WHAT IS KUBEFLOW? MACHINE LEARNING BASICS WITH KUBEFLOW



Photographers have Photoshop, architects have AutoCAD, and machine learning engineers have Kubeflow. Every artist benefits greatly by having software made specific to their discipline, and ML engineers are no different.

In this article, we'll explore how Kubeflow works and when to use it.

What is Kubeflow?

Kubeflow is a free, open-source machine learning platform that makes it possible for machine learning pipelines to orchestrate complicated workflows running on Kubernetes. Kubeflow was first released in 2017, built by developers from Google, Cisco, IBM, Red Hat, and more. The 1.0 version was officially released this year.

The Kubeflow suite of tools help an engineer:

- Build machine learning models
- Analyze a model's performance
- Tune hyper-parameters
- Version different models
- Manage compute power
- Deploy models to production

All of this can be done on any cloud provider and anywhere there is Kubernetes.

The "Kube" in Kubeflow comes from the server orchestration tool Kubernetes. Kubeflow runs on Kubernetes clusters either locally or in the cloud, easily enabling the power of training machine learning models on multiple computers, accelerating the time to train a model. "Flow" was given to signal that Kubeflow sits among other workflow schedulers like ML Flow, FBLearner Flow, and Airflow.

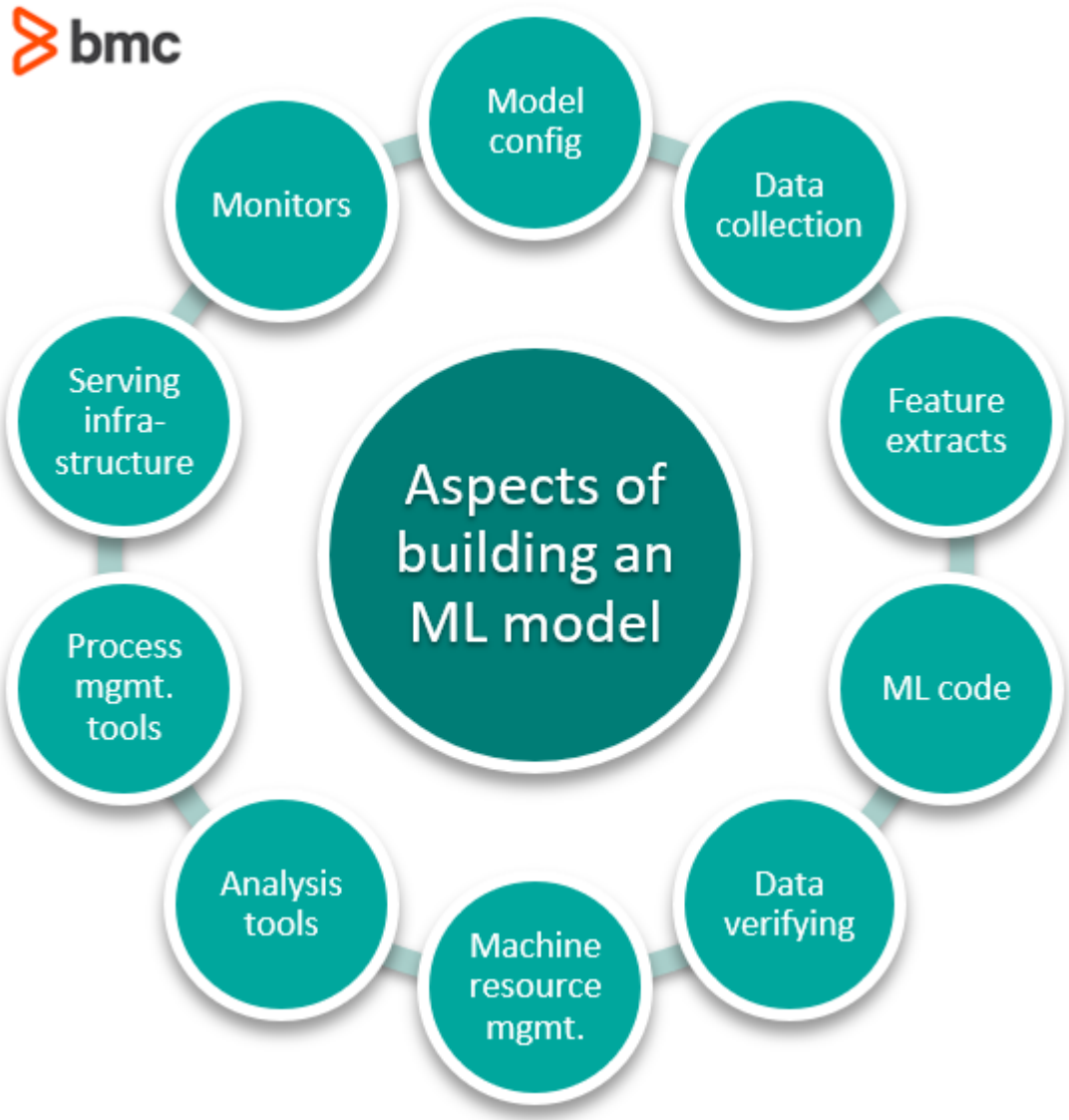
Kubeflow and machine learning

Kubeflow is a staple for MLOps teams. It helps organize projects, leverage cloud computing, and lets a ML Engineer really dive in and build the best models they can. Kubeflow Community Manager Thea Lamkin says,

> *"Kubeflow's goal is to make it easy for machine learning (ML) engineers and data scientists to leverage cloud assets (public or on-premise) for ML workloads."*

As the Kubeflow team regularly points out, there is a lot that goes into making a machine learning model happen that goes well beyond just understanding statistics and neural nets. It is not just writing ML code. Kubeflow helps manage all of it in a single piece of software.

For example, **all** these aspects are a part of building a machine learning model:



"Using Kubeflow, one of my engineers went from taking 8 weeks to build a production-ready model, to creating one in a single day." – A Spotify ML Engineer

Kubeflow 1.0 released

Kubeflow officially released its 1.0 version in March 2020. The team's progress can be seen on their GitHub Kanban board.

Kubeflow use cases

Kubeflow can be used for:

- Great documentation
- Multi-cloud framework
- Monitoring tools
- Workflow management
- Model deployment

Let's look at each in more detail.

Great documentation

First, open source projects do not have to have documentation. If its creators want others to use it, then documentation helps tell people what their software library does and how to use it. Documentation is useful in the open-source community, where projects are often passed via word-of-mouth and picked up and used through self-directed curiosity. If the package can be used, they stick around. If the package can't be used, then developers move on.

The writing in Kubeflow's documentation is of a quality that beginning projects usually forgo. The standard of excellence the writers maintain is wondrous and, as it would, does its purpose very well.

The Kubeflow documentation is easy to follow and understand. Whether a developer is just starting out, getting their first pipeline setup, or getting more involved, trying to rollback their production model to a previously trained model, the Kubeflow documentation is easy to read and search through.

Multi-cloud framework

Kubeflow is an open source tool. It was developed, initially as an internal tool at Google to use Google's Tensorflow Extended on Kubernetes. Too early in its development, it has yet to be a funded project of Google's, but its creators, contributors, and the ML engineers using it already reap from the value it provides and continue to expand the project.

Since its creation, Kubeflow runs anywhere there is Kubernetes and allows for any model architectures. For use on a cloud provider, Kubeflow has the documentation for getting setup:

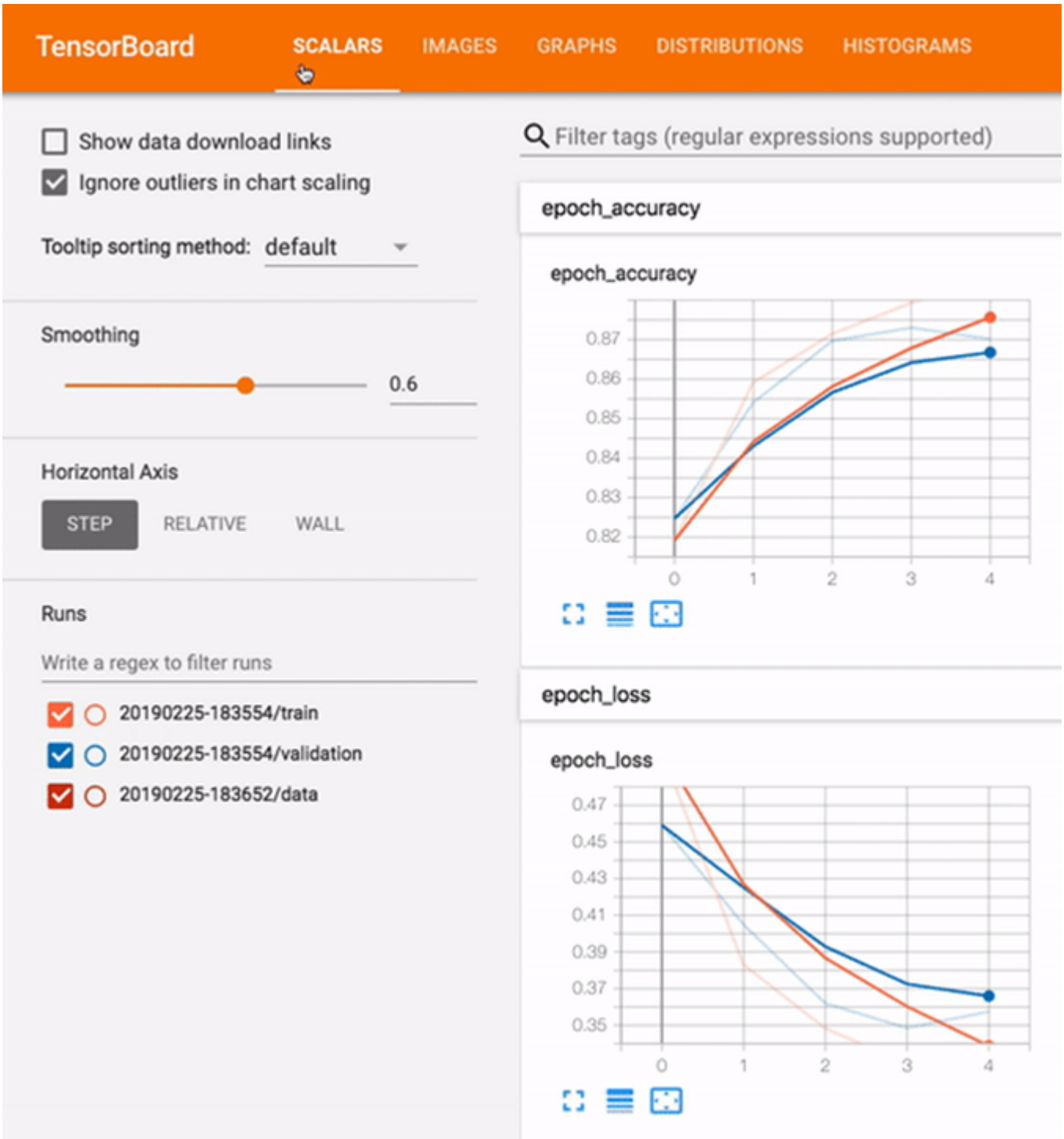
- Kubeflow on AWS
- Kubeflow on Azure
- Kubeflow on GCP
- Kubeflow on IBM Cloud

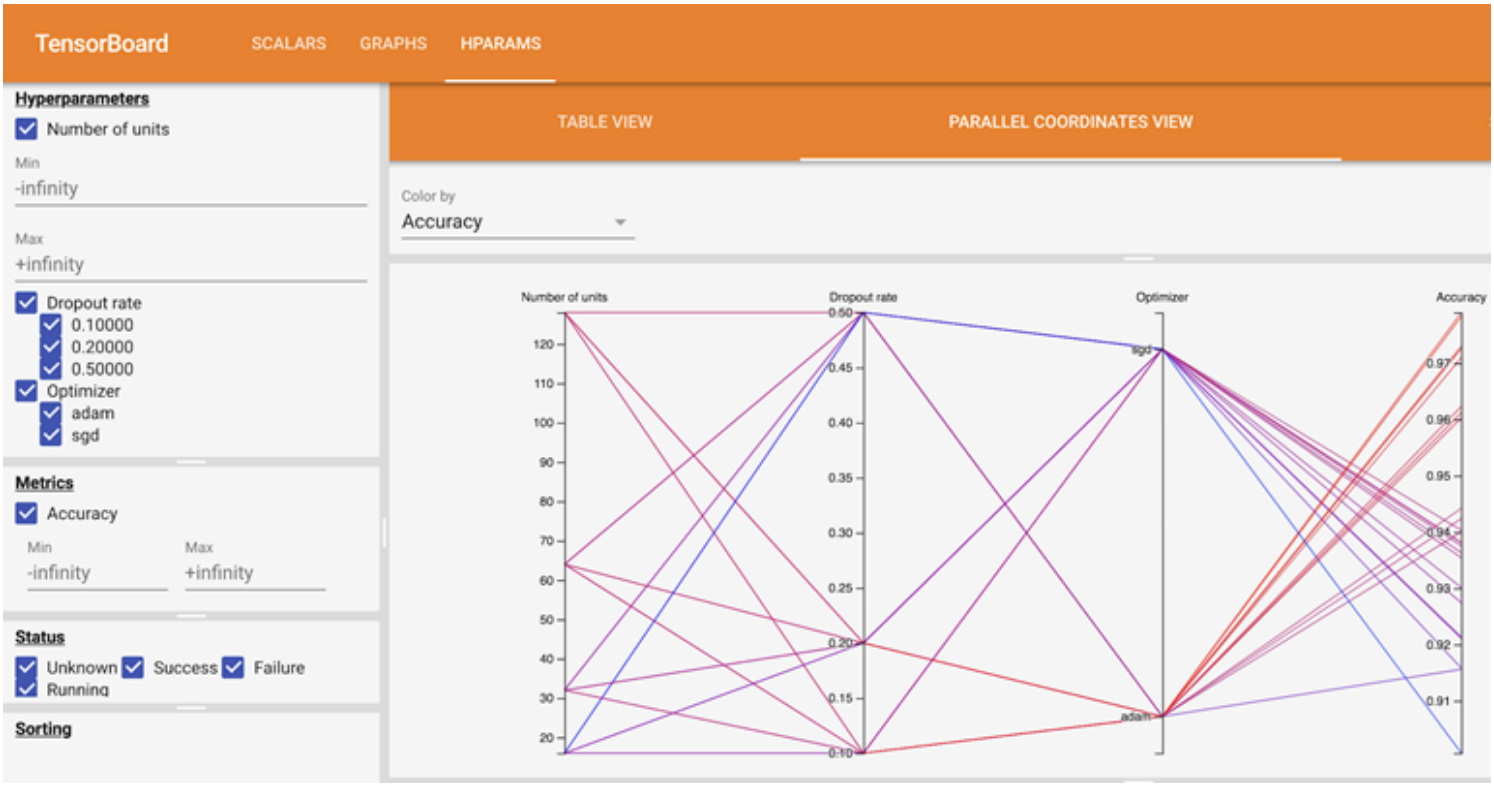
- Kubeflow on OpenShift

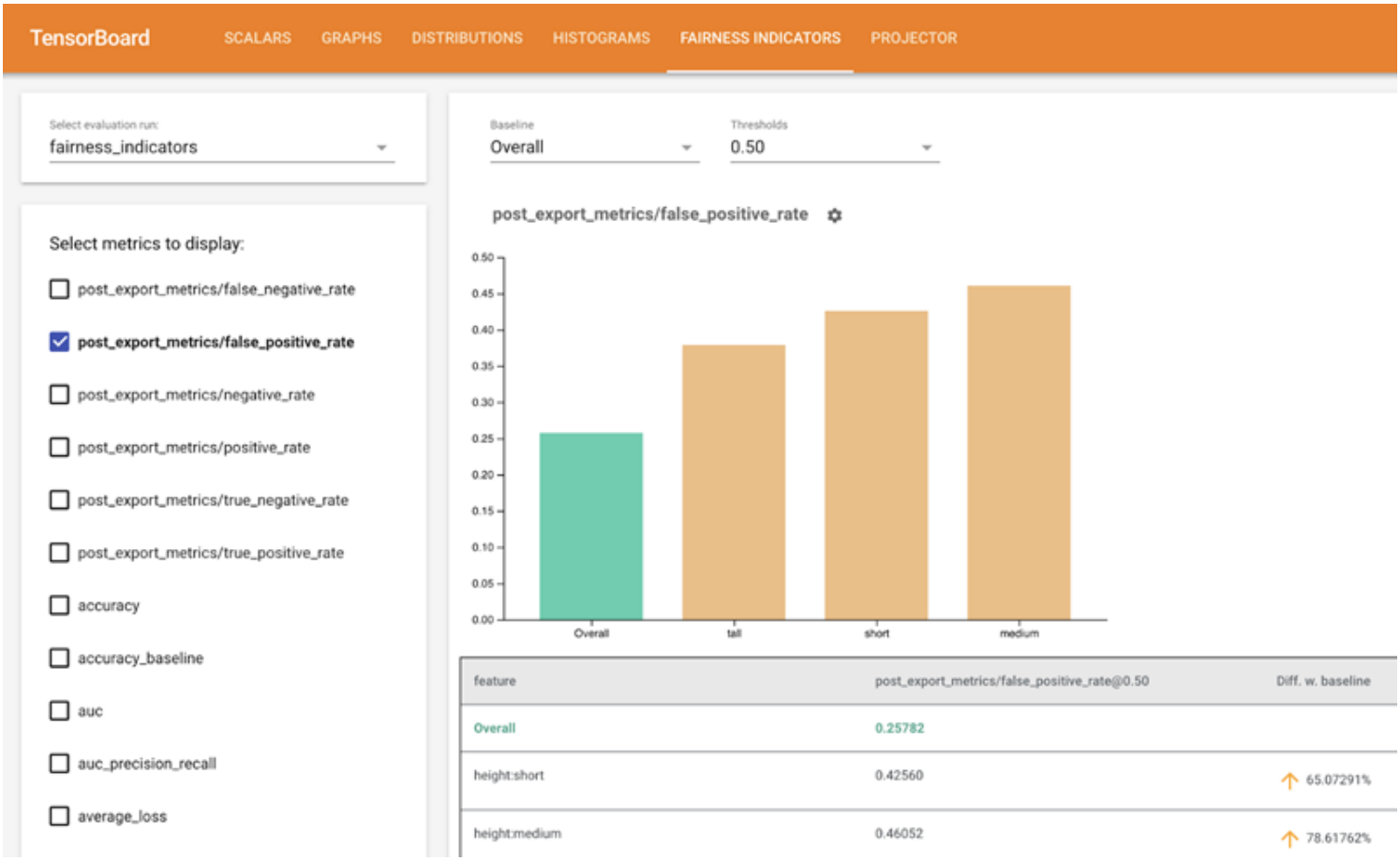
Monitoring tools

Kubeflow integrates Tensorboard into its service. Tensorboard provides the tooling to visualize the machine learning training process. As learned in Chaos Engineering, observability helps prevent failure. The ability to monitor the training process helps MLOps teams to:

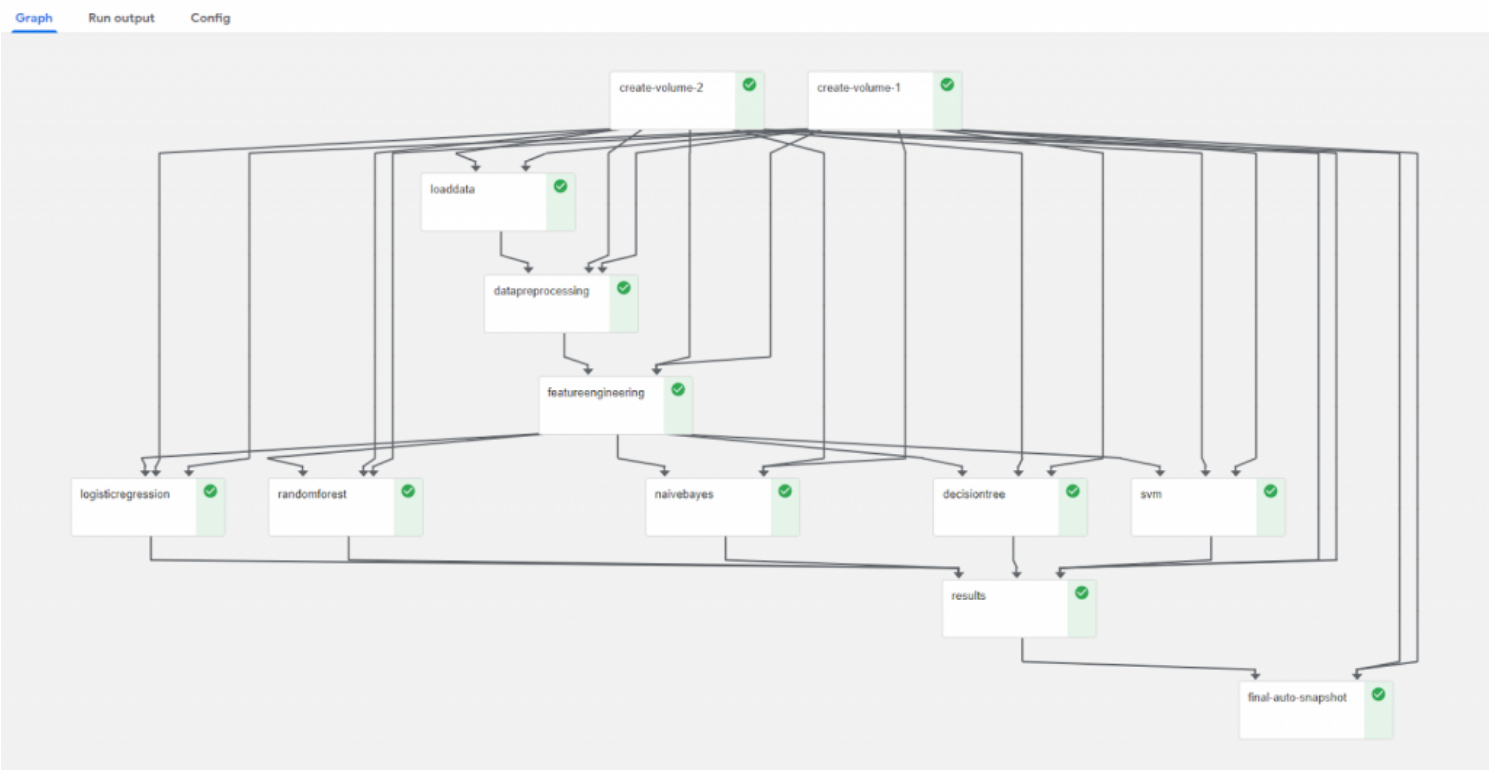
1. Fine-tune the model's parameters.
2. Save resources by stopping the training if it is seen the model does not work.
3. Accelerate building time through rapid iteration.







Workflow management



This chart indicates a completed job in Kubeflow

In a similar vein as monitoring, the workflow management tool allows the engineers to manage their own time. Like rendering exists for video editors, for ML engineers, their waiting period comes when moving data around and when training models. Kubeflow shows where their processing job is at in its list of tasks, and when jobs are finished Users have even utilized a Slack message to notify themselves or their team when their pipeline has completed.

Because of the flow tool, the engineer has a much better understanding of the *timing* involved to complete their machine learning jobs, enabling them to juggle multiple tasks at once.

Model deployment

Finally, no model is complete if it sits on a shelf or unused in a Jupyter notebook. Models need to be deployed, to be inferencing.

There are several ways to deploy a model from Kubeflow, first starting with the custom-built KFserving tool. Along with many others, KFServing runs on-prem and in the cloud. It runs using several machine learning frameworks, including:

- TensorFlow
- PyTorch
- SciKit Learn

Because it uses Kubernetes as its resource foundation, resources will be autoscaled as the model gets more use.

Additional resources

For more on this and related topics, browse these resources:

- BMC DevOps Blog

- BMC Machine Learning & Big Data Blog
- Machine Learning with TensorFlow & Keras, a multi-part tutorial
- Apache Spark Guide to Machine Learning
- Machine Learning: Hype vs Reality
- 5 Tribes of Machine Learning