Operationalizing Machine Learning Models

Target Complete Date	@December 16, 2022
Status	Completed
Reviewed	Reviewed
# Time to complete (Hours)	0.2
Type	Cloud Guru

Operationalize machine learning

> 💡 The process of deploying predictive models to a production environment, together with ongoing measurement, monitoring and improvement of those models. It is a set of best practices for managing a ML pipeline.

The key considerations are:

- Deployment
- Scoring
- Logging
- Monitoring
- Retraining

Deployment

How do you manage deployments?

- Separate infrastructure as your training

- Data preparation pipeline that is the same as the training pipeline
- Validations
- Design to run anywhere

Scoring

How can scoring improve the operation of models?

- Make model interfaces flexible
- Use standardized frameworks to score

Logging

How do you capture logs to identify problems with a model?

How do you capture logs to evaluate a model over time?

We should capture:

- The input request
- Output response
- Model version
- Data validation errors
- Output prediction
- Response times

Monitoring

How do you know how well the model is predicting?

How do you know how fast the trained models is serving predictions?

- Predictive performance
- Processing performance

- Could indicate issues on the underlying infrastructure that forms the pipeline

Retraining

How do you handle the decreasing predictive power of your model?

✏️ Data drift refers to the changes in distribution of the input data used to train your models. Data drift can occur due to many reasons such as changes in the underlying source generating system, external factors such as environmental changes. When data drift occurs the performance decreases.

✏️ Concept drift reflects changes in the underlying relationship between the input variables and the output prediction of the model. This can happen due to a variety of factors, including changes in user behavior, or external events.

- Retrain with same features
  - We can automate retraining based on a threshold.
- Rebuild with new features

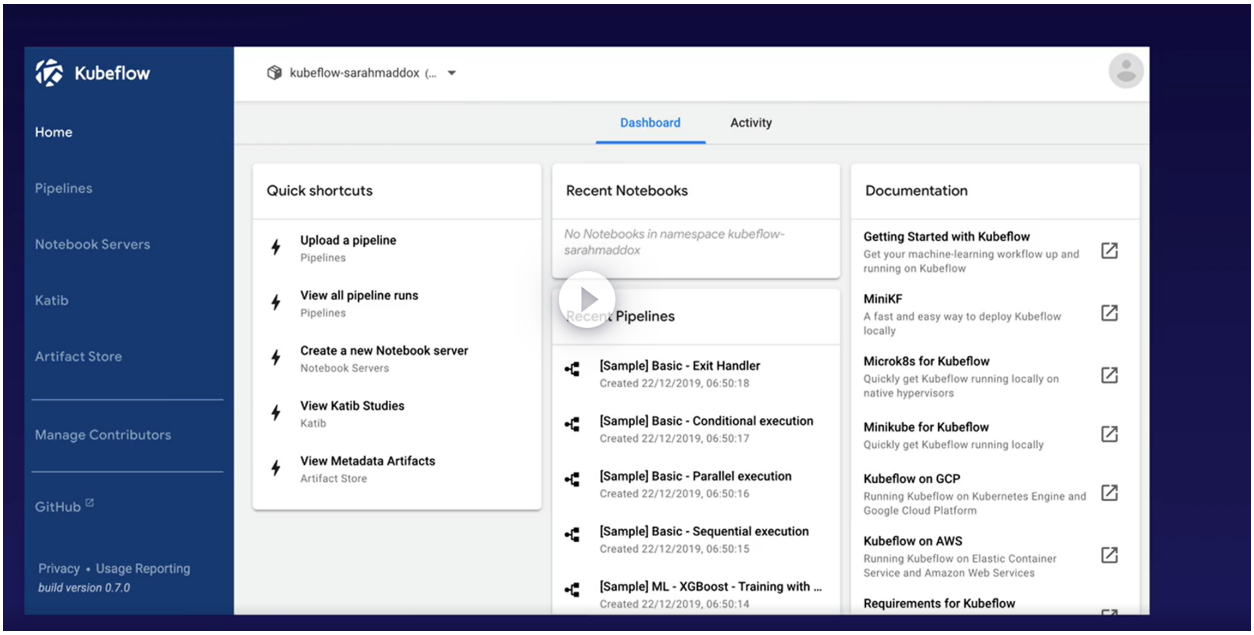
Kubeflow

💡 A complete platform for developing, deploying and operationalizing ML models. It is the ML toolkit for kubernetes in which it is built upon.

- Data modelling with Jupyter notebooks
- Tuning and training with tensorflow
- Model serving and monitoring

Interacting with Kubeflow

It has it’s own built in web UI, but also has it’s own APIs and Python SDKs.



Experimental phase

Developing your piplines

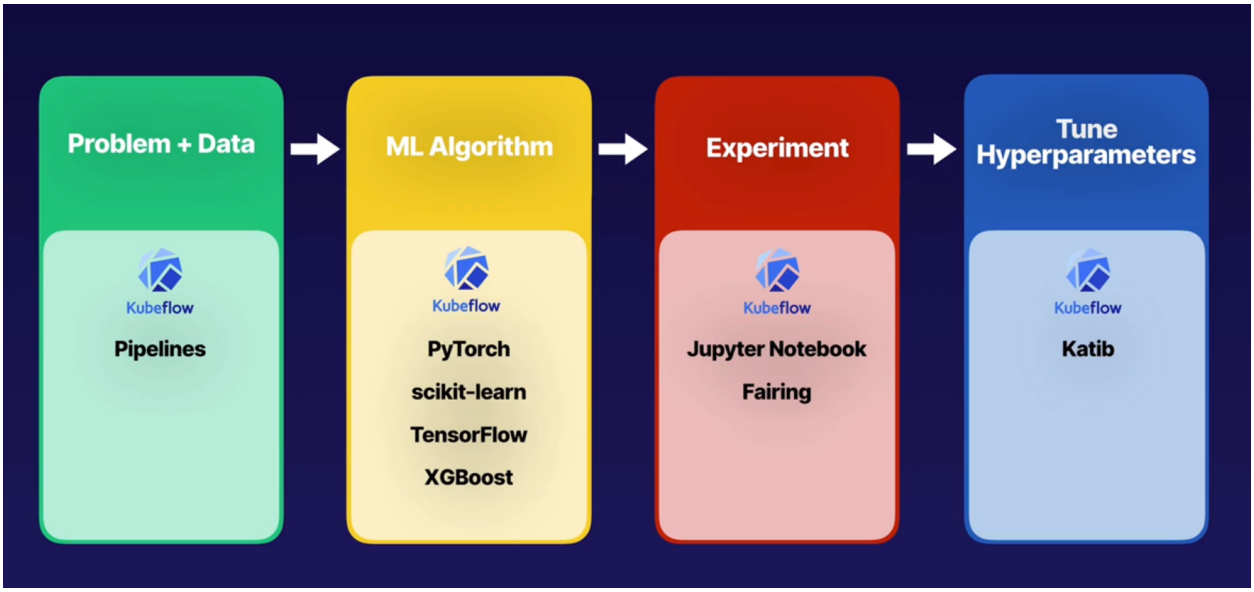
ML algorithm

Kubeflow provides the best ML libraries such as:

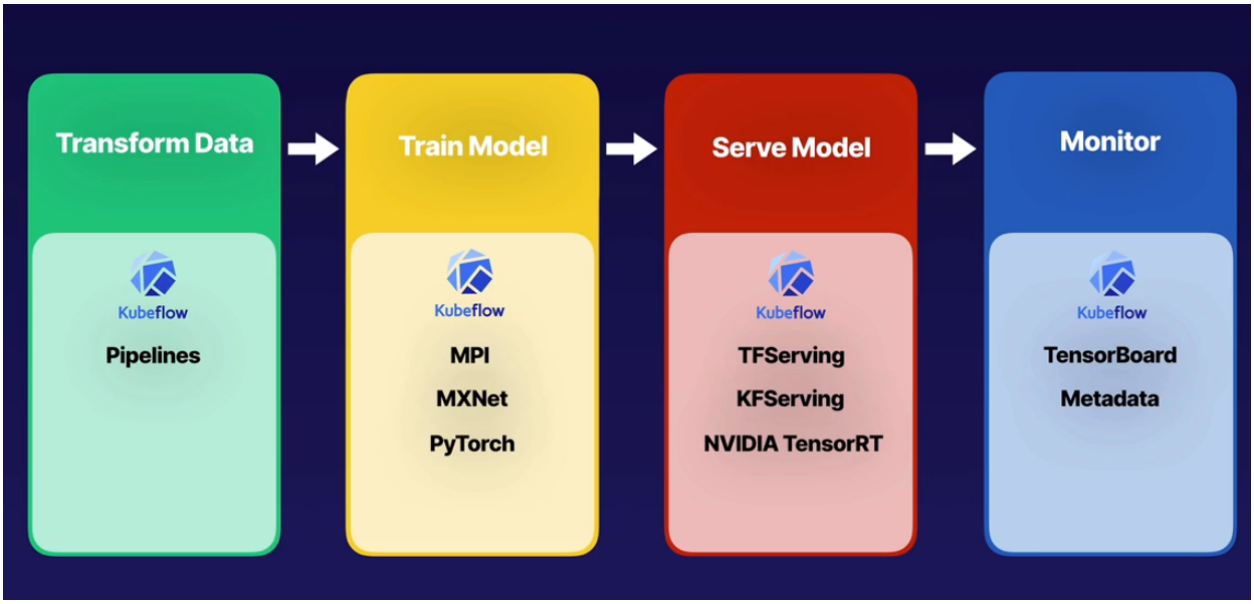
- PyTorch
- Scikit-learn
- Tensorflow
- XGBoost

Experiment with training data

Provides Juypter notebook and fairing.



Production phase



Pipelines

💡 A pipeline is a description of Machine Learning (ML) workflow, including all of the components in the workflow and how the components relate to each other in the form of. a graph.

💡 A pipeline component is a self-contrained set of user code, packaged as a container, that performs one step in the pipeline. For example data processing, data transformation and model training.

AI platform

💡 A suite of API services combined with other big data and ML products in GCP to make an operationalized ML pipeline.

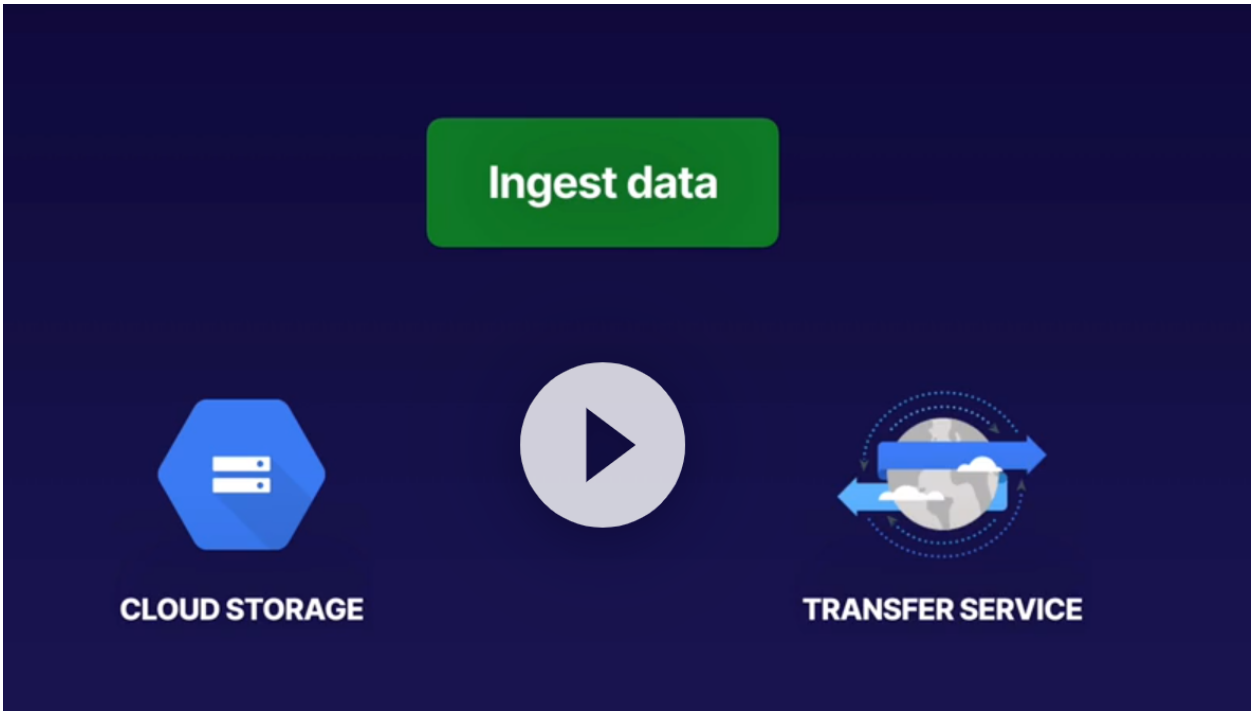
- Ingest data
- Prepare
- Preprocess
- Discover
- Develop
- Train
- Test
- Deploy

⚙️ Operational models can be deployed to VMs or K8s or managed tensforflow models.

AI hub

💡 Google’s hosted repo for AI plug and play AI components, including end to end AI piplines and out of the box algorithms.

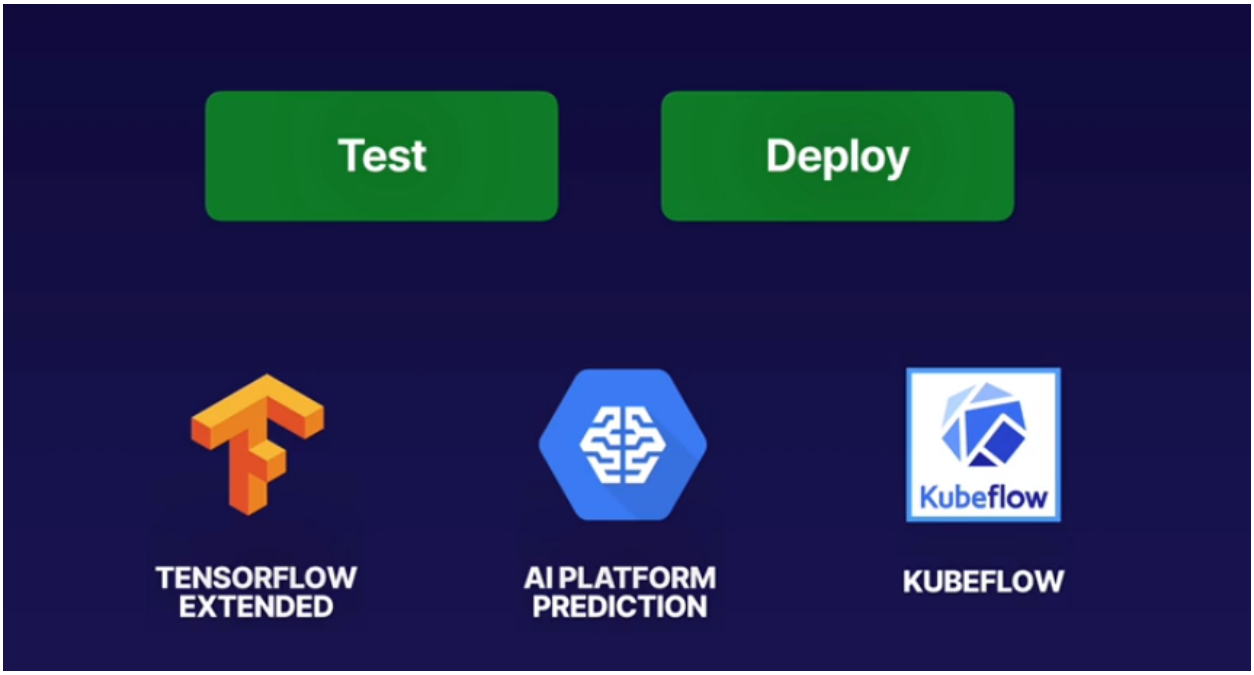
Ingest data



Prepare and preprocess

Develop and train models

Test and deploy models



Exam

AI platform is the easiest way to quickly run a ready to go model in a managed service.