How Google Does Machine Learning

Course Summary and Key Takeaways

Learning Objectives

- Describe Vertex AI Platform and how it's used to quickly build, train, and deploy
- AutoML machine learning models without writing a single line of code
- Describe best practices for implementing machine learning on Google Cloud
- Leverage Vertex AI for ML development
- Develop a data strategy around machine learning
- Examine use cases that are then reimagined through an ML lens
- Leverage Google Cloud Platform tools and environment to do ML
- Learn from Google's experience to avoid common pitfalls
- Carry out data science tasks in online collaborative notebooks
- Articulate Responsible AI best practices

Module Breakdown

- Module 0: Introduction to Course and Series
- Module 1: What It Means to be AI-First
- Module 2: How Google Does ML
- Module 3: Machine Learning Development with Vertex AI
- Module 4: Machine Learning Development with Vertex Notebooks
- Module 5: Best Practices for Implementing Machine Learning on Vertex AI
- Module 6: Responsible AI Development

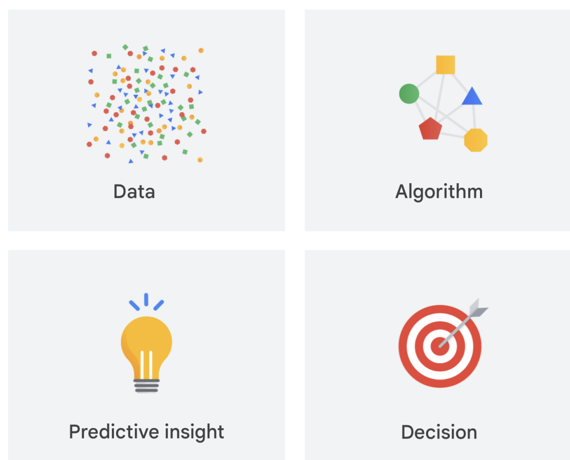
Summary

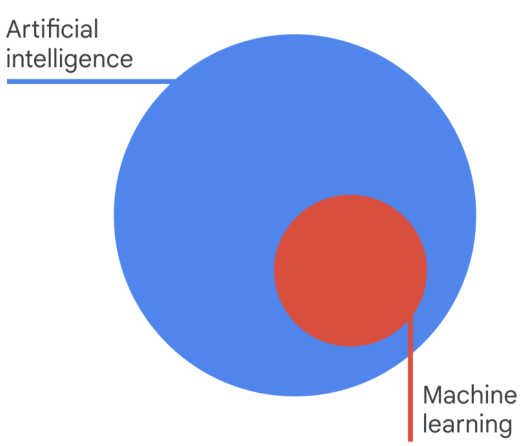
Machine learning at Google is about providing a unified platform for managed datasets, a feature store, a way to build, train, and deploy machine learning models without writing a single line of code, providing the ability to label data, create Workbench notebooks using frameworks such as TensorFlow, SciKit Learn, Pytorch, R, and others. Vertex AI Platform also includes the ability to train custom models, build component pipelines, and perform both online and batch predictions. This course reviews the five phases of converting a candidate use case to be driven by machine learning, and considers why it is important to not skip the phases. We end with a recognition of the biases that machine learning can amplify and how to recognize them.

Key takeaways

Module 1: What It Means to be AI-First

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





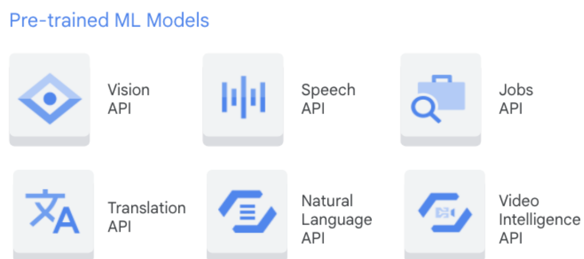
The difference between **machine learning (ML)** and **artificial intelligence (AI)**:

AI is a discipline that has to do with the theory and methods to build machines that think and act like humans. ML is a toolset that you can use machine learning to solve certain kinds of AI problems.

There are two stages of machine learning. Stage 1 is to train an ML model with examples. Stage 2 is to use the trained model to predict what it hasn't seen before. Training a model from examples is called **supervised learning**.

Google infuses ML into almost all its products. The idea behind ML is to take a bunch of examples and convert that knowledge into future predictions.

An easy way to add ML to your apps is to take advantage of pre-trained models. Google has several **pre-trained ML models**.



Given the choice between more data and more complex models, spend your energy collecting more data. That means, collecting not just more quantity, but also more variety.

So, how do you get started on machine learning? We suggest selecting a use case where you're doing manual data analysis today.

- Collect a lot of data
- Go through manual analysis
- Know your data
- Understand that ML is a journey towards automation and scale

---

Module 2: How Google Does Machine Learning

Avoid these ten ML pitfalls:

1. You thought training your own ML algorithm would be faster than writing the software
2. You don't collect enough data
3. You haven't looked at the data but assume it's ready to use
4. You forgot to put and keep humans in the loop
5. Your product launch focused on the ML algorithm
6. You optimized your ML algorithm for the wrong thing
7. You don't know if your ML is improving things in the real world

8. You didn't use pre-trained ML algorithm
9. You only trained your ML algorithm once
10. You designed your own perception or NLP algorithm

The path to ML has five phases:

— 1 — Individual contributor

— 2 — Delegation

— 3 — Digitization

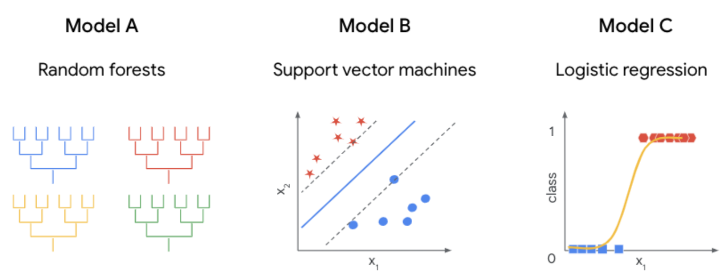
— 4 — Big data and analytics

— 5 — Machine learning

Module 3: Machine Learning Development with Vertex AI

At a high level, machine learning development addresses framing the problem, preparing the data, experimenting, and evaluating the model.

You build and compare many different models to determine which works best. For example, random forests, support vector machines, and logistic regression are just three models you could use.

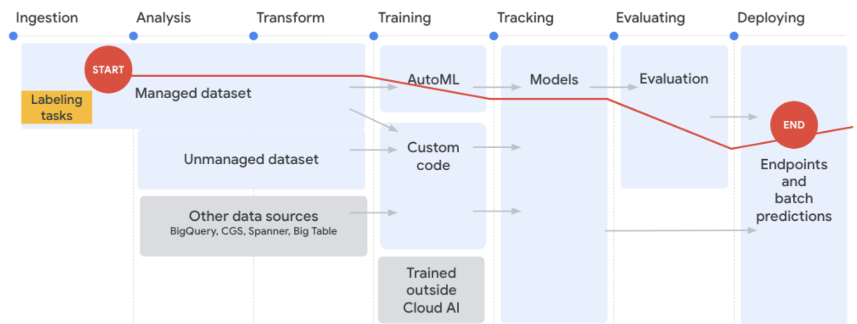


Moving from experimentation to production requires packaging, deploying, and monitoring your model.

Vertex AI provides unified definitions/implementations of four concepts:

1. A dataset can be structured or unstructured. It has managed metadata including annotations, and can be stored anywhere on Google Cloud.
2. A training pipeline is a series of containerized steps that can be used to train an ML model using a dataset. The containerization helps with generalization, reproducibility, and auditability.
3. A model is an ML model with metadata that was built with a Training Pipeline or directly loaded (only if it is in a compatible format).

4. An endpoint can be invoked by users for online predictions and explanations. It can have one or more models, and one or more versions of those models, with disambiguation carried out based on the request.



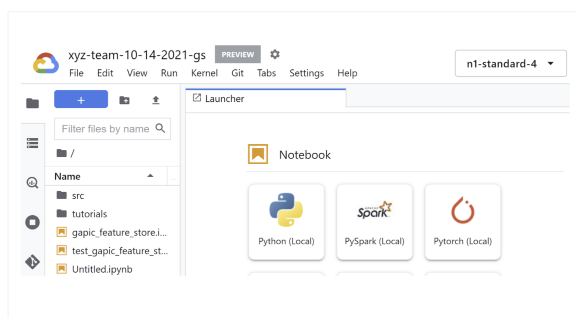
Vertex AI provides a unified set of APIs for the ML lifecycle. Diagram courtesy Henry Tappen and Brian Kobashikawa

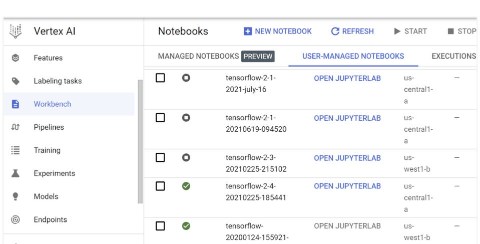
Tools to interact with Vertex AI include client libraries, VM images, REST API, and containers.

Module 4: Machine Learning Development with Vertex Notebooks

Vertex AI Workbench provides two Jupyter notebook-based options for your data science workflow: managed notebooks and user-managed notebooks.

Managed notebooks instances are Google-managed environments with integrations and features that help you set up and work in an end-to-end notebook-based production environment. Managed notebooks let you access your data without leaving the JupyterLab interface.





User-managed notebooks are Deep Learning VM Images instances that are heavily customizable and are ideal if you need a lot of control over your environment. User-managed notebooks can be a good choice for users who require extensive customization or who need a lot of control over their environment.

Module 5: Best Practices for Implementing Machine Learning on Vertex AI

Google has recommended best practices for:

- Machine learning development
- Data preprocessing
- Machine learning environment setup
- Model deployment and serving
- Model monitoring
- Vertex AI Pipeline
- Artifact organization

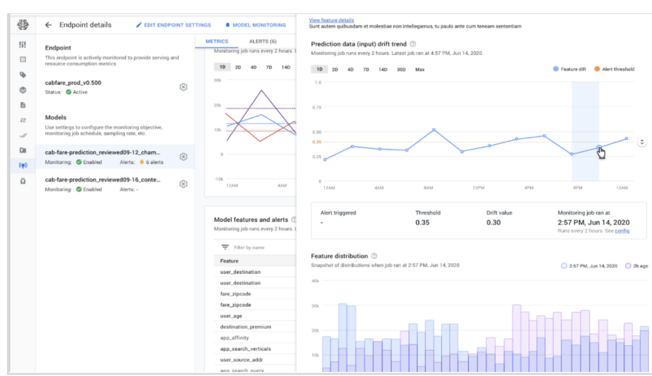
Machine learning development best practices

Preparing and storing data:

- Regardless of your data's origin, extract data from the source systems and convert to the format and storage (separate from the operational source) optimized for ML training.
- Structured data:
  - Store tabular data in BigQuery
    - Use Vertex AI Feature Store with structured data
- Unstructured data:
  - Store image, video, audio, and unstructured data in Cloud Storage
    - Use Vertex Data Labeling to provide labels

Training a model:

- For small datasets, train a model within the Notebooks instance.
- For large datasets, distributed training, or scheduled training, use the Vertex training service.
- Vertex AI Training provides a set of pre-built algorithms that allows users to bring their custom code to train models.

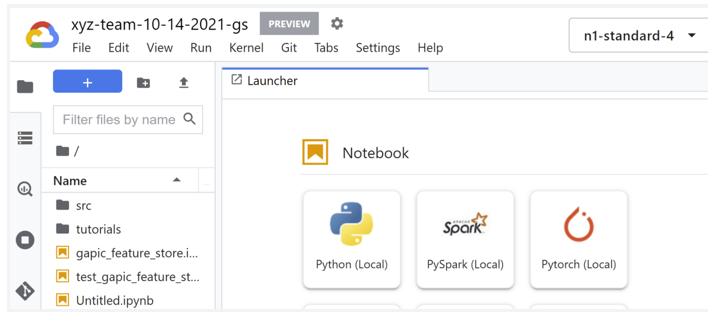


Best practices for **Explainable AI**:

- 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.

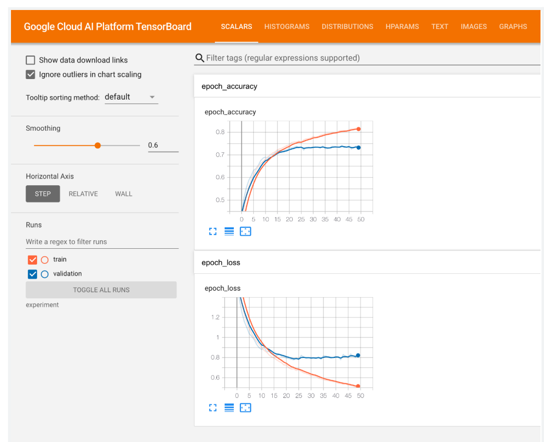
Best practices for using **Workbench Notebooks**:

- 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**:

- 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

- Use BigQuery to process tabular data and use Dataflow to process unstructured data.

ML environment setup best practices

- Use Notebooks for experimentation and development.
- Create a Notebooks instance for each team member.
- Help secure PII in Notebooks.
- Store prepared data and your model in the same project.
- Optimize performance and cost.
- Use Vertex SDK for Python.

Module 6: Responsible AI development

Unconscious biases exist in data. They can exist in both collecting and labeling data. These biases will then be reflected in your ML, and affect the entire pipeline.

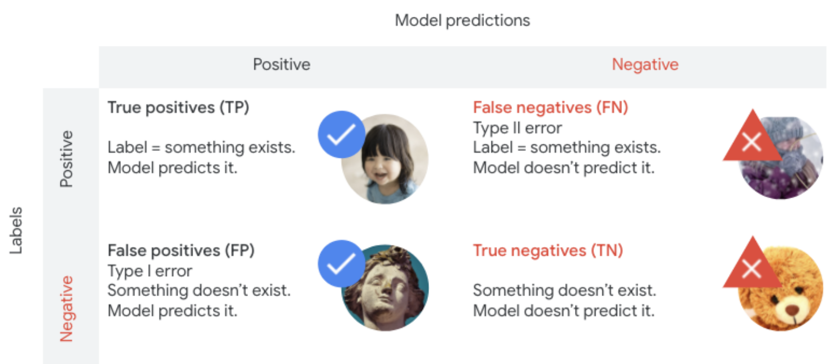
A checklist for bias-related issues:

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

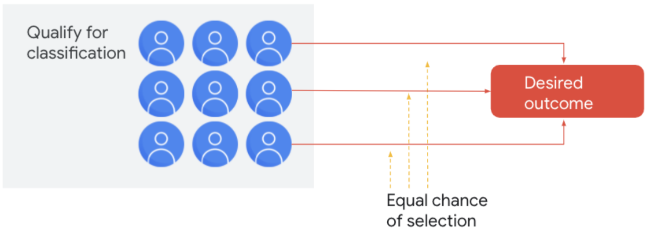
Use the **What-If tool** to help you diagnose fairness issues in your data, in your labels, and in the effects of predictions.

A **confusion matrix** helps in understanding inclusion and how to introduce inclusion across different kinds of groups across your data.

- *True positive:* When the label says something exists and the model says it exists.
- *True negative:* When the label says something doesn't exist and the model doesn't predict it.
- *False negative:* When the label says something exists and the model doesn't predict it.
- *False positive:* When the label says something doesn't exist, but the model says it exists.



Equality of opportunity is an approach that strives to give individuals an equal chance of the desired outcome. Incorporating this approach into your machine learning system gives you a way to scrutinize your model in order to discover possible areas of concerns. Once you identify opportunities for improvements, you can now make the necessary adjustments to strike a better tradeoff between accuracy and non-discrimination—which, in turn, could make your machine learning model more inclusive.



The **Facets** tool can help you make machine learning more inclusive.

- Facets Overview automatically gives you a quick understanding of the distribution of values across the features of their datasets.
- Facets Dive provides an easy-to-customize, intuitive interface for exploring the relationship between the data points across the different features of a dataset.