How Google Does Machine Learning

What it Means to be AI-first

How Google Does ML

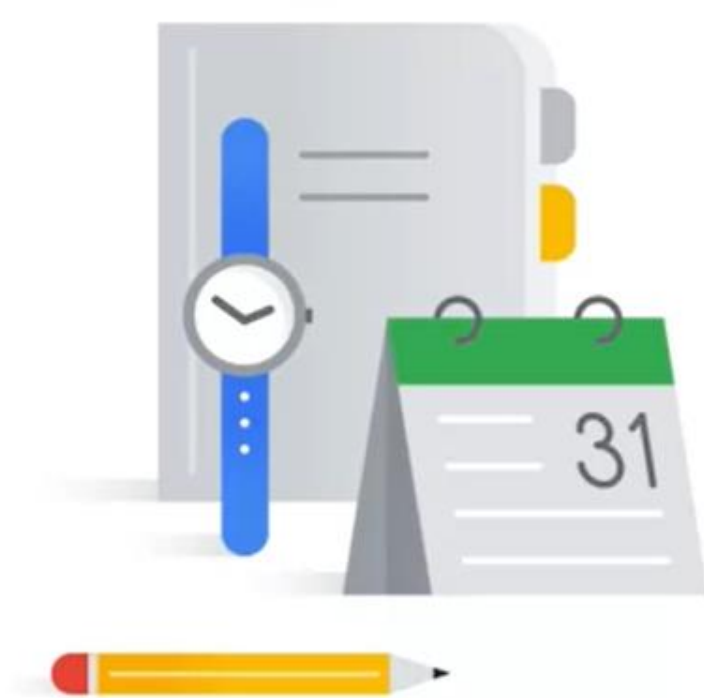
Introduction to Machine Learning Development with Vertex AI

ML Development with Vertex Notebooks

Best Practices for Implementing Machine Learning on Vertex AI

Responsible AI Development

Course Summary



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

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

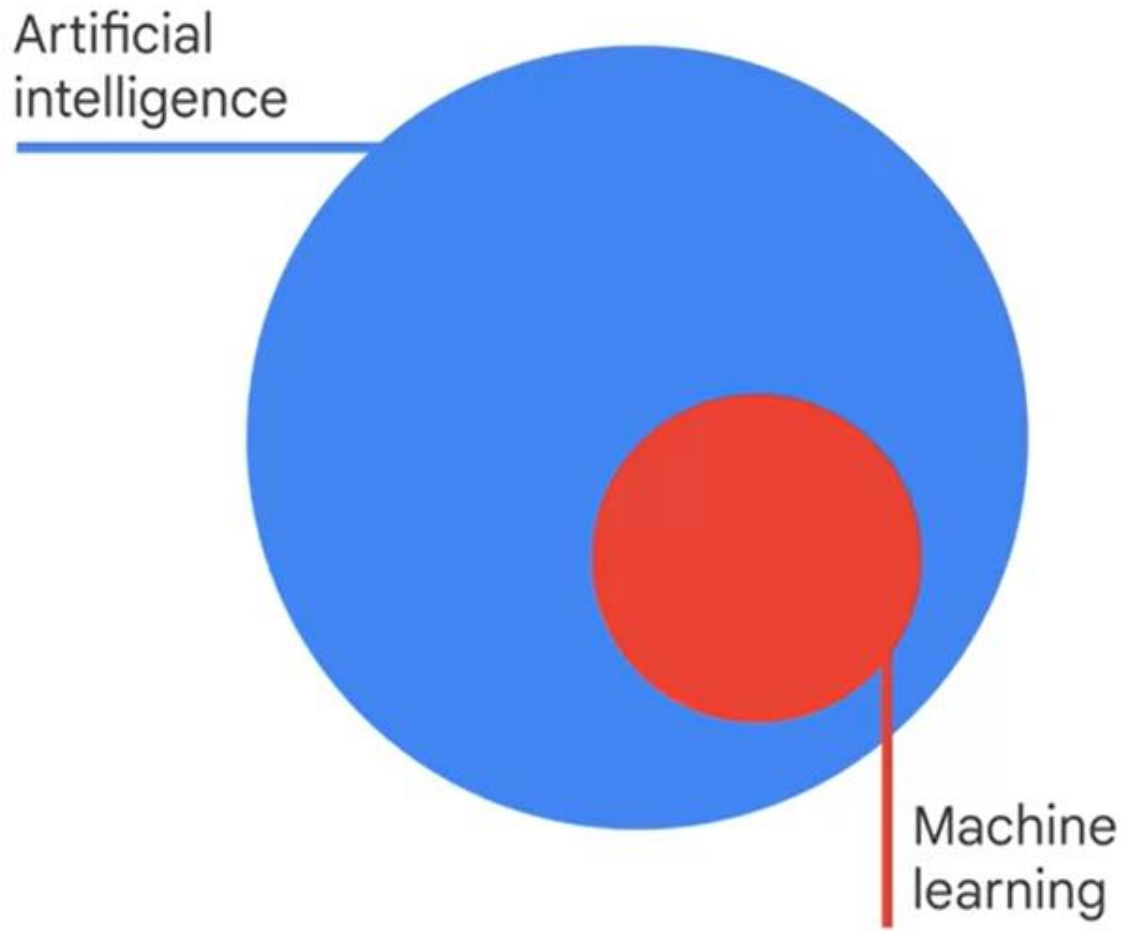
Data

Algorithm

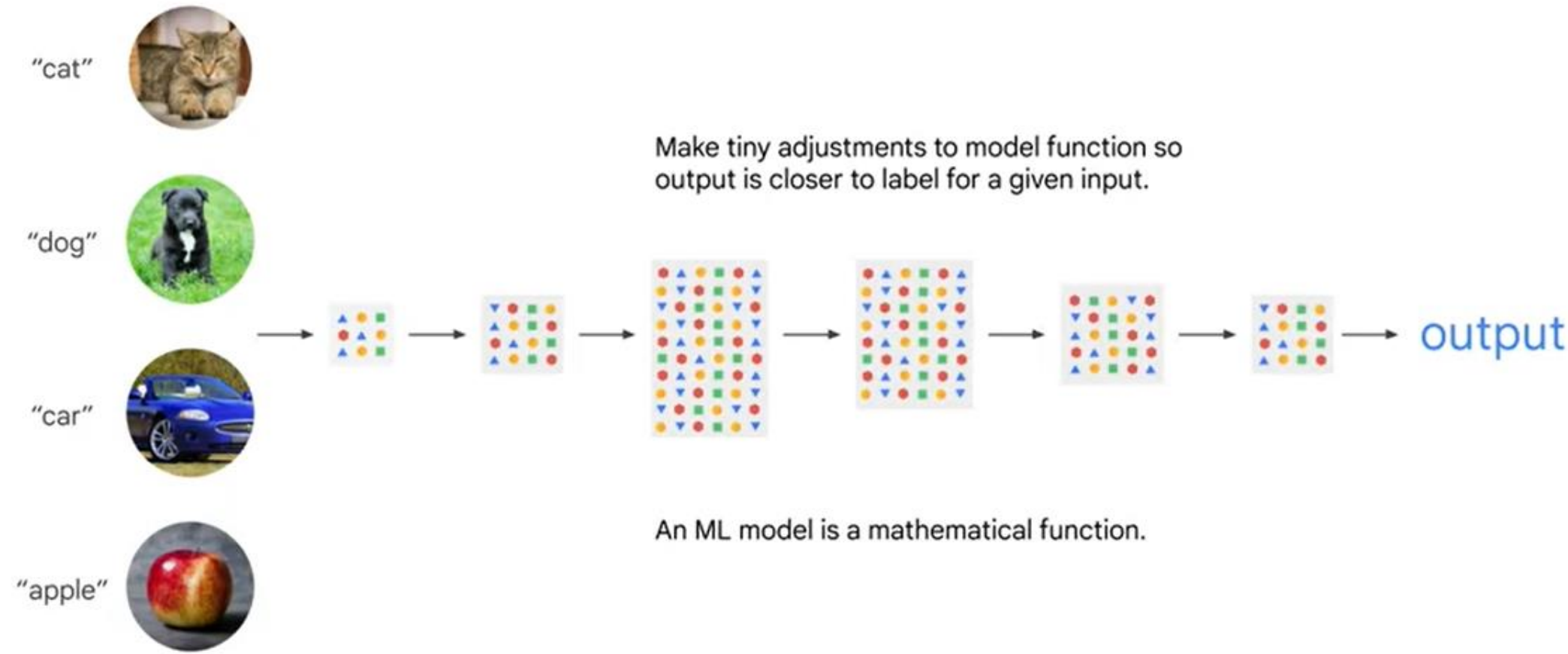
Predictive insight

Decision

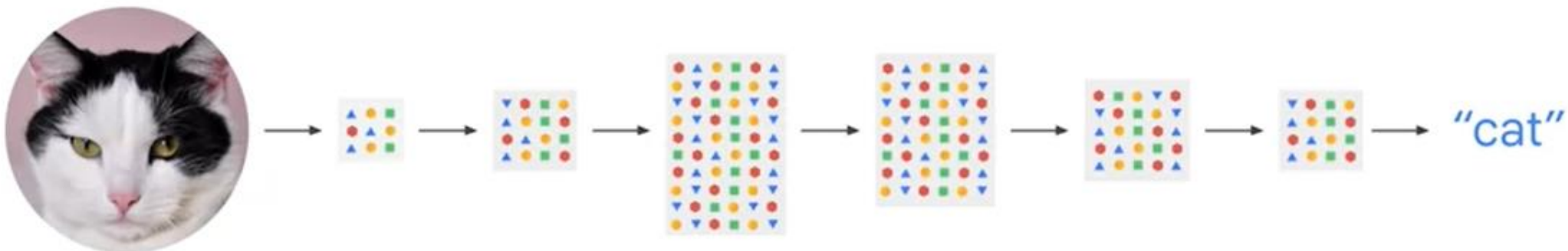
Artificial Intelligence is a discipline; machine learning is a specific way of solving AI problems.



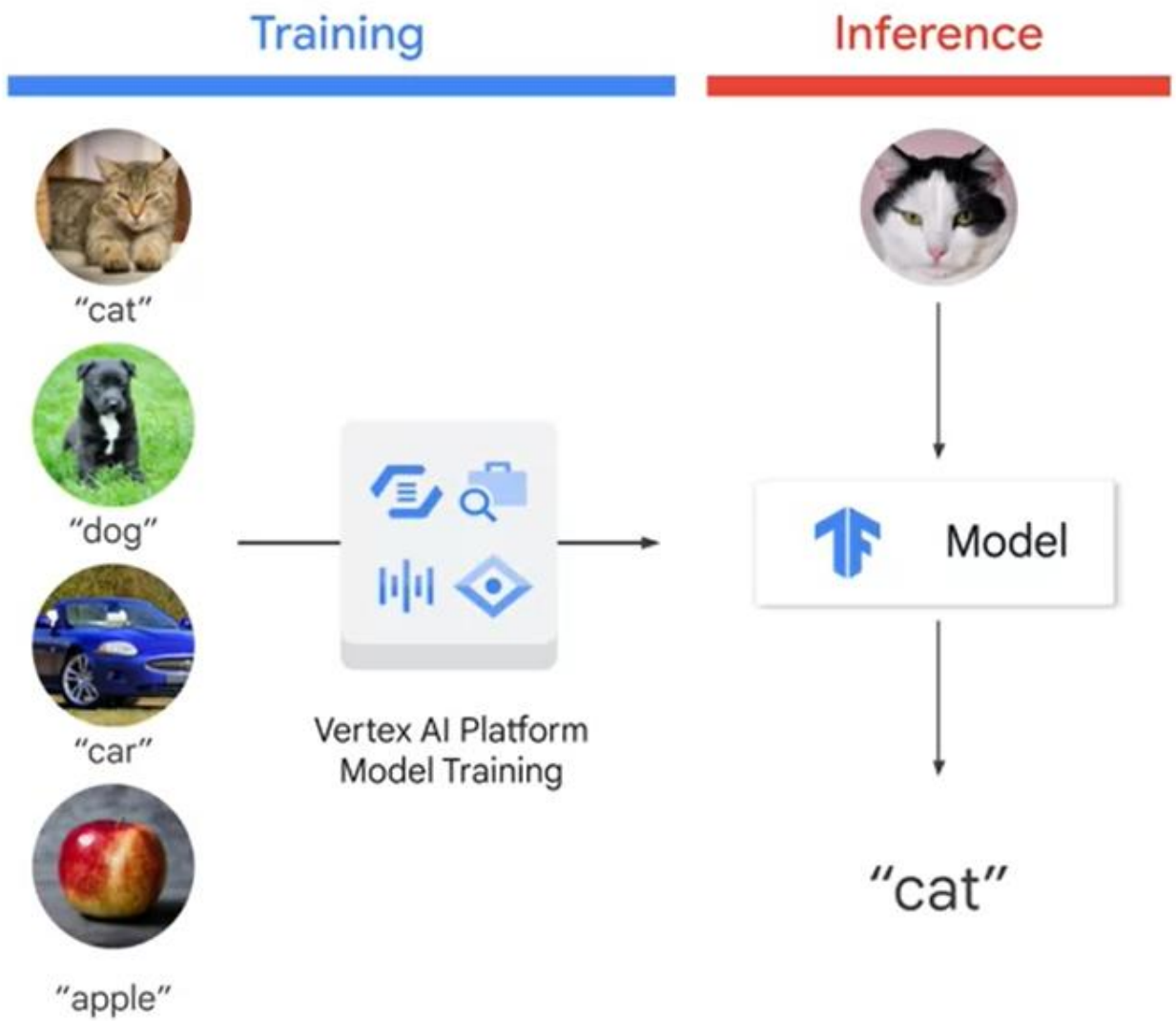
Stage 1: Train an ML model with examples



Stage 2: Predict with a trained model

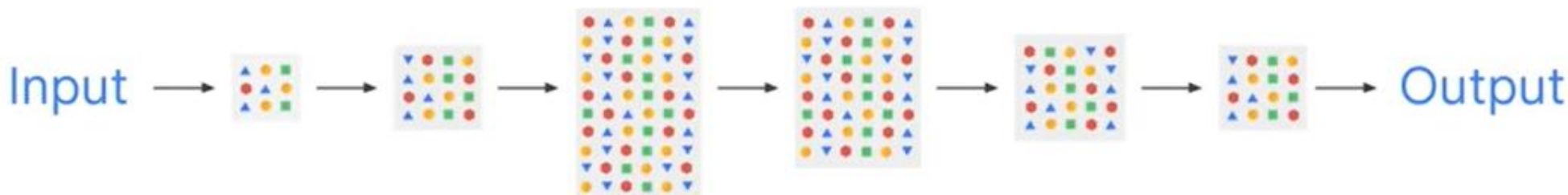


unlabeled photo



Focus on both the training and inference stages of ML

Neural networks are one important technology we use



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

Google Photos

illustrates how far ML has come.

Google Translate

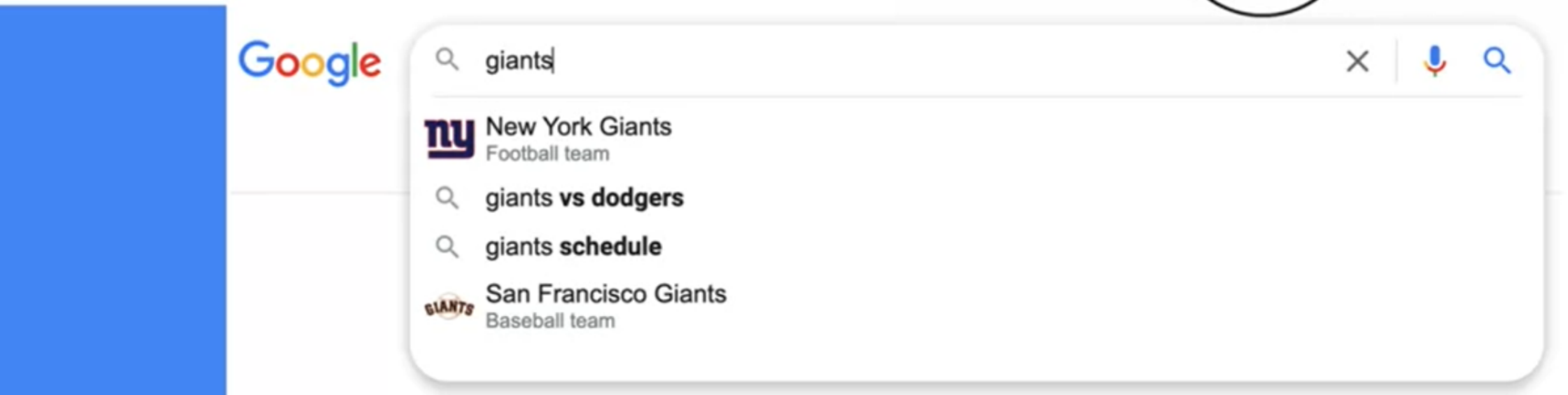
is a combination of several models.

Gmail

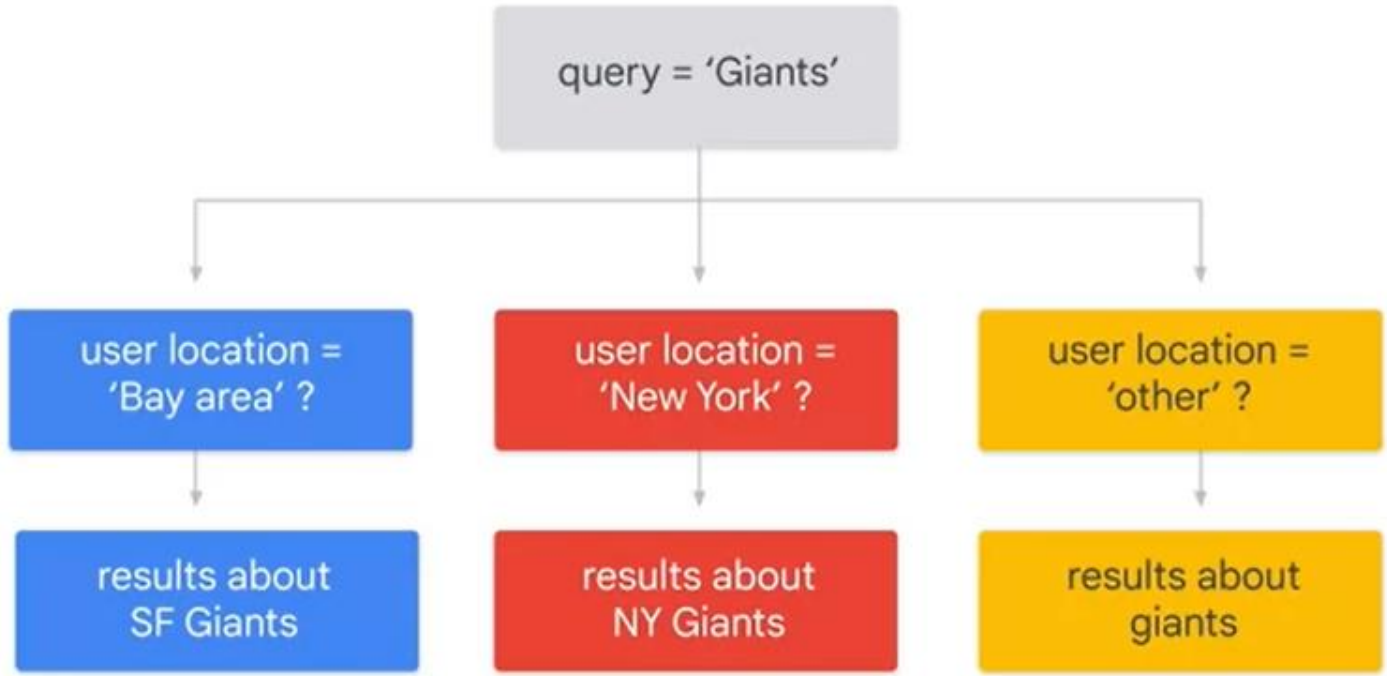
Smart Reply Inbox
20% of all responses sent on mobile.

Google Search,

our flagship application



Machine learning scales better than hand-coded rules



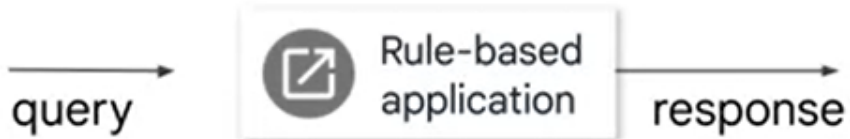
RankBrain (a deep neural network for search ranking) improved performance significantly

#3 Signal for search ranking, out of hundreds

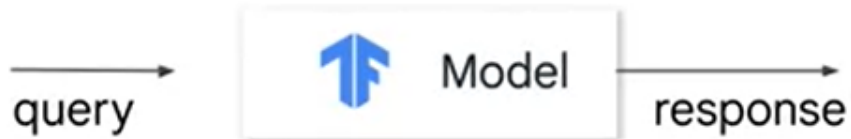
#1 Improvement to ranking quality in 2+ years

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



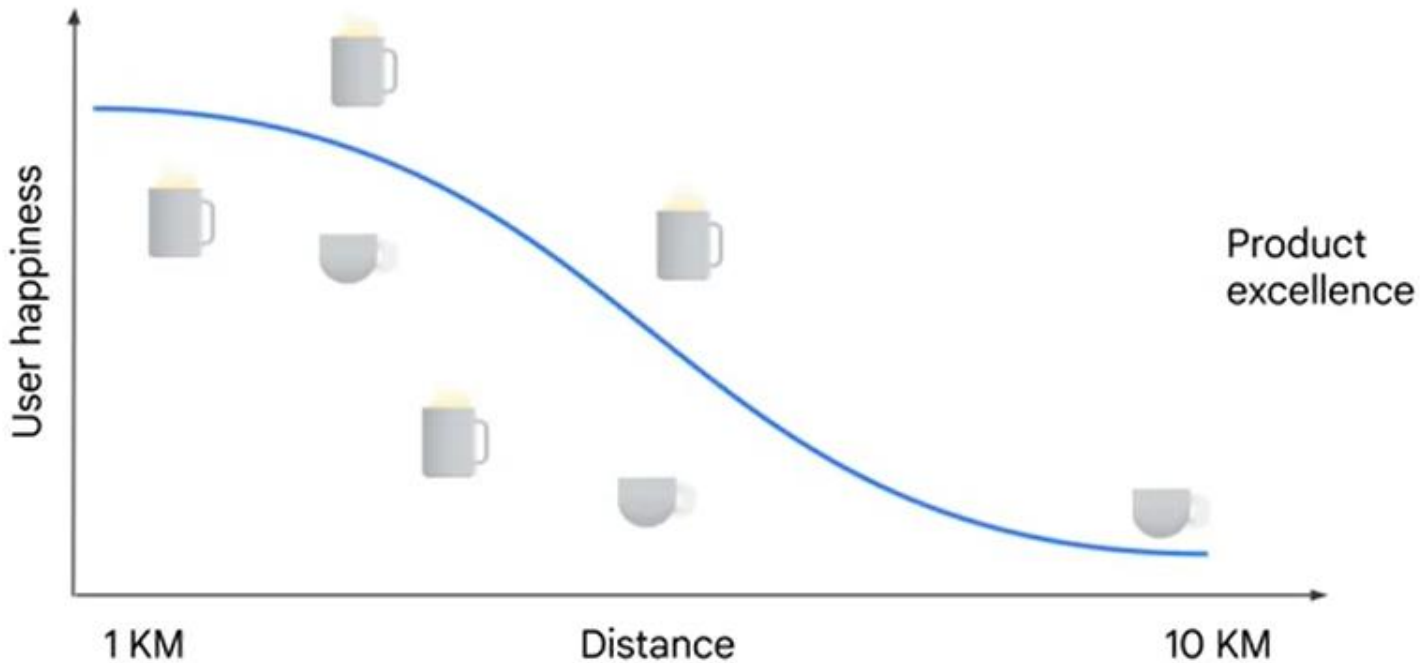
- Codeup rules based on human expertise
- Apply rules program to make decisions
- Add new rules in response to bug reports



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

ML converts examples into knowledge

Good learning involves blending all the users' preferences



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

Example solution:

As a software problem

predictDemand(widgetID, month=CurrentTime.month)

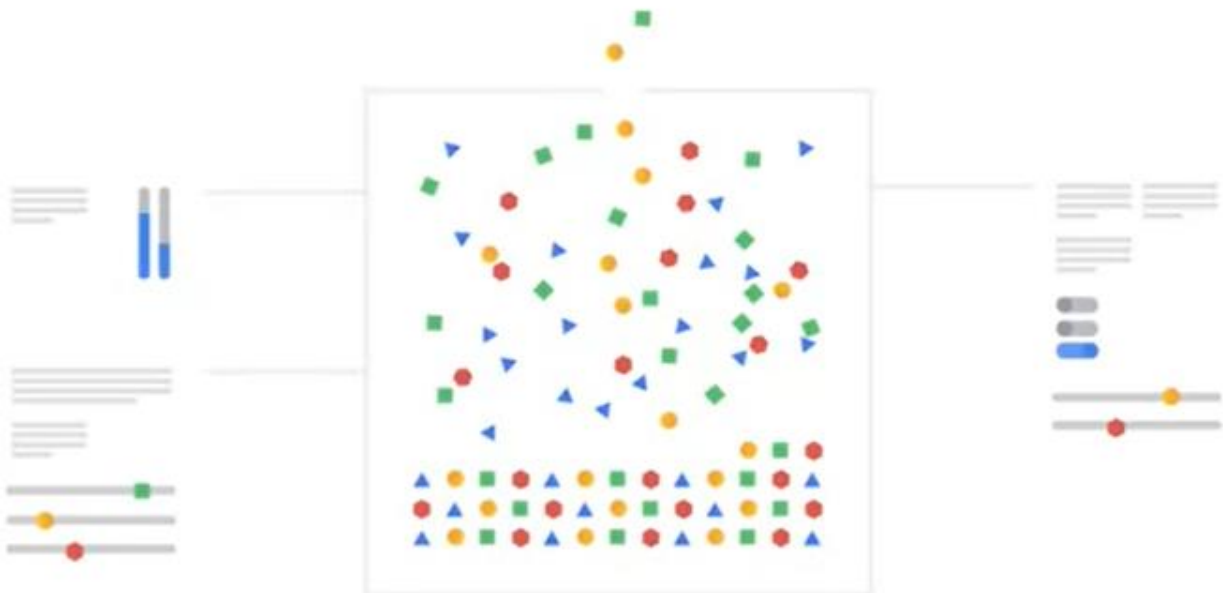
Who will use this service?

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



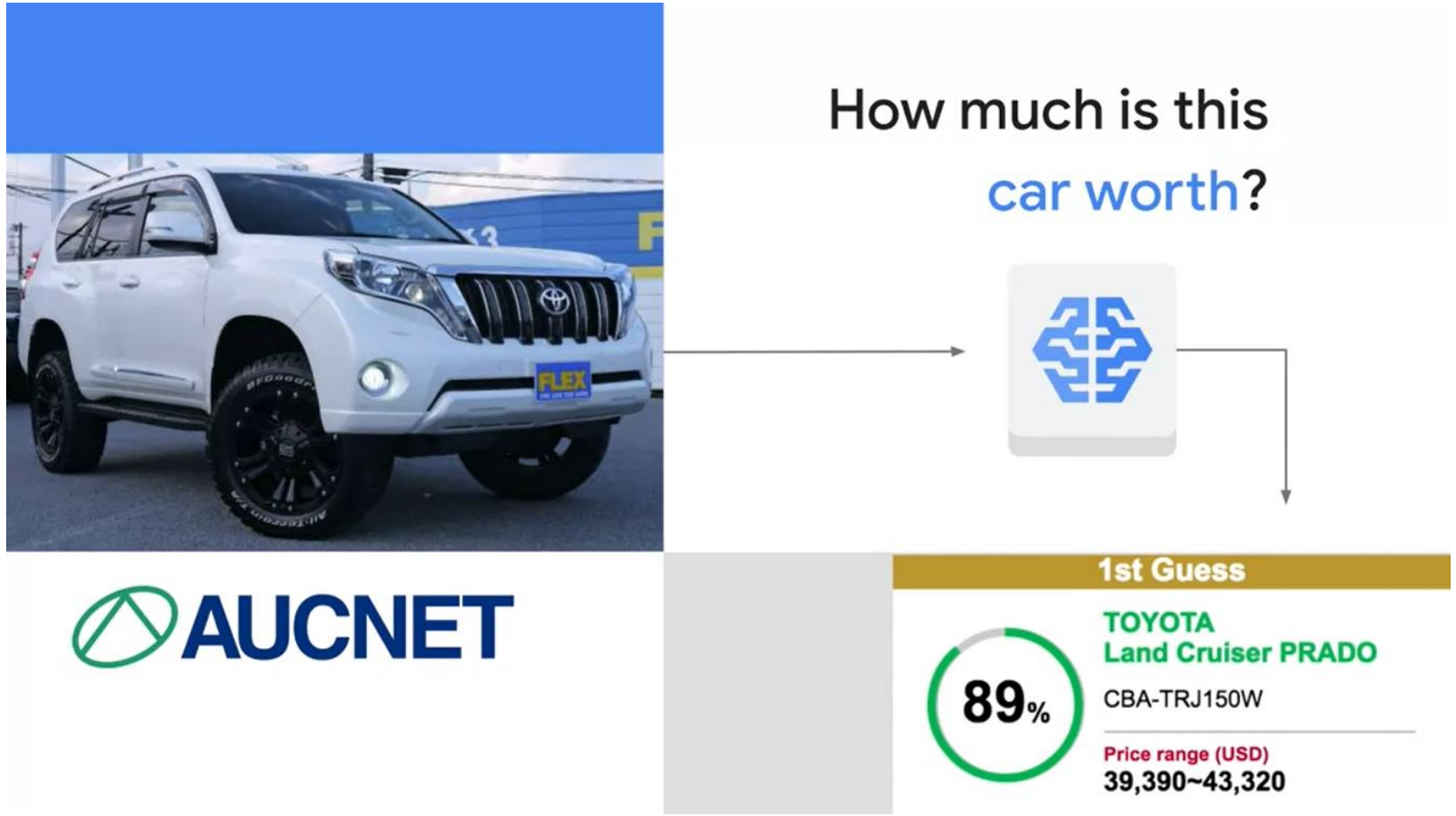
Data problem:

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?

Infuse your apps with ML



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

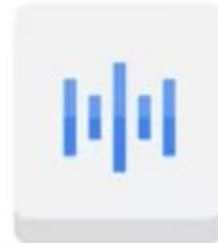
Custom ML models

Vertex AI

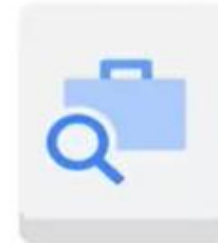
TensorFlow

Pre-trained ML Models

Vision API



Speech API



Jobs API

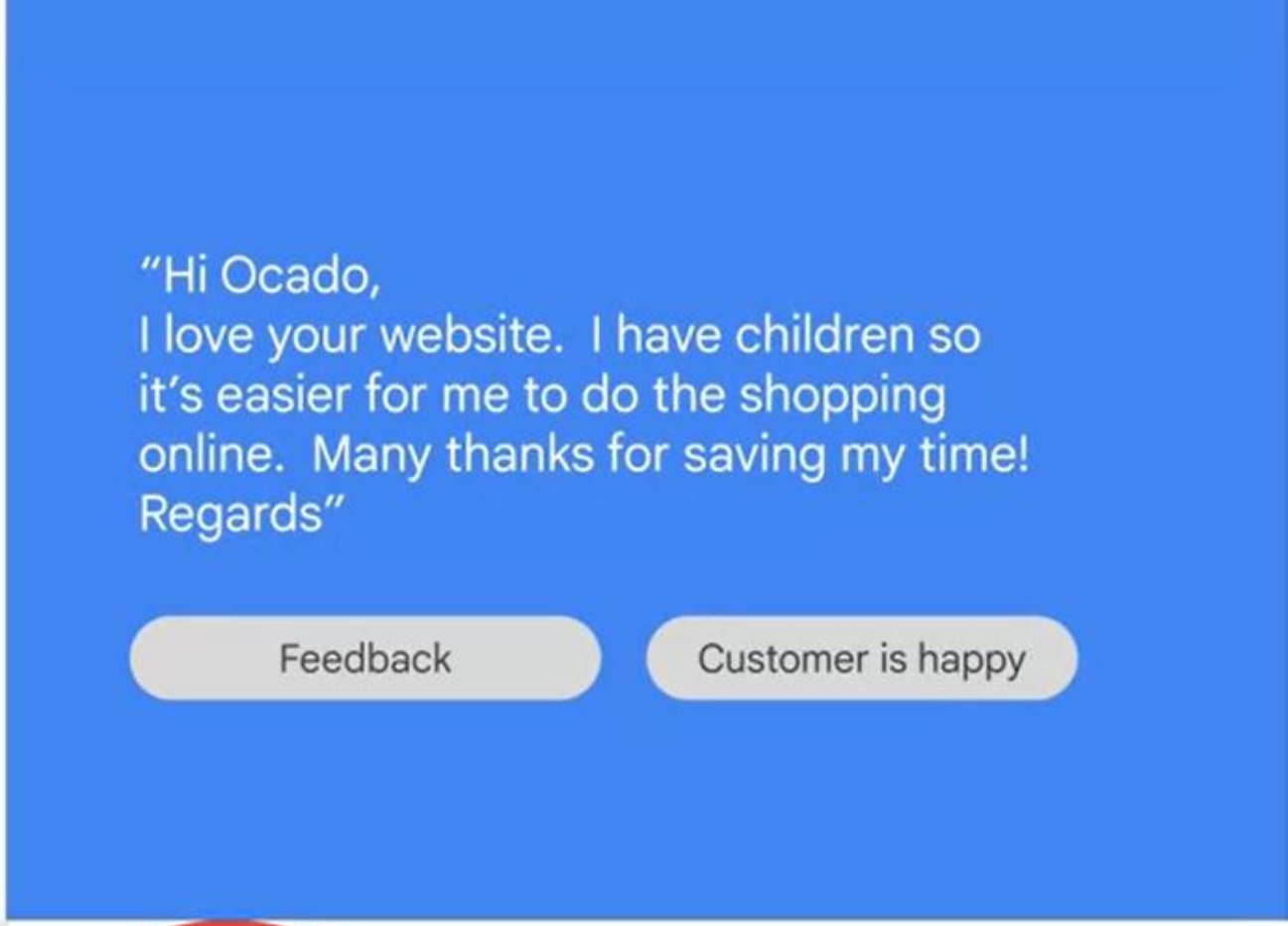
Translation API

Natural Language API

Video Intelligence API

Ocado routes emails based on NLP

Improves natural language processing of customer service claims



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

The ML marketplace is moving towards increasing levels of ML abstraction

Create a custom image model to price cars.

Build off NLP API to route customer emails.

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

Use Dialogflow to create a new shopping experience.

Build a data strategy around ML

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

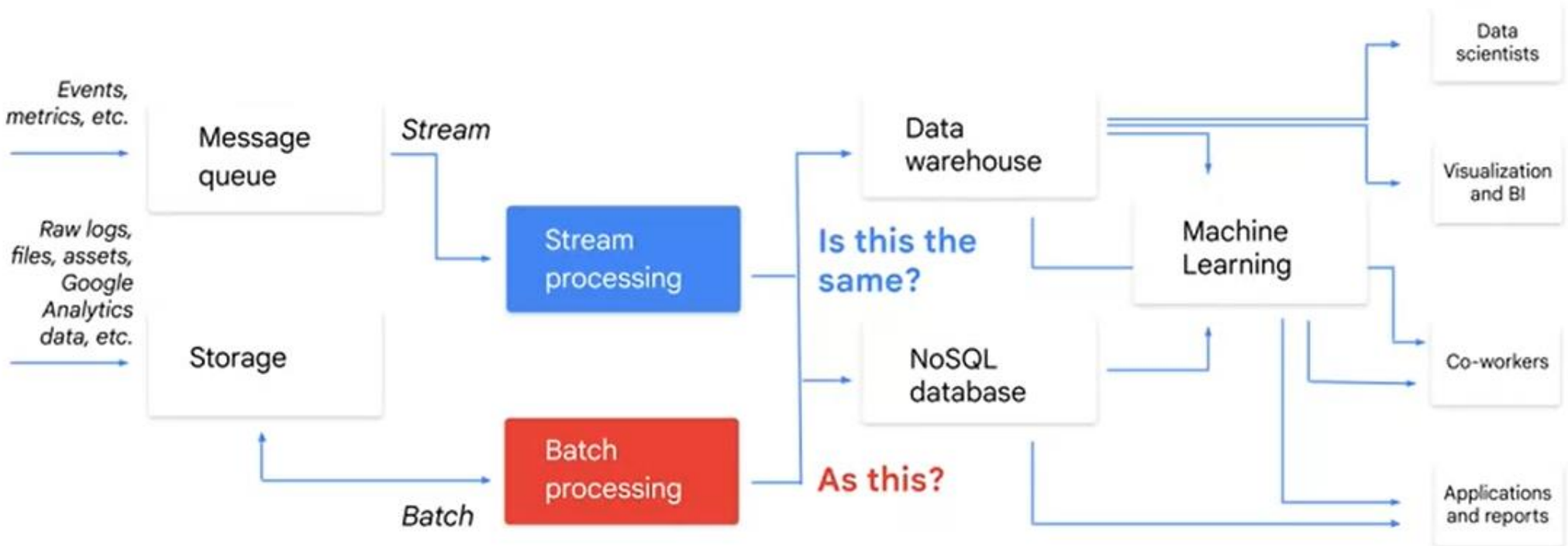
1. Collecting data is often the longest and hardest part of an ML project, and the one most likely to fail

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.

Freedom to experiment (and fail) is important

Take your time and succeed.

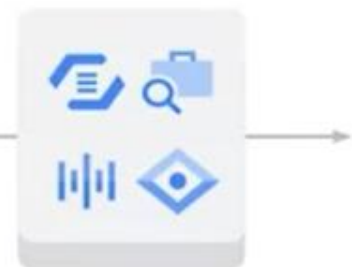
Fail fast and iterate.

Build on top of Google

Images
Audio
Video
Free-form text

ML API

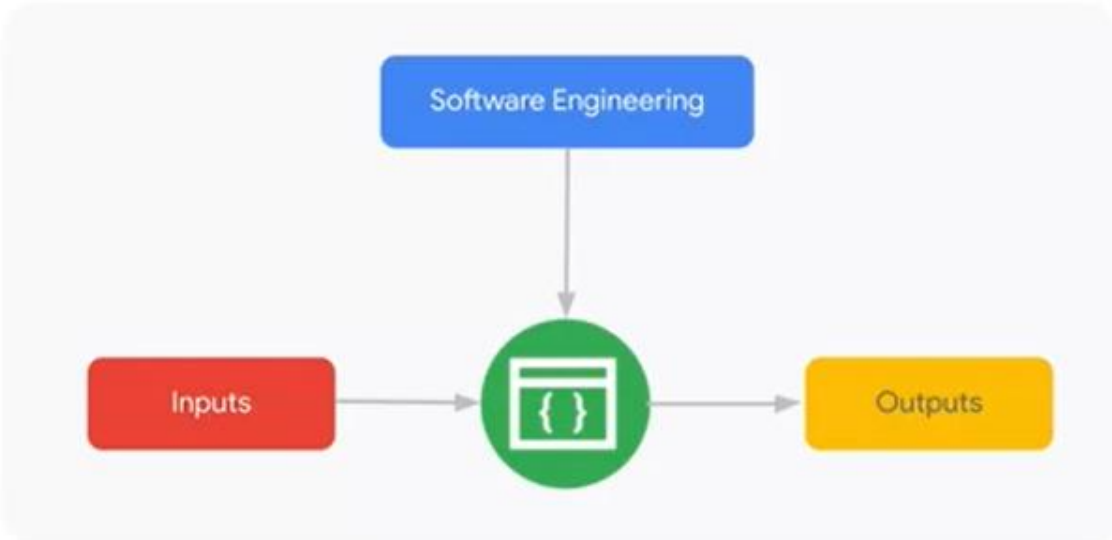
Places
Labels
People
Events
...



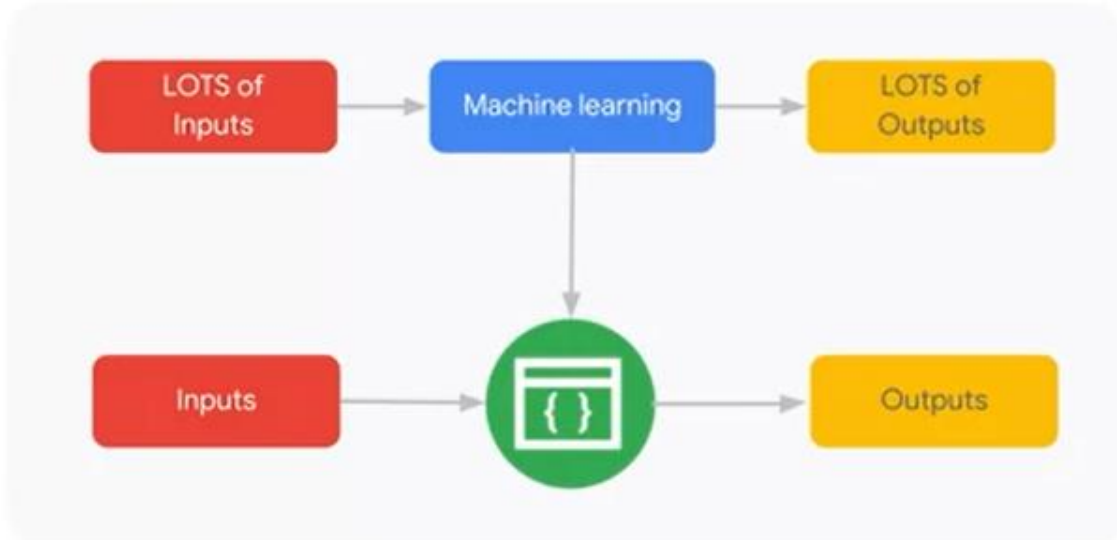
Vertex AI

- Module 1: What it Means to be AI First

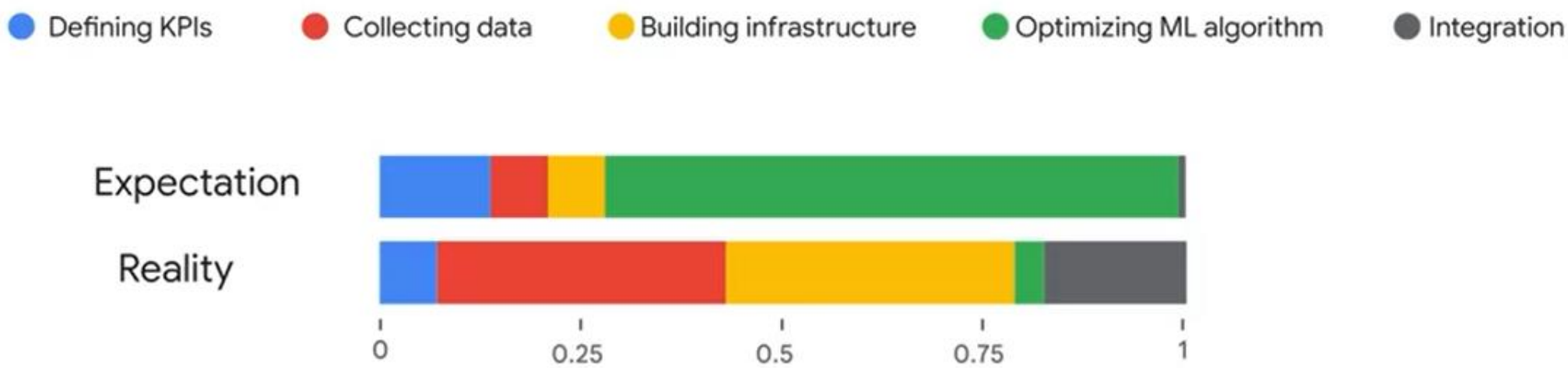
Software engineers write program rules



Machine learning figures out program rules



ML effort allocation



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

ML and business processes

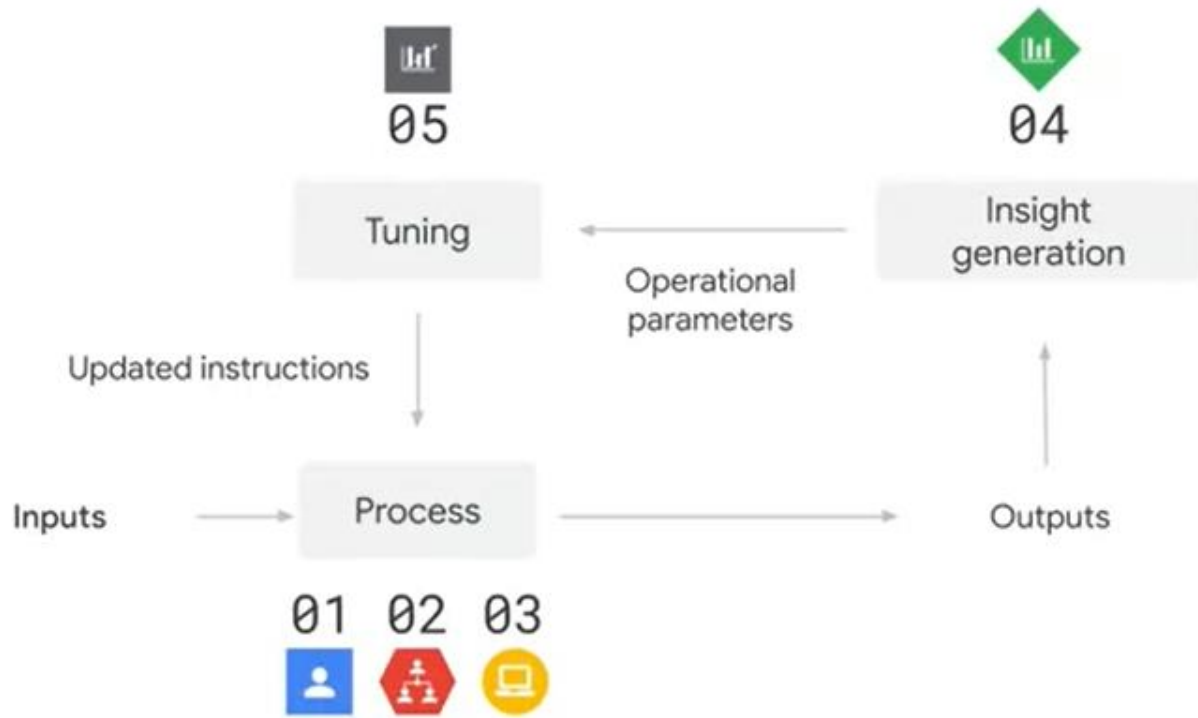
Example: Call center feedback loop

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



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

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

Typical ML development during experimentation

Framing the problem

Prepare training data

Experimenting

- Model A
- Model B
- Model C

Evaluating the model

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

- Numerical data sets
- Bivariate data sets
- Multivariate data sets
- Categorical data sets
- Correlation data sets

Different hyperparameters

Different hardware

Typical ML development during experimentation

Different architectures	Different input data sets	Different hyperparameters	Different hardware
		• Learning rate • Number of layers • Num_estimators • Max_depth	

Typical ML development during experimentation

Different architectures

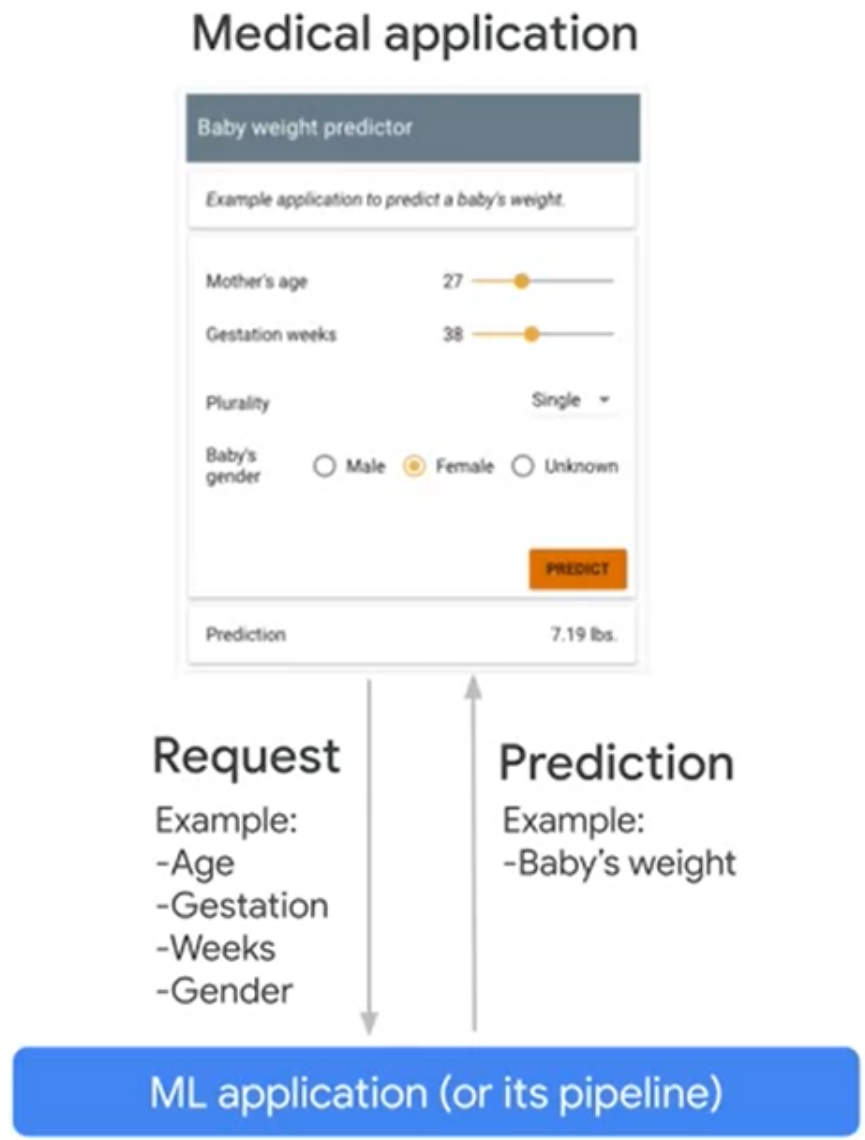
Different input data sets

Different hyperparameters

Different hardware

- CPUs
- GPUs
- TPUs

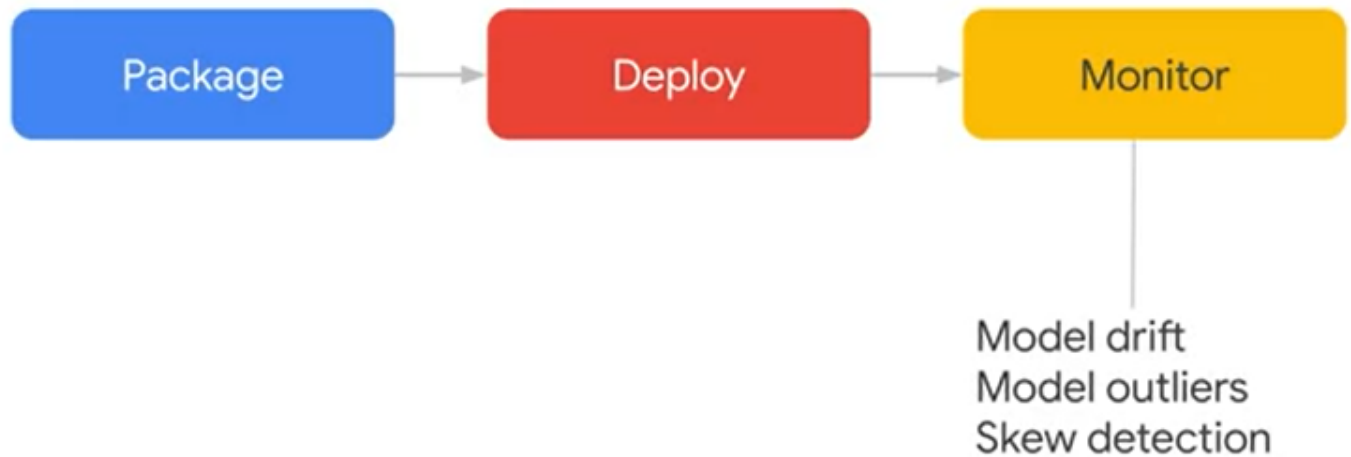
ML application generates a REST service for use by a medical application



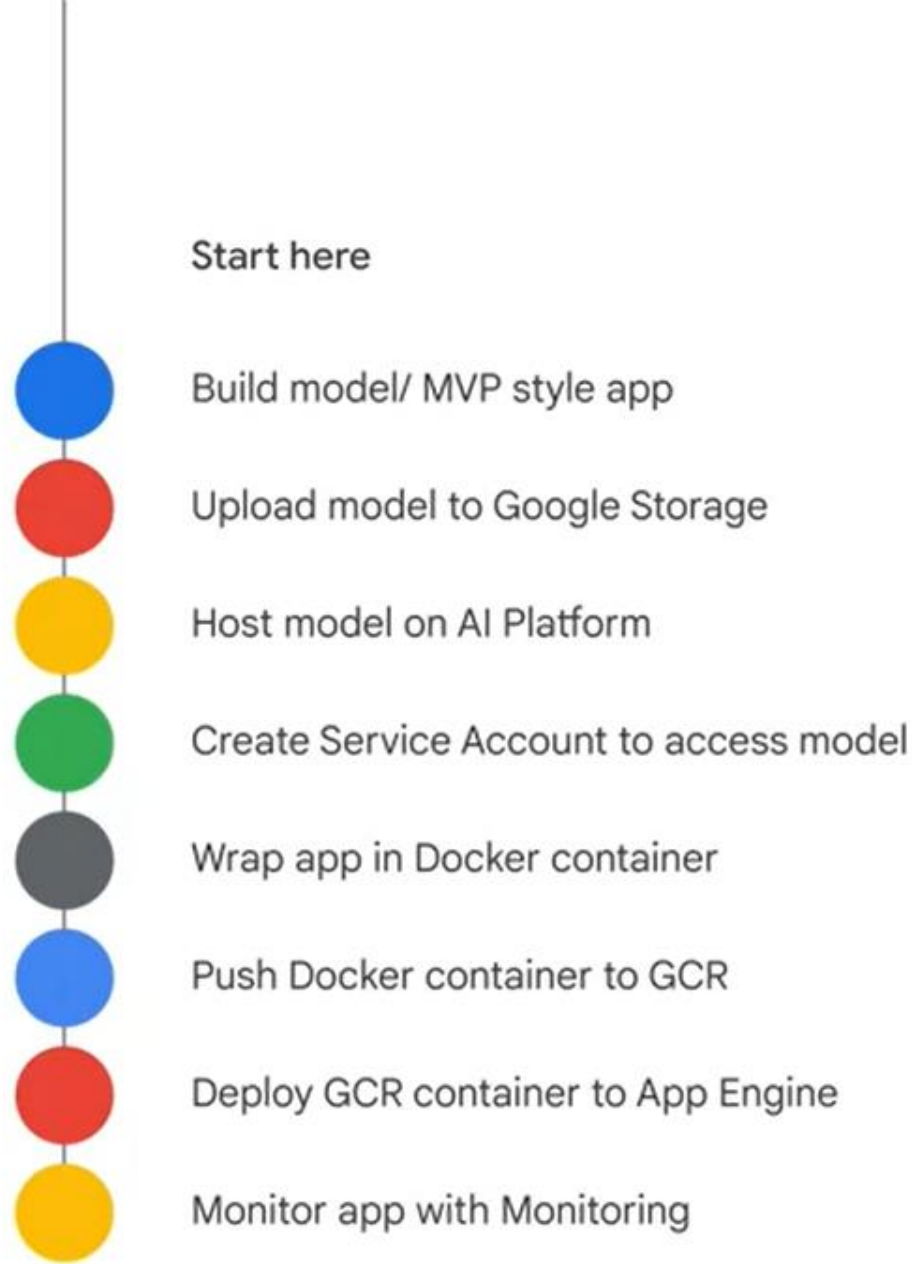
Why model monitoring

- 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



ML product knowledge required



What is there to unify?

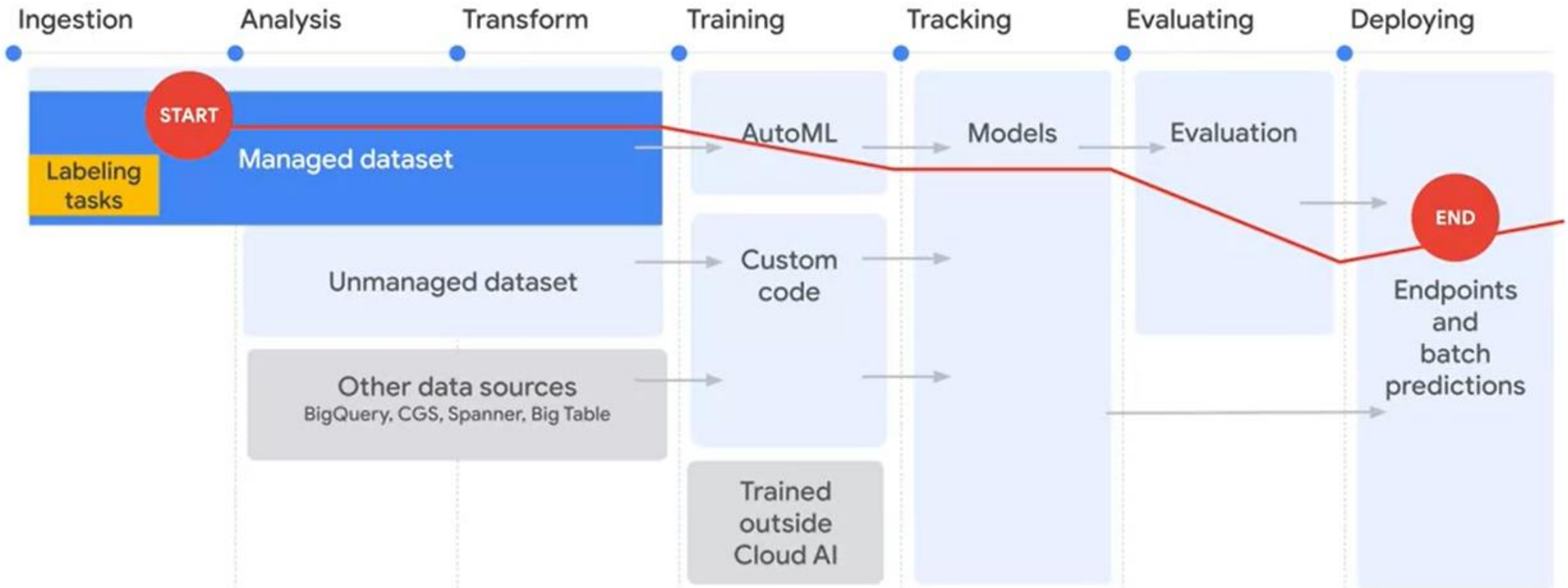
Dataset is

- 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.

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

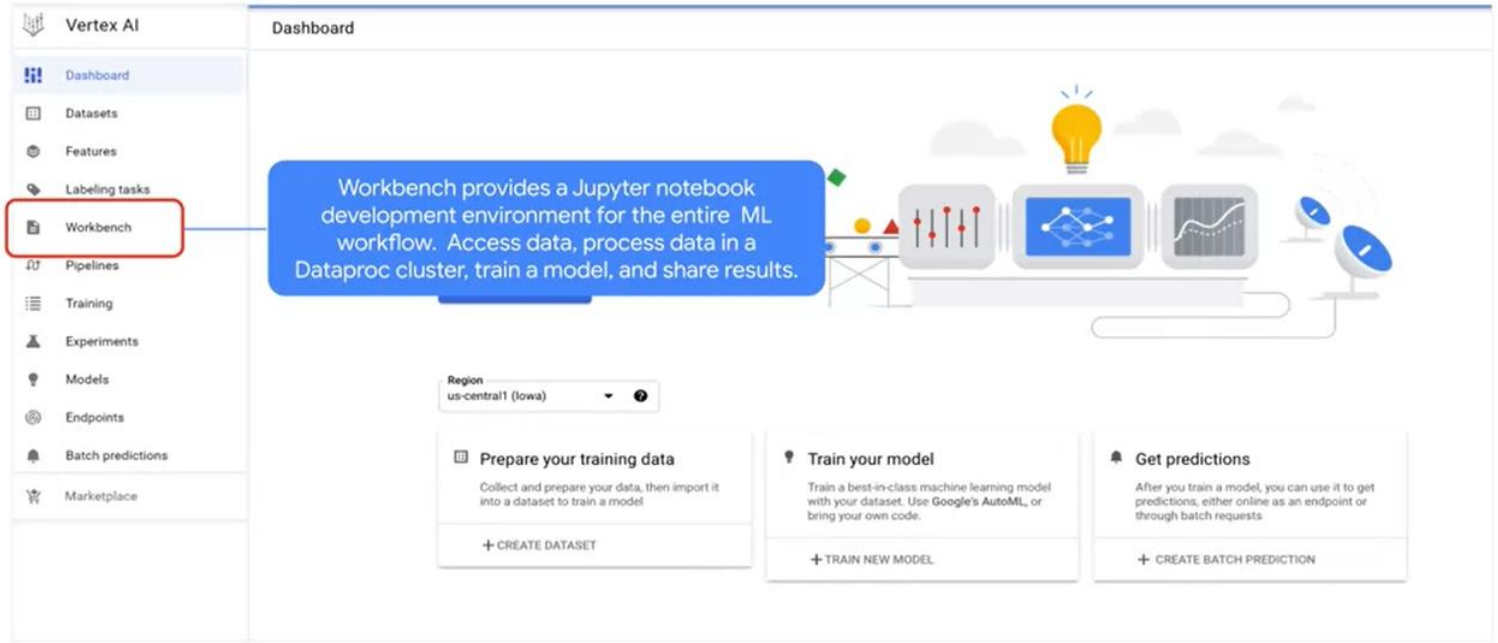
Fast experimentation

Accelerated deployment

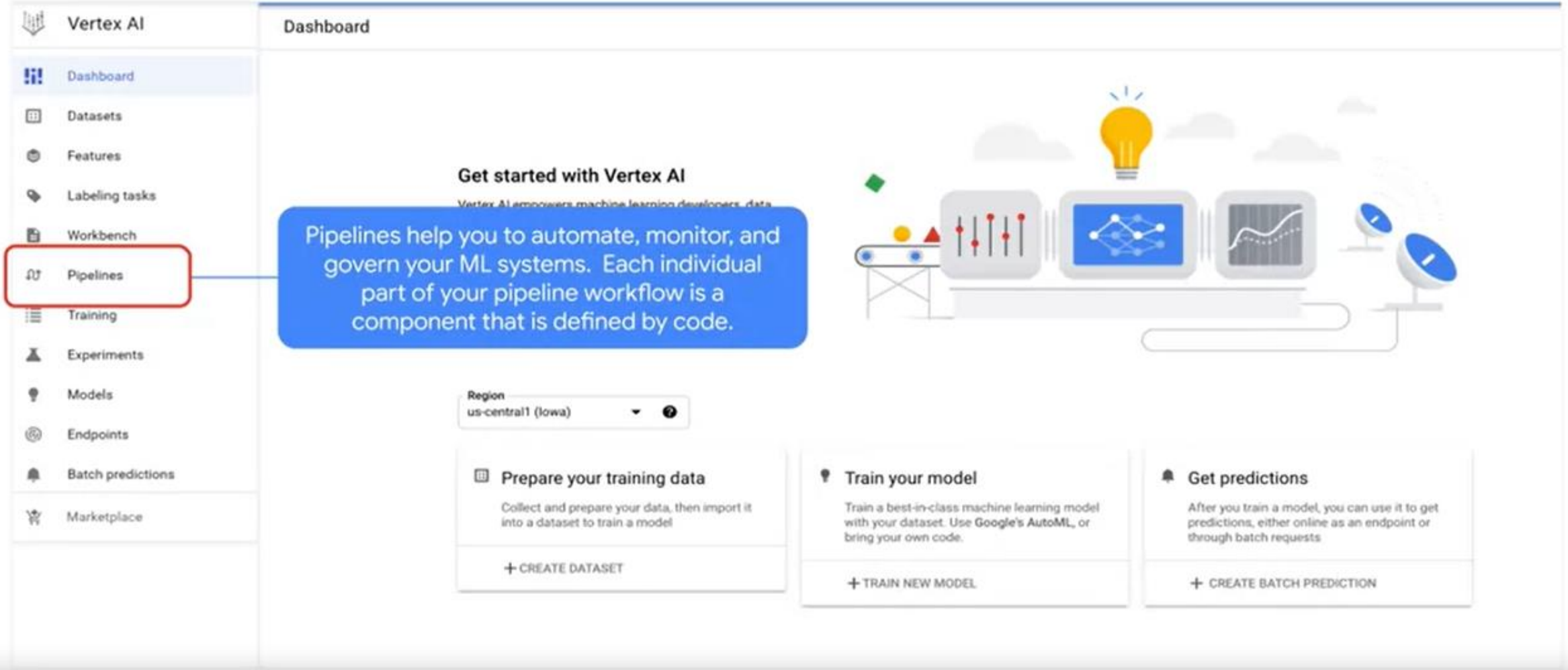
Simplified model management

Components of Vertex AI

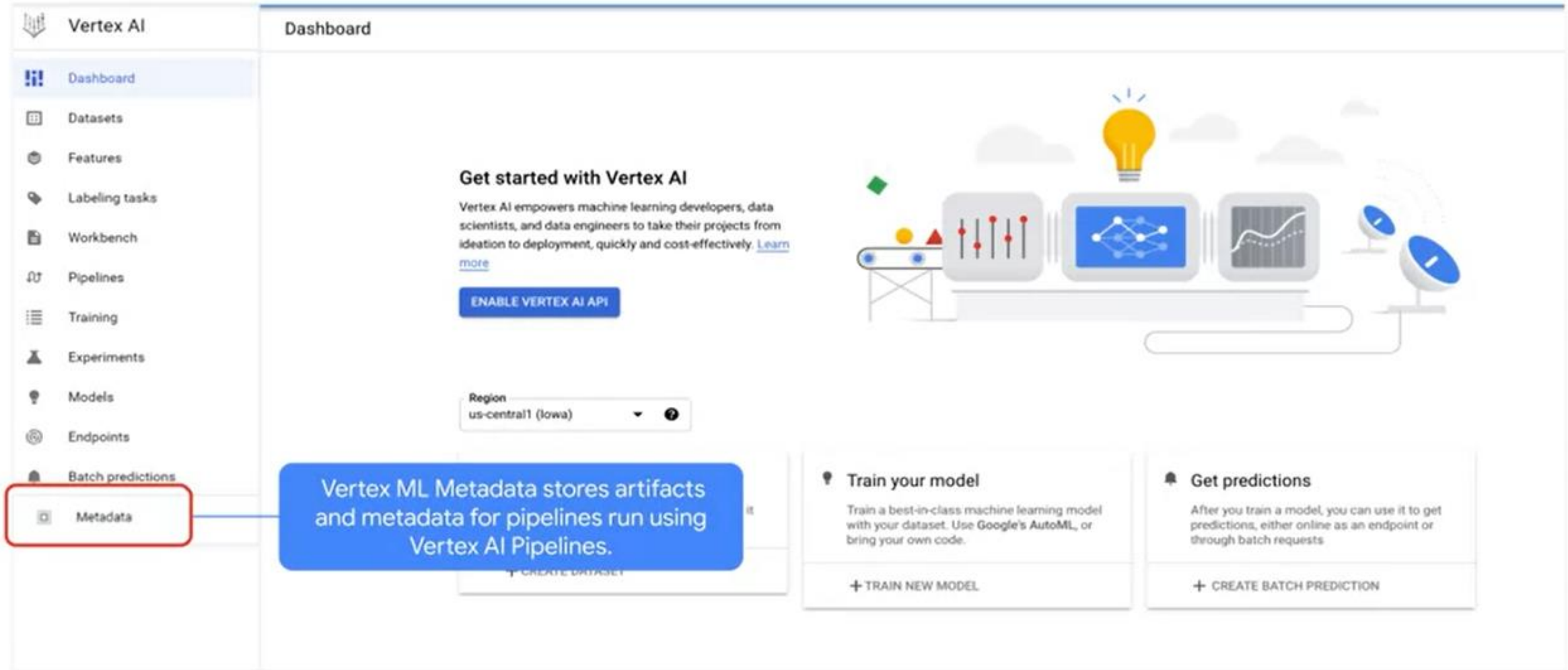
Vertex AI > Workbench



Vertex AI > Pipelines

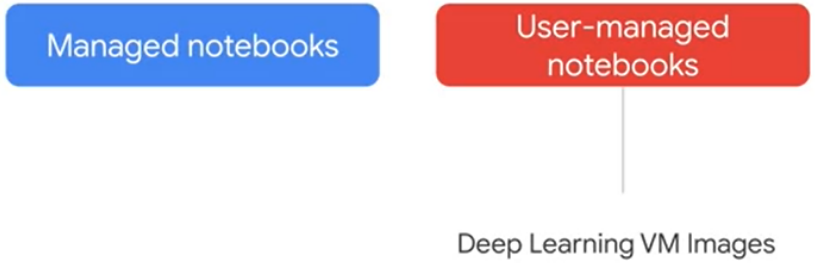


Vertex AI > Metadata



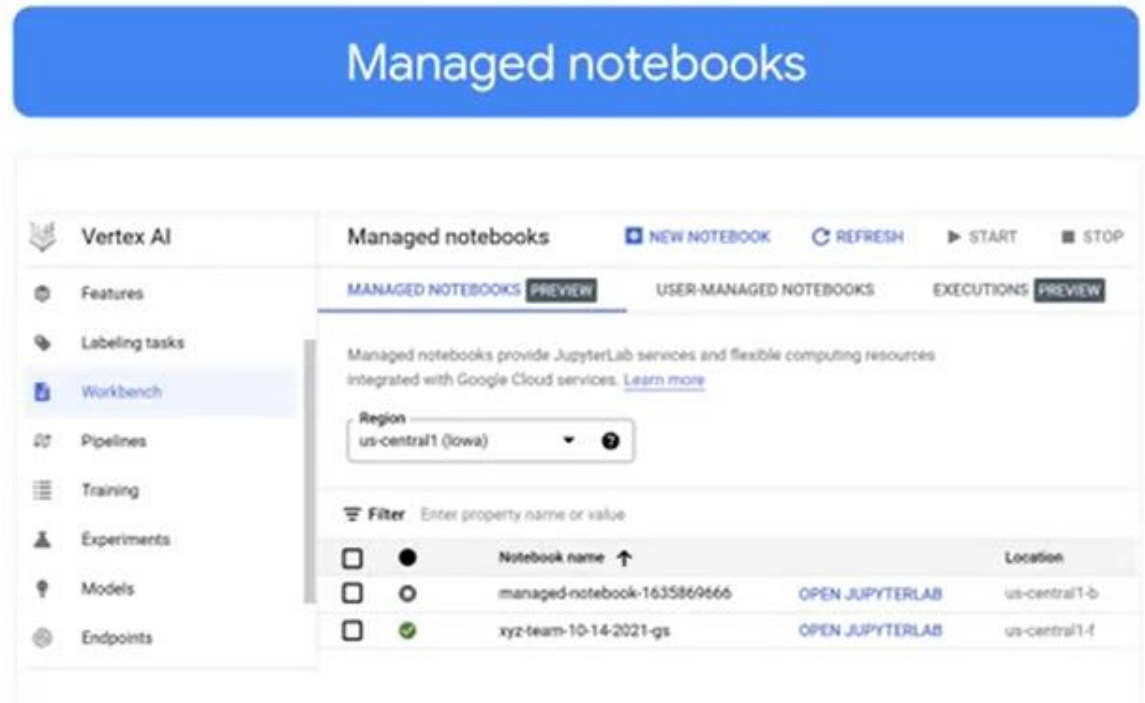
Machine learning development with Vertex Notebooks

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

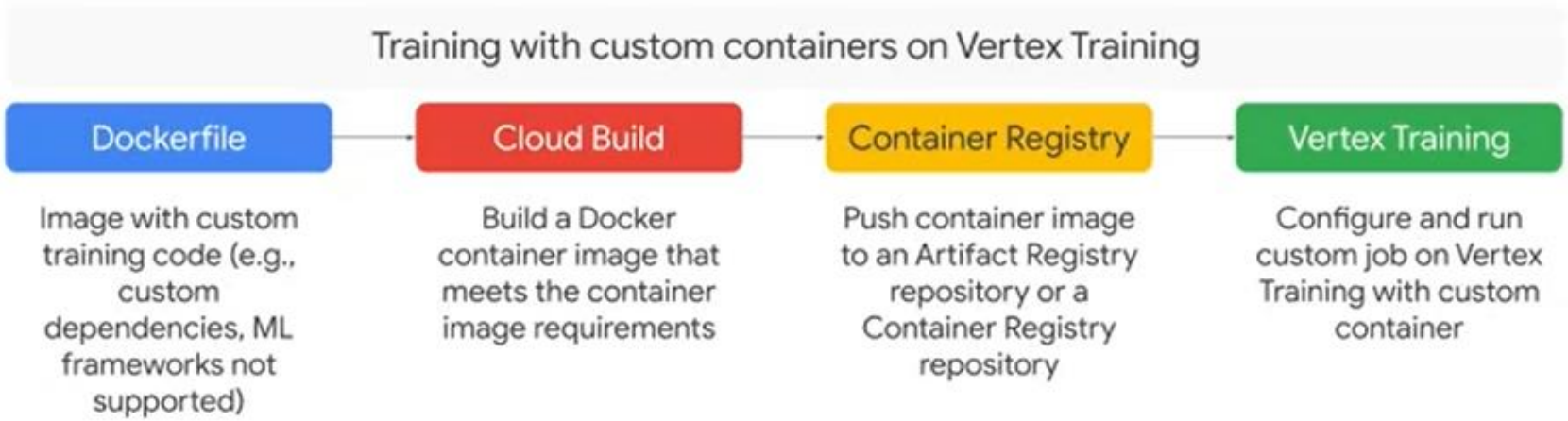


Good choice for data exploration, analysis, modeling, or as part of an end-to-end data science workflow.

Perform workflow-oriented tasks without leaving the JupyterLab interface.



Managed notebooks



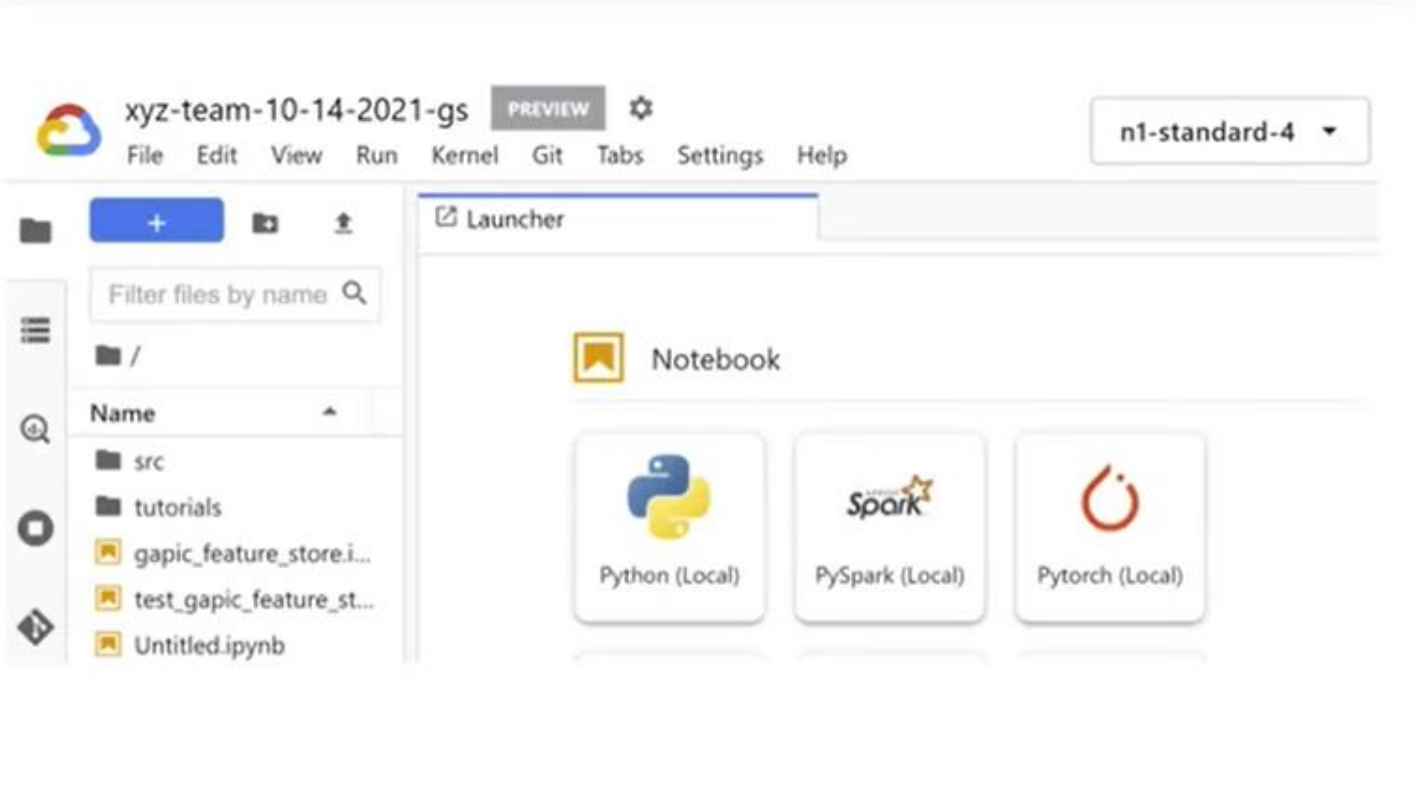
Control hardware

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

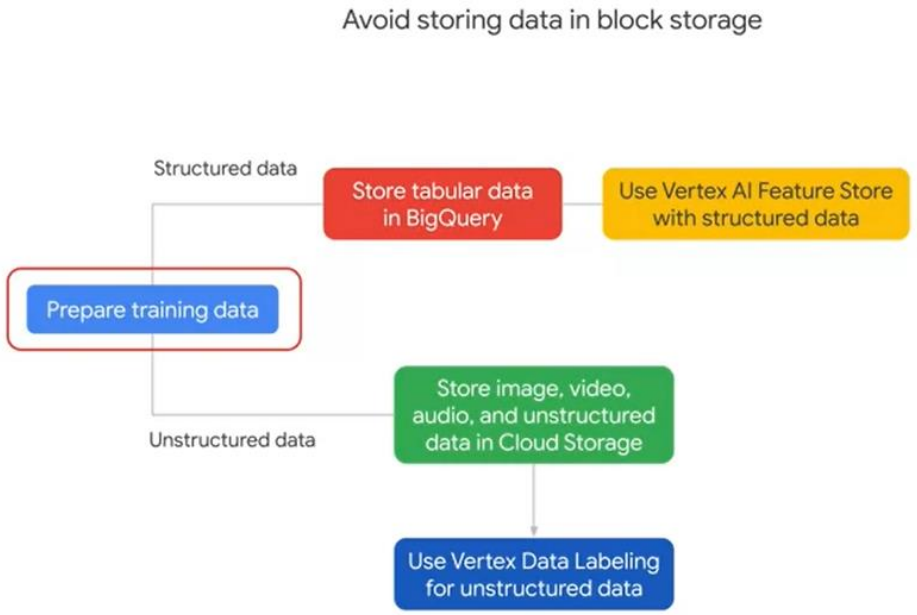
Process data quickly by running a notebook on a Dataproc cluster.

Best practices for machine learning development – Vertex AI

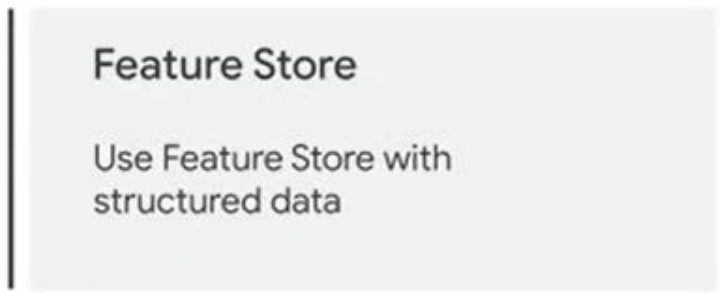
Best practices for preparing and storing data

Data

How it is prepared and stored



Vertex AI Feature Store



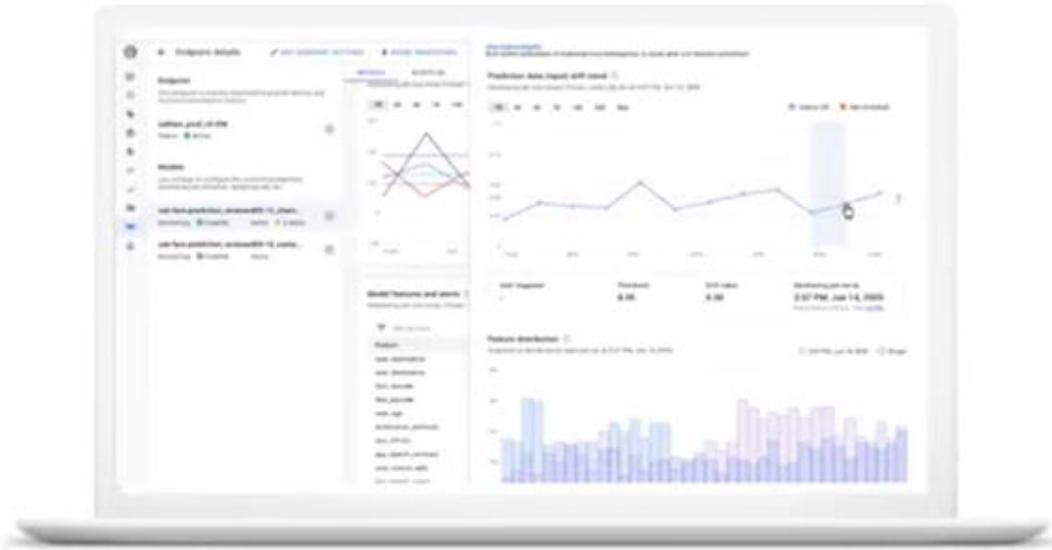
Follow these steps:

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

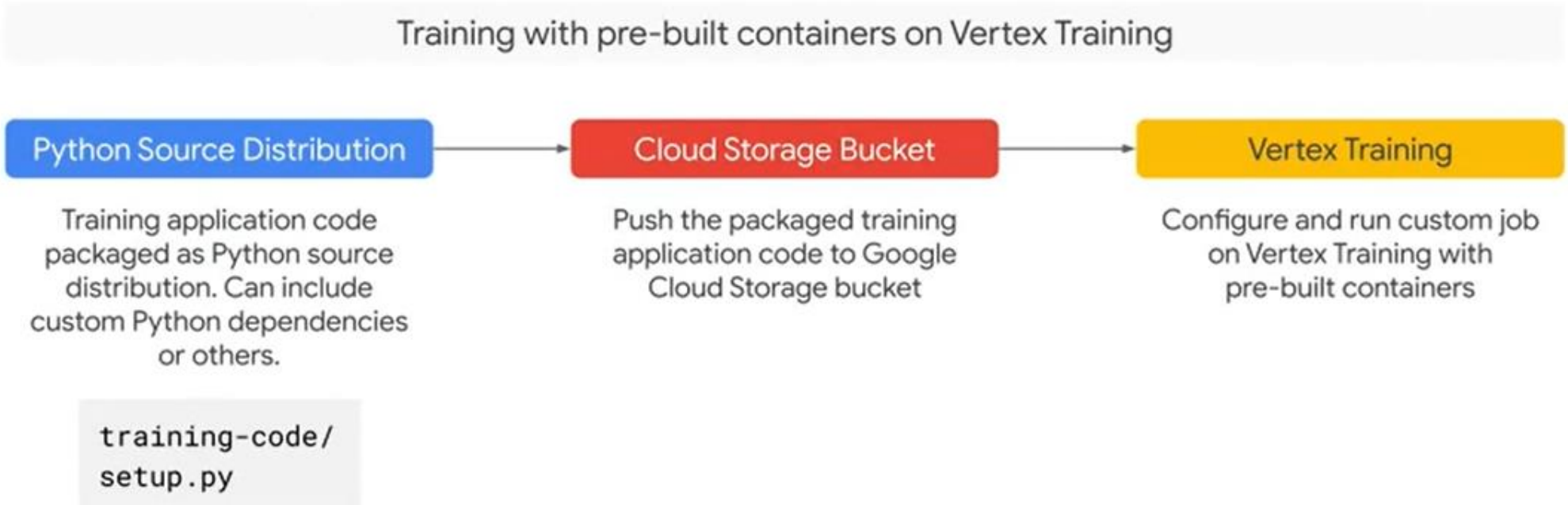
Tips for training, maximizing predictive accuracy, and feature attributions for insights



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

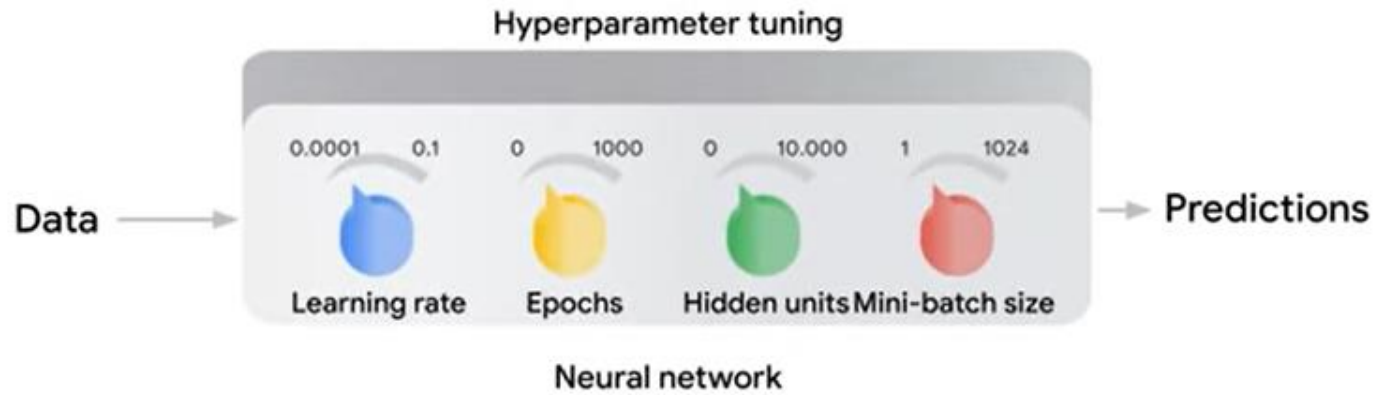
Offers feature attributions to provide insights into why models generate predictions.

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



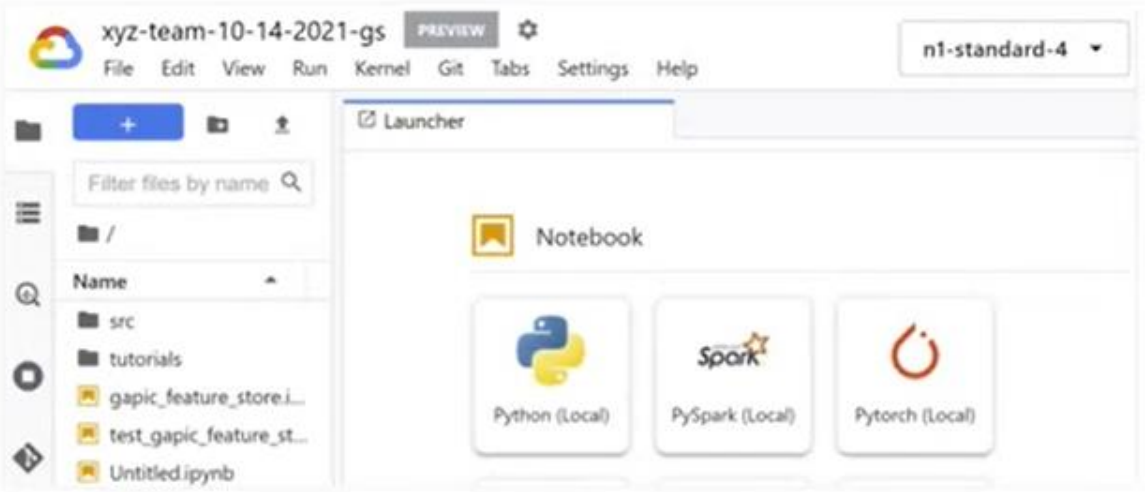
The hyperparameters are knobs that act as the network-human interface.

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

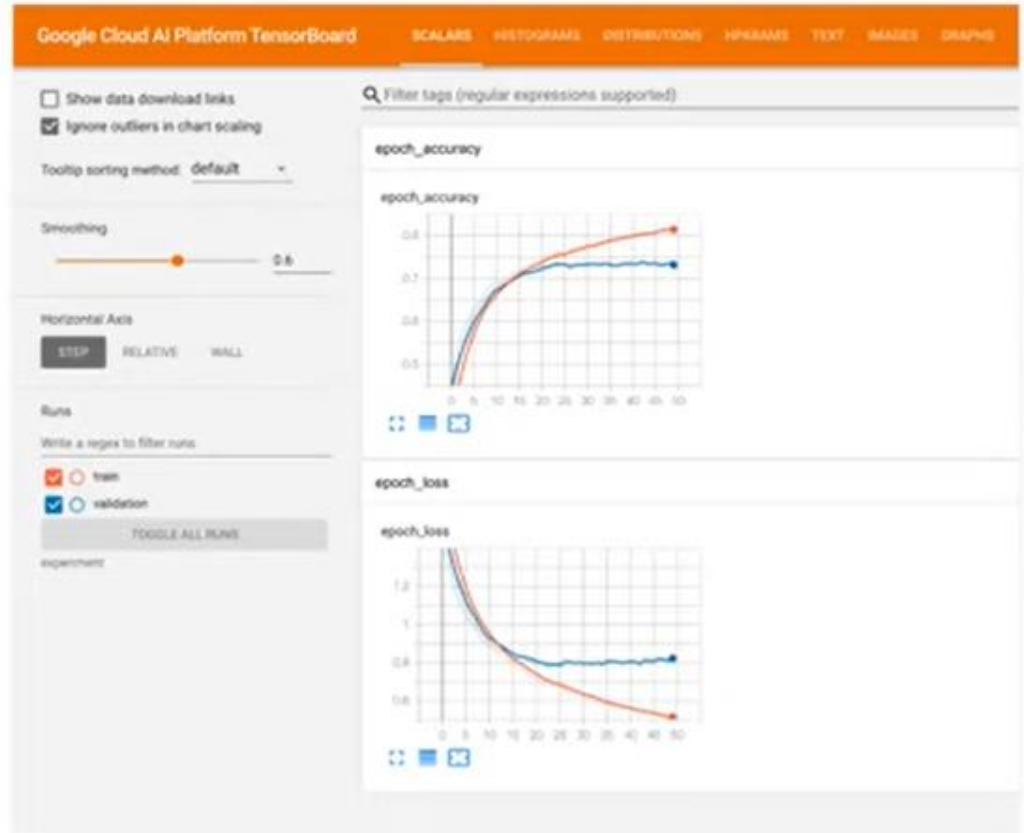
> **Workbench Notebooks**
>
> Use Notebooks to evaluate and understand your models.



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

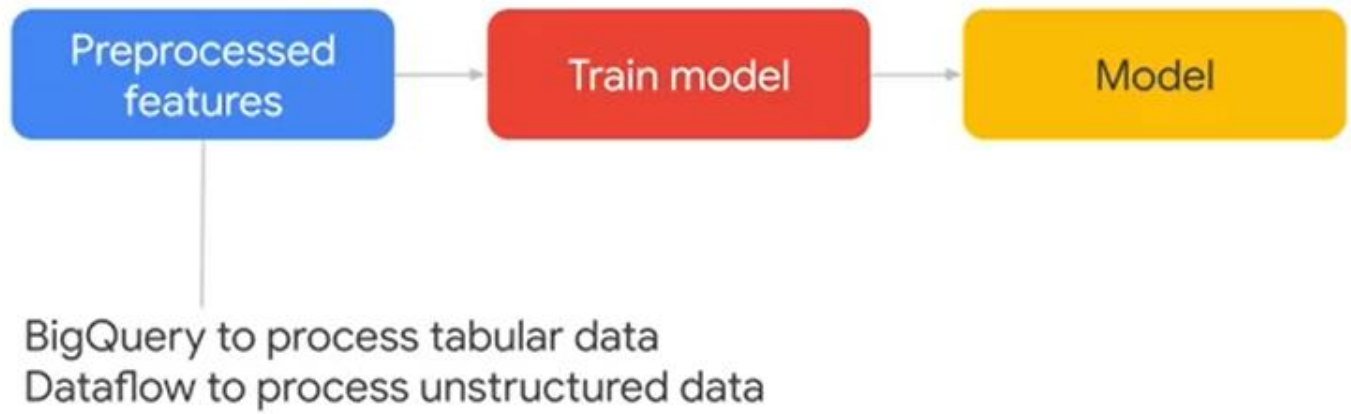
Best practices for using Vertex AI TensorBoard

> **TensorBoard**
>
> Use Vertex AI TensorBoard to visualize experiments.

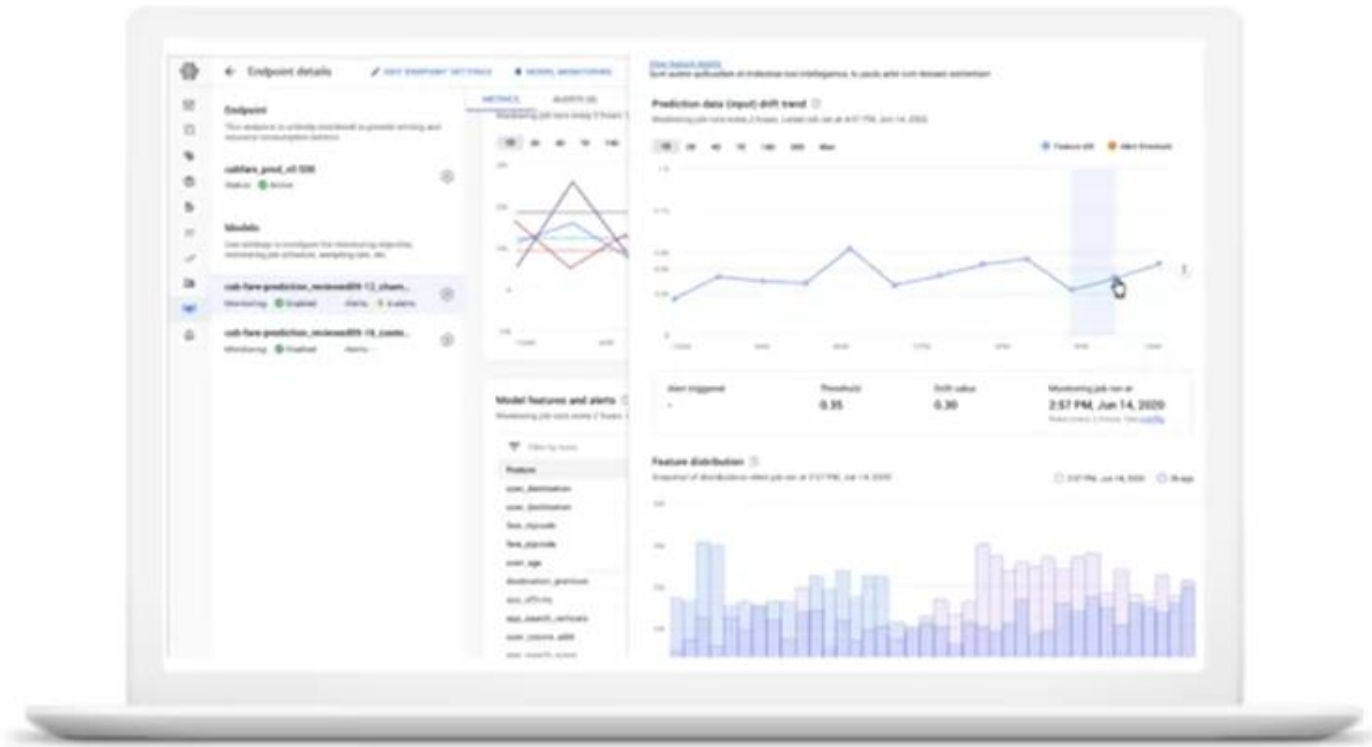


Vertex AI TensorBoard service lets you track experiment metrics such as loss and accuracy over time, visualize a model graph, project embeddings to a lower dimensional space, and much more.

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

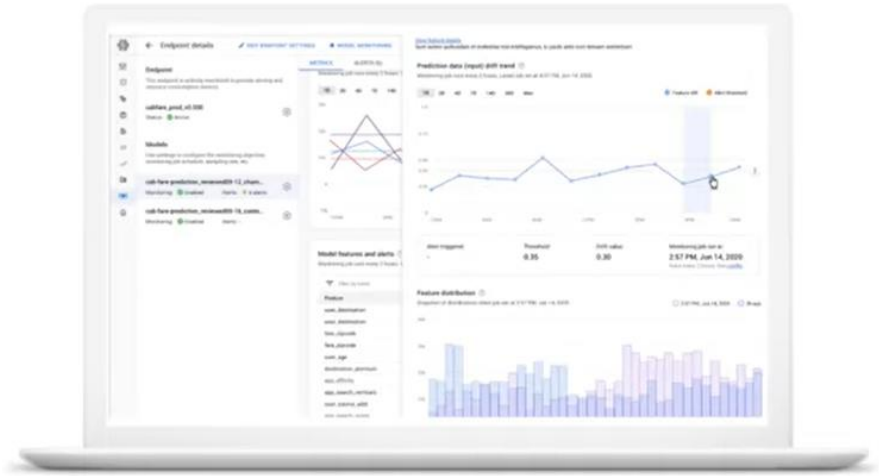
> **TensorFlow Extended**
>
> Use TensorFlow Extended when leveraging TensorFlow ecosystem.

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 machine learning 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

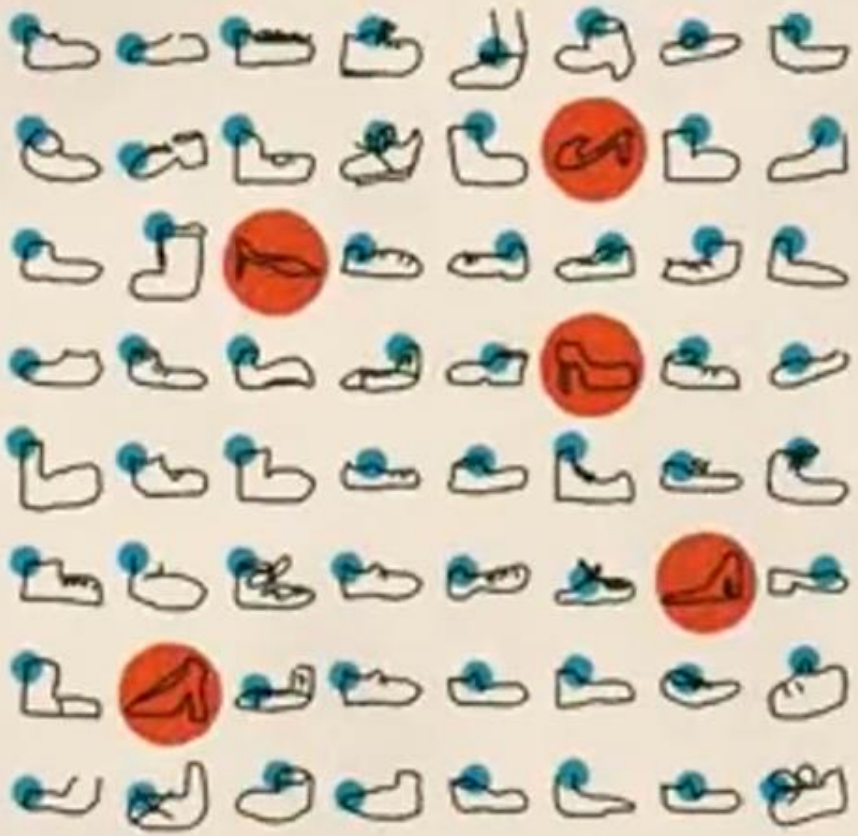
Secure PII in Notebooks.

Data & model

Store prepared data and model in same project.

Optimize performance & cost

Optimize performance and cost.



interaction bias

latent bias






selection bias

Biases in data

Unconscious biases exist in data

Examples of human biases in data
- Reporting bias
- Selection bias

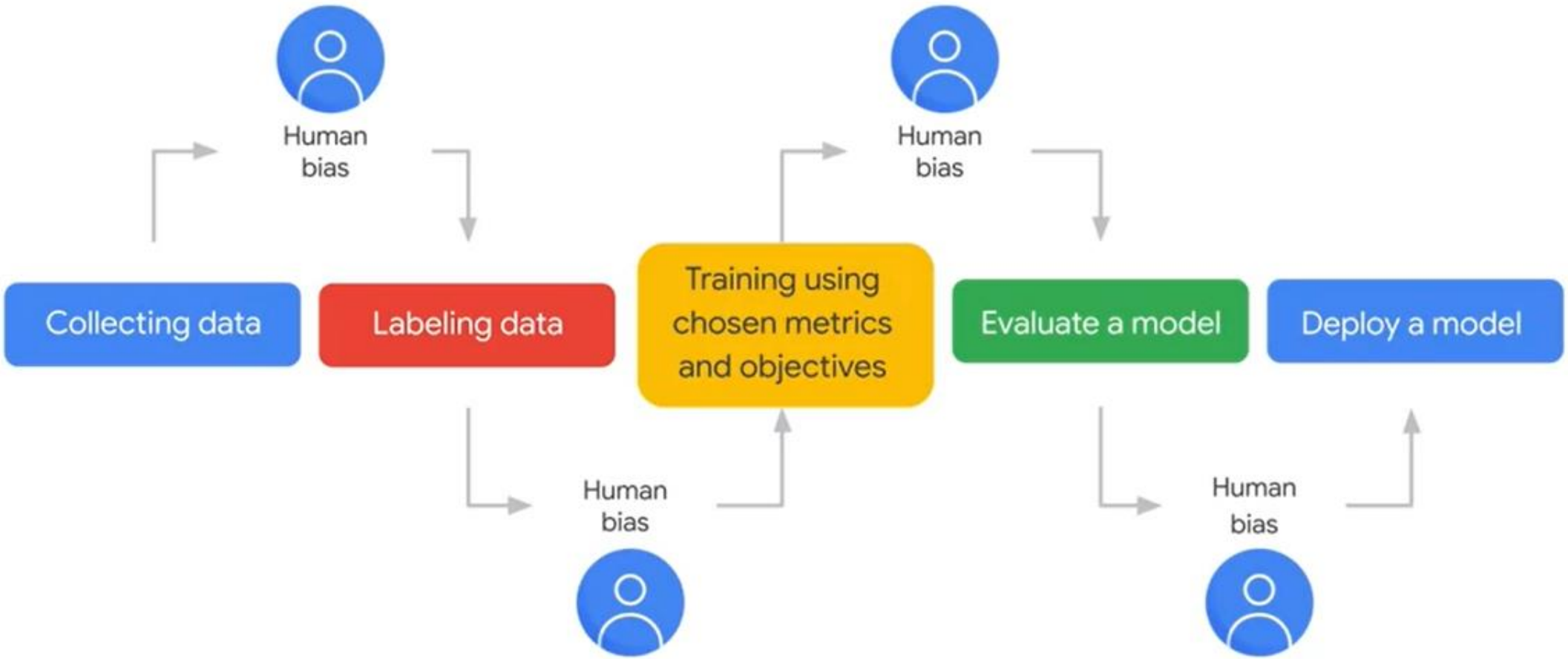
Examples of human biases in collection and labeling
- Confirmation bias
- Automation bias

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

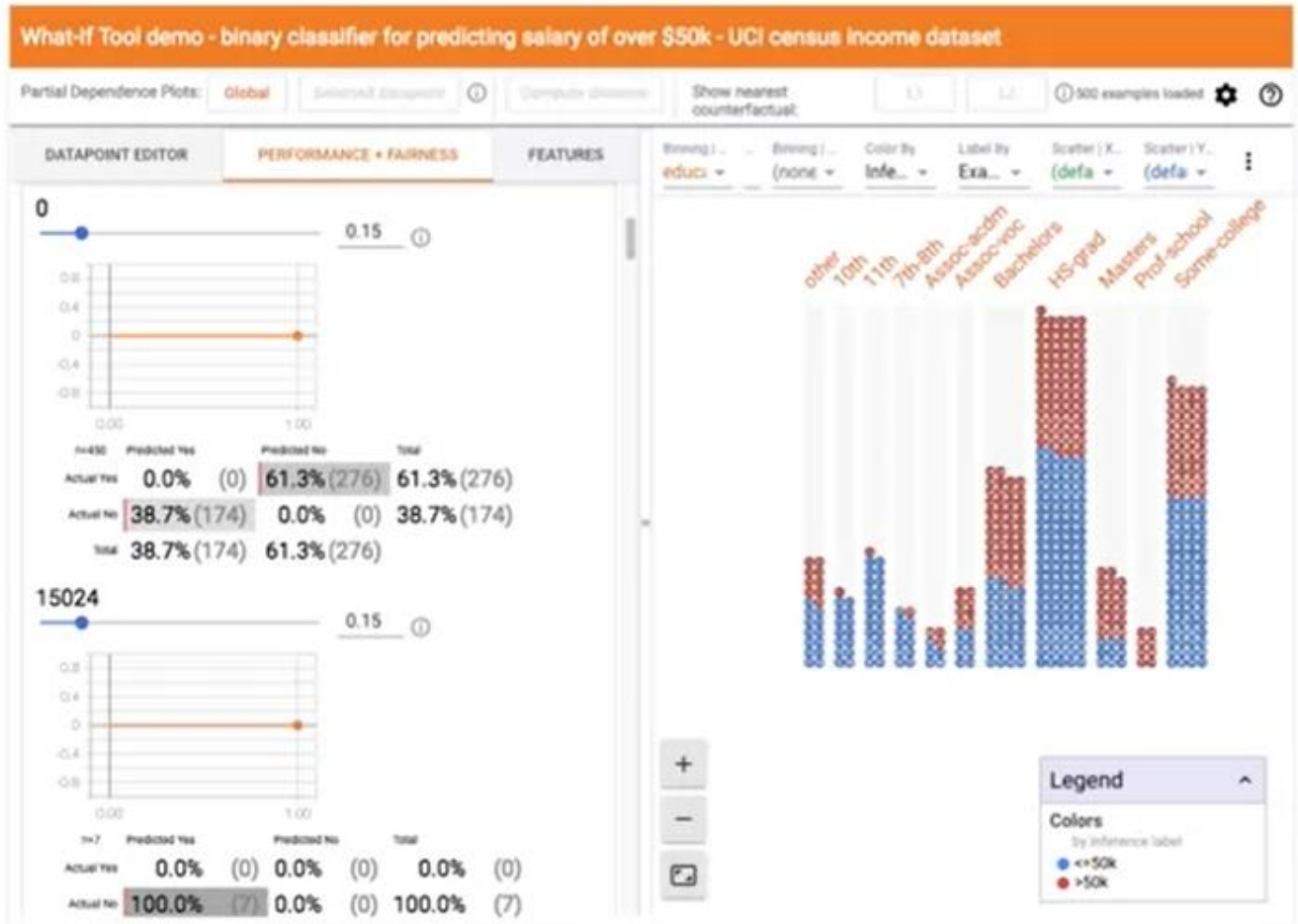
ML models learn from existing data collected from the real world, and so an accurate model may learn or even amplify problematic pre-existing biases in the data based on race, gender, religion, or other characteristics.

ai.google/principles

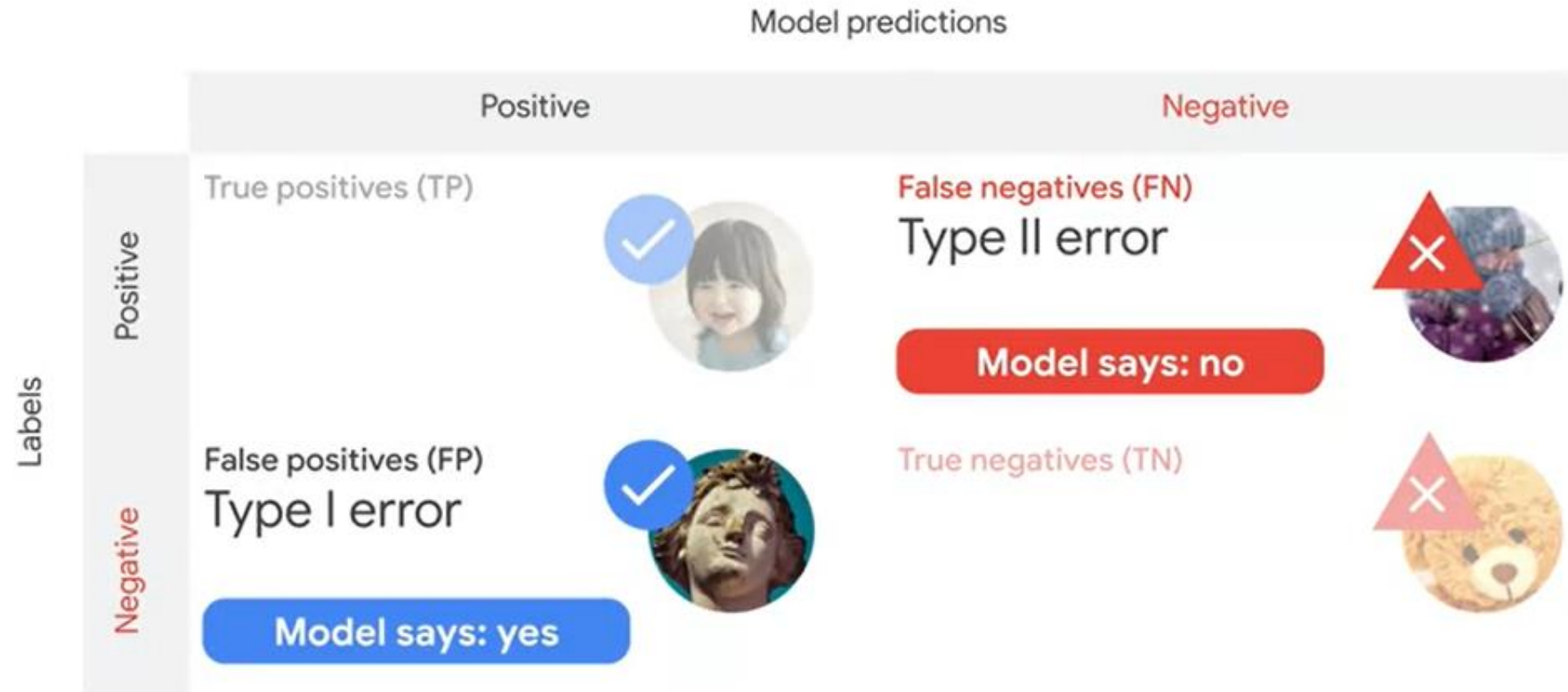
A checklist for bias-related issues

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

Tools for responsible AI



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



How to find errors in your dataset using Facets