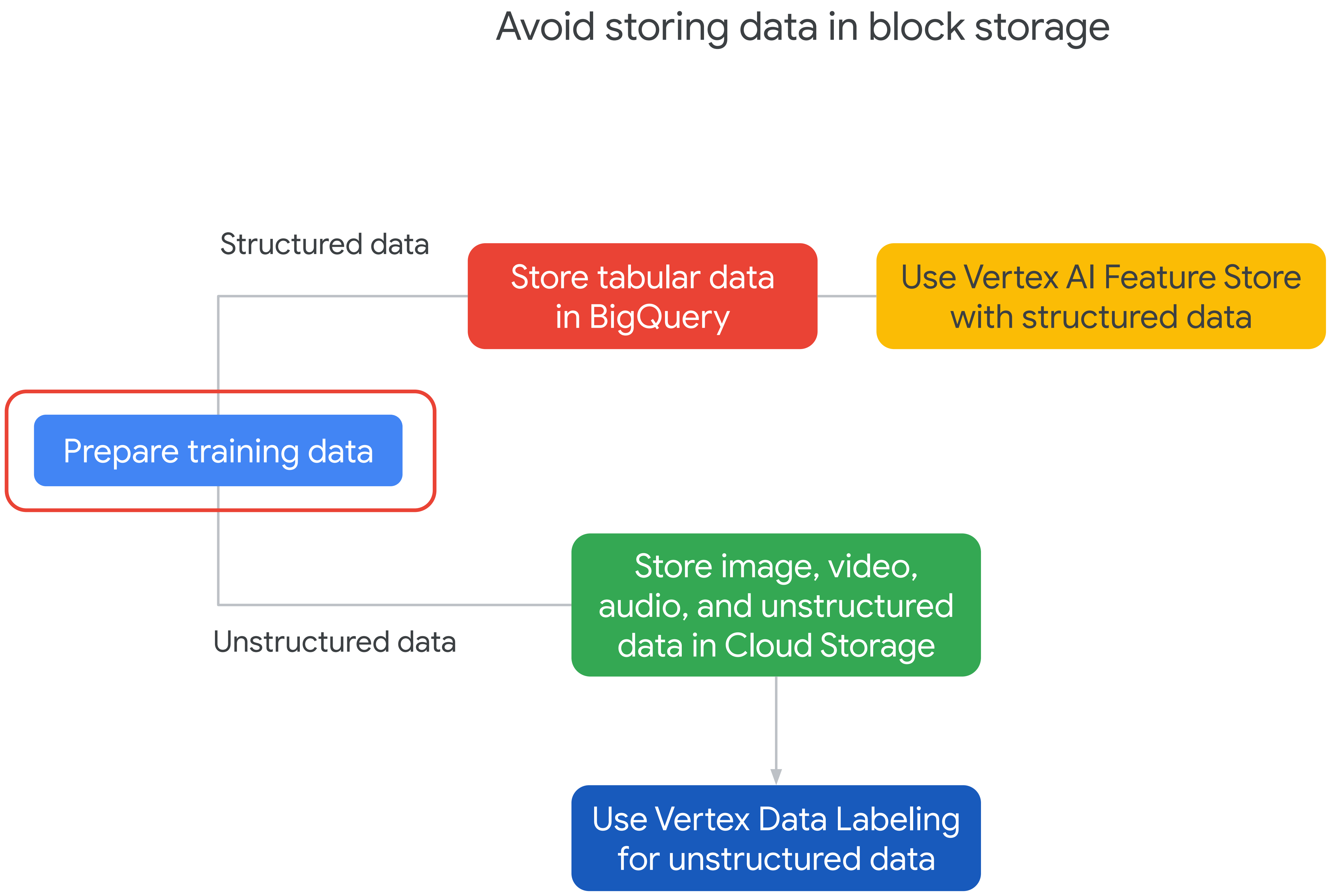
Tensorboard

Using Vertex AI TensorBoard to visualize experiments

Best practices for preparing and storing data

Data

How it is prepared and stored

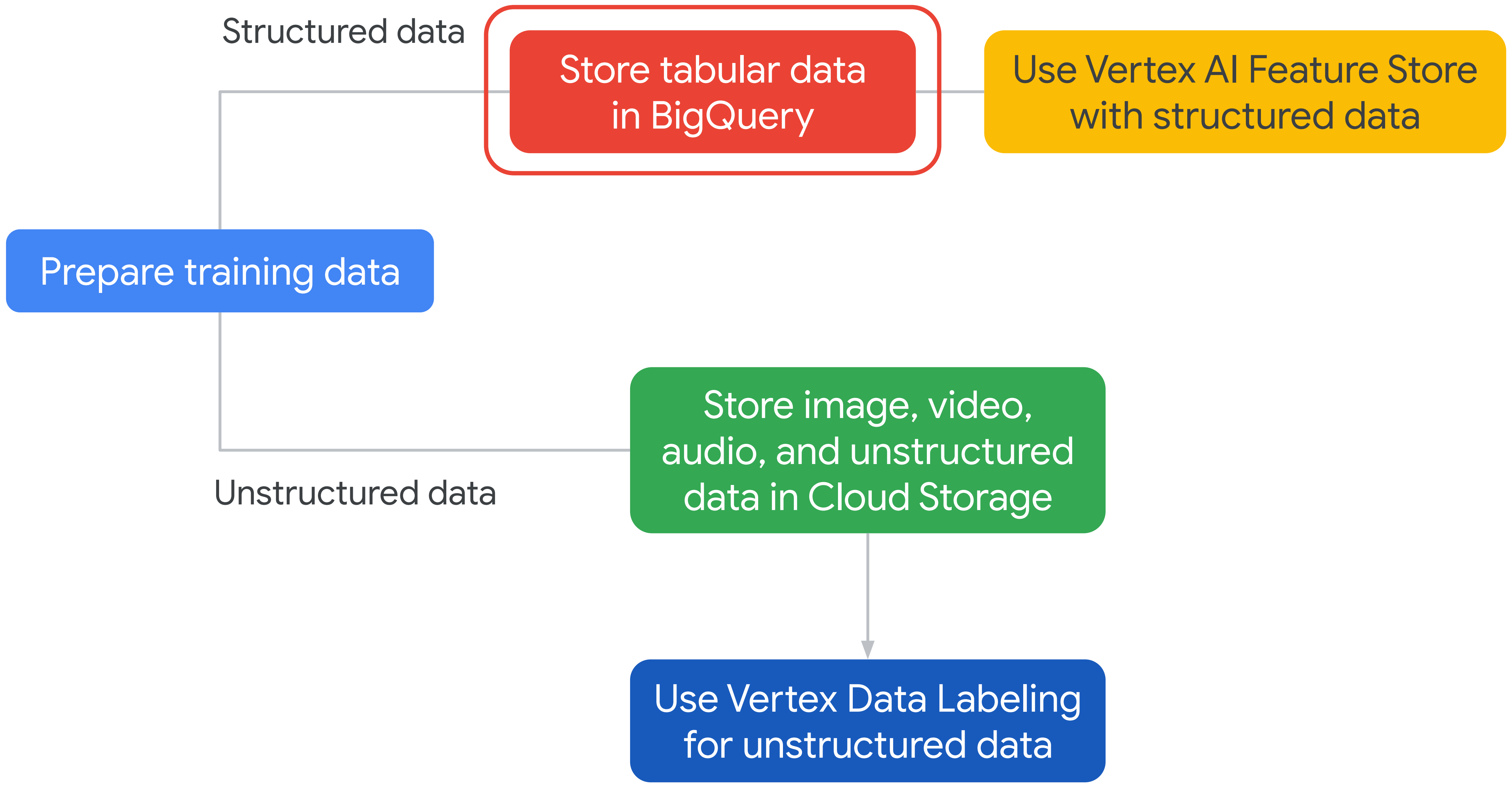


Best practices for preparing and storing data

Data

How it is prepared and stored

Avoid storing data in block storage

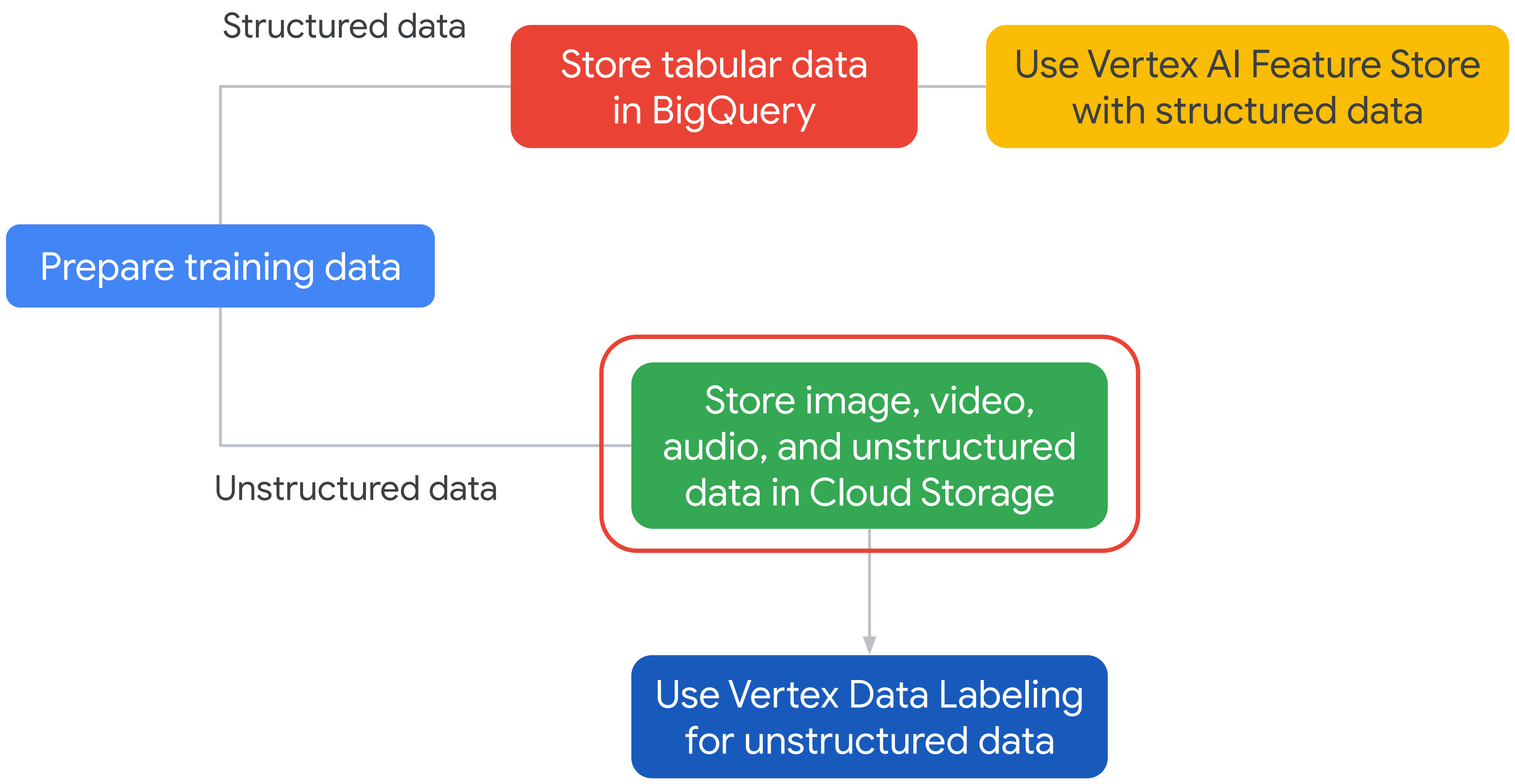


Best practices for preparing and storing data

Data

How it is prepared and stored

Avoid storing data in block storage

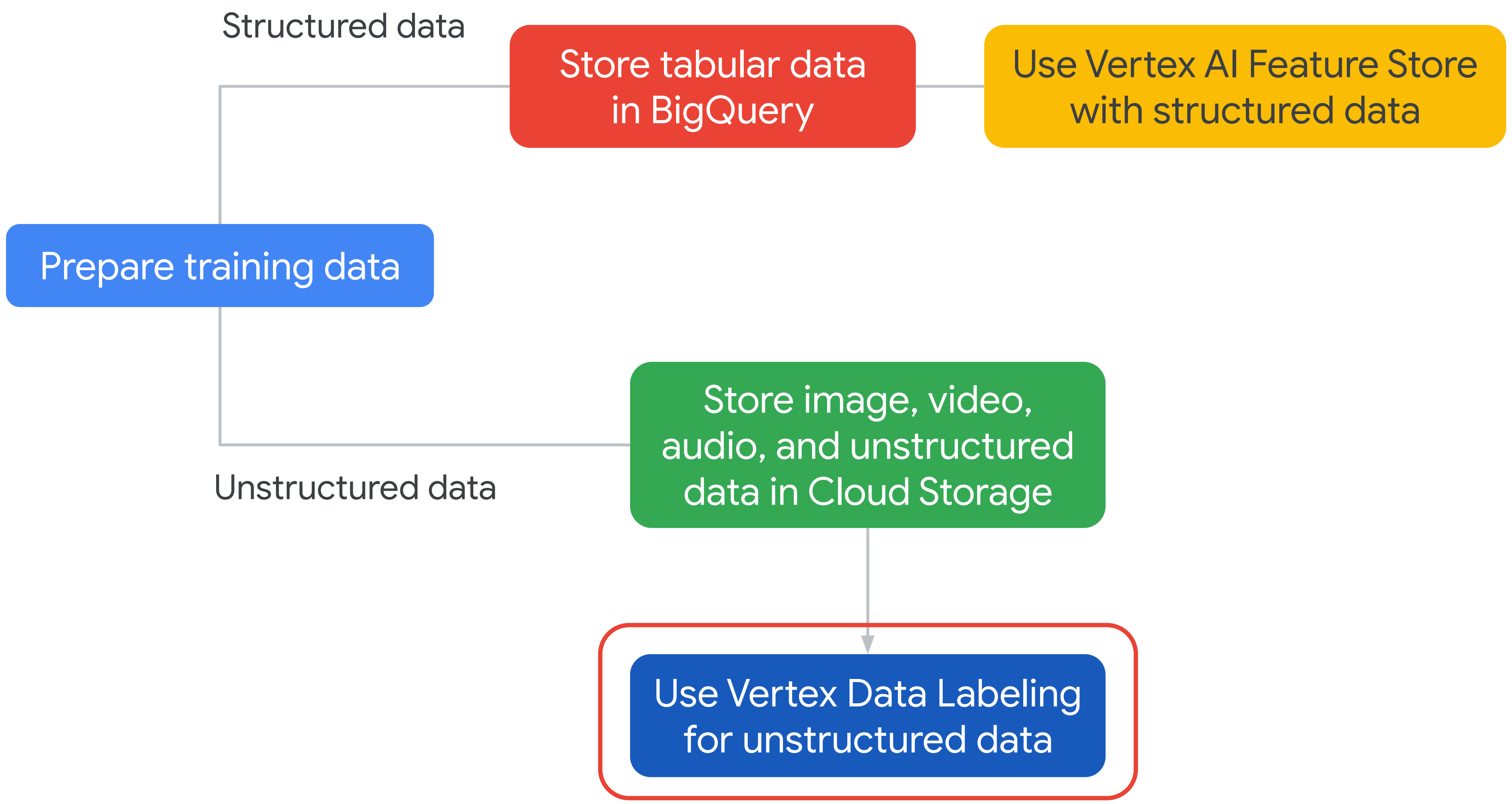


Best practices for preparing and storing data

Data

How it is prepared and stored

Avoid storing data in block storage

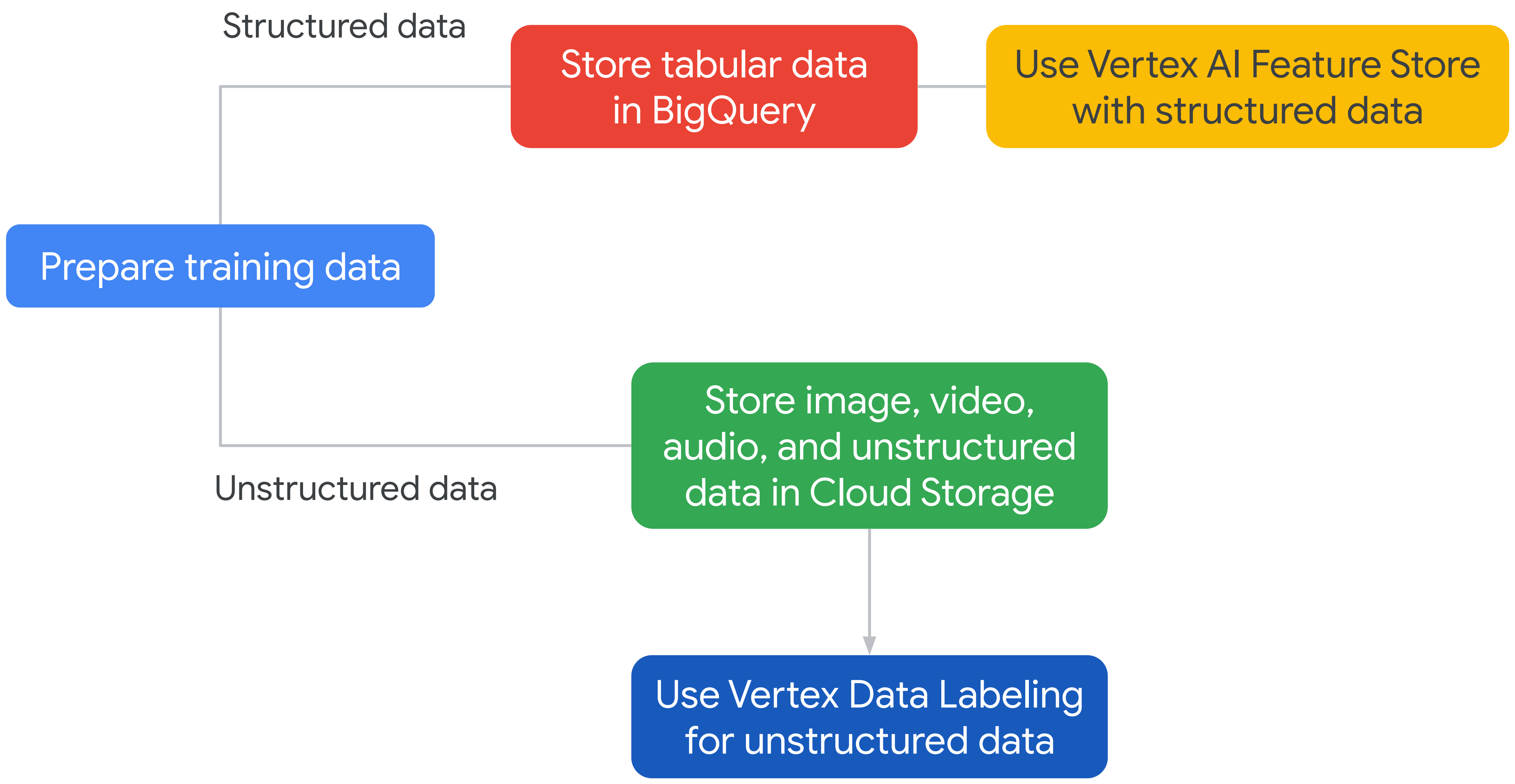


Best practices for preparing and storing data

Data

How it is prepared and stored

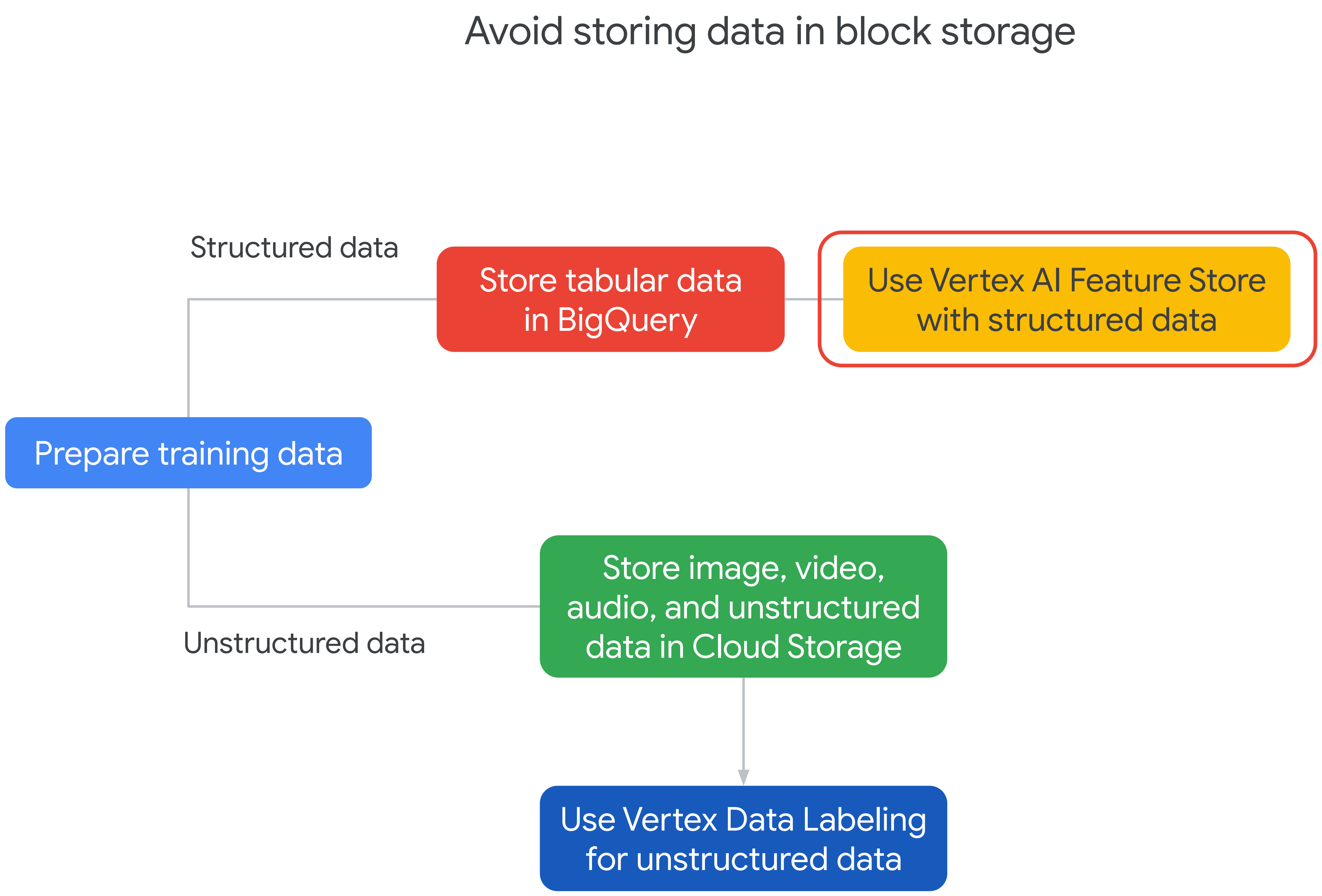
Avoid storing data in block storage



Best practices for preparing and storing data

Data

How it is prepared and stored



Vertex AI Feature Store

Feature Store

Use Feature Store with structured data

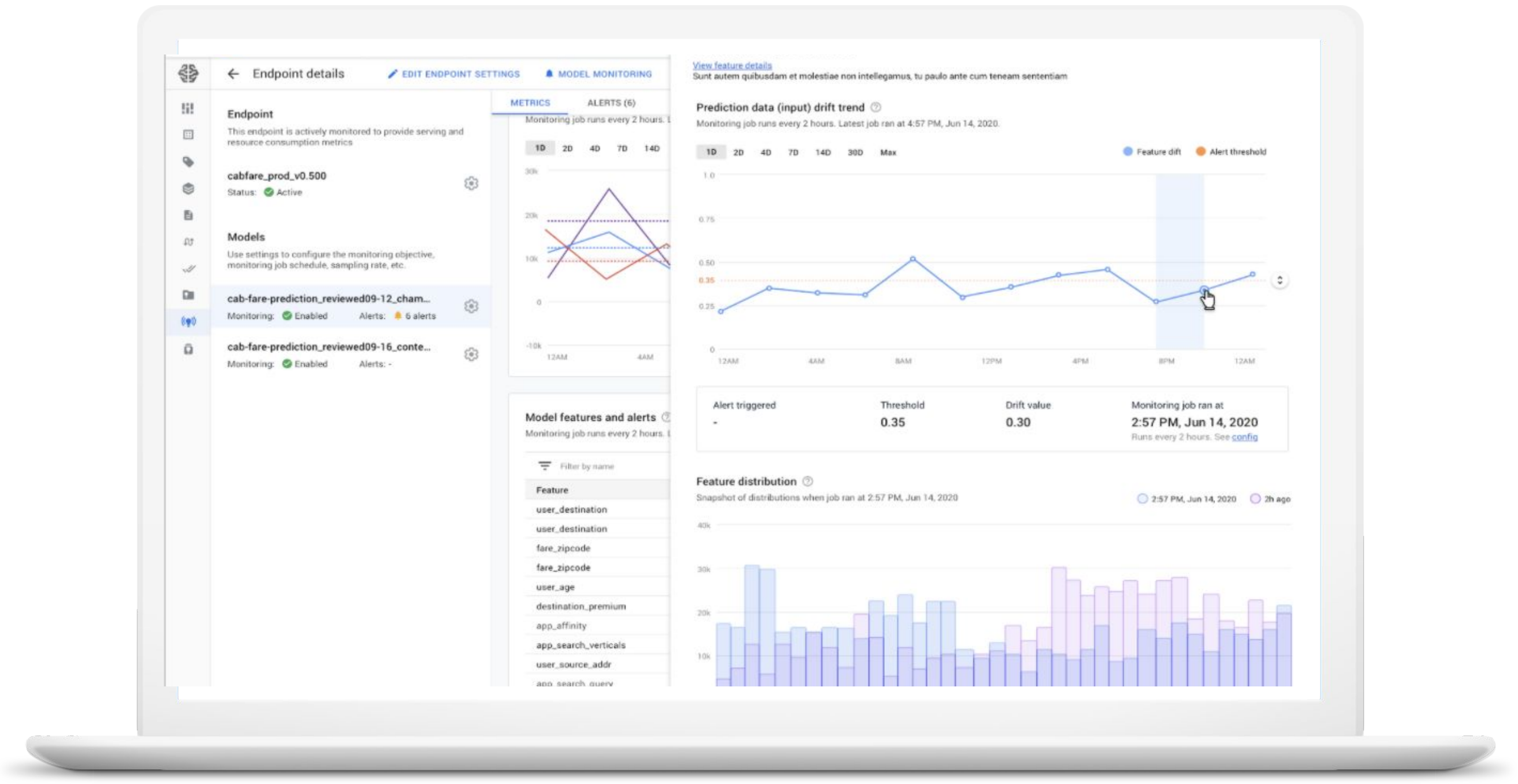
Follow these steps:

1. Search Vertex AI Feature Store
    a. Search to see if a feature already exists.
    b. Fetch those features for your training labels using Vertex AI Feature Store's batch serving capability.
2. Create a new feature
    a. Create a new feature using your Cloud Storage bucket or BigQuery location. OR
    b. Fetch raw data from your data lake and write your scripts to perform feature processing.
    c. Join the feature values and the new feature values. Merging those feature values produces the training data set.

Best practices for training a model

Model

Tips for training, maximizing predictive accuracy, and feature attributions for insights



For small datasets, train a model within the Notebooks instance.

For large datasets, distributed training, or scheduled training, use the Vertex training service.

Training with pre-built containers on Vertex AI

Training with pre-built containers on Vertex Training

Python Source Distribution → **Cloud Storage Bucket** → **Vertex Training**

Python Source Distribution

Training application code packaged as Python source distribution. Can include custom Python dependencies or others.

```
training-code/
setup.py
```

Cloud Storage Bucket

Push the packaged training application code to Google Cloud Storage bucket

Vertex Training

Configure and run custom job on Vertex Training with pre-built containers

Best practices for Explainable AI

Model

Tips for training, maximizing predictive accuracy, and feature attributions for insights

Offers feature attributions to provide insights into why models generate predictions.

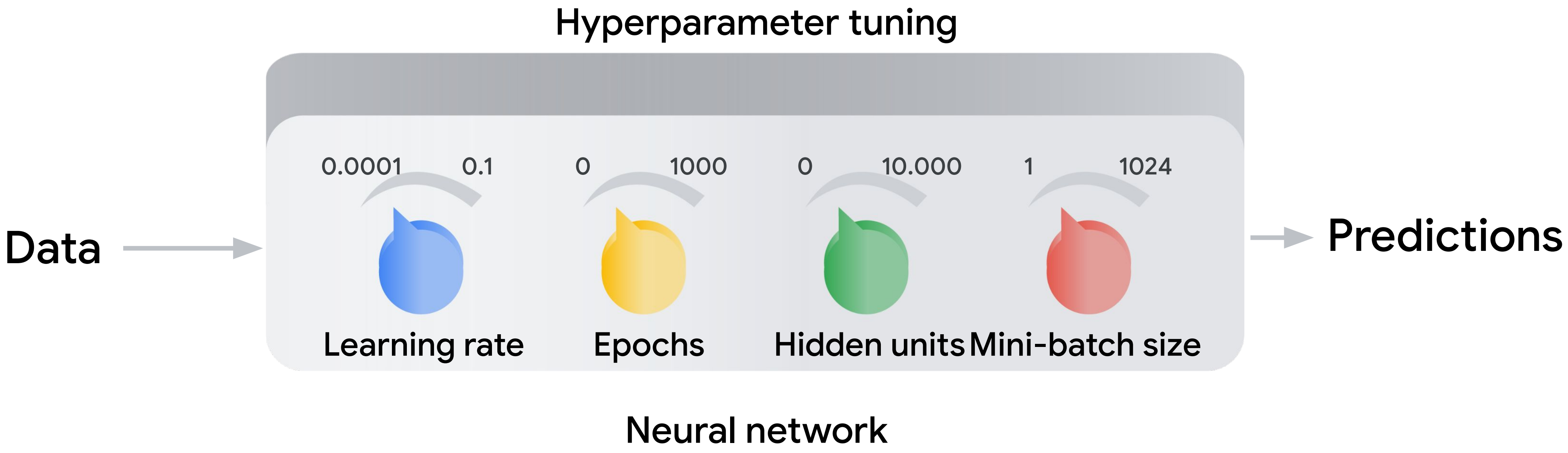
Details the importance of each feature that a model uses as input to make a prediction.

Supports custom-trained models based on tabular and image data.

Hyperparameter tuning with Vertex Training

Model

Maximize your model's predictive accuracy with hyperparameter tuning

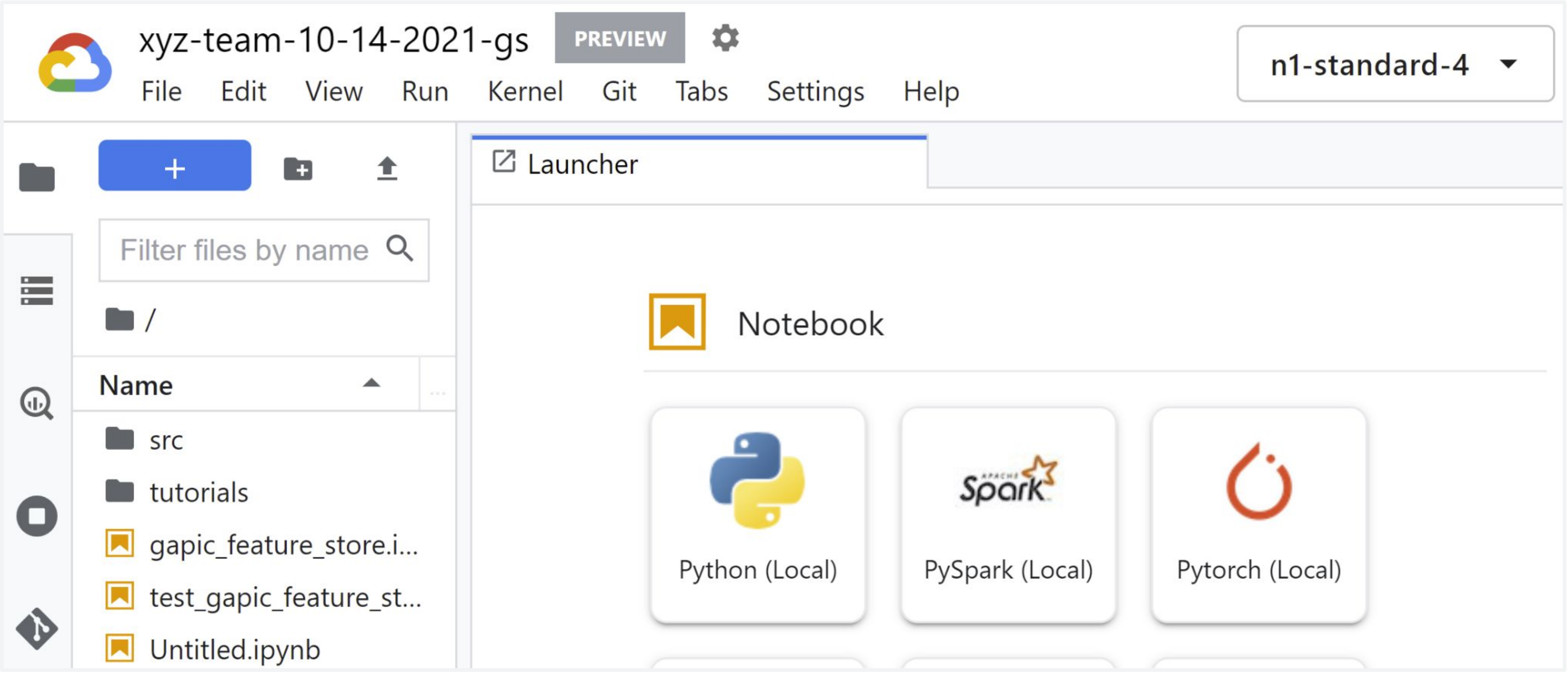


The hyperparameters are knobs that act as the network-human interface.

Maximize a model's predictive accuracy. Vertex Training provides an automated model enhancer to test different hyperparameter configurations when training your model.

No need to manually adjust hyperparameters over the course of numerous training runs to arrive at the optimal values.

Best practices for using Workbench Notebooks



Workbench Notebooks

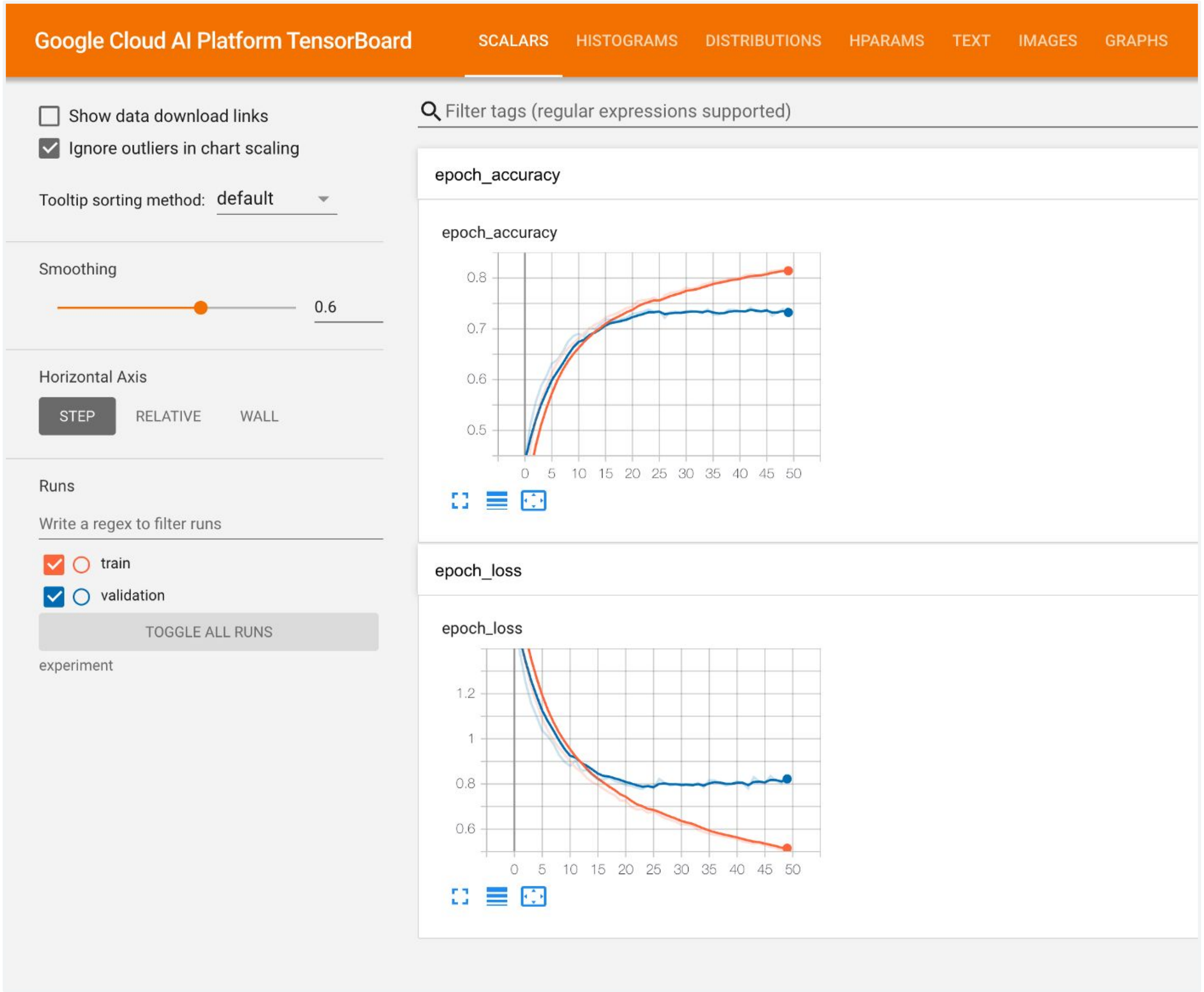
Use Notebooks to evaluate and understand your models.

Use Notebooks to evaluate and understand your models. In addition to built-in common libraries like scikit-learn, Notebooks offers What-if Tool (WIT) and Language Interpretability Tool (LIT).

Best practices for using Vertex AI TensorBoard

TensorBoard

Use Vertex AI TensorBoard to visualize experiments.

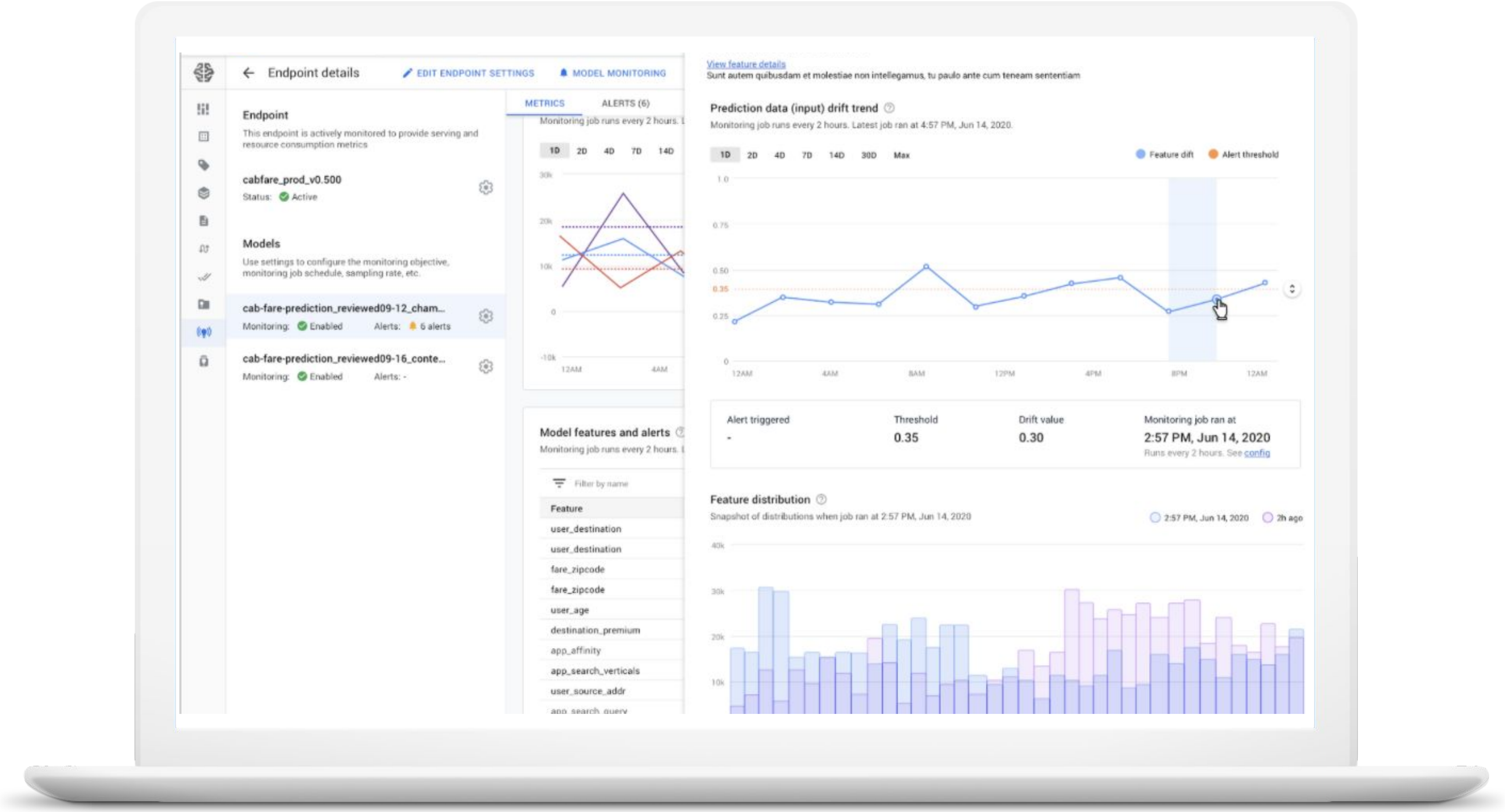


[Vertex AI TensorBoard](#) service lets you track experiment metrics such as loss and accuracy over time, visualize a model graph, project embeddings to a lower dimensional space, and much more.

Data preprocessing best practices

For training and evaluation, we need preprocessed features

Best practices: Data preprocessing



Dataflow

Use Dataflow to process unstructured data.

TensorFlow Extended

Use TensorFlow Extended when leveraging TensorFlow ecosystem.

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.

Using managed datasets in Vertex AI

Managed datasets

Use managed datasets to link data to your models.

Managed datasets:

- Enable you to create a clear link between your data and custom-trained models,
- Provide descriptive statistics and automatic or manual splitting into train, test, and validation sets.
- Are not required to use Vertex AI.

Transforming unstructured data with Dataflow

Dataflow

Use Dataflow to process 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.

TensorFlow Extended

TensorFlow Extended

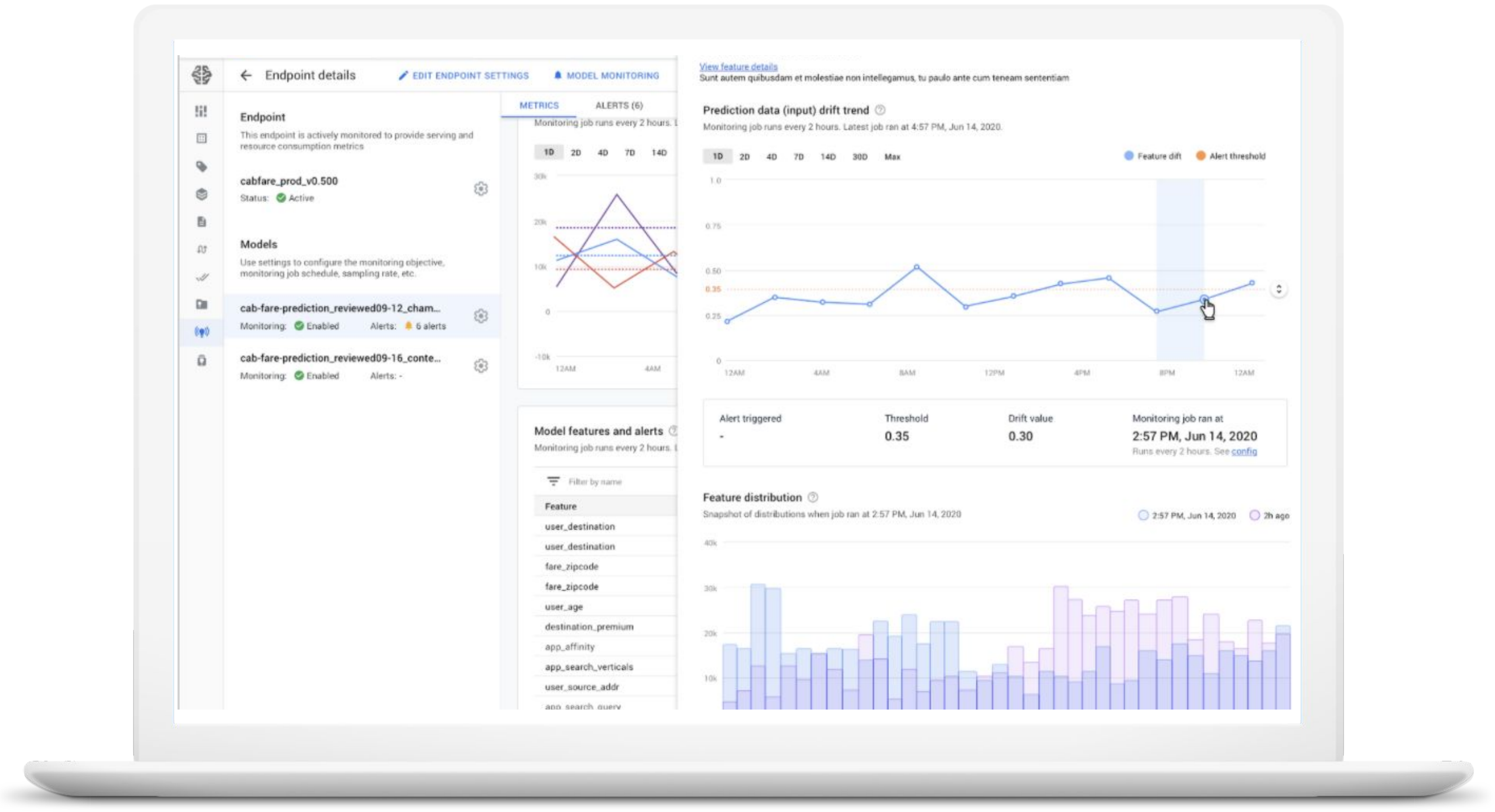
Use TensorFlow Extended when leveraging TensorFlow ecosystem.

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.

Best practices for ML environment setup

Best practices: ML environment setup



Workbench Notebooks

Use for development and experimentation. Create NB for each team member. Use Vertex SDK for Python.

Security

Secure PII in Notebooks.

Data & model

Store prepared data and model in same project.

Optimize performance & cost

Optimize performance and cost.

Workbench Notebooks

> **Workbench Notebooks**
> Use for development and experimentation. Create NB for each team member. Use Vertex SDK for Python

Use Notebooks for experimentation and development, including writing code, starting jobs, running queries, and checking status.

Create a new notebook instance for each member of your data science team.

Secure PII in Notebooks

Security

Secure PII in Notebooks

Apply data governance and security policies to help protect your Notebooks that contain personally identifiable information (PII) data - see Notebooks security blueprint: Protecting PII data guide.

Best practices: ML environment setup

Data & model

Store prepared data and model in same project.

Your Google Cloud project

Your prepared data
Cloud Storage or BigQuery

Access all of the datasets required for modeling. Store prepared data in your Google Cloud project.

However, different parts of your organization might store their data in different projects, then rely on raw data from different projects.

Best practices: ML environment setup

Optimize performance & cost

Optimize performance and cost.

Enhancing the performance and decreasing the cost of your machine learning workloads is a comprehensive subject, and out of scope for this course.

Responsible AI Development

In this module, you learn to ...

01	Articulate Responsible AI best practices (ML fairness, explainability, privacy, security)
02	Recognize biases that ML can amplify (ML Fairness)
03	Categorize explainable AI methods through taxonomy

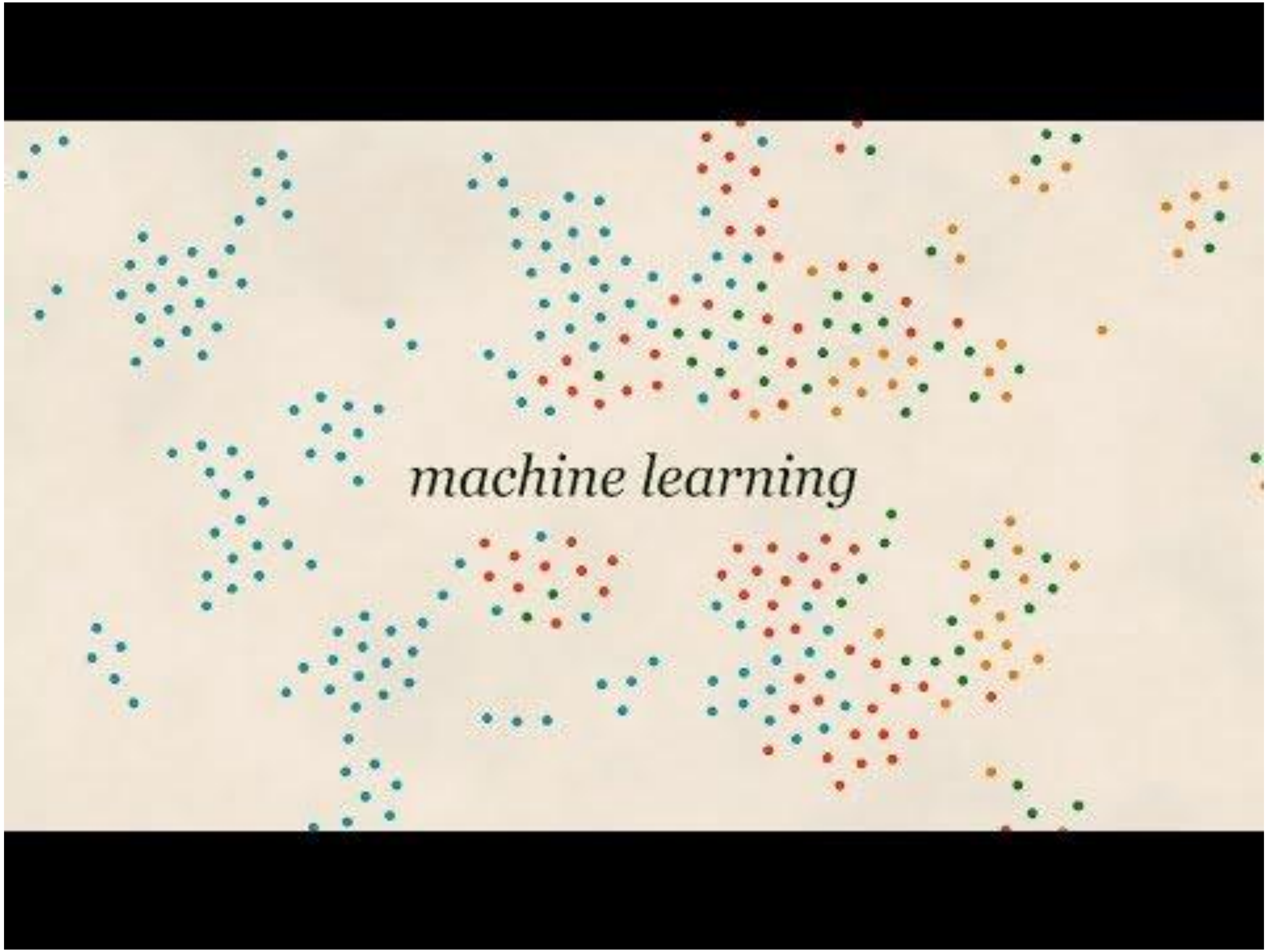
1. Origin of bias in ML models

2. ML trade-offs

3. Equality of opportunity

4. Understand your data

5. Find errors in your dataset


machine learning

Unconscious biases exist in data

Examples of human biases in data

- Reporting bias
- Selection bias

Examples of human biases in collection and labeling

- Confirmation bias
- Automation bias

Unconscious biases exist in data

Examples of human biases in data

- Reporting bias
- Selection bias

Examples of human biases in collection and labeling

- Confirmation bias
- Automation bias

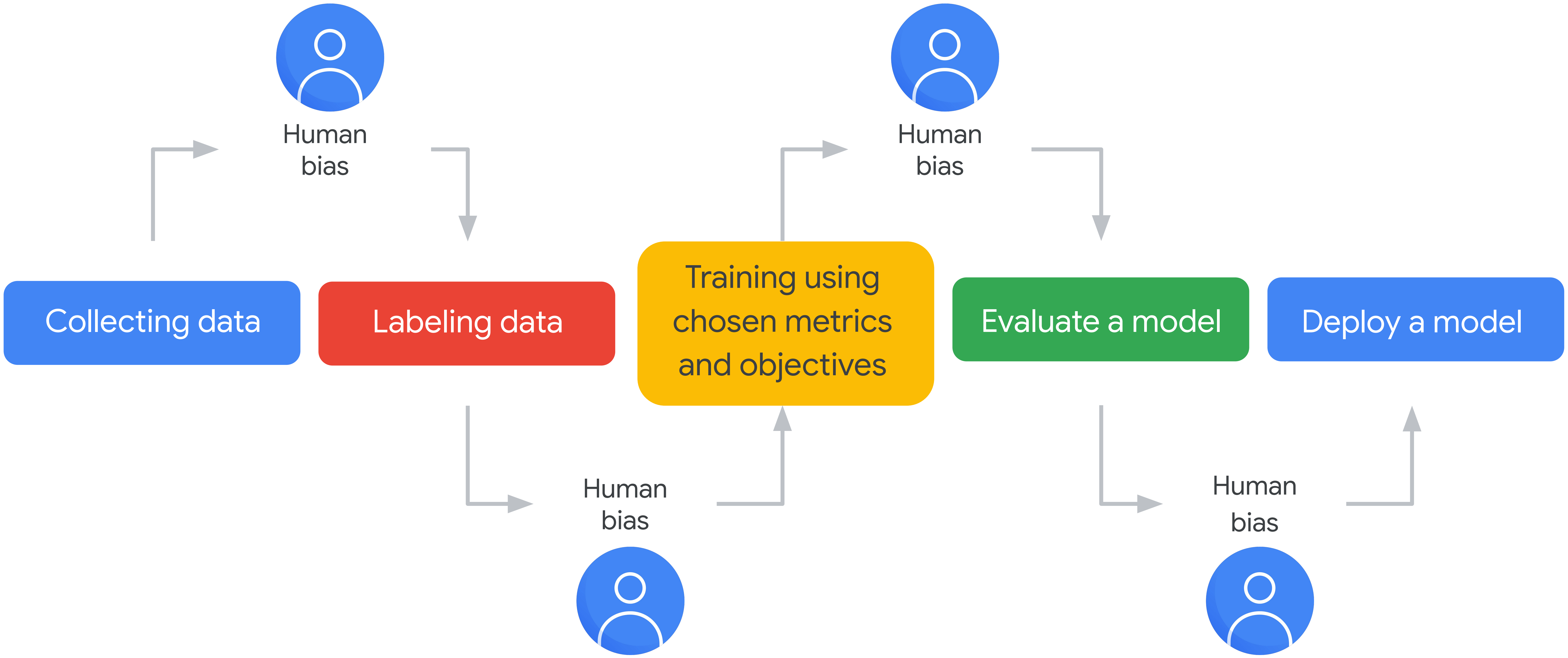
Unconscious bias from “the world” that we might reflect in ML when using existing data

Collecting data

Labeling data

Unconscious bias in our procedures that we might reflect in our ML

A typical ML pipeline with bias



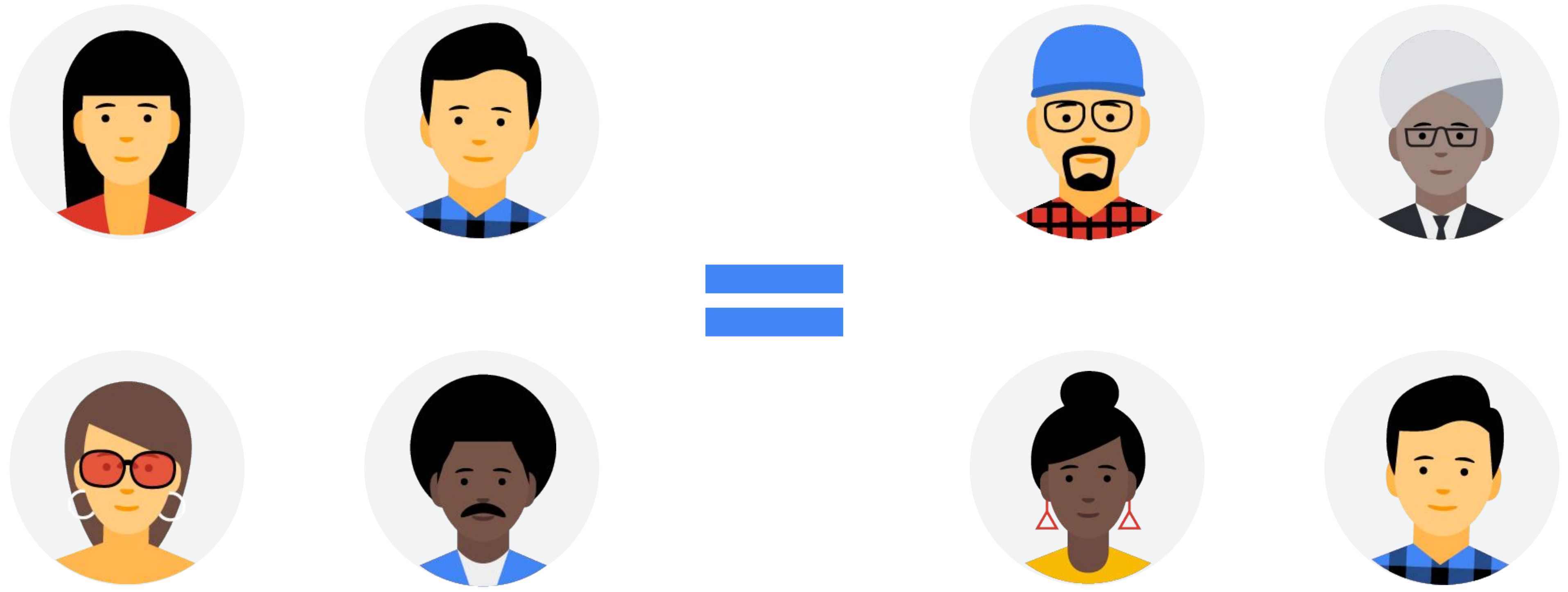
Avoid creating or reinforcing unfair bias

ML models learn from existing data collected from the real world, and so an accurate model may learn or even amplify problematic pre-existing biases in the data based on race, gender, religion, or other characteristics.

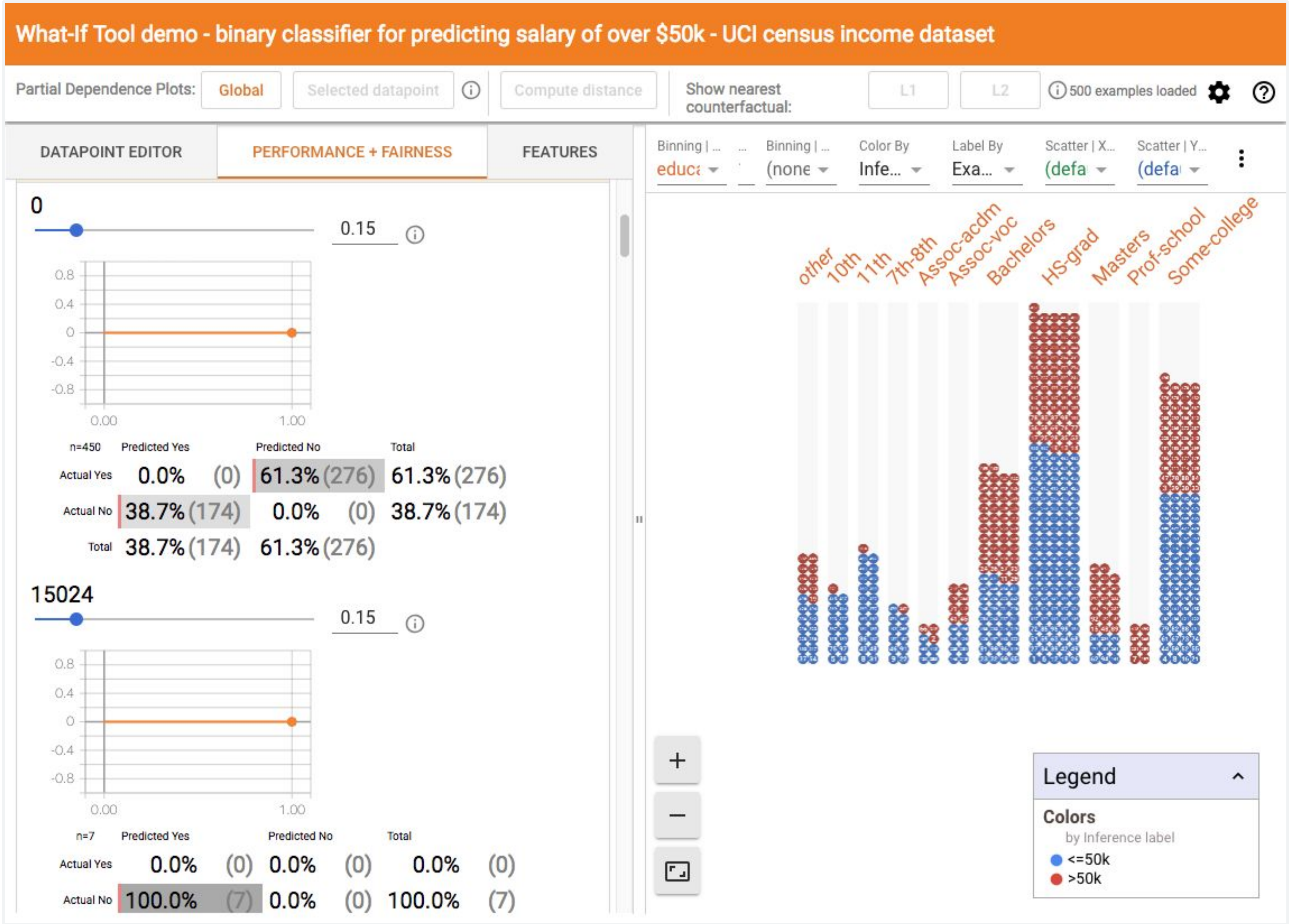
ai.google/principles

A checklist for bias-related issues

- Biometrics
- Race
- Skin color
- Religion
- Sexual orientation
- Socioeconomic status
- Income
- Country
- Location
- Health
- Language
- Dialect



Tools for responsible AI



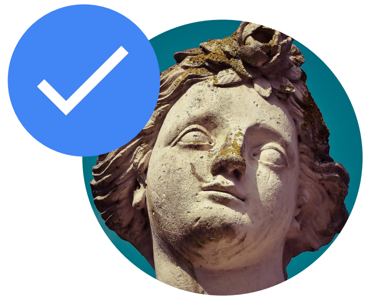
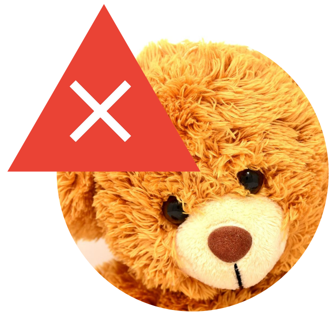
Understand the confusion matrix

Evaluate your model over subgroups also

The confusion matrix leads to evaluation metric insights

		Model predictions	
		Positive	Negative
Labels	Positive	**True positives (TP)** Label says something exists. Model predicts it. 	
	Negative		

The confusion matrix leads to evaluation metric insights

		Model predictions	
		Positive	Negative
Labels	Positive	**True positives (TP)** Label = something exists. Model predicts it. 	**False negatives (FN)** Type II error Label = something exists. Model doesn’t predict it. 
	Negative	**False positives (FP)** Type I error Something doesn’t exist. Model predicts it. 	**True negatives (TN)** Something doesn’t exist. Model doesn’t predict it. 

False positives and false negatives errors occur when predictions and labels disagree

Labels		Model predictions: Positive	Model predictions: Negative
	Positive	True positives (TP) 	False negatives (FN) Type II error **Model says: no** 
	Negative	False positives (FP) Type I error **Model says: yes** 	True negatives (TN) 

Evaluation metrics can help highlight areas where machine learning could be more inclusive

Model predictions

Labels		Positive	Negative
	Positive	**True positives (TP)** Label says something exists. The model predicts it.	**False negatives (FN)** Type II error Label says something exists. Model doesn't predict it. **Model says: no**
	Negative	**False positives (FP)** Type I error Label says something doesn't exist. Model predicts it. **Model says: yes**	**True negatives (TN)** Label says something doesn't exist. Model doesn't predict it.

False negative rate is the fraction of true faces that are not detected by the ML system

Model predictions

Labels	Positive	Negative
Positive	**True positives (TP)** Label says something exists. The model predicts it.	Type II error Label says something exists. Model doesn't predict it.
Negative		

$$\text{False negative rate} = \frac{\text{False negatives}}{\text{False negatives + True positives}}$$

False positive rate is the fraction of the faces that the ML model detects that are not really faces

		Model predictions	
		Positive	Negative
Labels	Positive	**True positives (TP)** Label says something exists. The model predicts it.	False positive rate = $\dfrac{\text{False positives}}{\text{False positives + True negatives}}$
	Negative	Type I error Label says something doesn't exist. Model predicts it.	

Sometimes, false positives are better than false negatives

Privacy in images

False positive

False negative

Sometimes, false negatives are better than false positives

False negative:

E-mail that is SPAM is not caught, so you see it in your inbox.

Sometimes, false negatives are better than false positives

False negative:

E-mail that is SPAM is not caught, so you see it in your inbox.

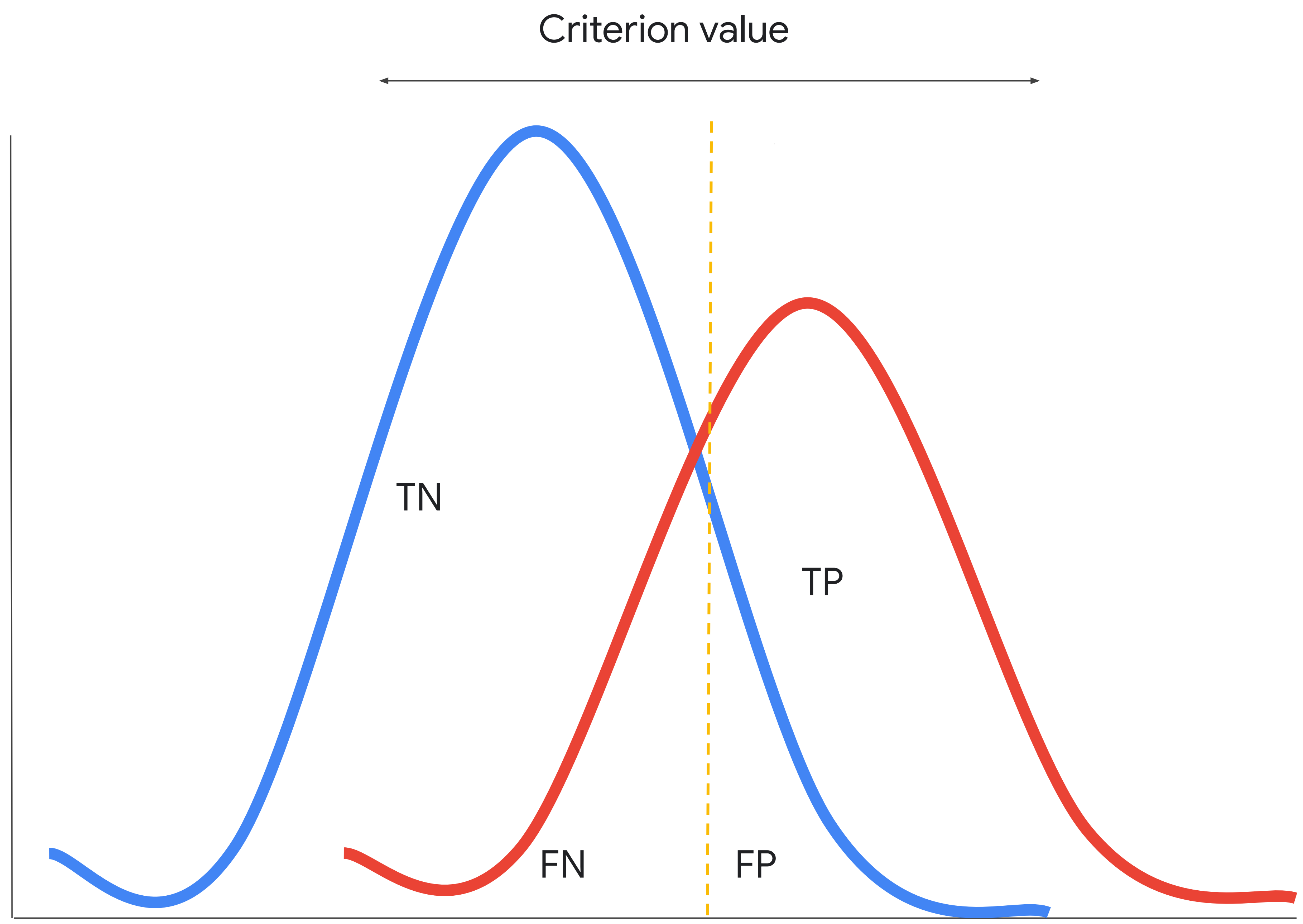
Jan Smith	Win the lottery with these numbers!

False positive:

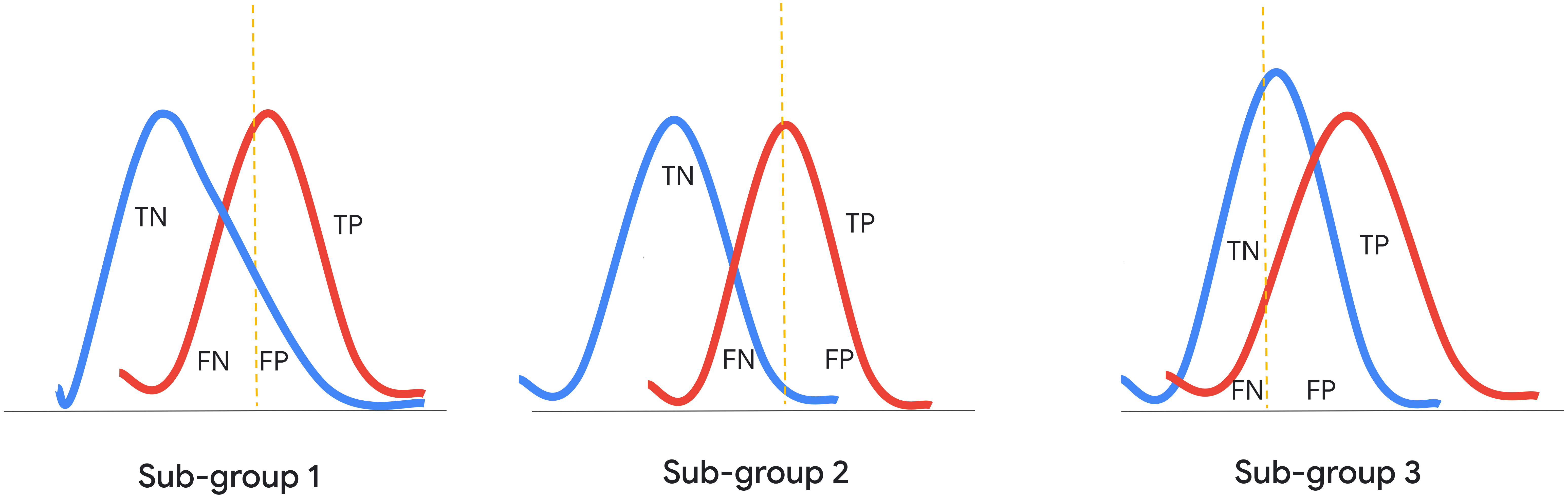
E-mail flagged as SPAM is removed from your inbox.

Karla Brown	Lunch today?

Find the threshold that brings the precision or recall to acceptable values



Check the precision/recall you obtain with that threshold in each of your subgroups



Evaluating metrics are some of the key things you can do to measure how inclusive an ML system is

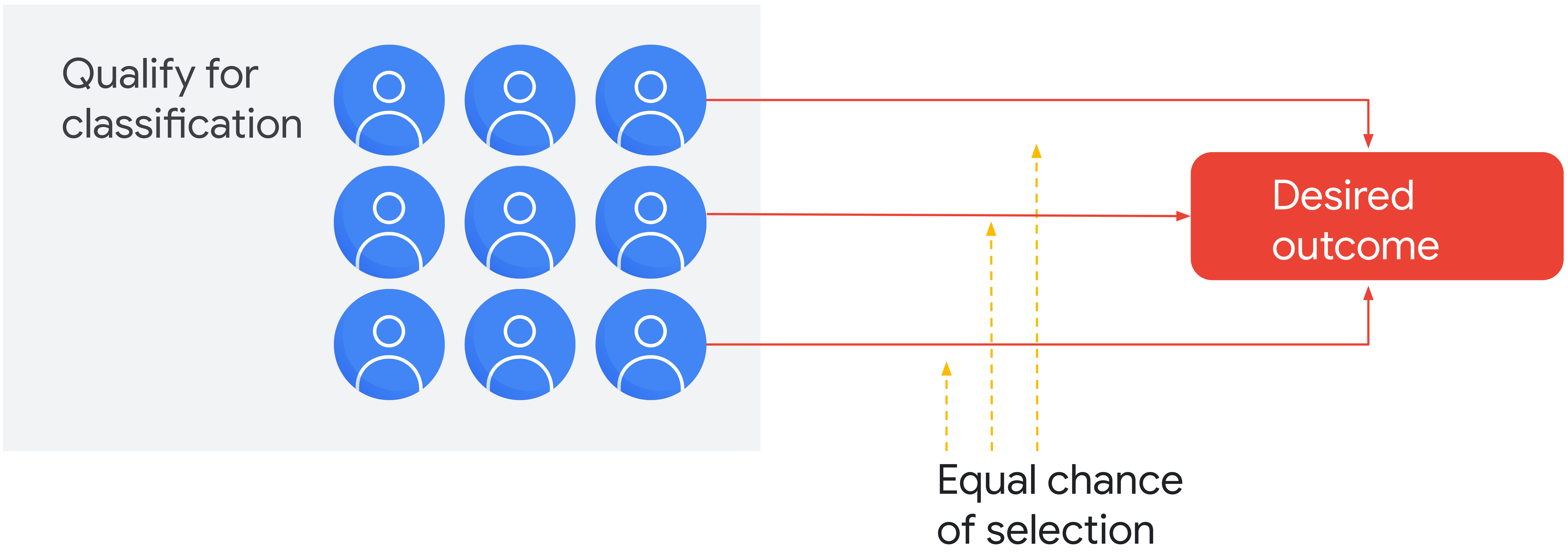
Choose your evaluation metrics in light of acceptable tradeoffs between false positives and false negatives.

Understand errors

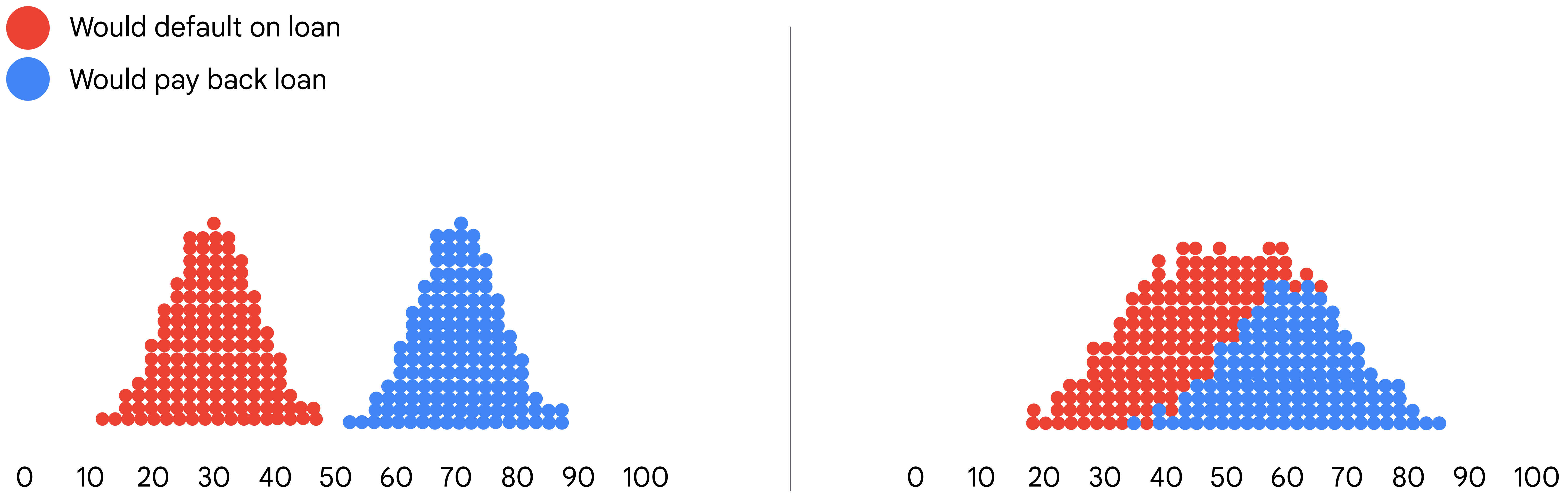
Understand errors

Evaluate inclusion

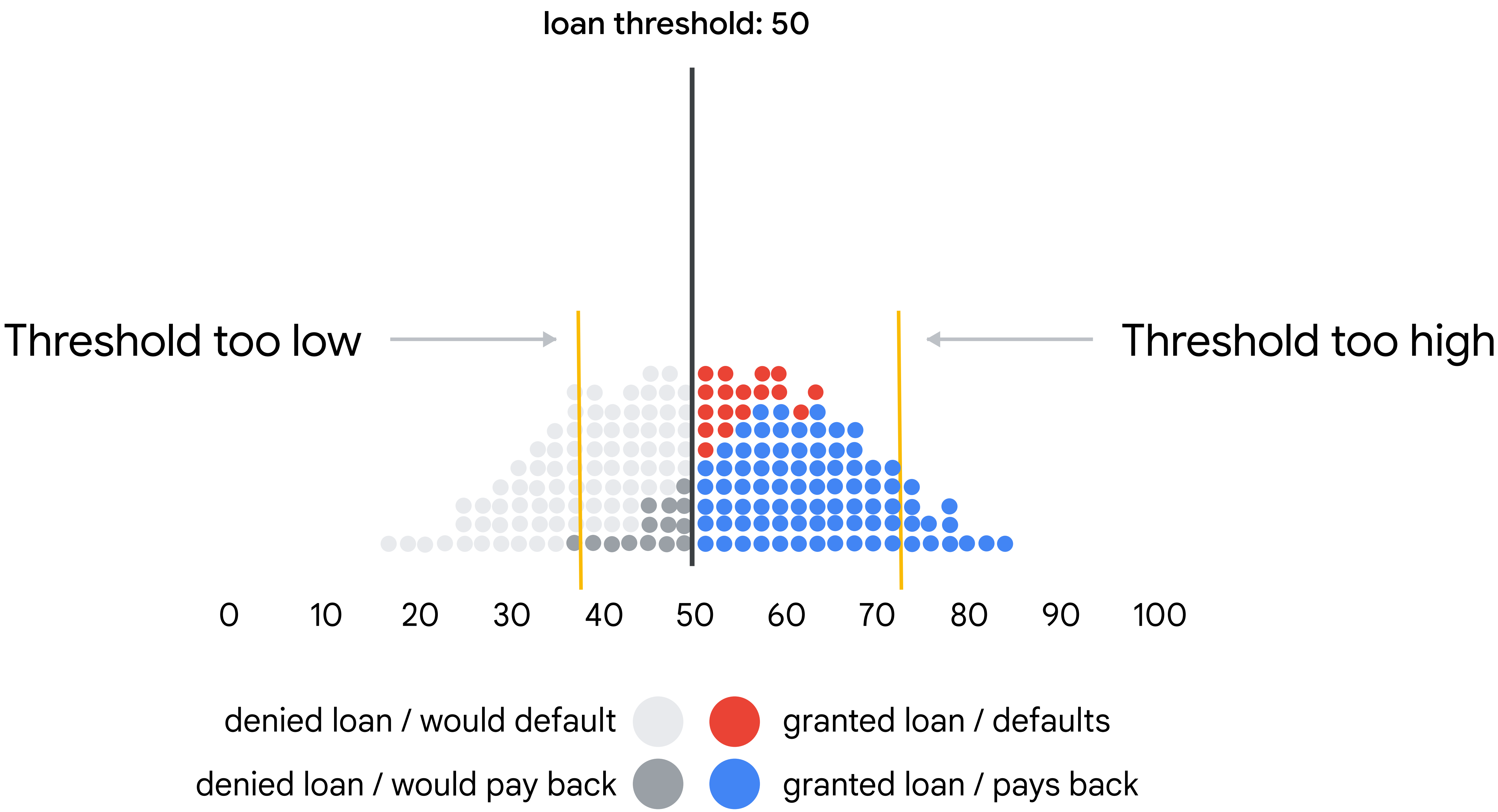
The equality of opportunity approach strives to give individuals an equal chance of desired outcome



A toy classifier to predict who will pay back their loan involves two populations that might overlap

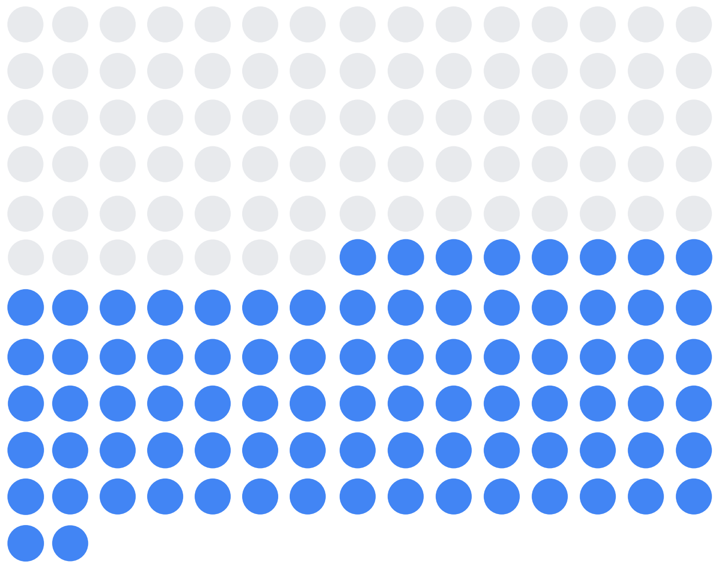


Picking a credit score threshold involves a tradeoff

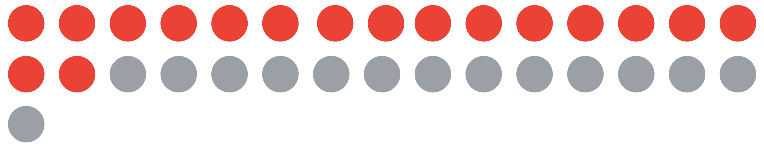


The impact of a threshold on credit score is evaluated based on its impact on customers and loan repayment

Correct 84%
Loans granted to paying applicants and denied to defaulters



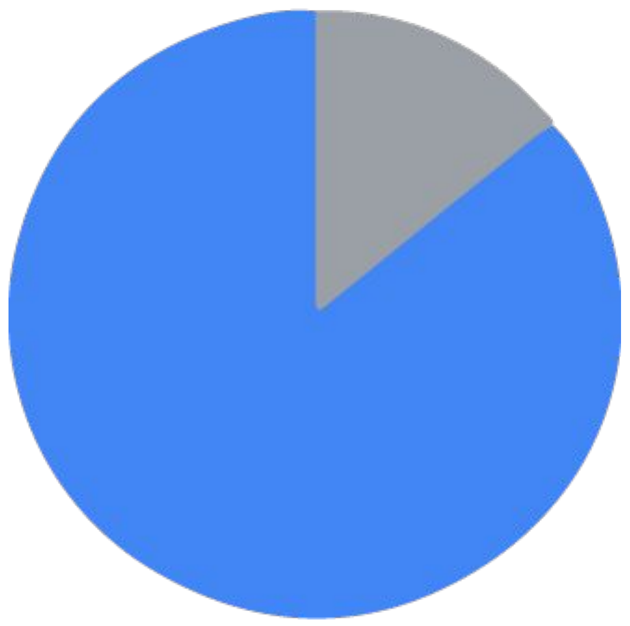
Incorrect 16%
Loans denied to paying applicants and granted to defaulters



Simulating the impact of a threshold on profit

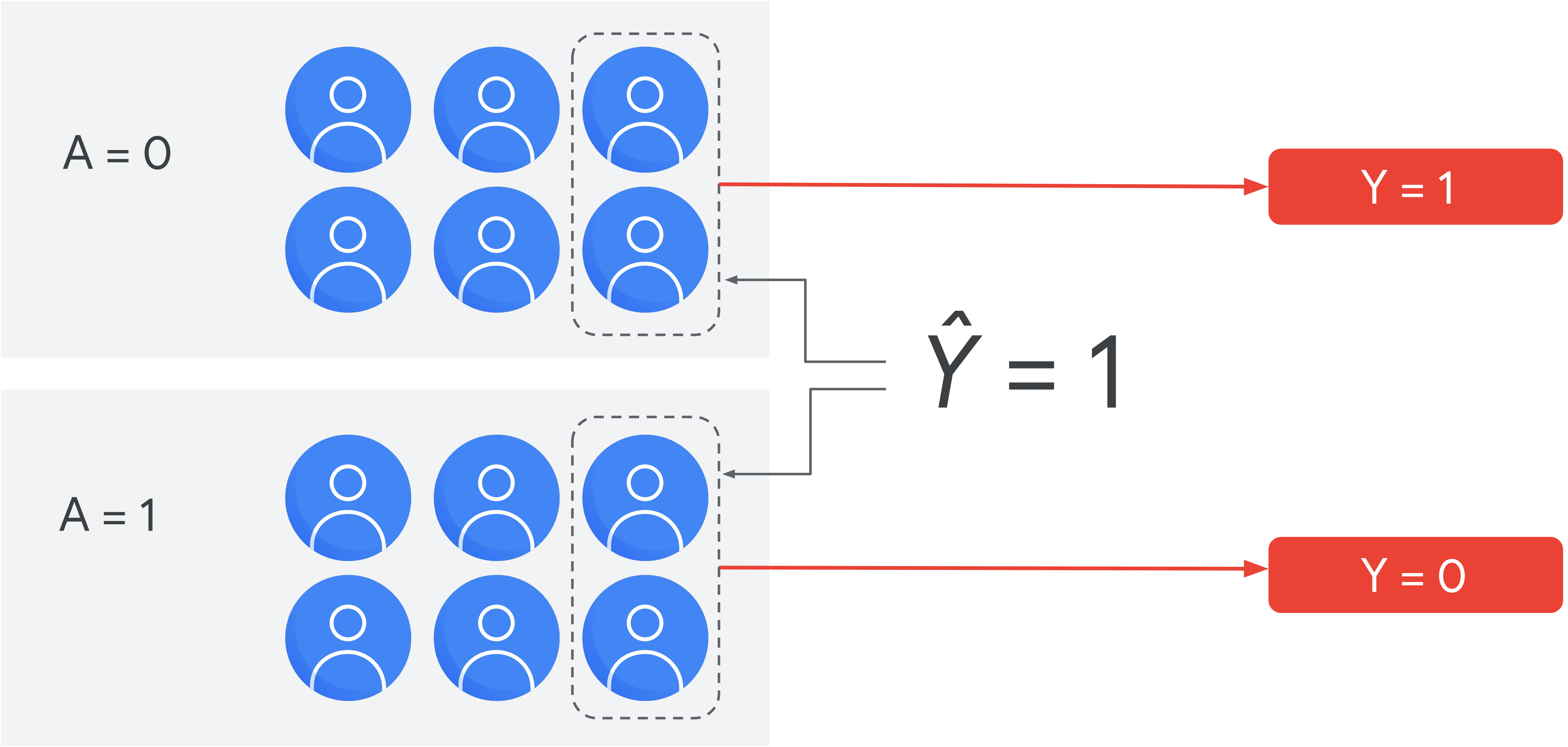
Profit: 13600

True positive rate 86%
Percentage of paying applications getting loans

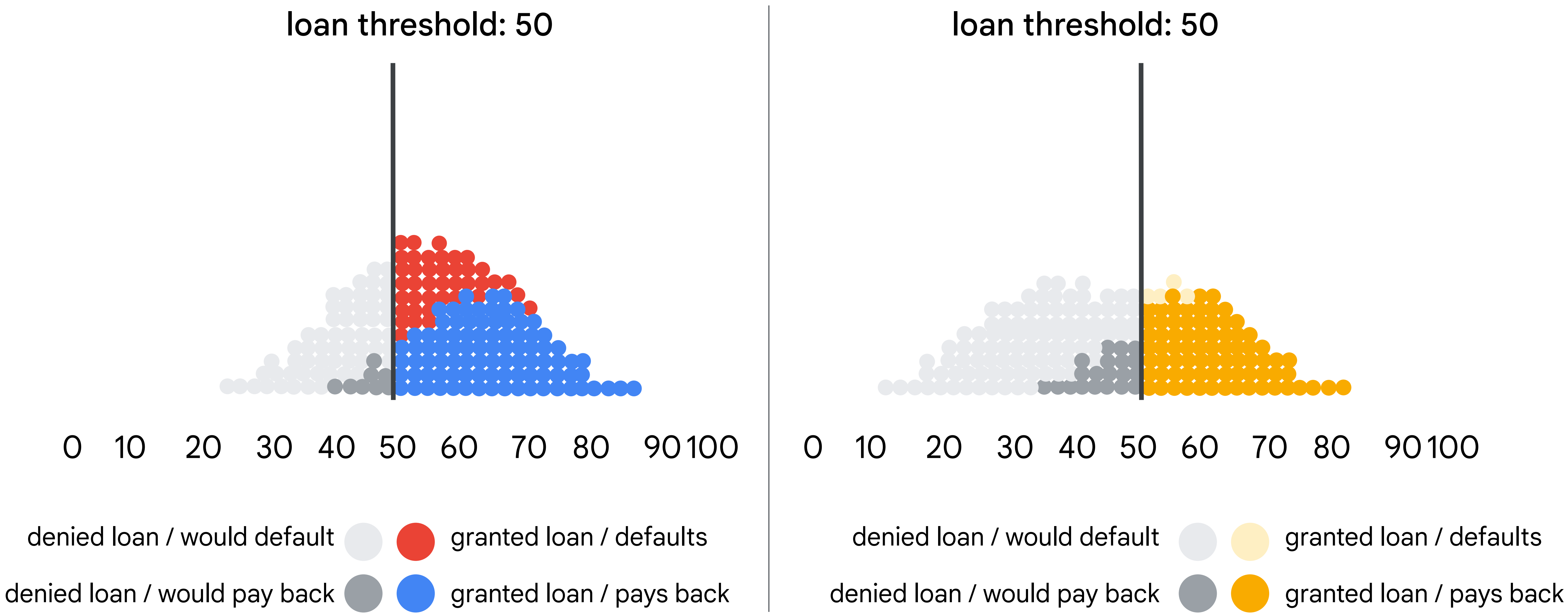


Positive rate 52%
Percentage of all applications getting loans

Classification and discrimination must obey the equality of opportunity principle

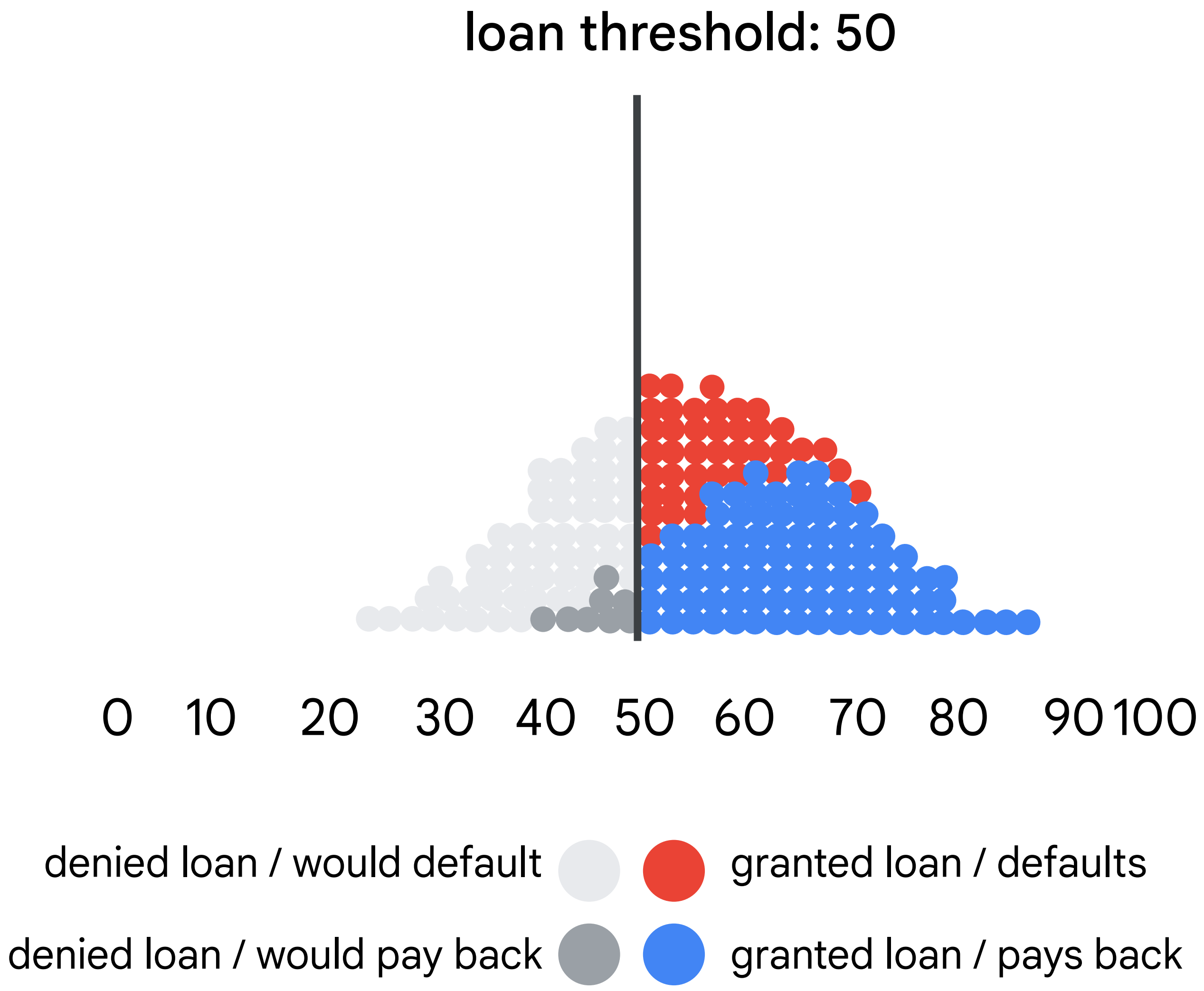


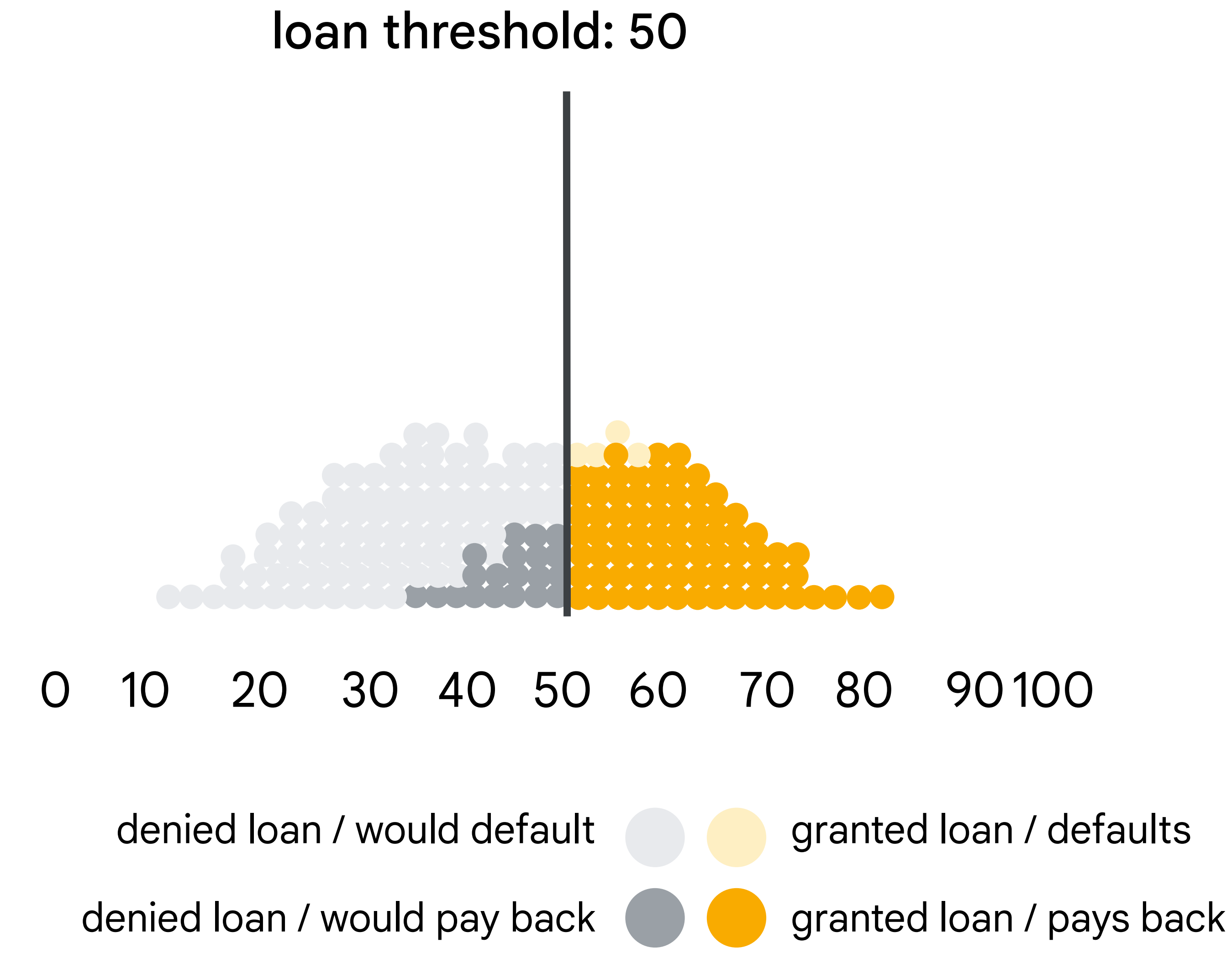
Simulating decisions with no constraints can lead to unequal distribution



Simulating decisions with no constraints can lead to unequal distribution

- A successful loan makes $300
- An unsuccessful loan costs $700
- Credit scores are between 0 - 100





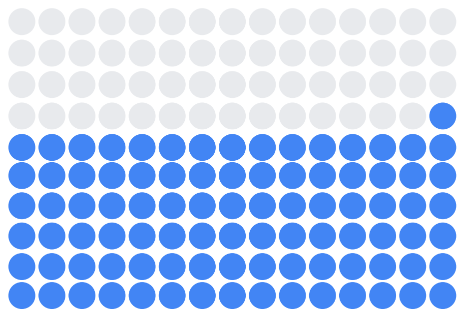
Simulating decisions with no constraints can lead to unequal distribution

Threshold

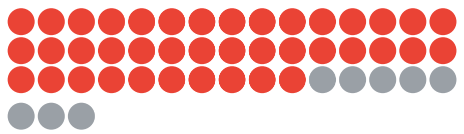
- Credit score of 50 for blue group
- Credit score of 50 for orange group

Total profit: 19600

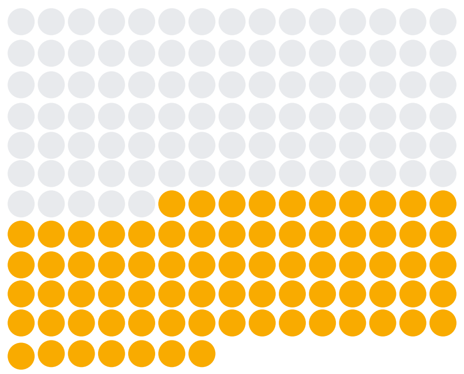
Correct 76%
Loans granted to paying applicants and denied to defaulters



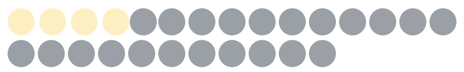
Incorrect 24%
Loans denied to paying applicants and granted to defaulters



Correct 87%
Loans granted to paying applicants and denied to defaulters



Incorrect 13%
Loans denied to paying applicants and granted to defaulters



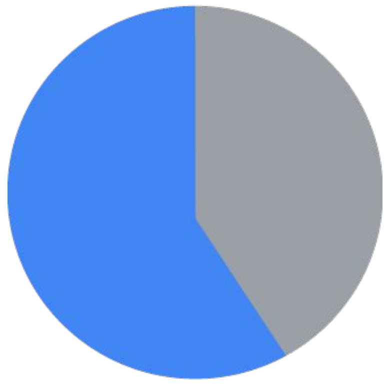
Simulating decisions for max profit result in unequal standards

Threshold

- Credit score of 50 for blue group
- Credit score of 50 for orange group

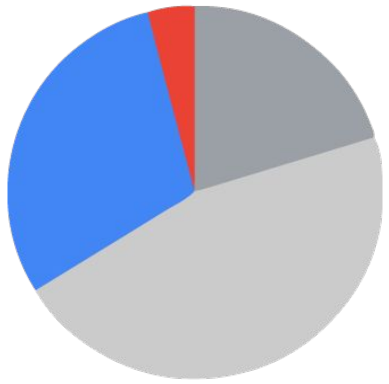
Total profit: 32400

True positive rate 60%
Percentage of paying applicants getting loans

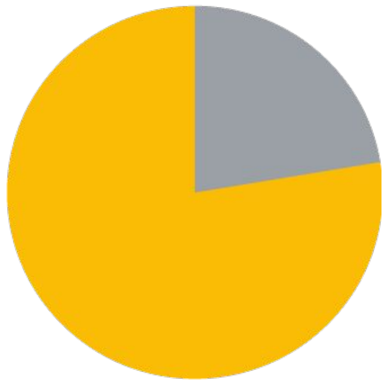


Profit: **12100**

Positive rate 34%
Percentage of all applicants getting loans



True positive rate 78%
Percentage of paying applicants getting loans

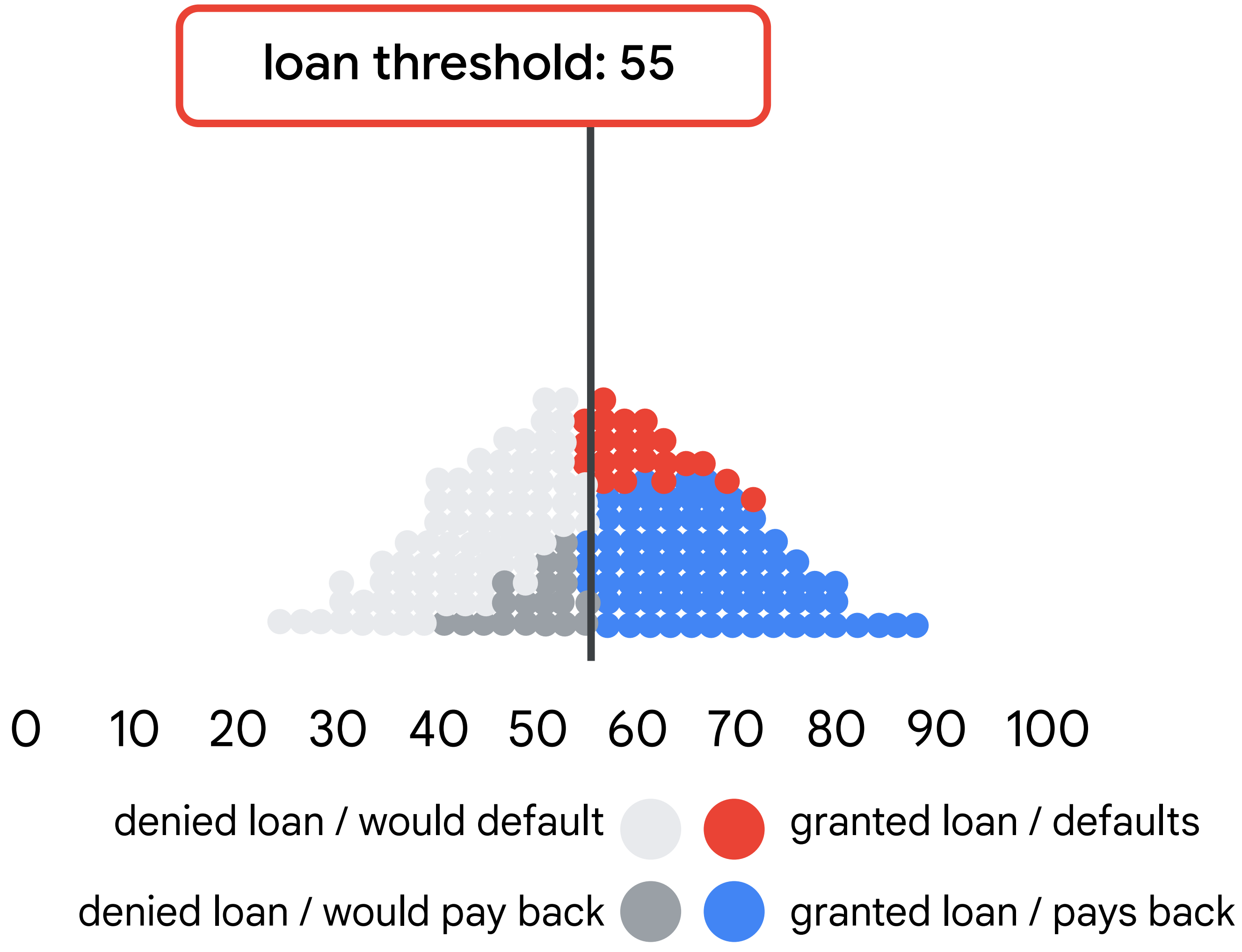


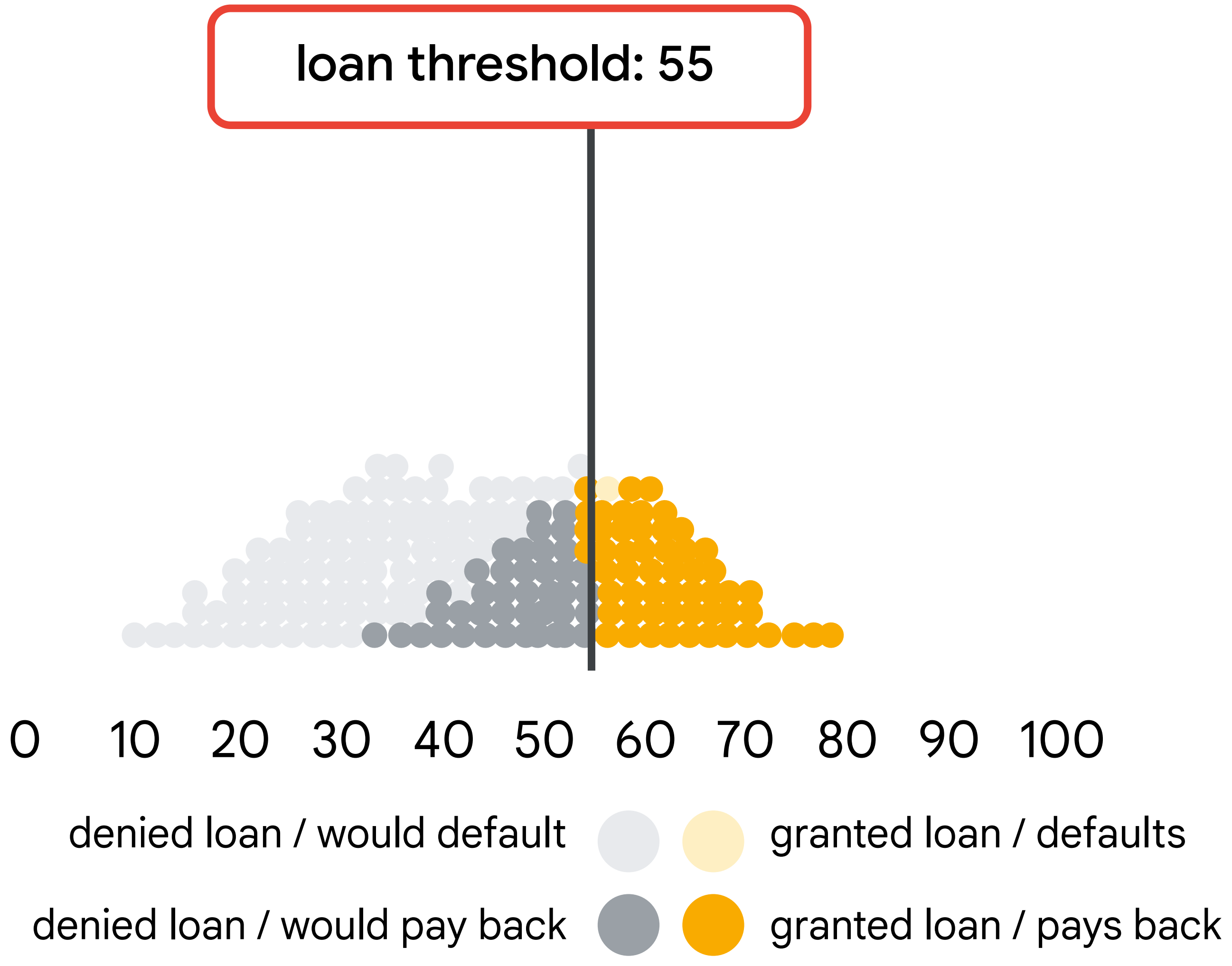
Profit: **20300**

Positive rate 41%
Percentage of all applicants getting loans

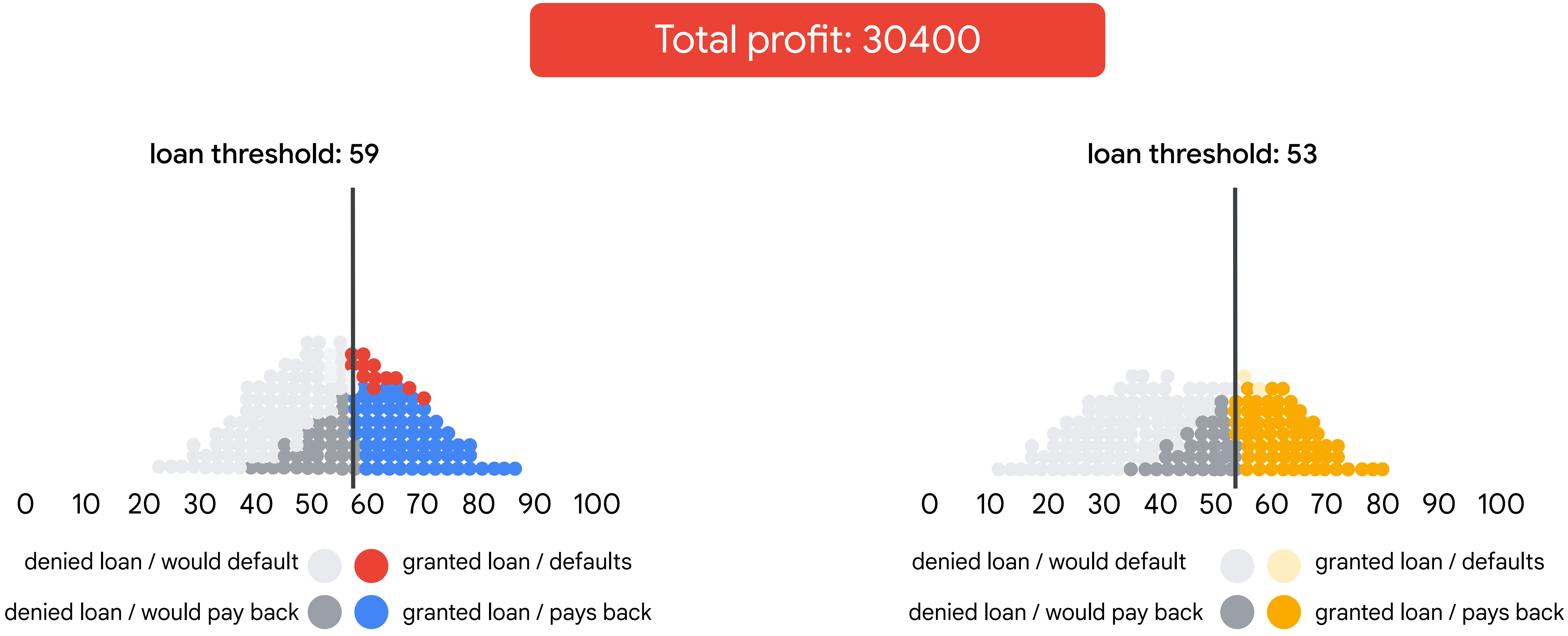
Simulating decisions with group unaware holds everyone to the same standard, which can be unfair to some groups

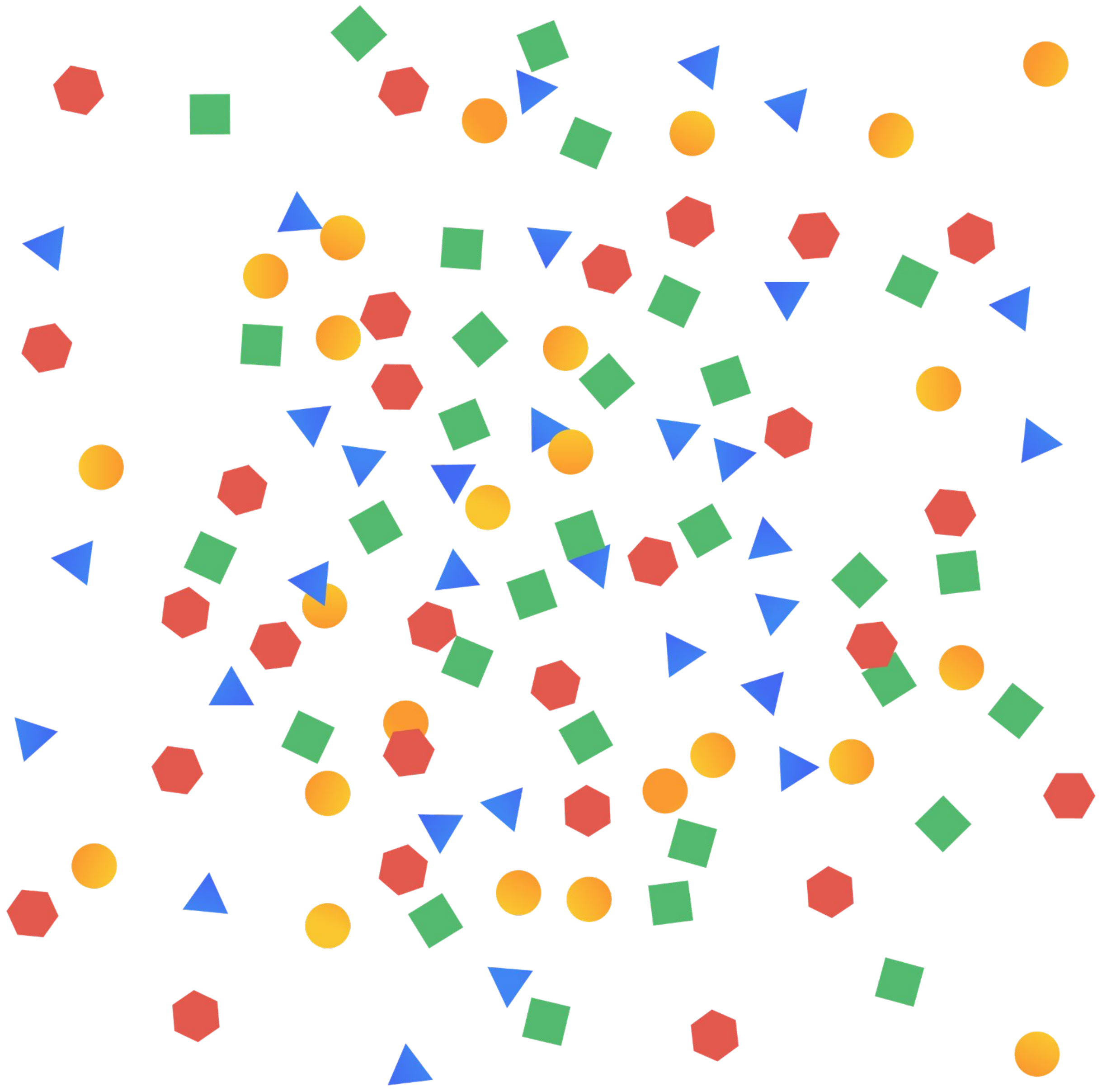
Total profit: 25600

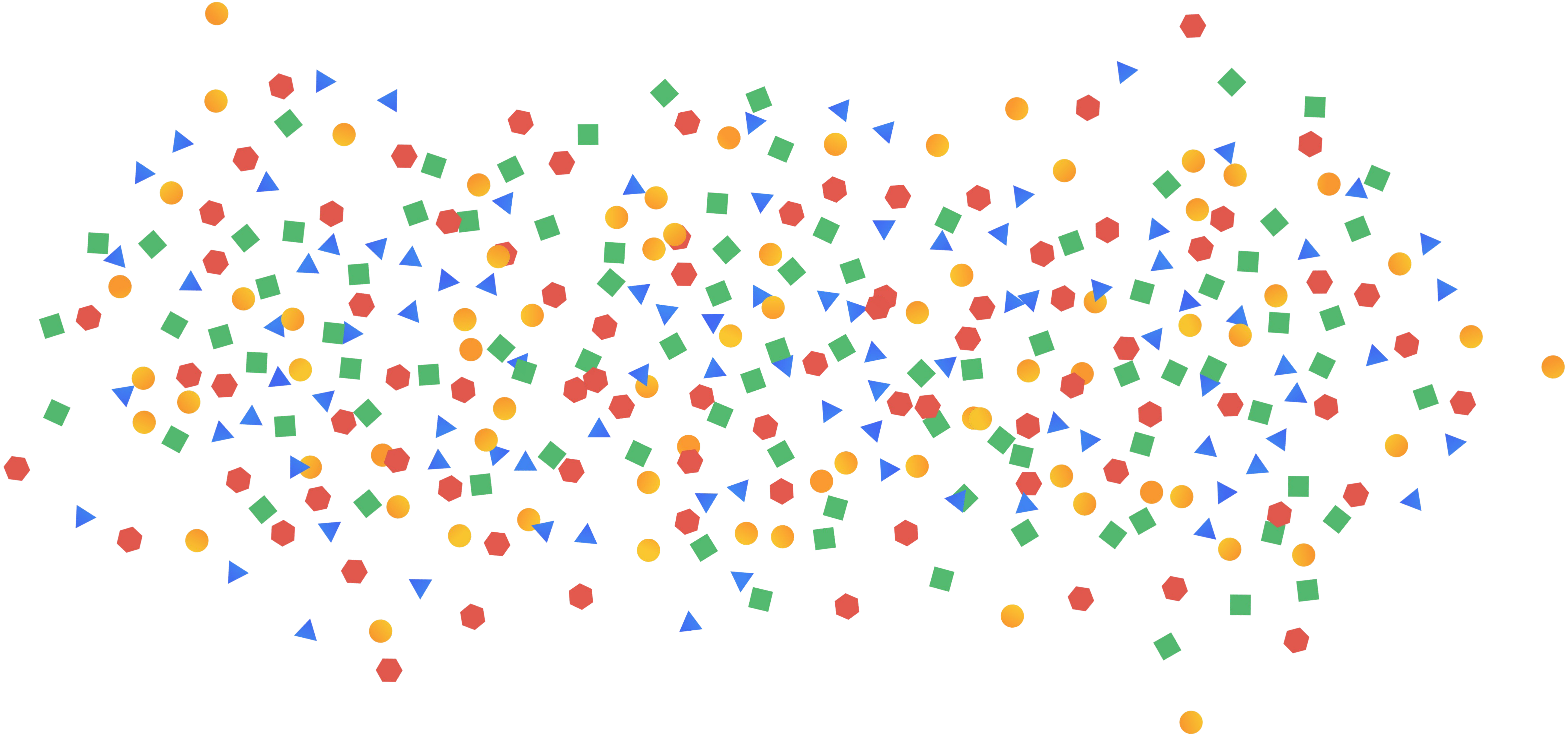




Simulating decisions equal opportunity results in an identical true positive rate for all groups

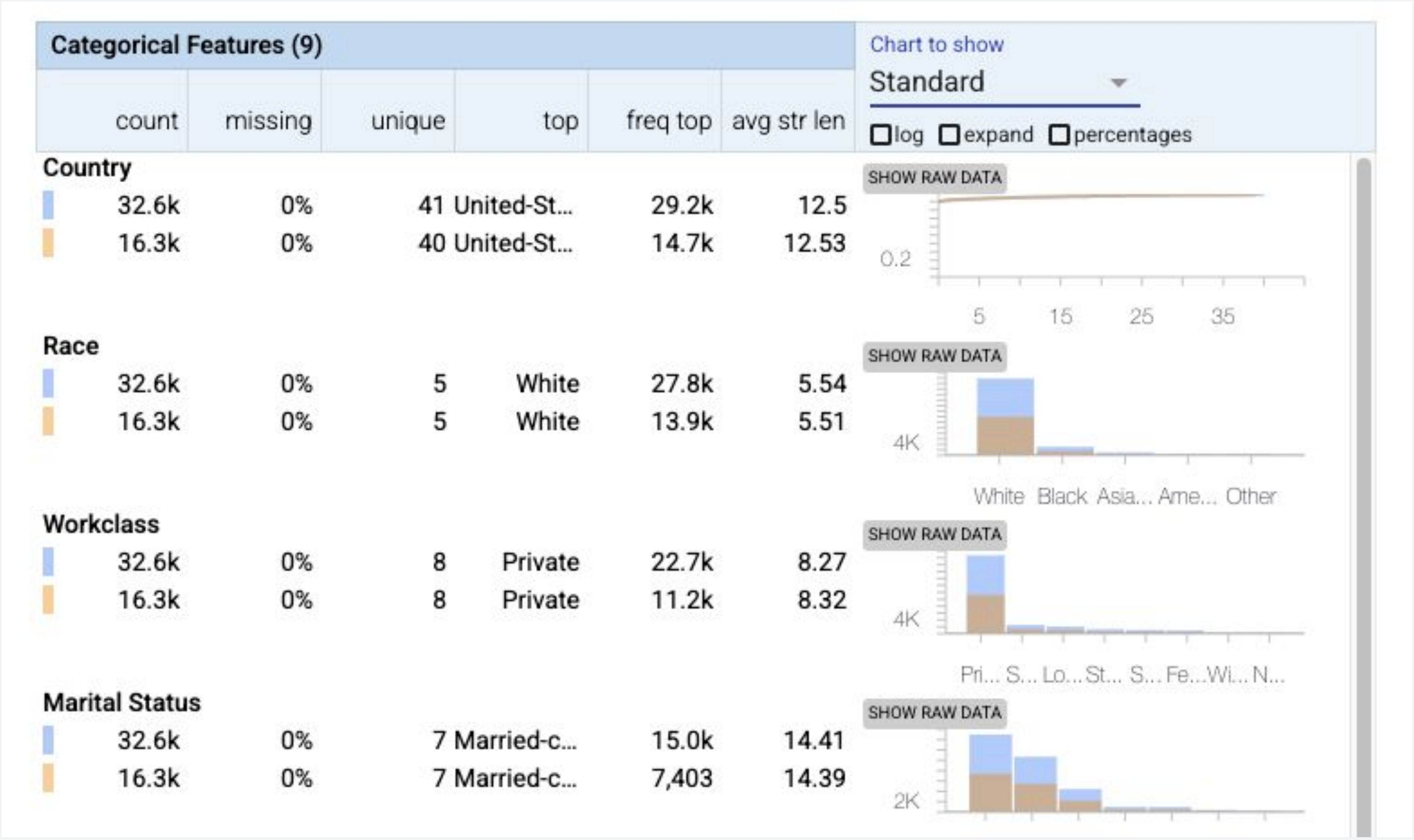




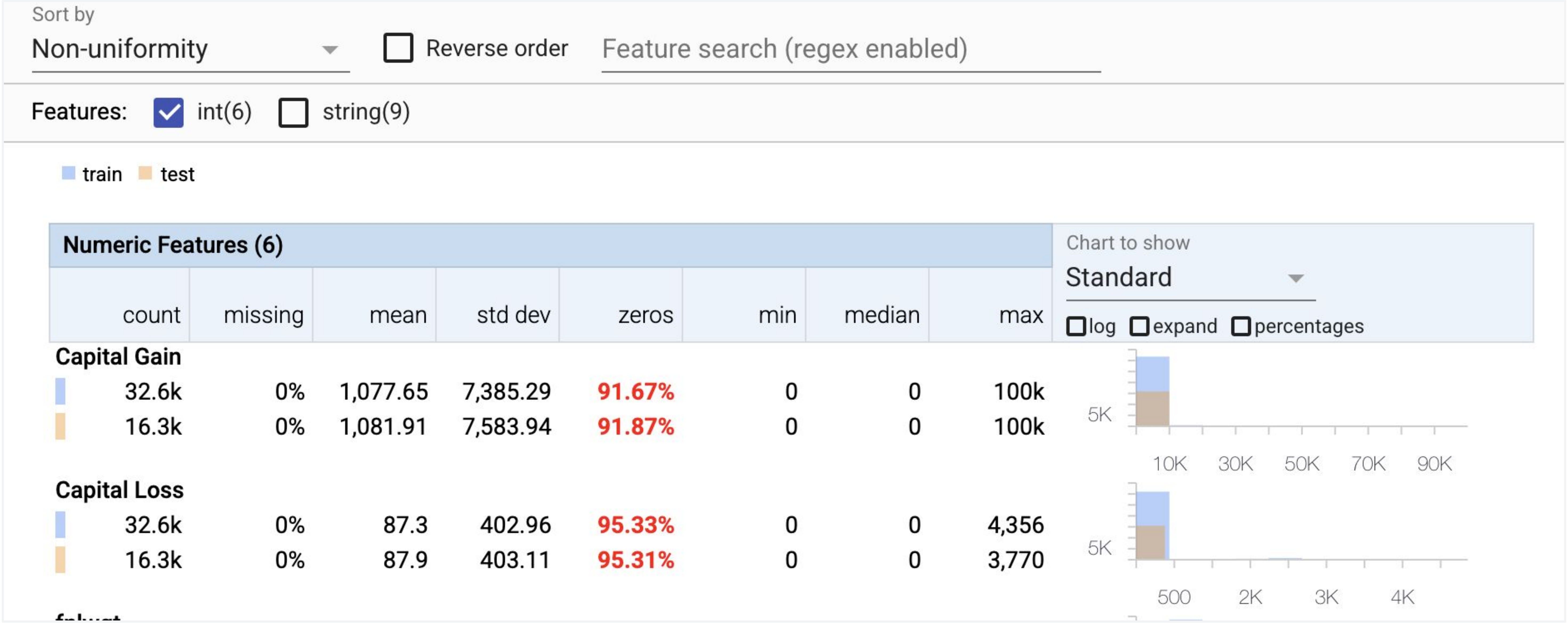


Facets

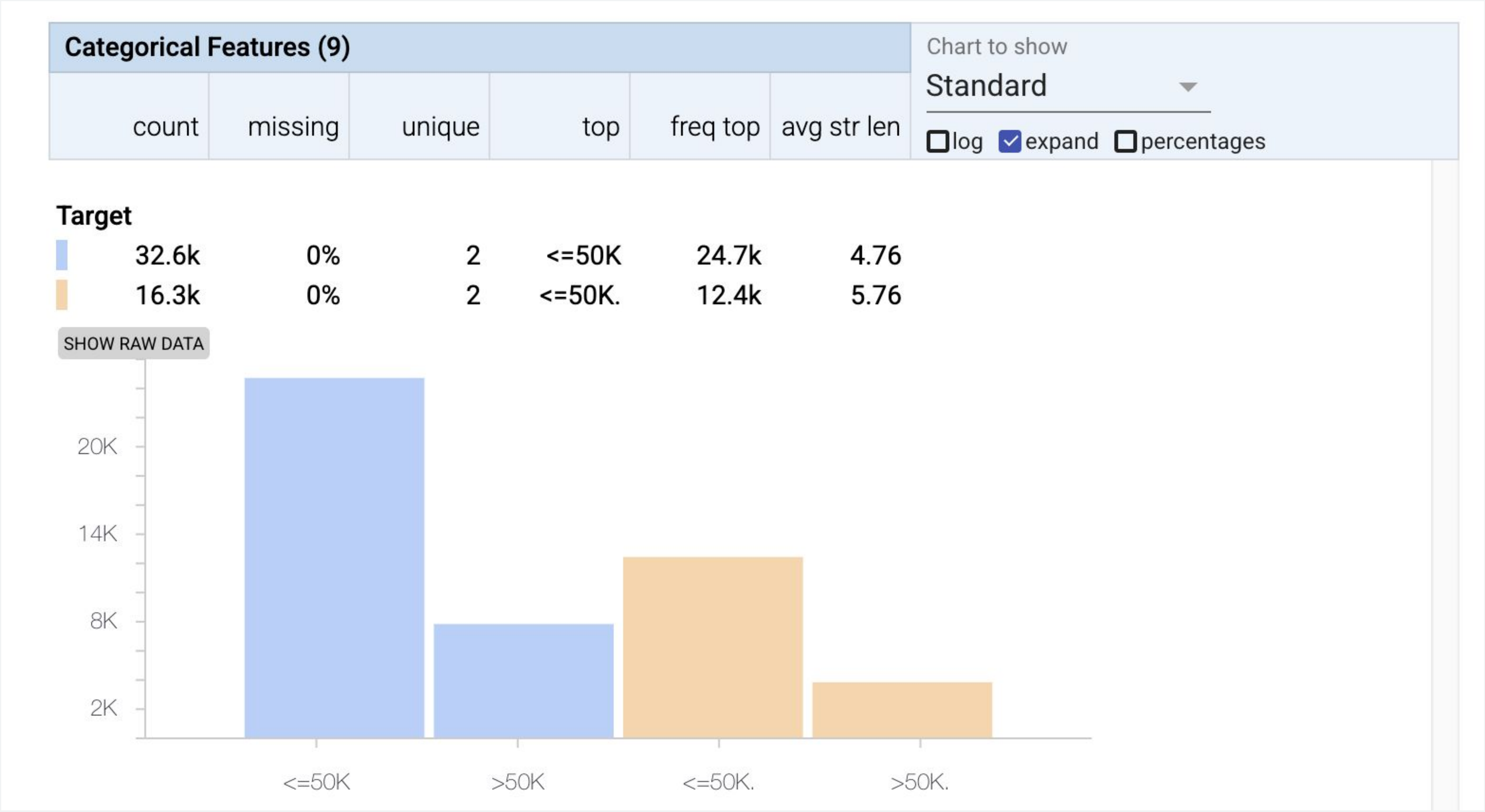
Facets gives users a quick understanding of the distribution of values across features



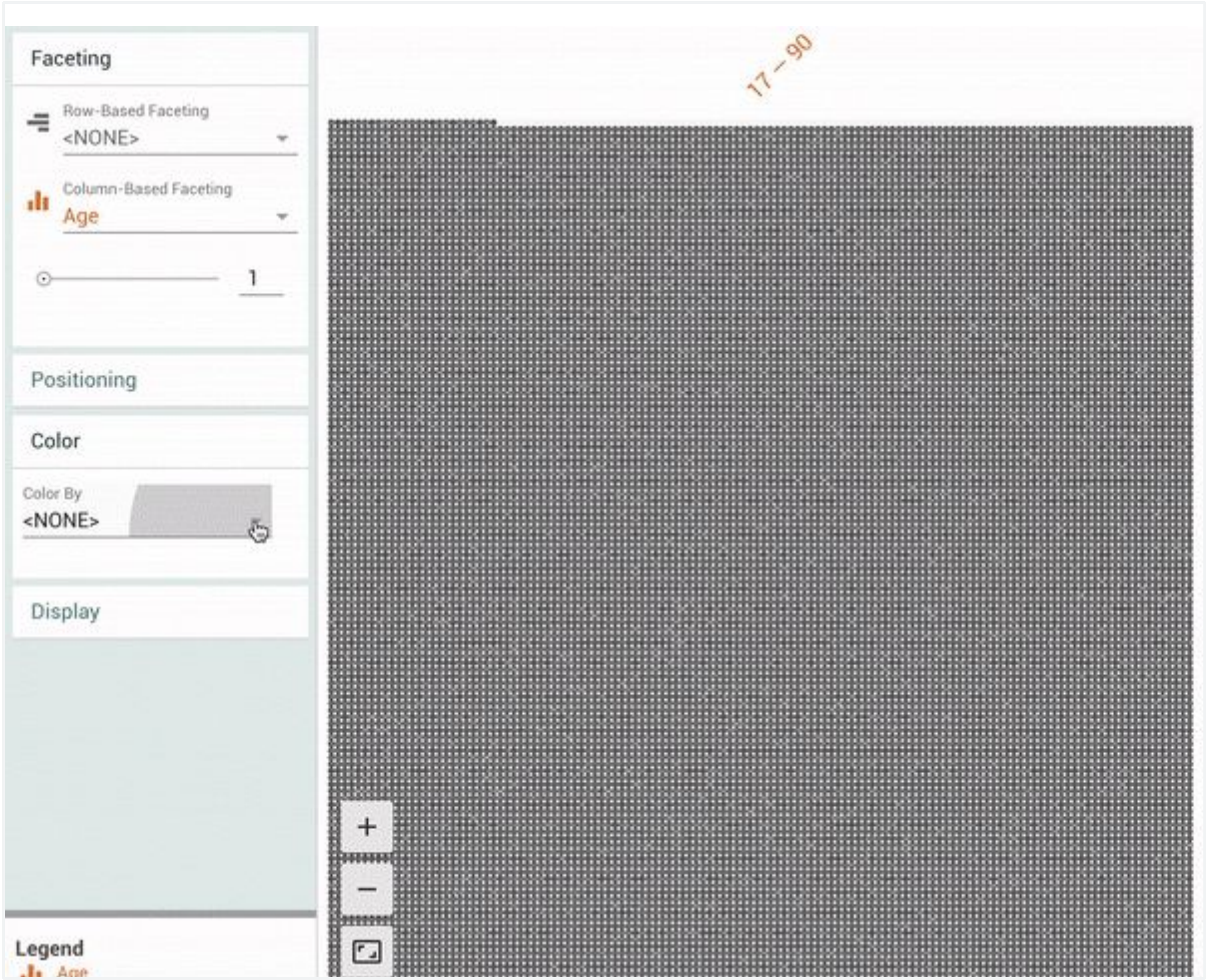
Facets gives users a quick understanding of the distribution of values across features



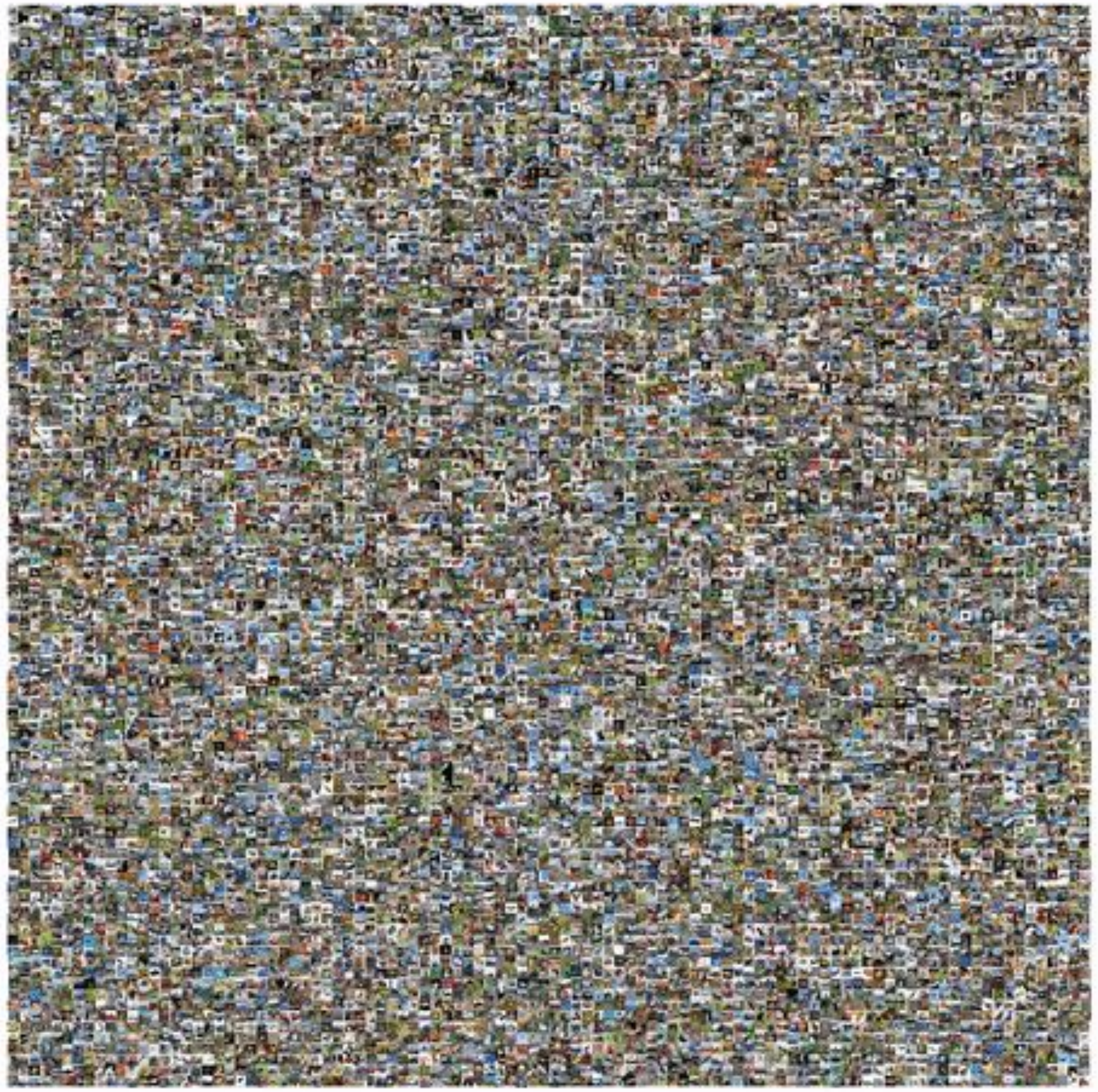
Facets features are sorted by distribution distance



Facets Dive provides an easy-to-customize, intuitive interface



Explore CIFAR-10 for errors using Facets Dive



Facets help you discover new and interesting things about your data