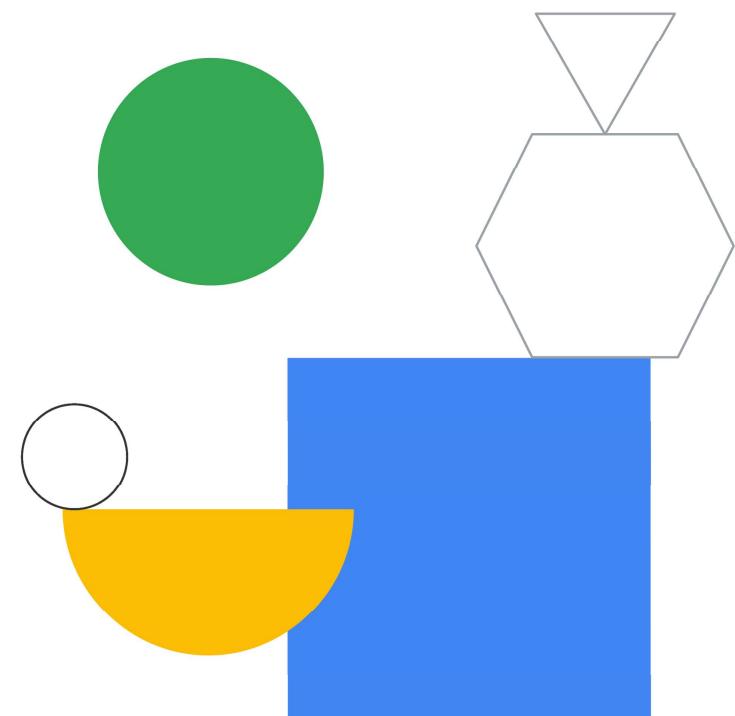
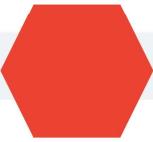




# How Google Does Machine Learning



# Practical, real-world introduction to ML



Data Analysts  
Citizen Data Scientists

Build without writing a  
single piece of code



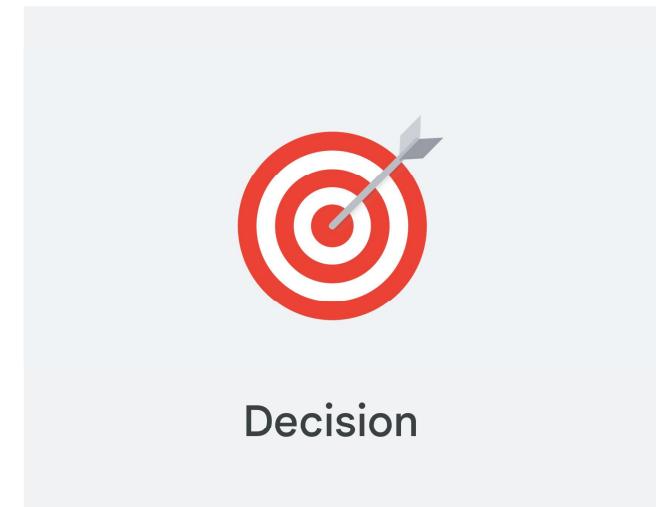
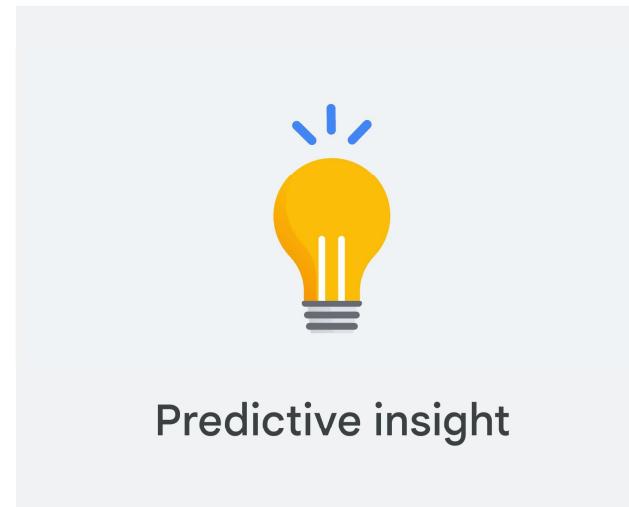
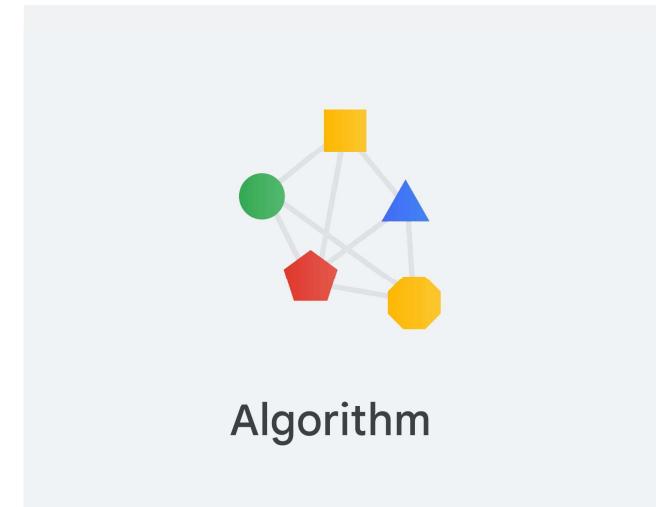
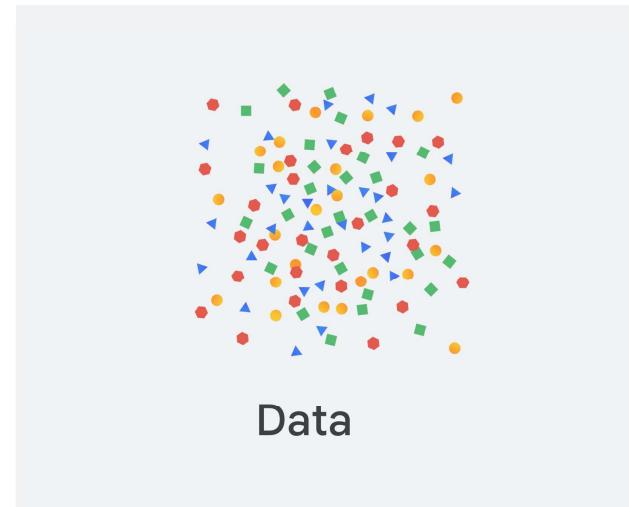
ML Engineers  
ML Scientists

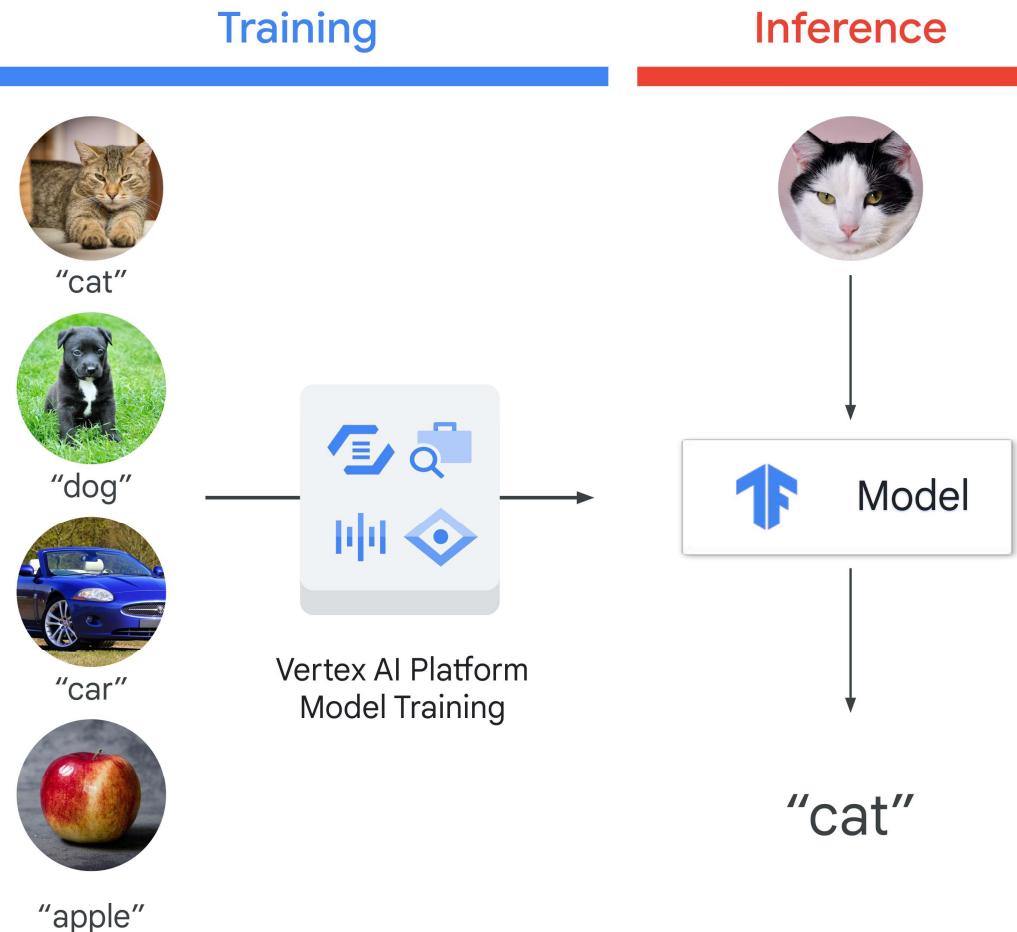
Go “code deep” with  
custom training

What is machine learning?

What does it mean to be AI-first?

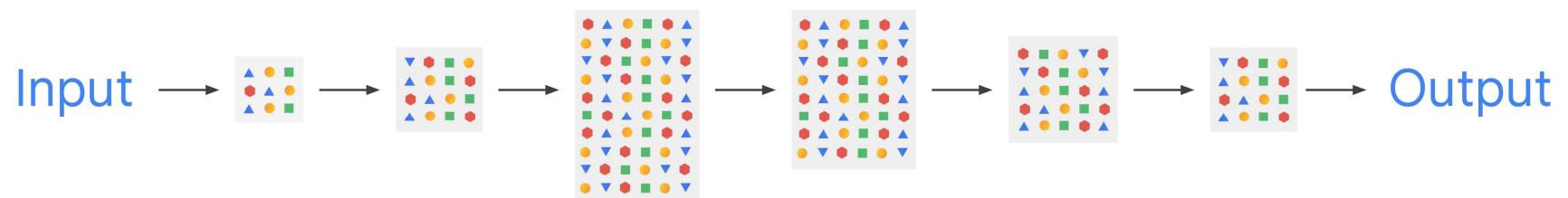
**Machine learning is a way to use standard algorithms to derive predictive insights from data and make repeated decisions.**



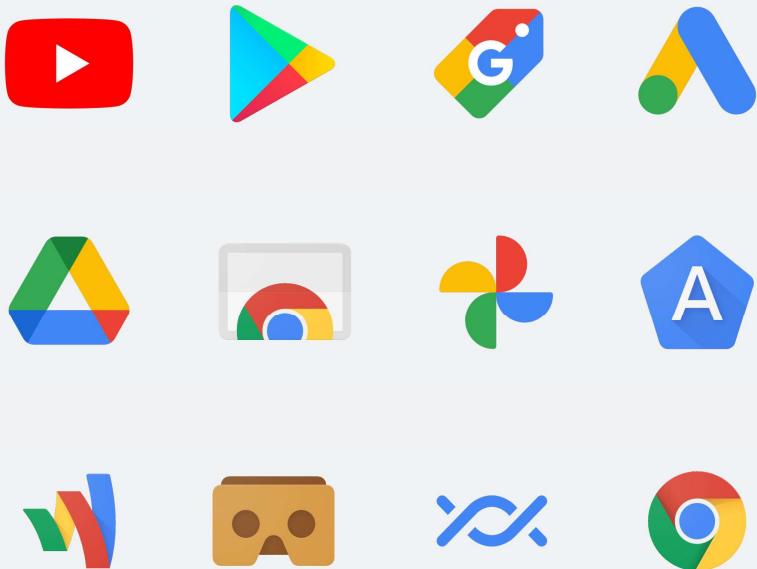


**Focus on both the  
training and  
inference stages  
of ML**

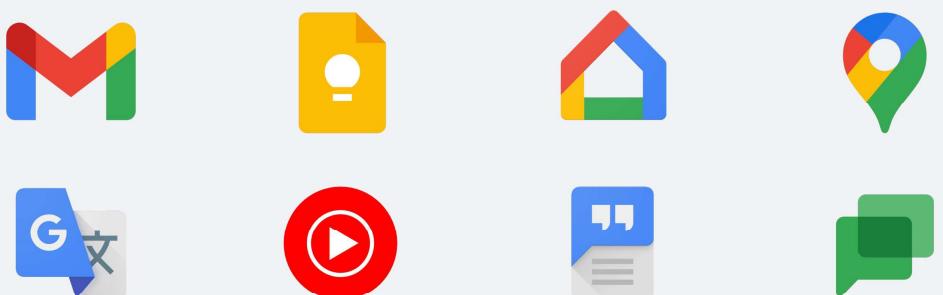
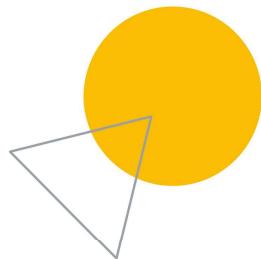
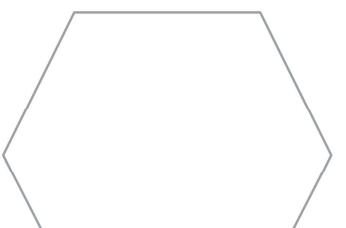
# Neural networks are one important technology we use



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.

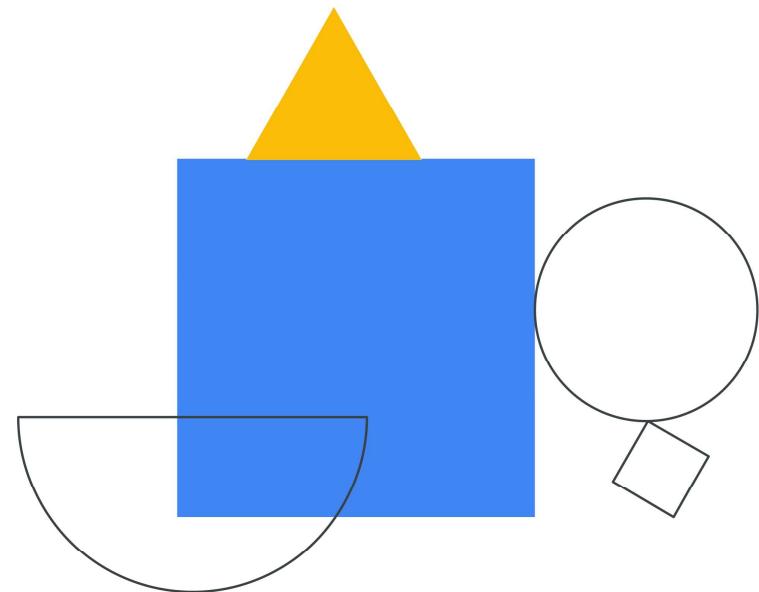


**Gmail**

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

# 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

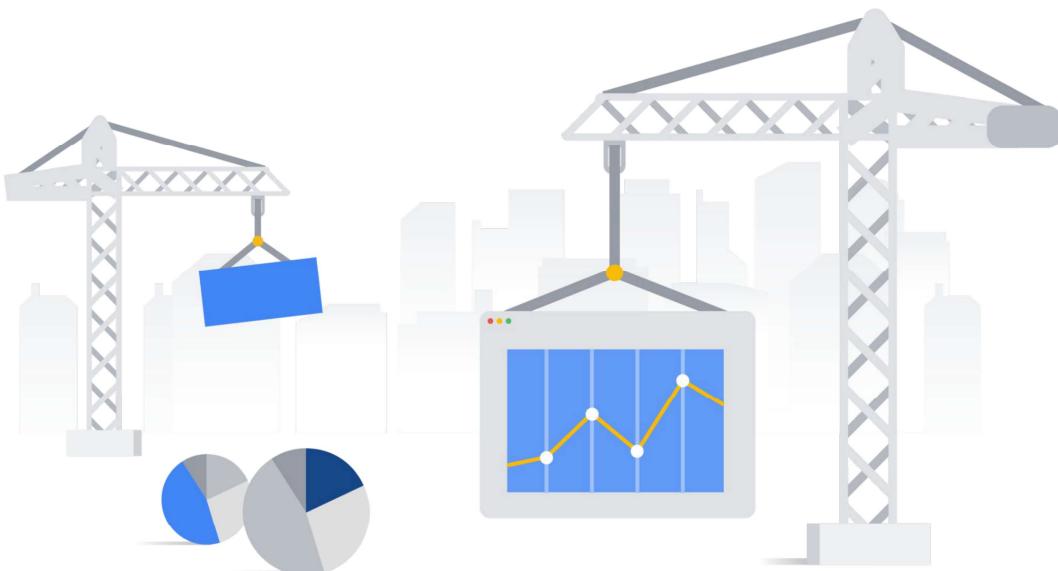


## Energy, Feedstock and Utilities

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

# Example solution:

## Demand forecasting in manufacturing



ML problem:

What is being predicted?

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

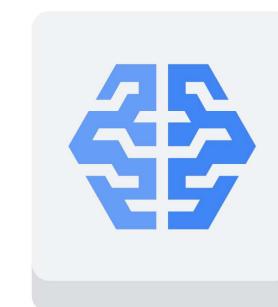
What data is needed?

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

# **Pre-trained models**



How much is this  
car worth?



AUCNET

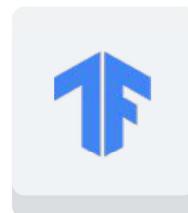


# There are pre-trained machine learning services available on Google Cloud

## Custom ML models

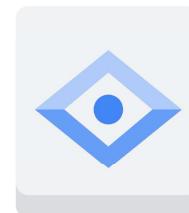


Vertex AI

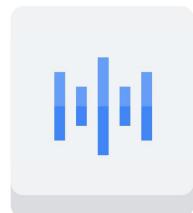


TensorFlow

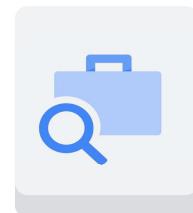
## Pre-trained ML Models



Vision API



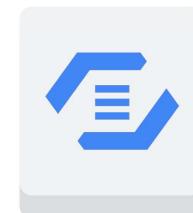
Speech API



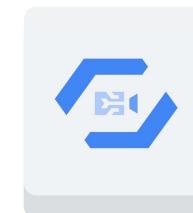
Jobs API



Translation API



Natural Language API

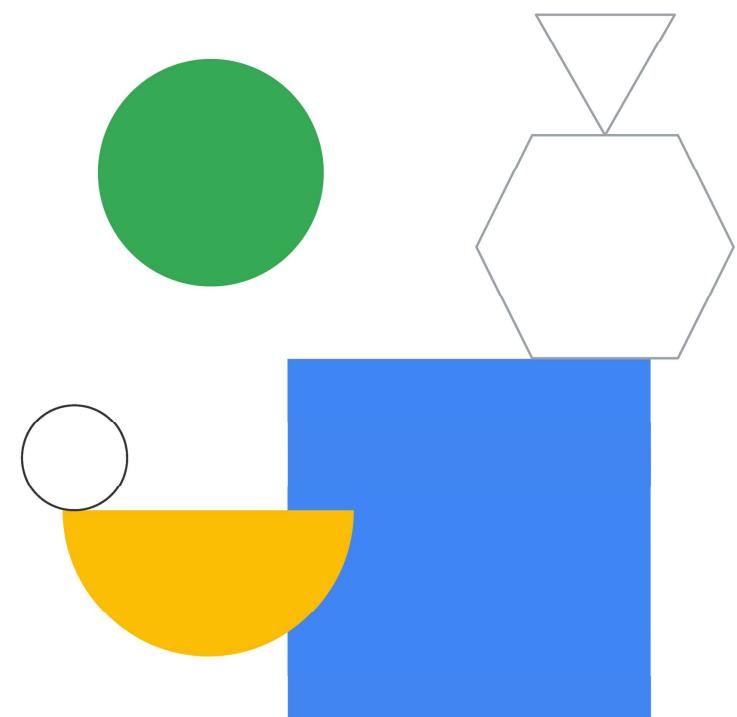


Video Intelligence API

- 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



# How Google Does Machine Learning



December 2021

# 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?

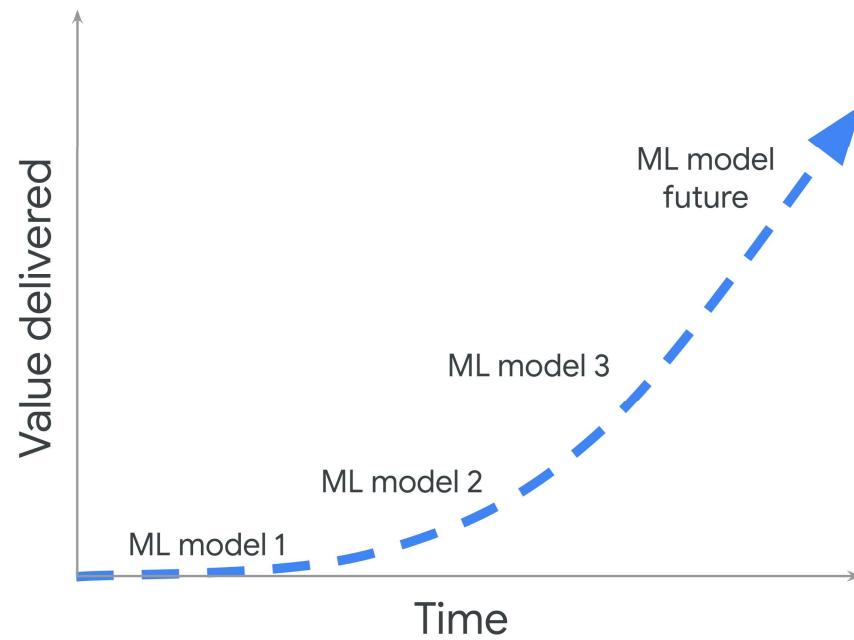
Most ML value  
comes along  
the way

ML improves  
almost everything  
it touches

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

ML is a great  
differentiator

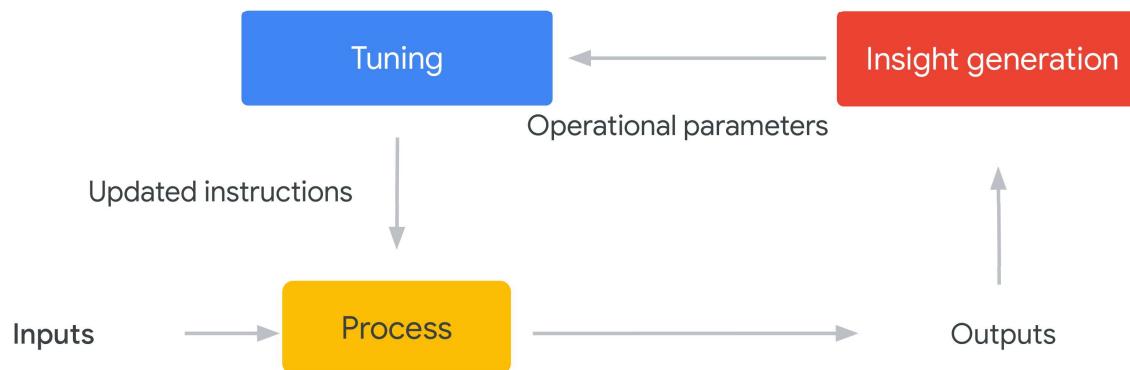
**Value comes  
along the way**



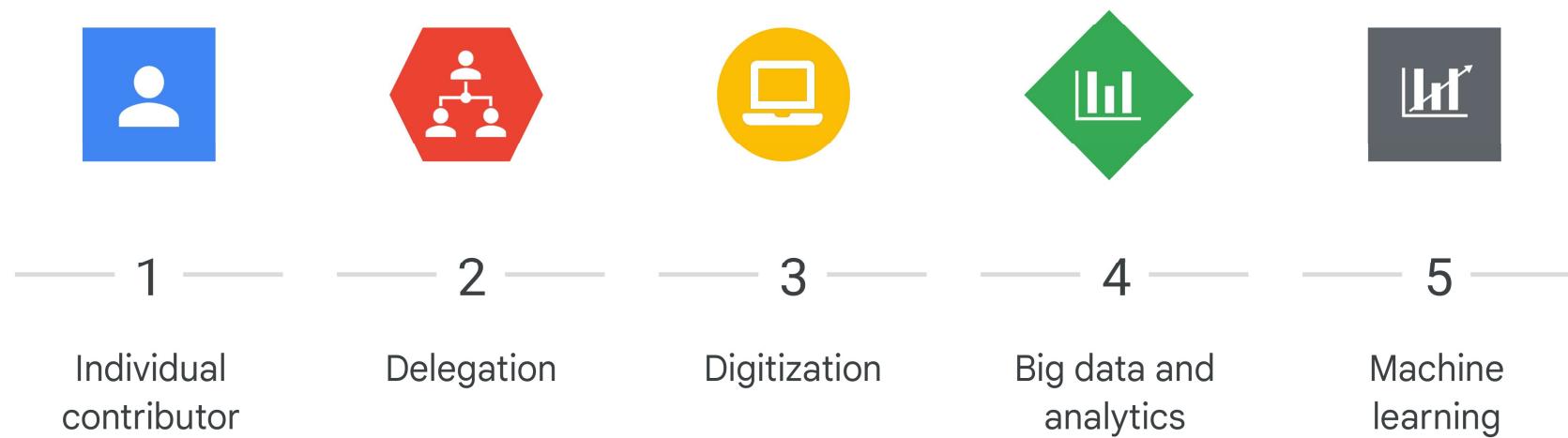
# Evolution of a business process



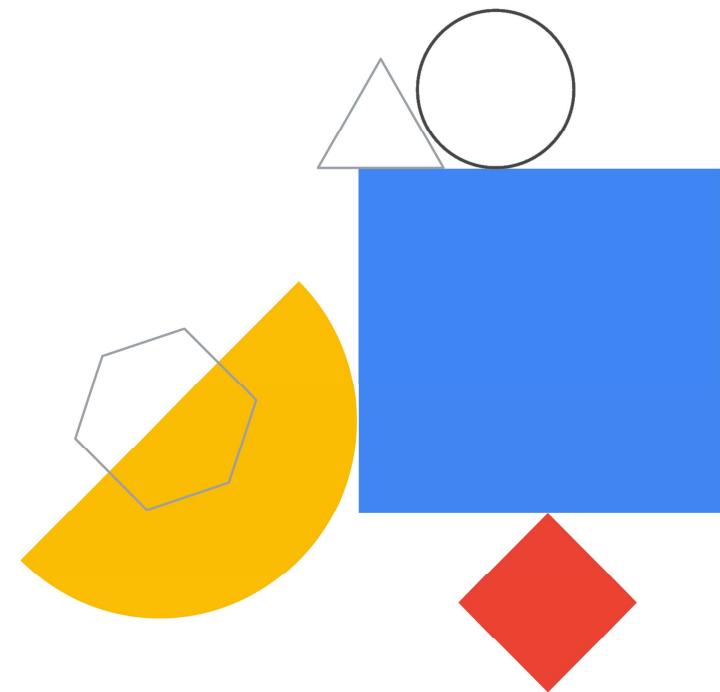
# Path to ML: Your turn



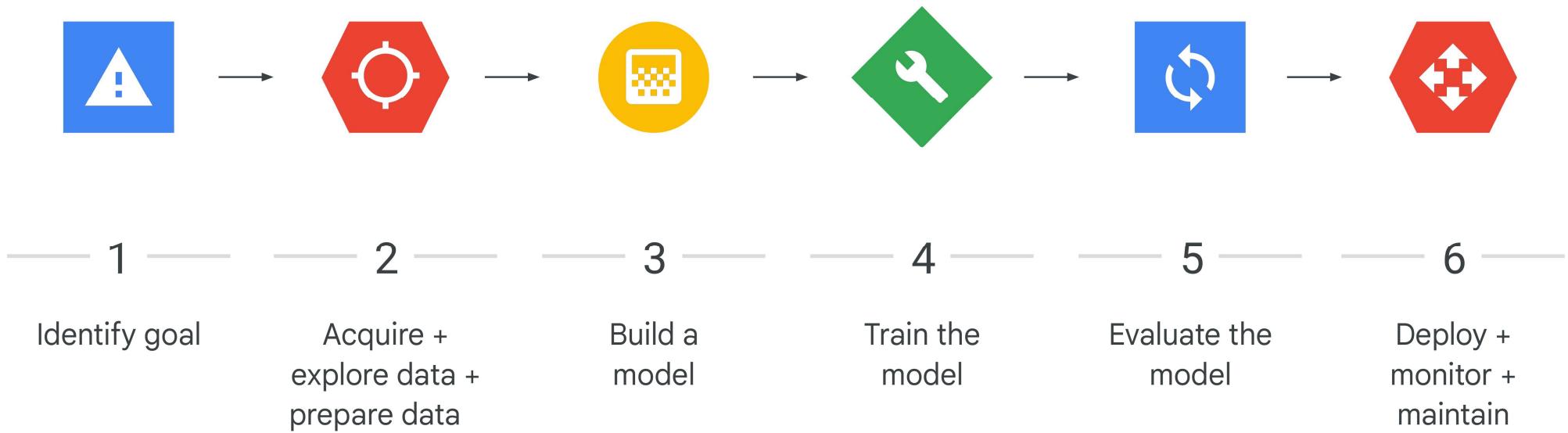
# The path to ML



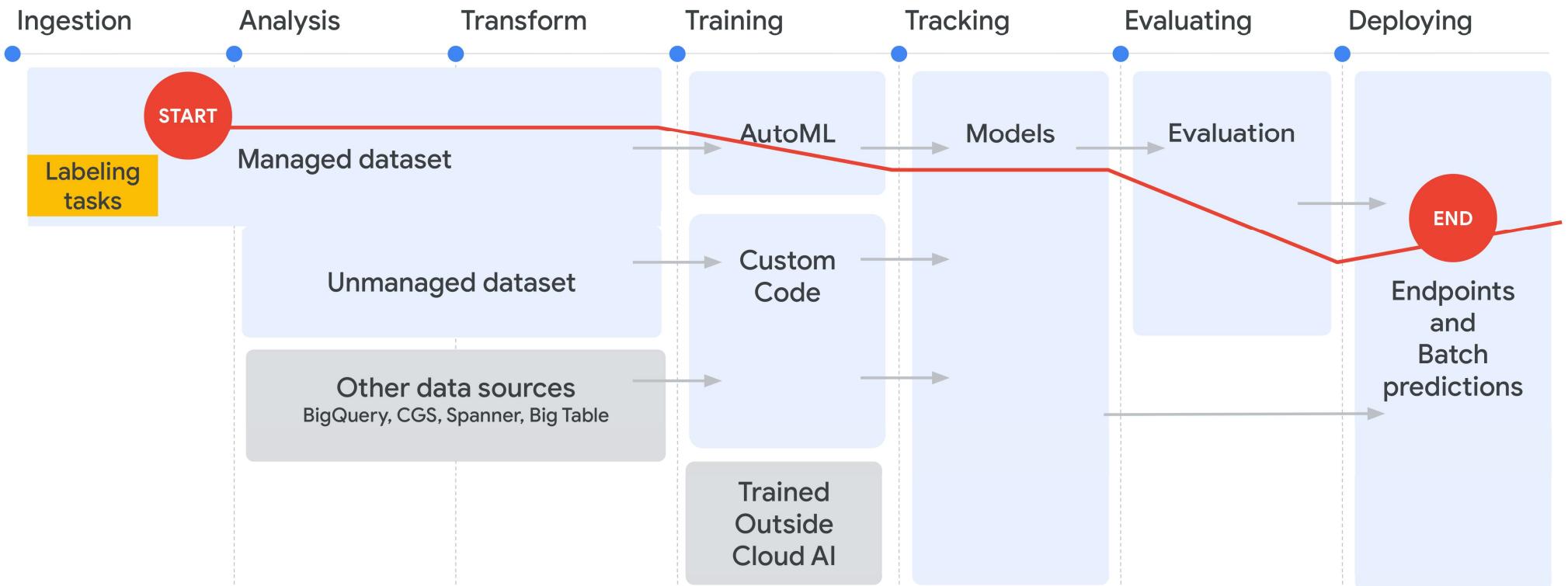
# Machine Learning Development with Vertex AI



# To build a machine learning model for production



# Vertex AI



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

# Choose a training method

## AutoML

---

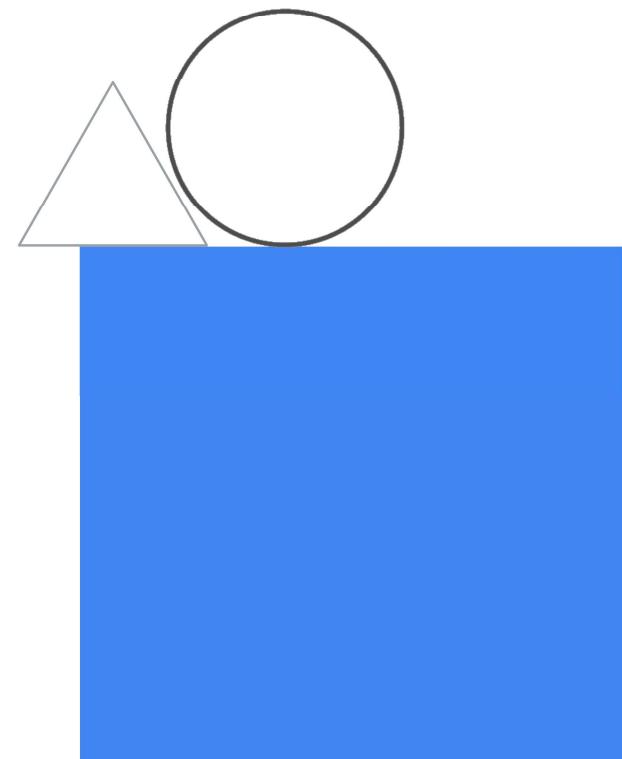
- Create and train a model with minimal technical effort.
- Quickly prototype models or explore datasets before developing in a custom training application.

## 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 Components



# Vertex AI > Dashboard

Vertex AI

- Dashboard
- Datasets
- Features
- Labeling tasks
- Workbench
- Pipelines
- Training
- Experiments
- Models
- Endpoints
- Batch predictions
- Marketplace

## Dashboard

**Get started with Vertex AI**

Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more](#)

**ENABLE VERTEX AI API**

Region: us-central1 (Iowa)

**Prepare your training data**

Collect and prepare your data, then import it into a dataset to train a model

**+ CREATE DATASET**

**Train your model**

Train a best-in-class machine learning model with your dataset. Use Google's AutoML, or bring your own code.

**+ TRAIN NEW MODEL**

**Get predictions**

After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests

**+ CREATE BATCH PREDICTION**

# Vertex AI > Datasets

Any dataset loaded into Vertex AI becomes “managed” and “available” to other components.

**Get started with Vertex AI**

Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more](#)

[ENABLE VERTEX AI API](#)

Region: us-central1 (Iowa)

**Prepare your training data**  
Collect and prepare your data, then import it into a dataset to train a model  
[+ CREATE DATASET](#)

**Train your model**  
Train a best-in-class machine learning model with your dataset. Use Google's AutoML, or bring your own code.  
[+ TRAIN NEW MODEL](#)

**Get predictions**  
After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests  
[+ CREATE BATCH PREDICTION](#)

The dashboard features a central blue callout box stating: "Any dataset loaded into Vertex AI becomes ‘managed’ and ‘available’ to other components." Below this, there's a section titled "Get started with Vertex AI" containing a brief description and a "ENABLE VERTEX AI API" button. The sidebar on the left is titled "Vertex AI" and includes links for Dashboard, Datasets (which is highlighted with a red box), Features, Labeling tasks, Workbench, Pipelines, Training, Experiments, Models, Endpoints, Batch predictions, and Marketplace. The main content area has a "Region" dropdown set to "us-central1 (Iowa)". It also contains three main cards: "Prepare your training data" (with a "CREATE DATASET" button), "Train your model" (with a "TRAIN NEW MODEL" button), and "Get predictions" (with a "CREATE BATCH PREDICTION" button). A decorative graphic at the bottom right shows a lightbulb, a neural network, a graph, and satellite dishes, symbolizing the flow from idea to deployed machine learning models.

# Vertex AI > Features

The screenshot shows the Vertex AI web interface. On the left is a sidebar with navigation links: Vertex AI, Dashboard, Datasets, **Features** (which is highlighted with a red box), Labeling tasks, Workbench, Pipelines, Training, Experiments, Models, Endpoints, Batch predictions, and Marketplace. A blue callout box points from the 'Features' link to the main content area.

**Dashboard**

Feature Store is a repository where you can ingest, serve, and share ML feature values. Feature Store manages all of the underlying infrastructure for you.

scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more](#)

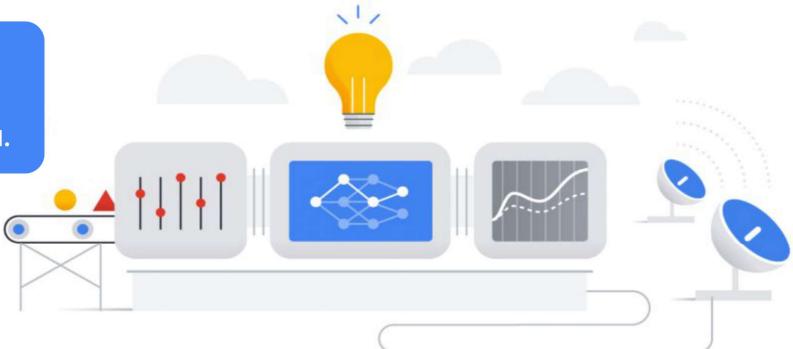
**ENABLE VERTEX AI API**

Region: us-central1 (Iowa)

**Prepare your training data**  
Collect and prepare your data, then import it into a dataset to train a model  
**+ CREATE DATASET**

**Train your model**  
Train a best-in-class machine learning model with your dataset. Use Google's AutoML, or bring your own code.  
**+ TRAIN NEW MODEL**

**Get predictions**  
After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests  
**+ CREATE BATCH PREDICTION**



# Vertex AI > Labeling tasks

The screenshot shows the Vertex AI dashboard. On the left, a sidebar menu lists various options: Dashboard, Datasets, Features, **Labeling tasks** (which is highlighted with a red box), Workbench, Pipelines, Training, Experiments, Models, Endpoints, Batch predictions, and Marketplace. A blue callout box over the 'Labeling tasks' item contains the text: "Data labeling jobs let you request human labeling for a dataset that you plan to use to train a custom machine learning model." Below this is a "ENABLE VERTEX AI API" button. The main dashboard area has a "Region" dropdown set to "us-central1 (Iowa)". It features three main cards: "Prepare your training data" (with a "+ CREATE DATASET" button), "Train your model" (with a "+ TRAIN NEW MODEL" button), and "Get predictions" (with a "+ CREATE BATCH PREDICTION" button). To the right of these cards is a decorative graphic depicting a lightbulb, clouds, and satellite dish antennas.

# Vertex AI > Workbench

The screenshot shows the Vertex AI Workbench dashboard. On the left, a sidebar menu lists various options: Dashboard, Datasets, Features, Labeling tasks, **Workbench** (which is highlighted with a red box), Pipelines, Training, Experiments, Models, Endpoints, Batch predictions, and Marketplace. The main area is titled "Dashboard" and features a central illustration of a machine learning pipeline: data flows from a table through a neural network, a chart, and a satellite dish into a lightbulb. A blue callout box over this illustration states: "Workbench provides a Jupyter notebook development environment for the entire ML workflow. Access data, process data in a Dataproc cluster, train a model, and share results." Below the illustration, there is a "Region" dropdown set to "us-central1 (Iowa)". The dashboard is divided into three main sections: "Prepare your training data", "Train your model", and "Get predictions".

Region  
us-central1 (Iowa)

Workbench provides a Jupyter notebook development environment for the entire ML workflow. Access data, process data in a Dataproc cluster, train a model, and share results.

Region  
us-central1 (Iowa)

**Prepare your training data**  
Collect and prepare your data, then import it into a dataset to train a model  
+ CREATE DATASET

**Train your model**  
Train a best-in-class machine learning model with your dataset. Use Google's AutoML, or bring your own code.  
+ TRAIN NEW MODEL

**Get predictions**  
After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests  
+ CREATE BATCH PREDICTION

# Vertex AI > Pipelines

The screenshot shows the Vertex AI Pipelines dashboard. On the left, a sidebar menu lists various options: Dashboard, Datasets, Features, Labeling tasks, Workbench, Pipelines (which is highlighted with a red box), Training, Experiments, Models, Endpoints, Batch predictions, and Marketplace. The main area is titled "Dashboard" and features a "Get started with Vertex AI" section. This section includes a sub-section about Pipelines: "Pipelines help you to automate, monitor, and govern your ML systems. Each individual part of your pipeline workflow is a component that is defined by code." Below this, there's a "Region" dropdown set to "us-central1 (Iowa)". The dashboard is divided into three main sections: "Prepare your training data", "Train your model", and "Get predictions". Each section contains a brief description and a "CREATE" button.

**Pipelines** help you to automate, monitor, and govern your ML systems. Each individual part of your pipeline workflow is a component that is defined by code.

Region  
us-central1 (Iowa)

**Prepare your training data**  
Collect and prepare your data, then import it into a dataset to train a model  
+ CREATE DATASET

**Train your model**  
Train a best-in-class machine learning model with your dataset. Use Google's AutoML, or bring your own code.  
+ TRAIN NEW MODEL

**Get predictions**  
After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests  
+ CREATE BATCH PREDICTION

# Vertex AI > Training

The screenshot shows the Vertex AI Training dashboard. On the left, a sidebar menu lists various options: Dashboard, Datasets, Features, Labeling tasks, Workbench, Pipelines, **Training** (which is highlighted with a red box and has a blue callout), Experiments, Models, Endpoints, Batch predictions, and Marketplace. The main content area is titled "Dashboard" and features a "Get started with Vertex AI" section. This section includes a brief description of Vertex AI's capabilities and a callout stating: "You can train models on Vertex AI using AutoML, or use custom training if you need the wider range of customization options available in AI Platform Training." Below this are three cards: "Prepare your training data" (with a "CREATE DATASET" button), "Train your model" (with a "TRAIN NEW MODEL" button), and "Get predictions" (with a "CREATE BATCH PREDICTION" button). A large, stylized graphic on the right depicts a lightbulb above a neural network, with clouds and satellite dish icons, symbolizing AI and machine learning.

Vertex AI

Dashboard

Datasets

Features

Labeling tasks

Workbench

Pipelines

**Training**

Experiments

Models

Endpoints

Batch predictions

Marketplace

Region  
us-central1 (Iowa)

Get started with Vertex AI

You can train models on Vertex AI using AutoML, or use custom training if you need the wider range of customization options available in AI Platform Training.

Prepare your training data

Collect and prepare your data, then import it into a dataset to train a model

+ CREATE DATASET

Train your model

Train a best-in-class machine learning model with your dataset. Use Google's AutoML, or bring your own code.

+ TRAIN NEW MODEL

Get predictions

After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests

+ CREATE BATCH PREDICTION

# Vertex AI > Experiments

The screenshot shows the Vertex AI interface with the 'Experiments' section highlighted.

**Left Sidebar:**

- Vertex AI
- Dashboard
- Datasets
- Features
- Labeling tasks
- Workbench
- Pipelines
- Training
- Experiments** (highlighted with a red box)
- Models
- Endpoints
- Batch predictions
- Marketplace

**Dashboard Content:**

**Get started with Vertex AI**  
Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more](#)

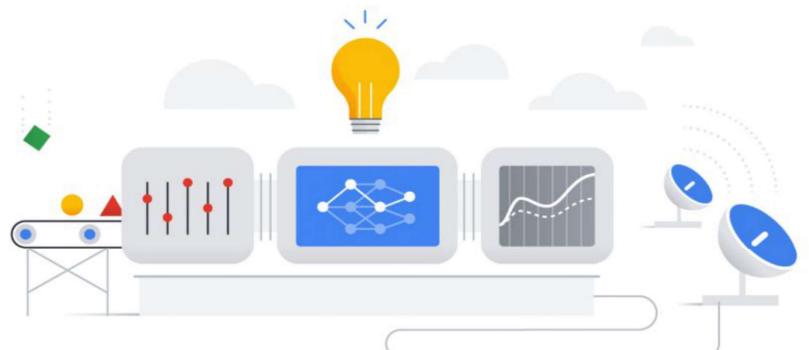
**Experiments includes studies, hyperparameter tuning, and TensorBoard.**

**us-central1 (Iowa)**

**Prepare your training data**  
Collect and prepare your data, then import it into a dataset to train a model  
[+ CREATE DATASET](#)

**Train your model**  
Train a best-in-class machine learning model with your dataset. Use Google's AutoML, or bring your own code.  
[+ TRAIN NEW MODEL](#)

**Get predictions**  
After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests  
[+ CREATE BATCH PREDICTION](#)



# Vertex AI > Models

Vertex AI

- Dashboard
- Datasets
- Features
- Labeling tasks
- Workbench
- Pipelines
- Training
- Experiments
- Models**
- Endpoints
- Batch predictions
- Marketplace

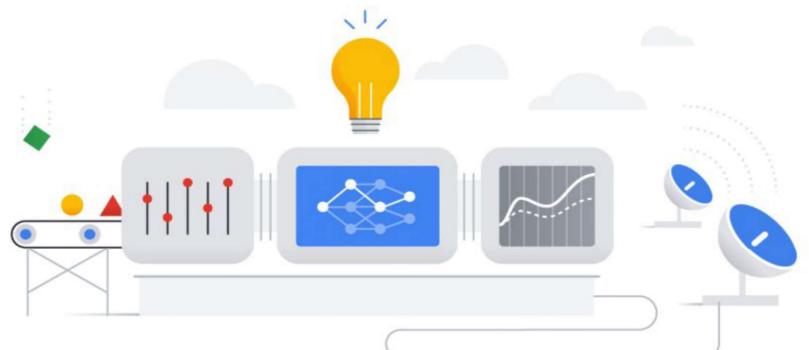
## Dashboard

**Get started with Vertex AI**

Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more](#)

**ENABLE VERTEX AI API**

Models are built from your datasets or unmanaged data sources.



- Prepare your training data**

Collect and prepare your data, then import it into a dataset to train a model

**+ CREATE DATASET**
- Train your model**

Train a best-in-class machine learning model with your dataset. Use Google's AutoML, or bring your own code.

**+ TRAIN NEW MODEL**
- Get predictions**

After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests

**+ CREATE BATCH PREDICTION**

# Vertex AI > Endpoints

Vertex AI

- Dashboard
- Datasets
- Features
- Labeling tasks
- Workbench
- Pipelines
- Training
- Experiments
- Models
- Endpoints**
- Batch predictions
- Marketplace

## Dashboard

**Get started with Vertex AI**

Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more](#)

**ENABLE VERTEX AI API**

You can deploy models for prediction on Vertex AI and get an endpoint to serve predictions on Vertex AI.

Collect and prepare your data, then import it into a dataset to train a model

**+ CREATE DATASET**

**Train your model**

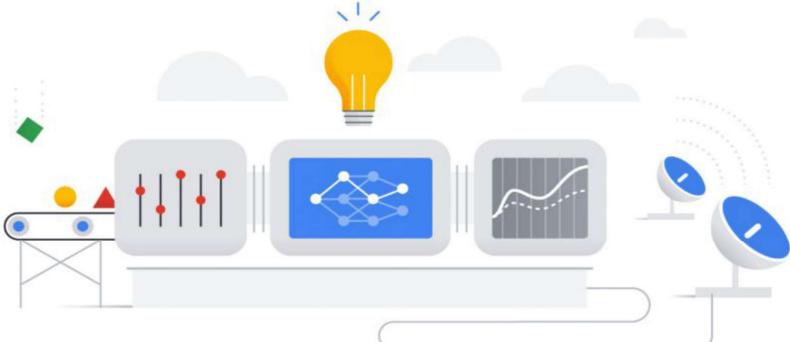
Train a best-in-class machine learning model with your dataset. Use Google's AutoML, or bring your own code.

**+ TRAIN NEW MODEL**

**Get predictions**

After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests

**+ CREATE BATCH PREDICTION**



# Vertex AI > Batch predictions

Vertex AI

- Dashboard
- Datasets
- Features
- Labeling tasks
- Workbench
- Pipelines
- Training
- Experiments
- Models
- Endpoints
- Batch predictions**
- Marketplace

## Dashboard

**Get started with Vertex AI**

Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more](#)

**ENABLE VERTEX AI API**

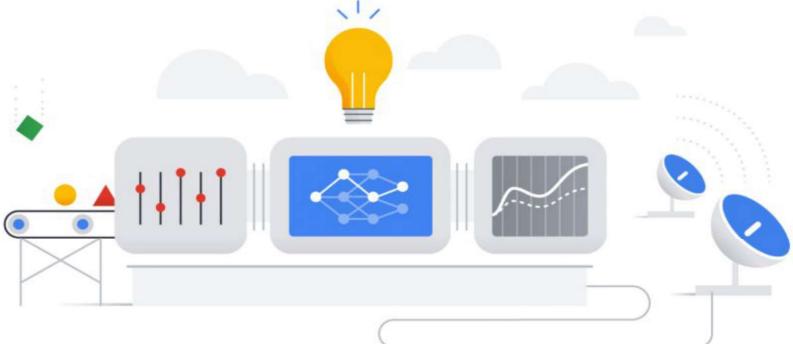
Region: us-central1 (Iowa)

Batch prediction intakes a group of prediction requests and outputs the results to a specified location.

+ CREATE DATASET

+ TRAIN NEW MODEL

+ CREATE BATCH PREDICTION



# Vertex AI > Metadata

The screenshot shows the Vertex AI Dashboard. The left sidebar menu includes:

- Dashboard
- Datasets
- Features
- Labeling tasks
- Workbench
- Pipelines
- Training
- Experiments
- Models
- Endpoints
- Batch predictions
- Metadata** (highlighted with a red box)

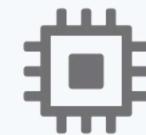
The main dashboard area features:

- Get started with Vertex AI**: A section with a brief description and a blue "ENABLE VERTEX AI API" button.
- Region**: Set to "us-central1 (Iowa)" with a dropdown arrow and a question mark icon.
- A central graphic illustrating the machine learning pipeline: a lightbulb above a neural network, which is connected to a line graph and two satellite dish icons.
- Vertex ML Metadata stores artifacts and metadata for pipelines run using Vertex AI Pipelines.** (Text in a blue callout box)
- Train your model**: Description and a blue "+ TRAIN NEW MODEL" button.
- Get predictions**: Description and a blue "+ CREATE BATCH PREDICTION" button.

# Tools to interact with Vertex AI



Client Libraries



VM Images



REST API



Containers

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

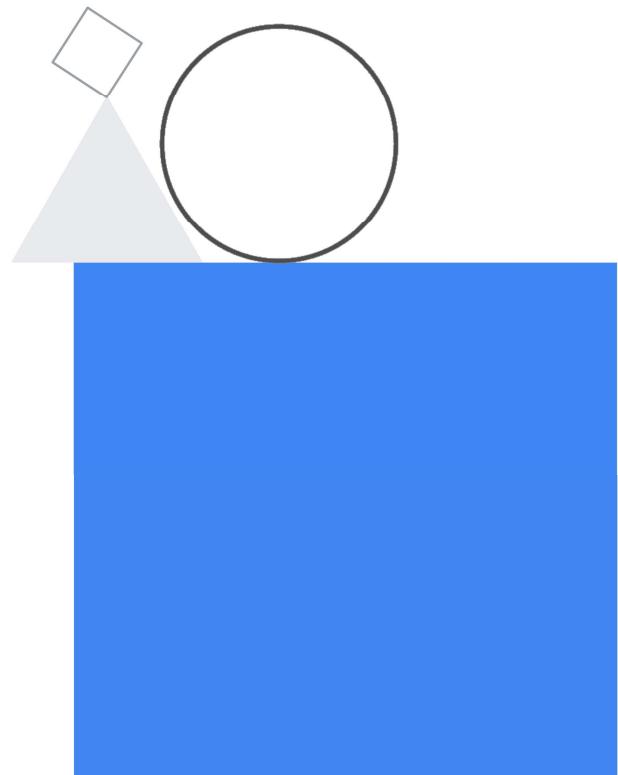
Managed notebooks

User-managed  
notebooks

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

[Deep Learning VM Images](#)  
customizable environment

# **Best Practices for Implementing Machine Learning on Vertex AI**

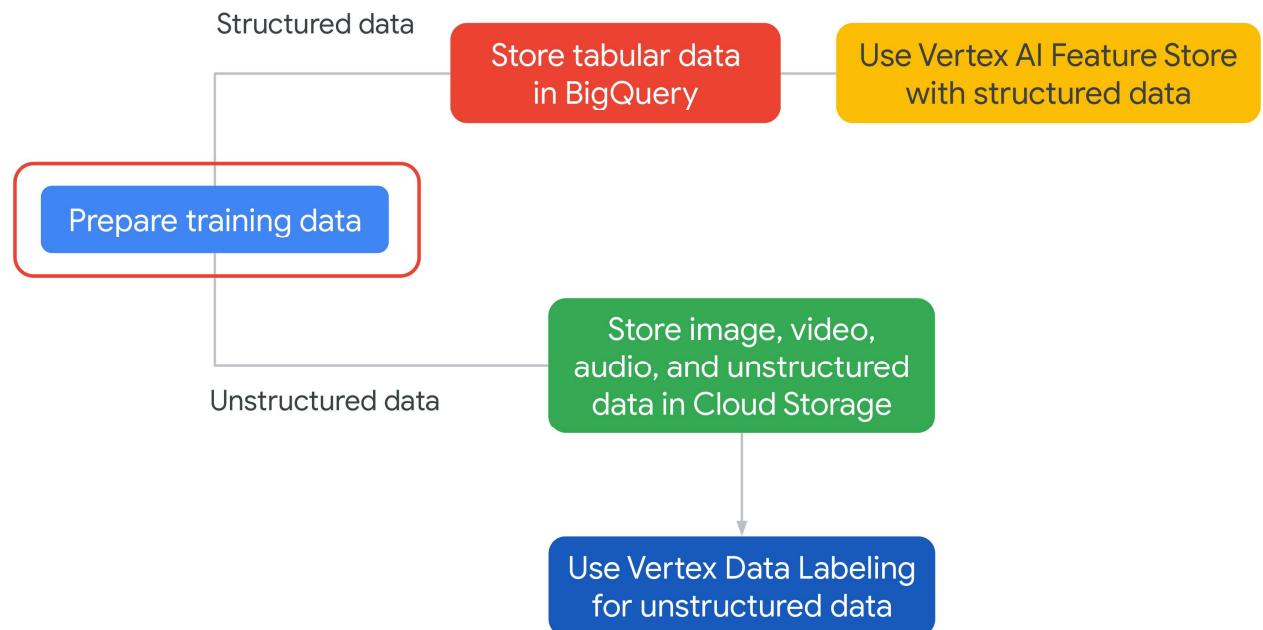


# Best practices for preparing and storing data

## Data

How it is prepared and stored

Avoid storing data in block storage



# Vertex AI Feature Store

## Feature Store

Use Feature Store with structured data

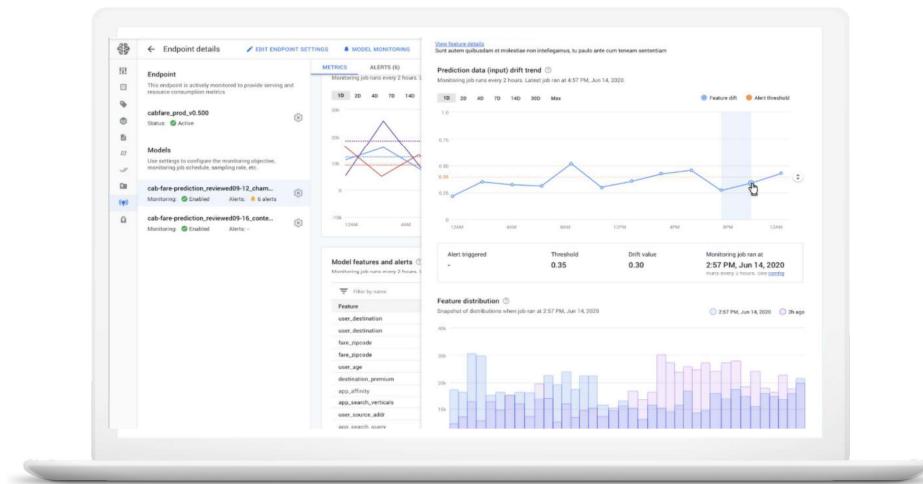
Follow these steps:

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

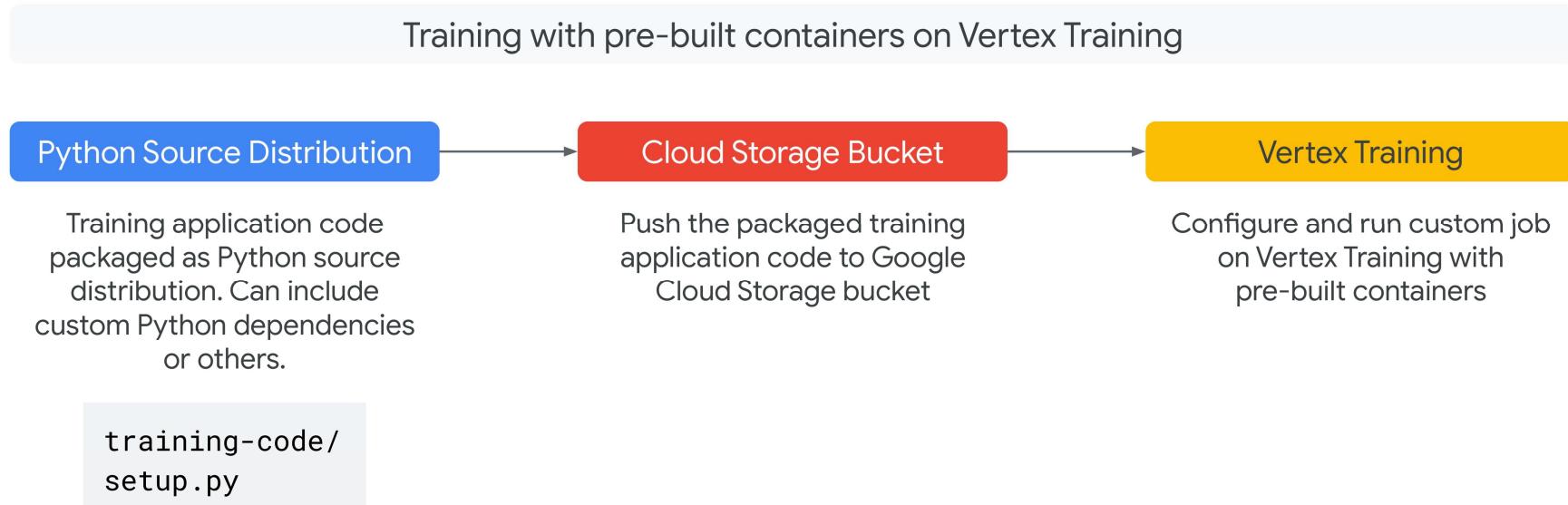
Tips for training, maximizing predictive accuracy, and feature attributions for insights



For small datasets, train a model within the [Notebooks instance](#).

For large datasets, distributed training, or scheduled training, use the [Vertex training service](#).

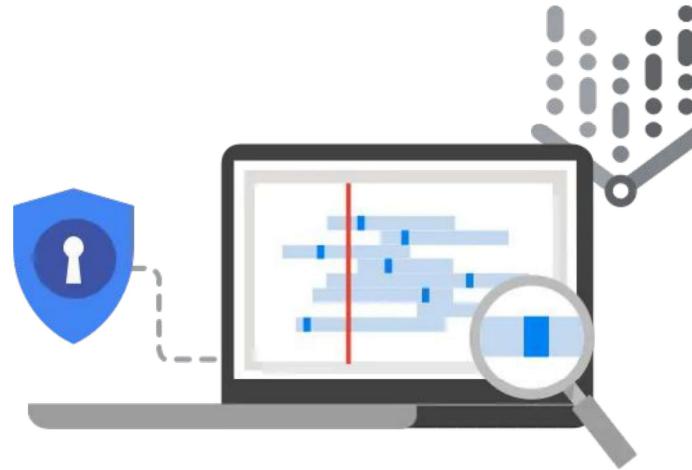
# Training with pre-built containers on Vertex AI



# 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.

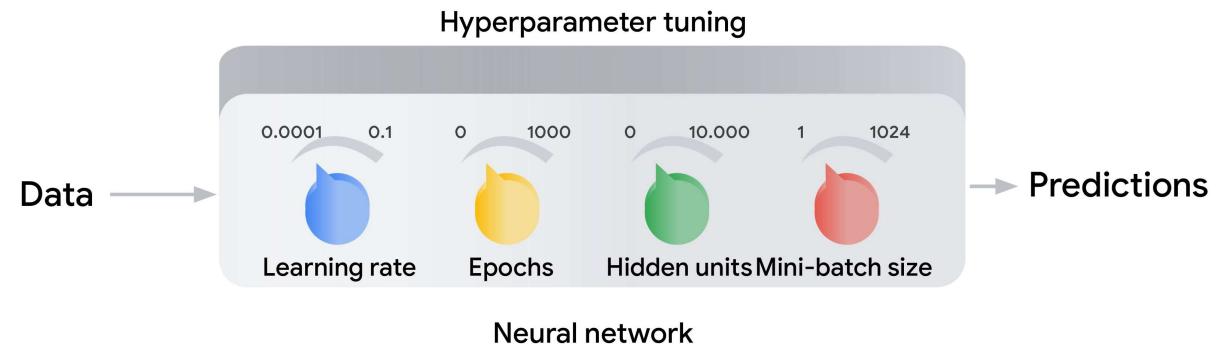
Details the importance of each feature that a model uses as input to make a prediction.

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.

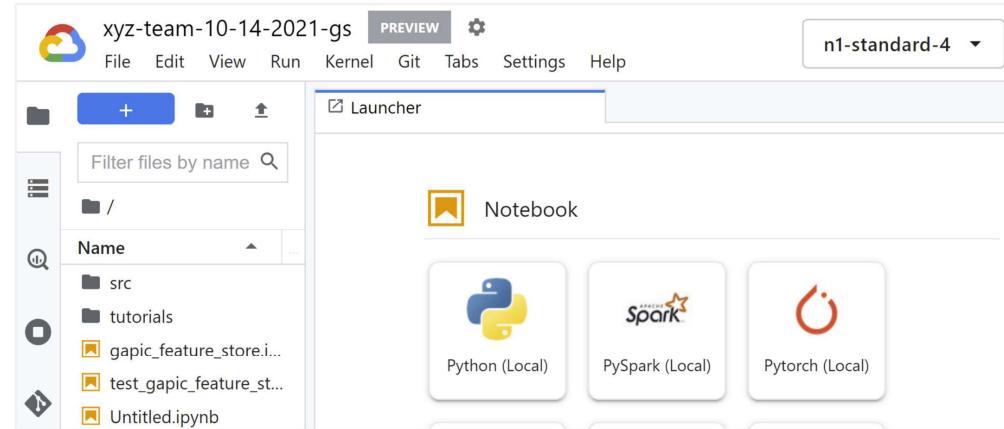
Maximize a model's predictive accuracy. [Vertex Training](#) provides an automated model enhancer to test different hyperparameter configurations when training your model.

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

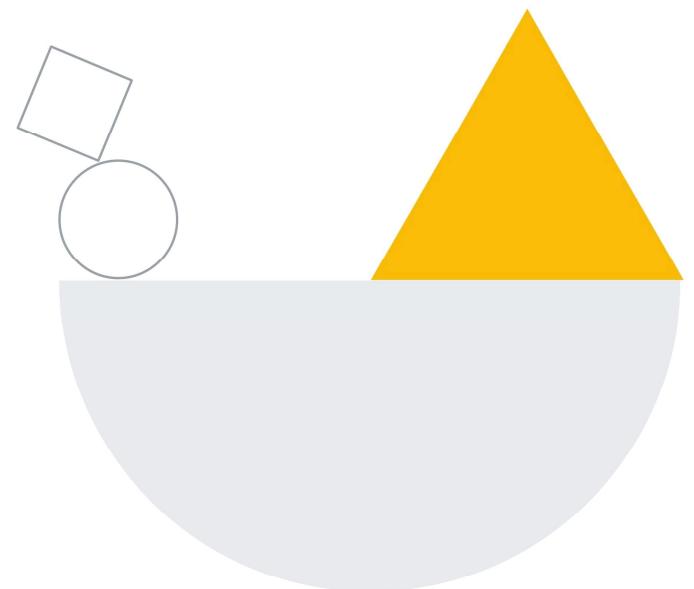
## Workbench Notebooks

Use Notebooks to evaluate and understand your models.

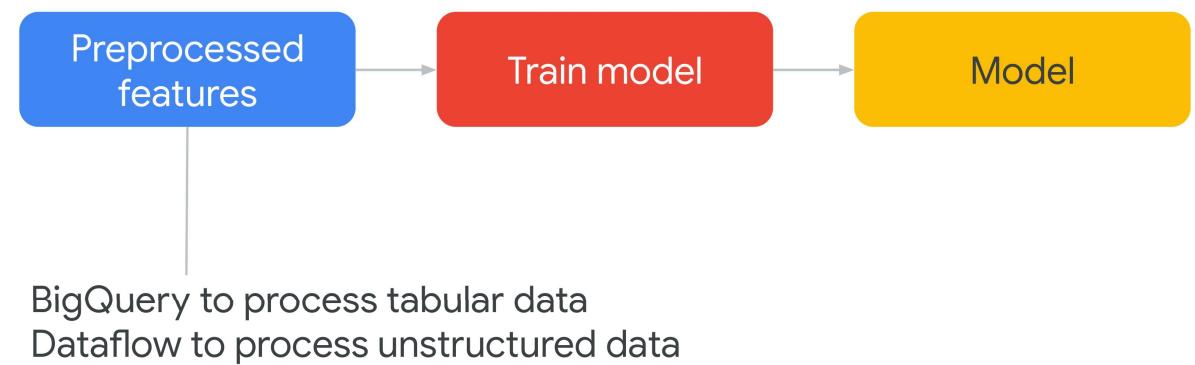


Use [Notebooks](#) to evaluate and understand your models.

# Data preprocessing best practices



For training and evaluation, we need preprocessed features



# 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.

# TensorFlow Extended

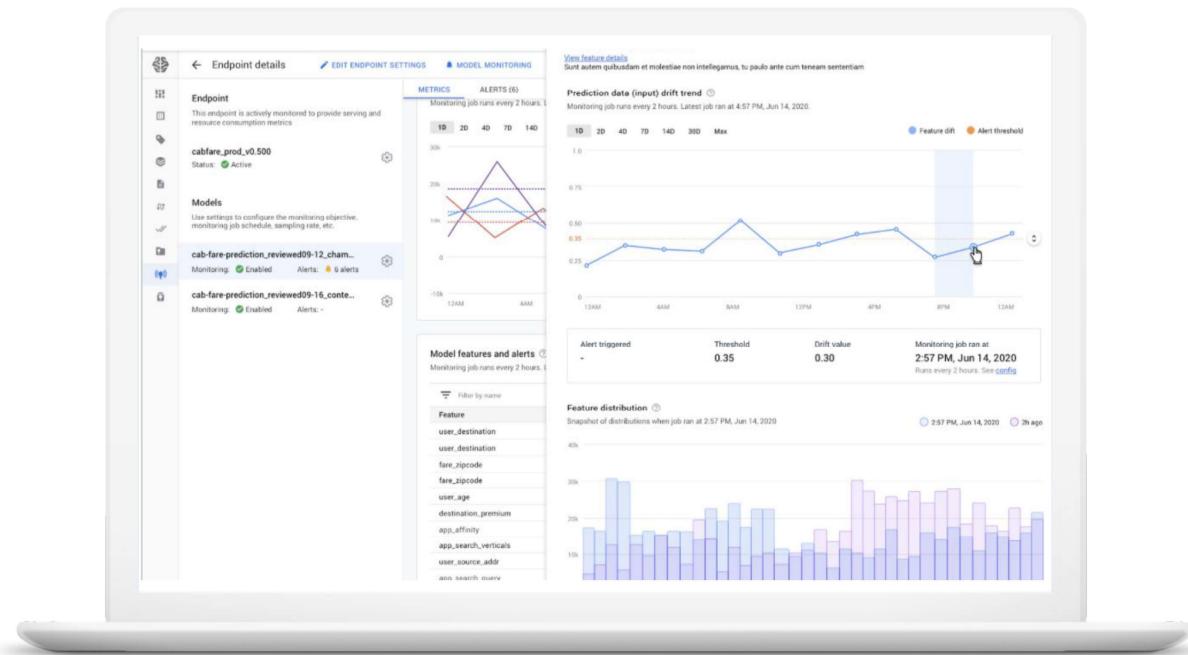
## TensorFlow Extended

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: ML environment setup



## Workbench Notebooks

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

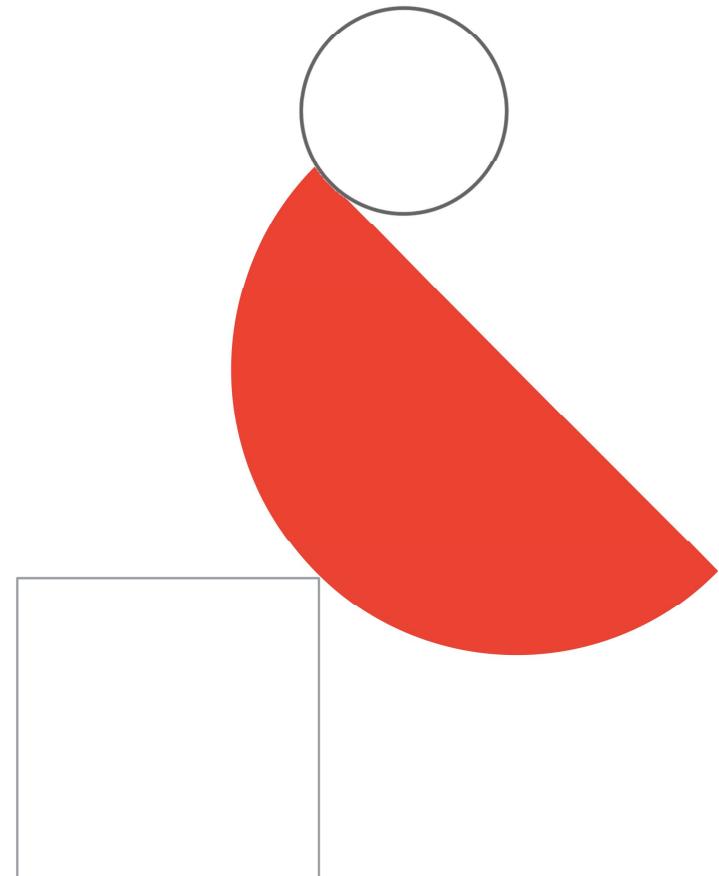
## Data & model

Store prepared data and model in same project.

## Optimize performance & cost

Optimize performance and cost.

# **Responsible AI Development**



# 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

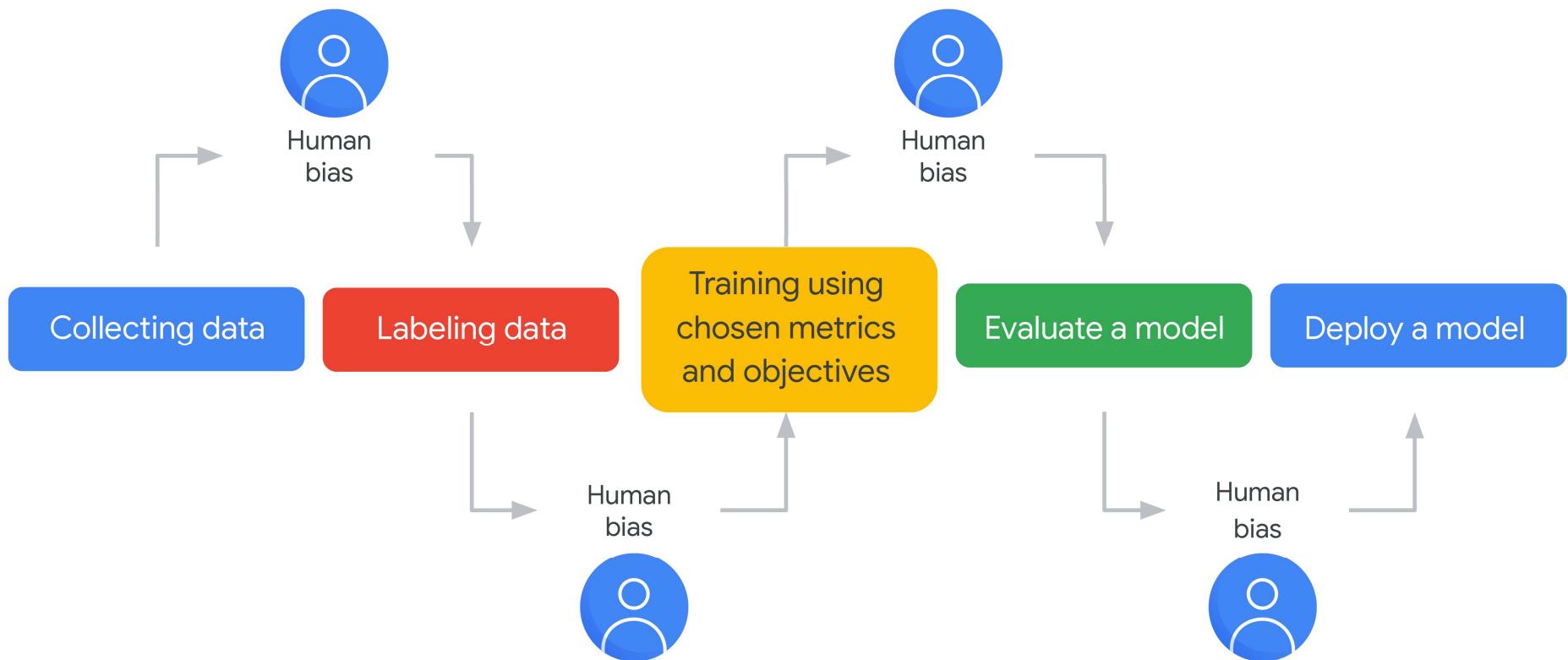
Unconscious bias from “the world” that we might reflect in ML when using existing data

Collecting data

Labeling data

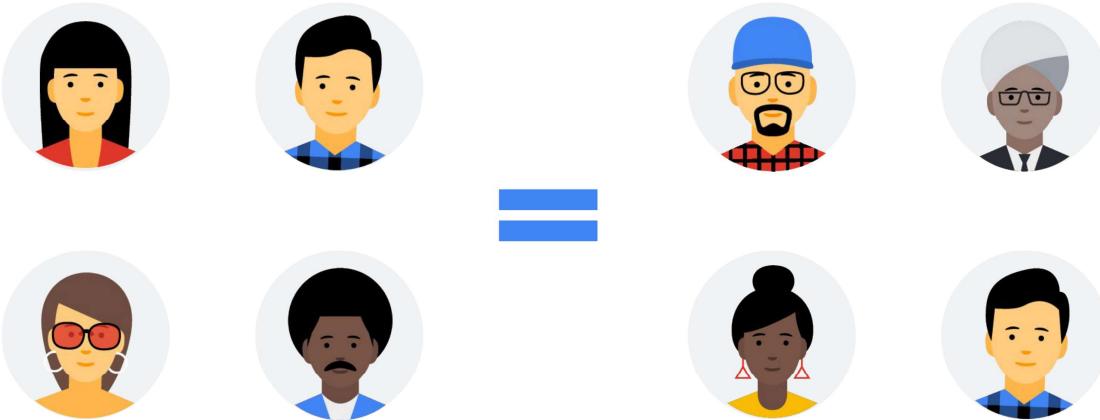
Unconscious bias in our procedures that we might reflect in our ML

# A typical ML pipeline with bias



# A checklist for bias-related issues

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



Understand the **confusion matrix**

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

		Model predictions	
		Positive	Negative
Labels	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.  <b>Model says: no</b>
	Negative	False positives (FP)  Type I error Label says something doesn't exist. Model predicts it.  <b>Model says: yes</b>	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	
		Positive	Negative
Labels	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	False negative rate	$\frac{\text{False negatives}}{\text{False negatives} + \text{True positives}}$

# 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  =  $\frac{\text{False positives}}{\text{False positives} + \text{True negatives}}$
	Negative	Type I error Label says something doesn't exist. Model predicts it.	

# 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?

---

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**.

Thanks