



## Production ML Pipelines with Kubeflow

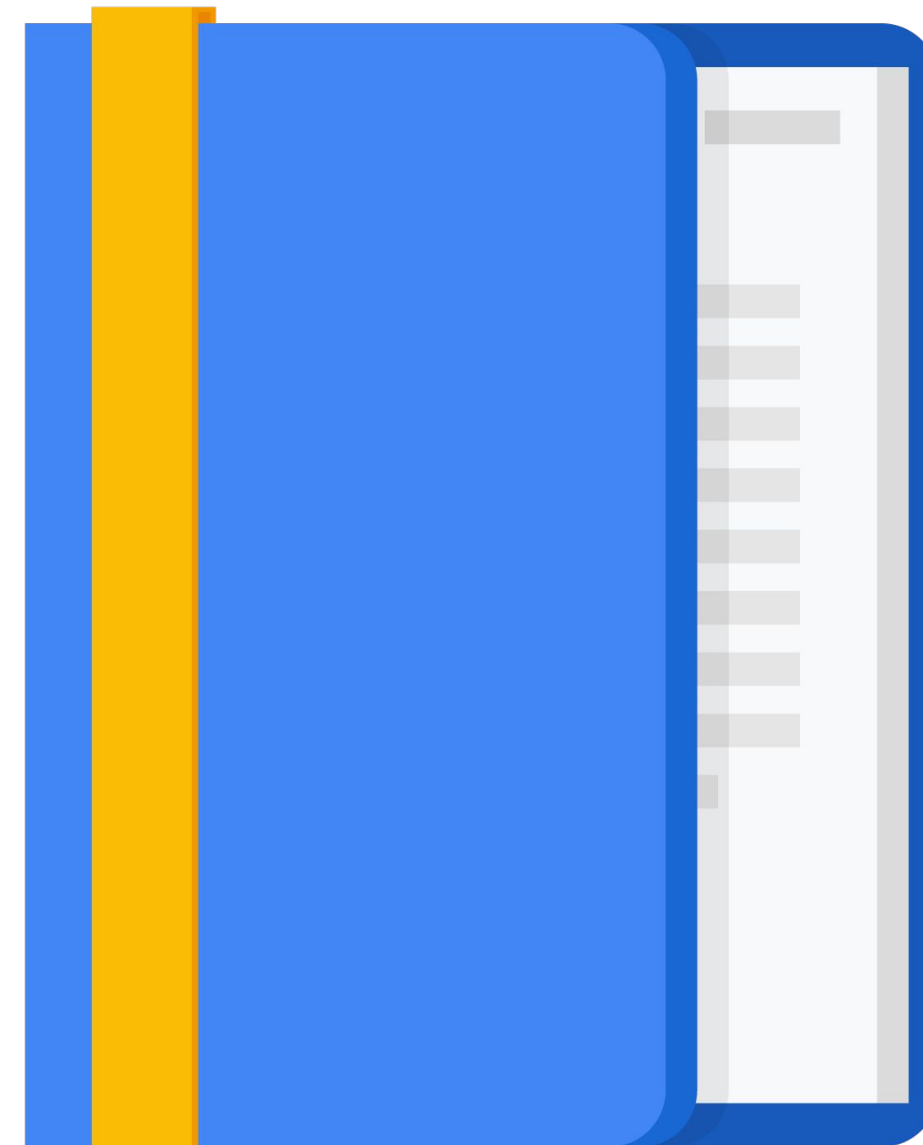
# Agenda

---

Ways to do ML on GCP

Kubeflow

AI Hub



# Create and deploy custom models with Kubeflow

## Build a Custom Model



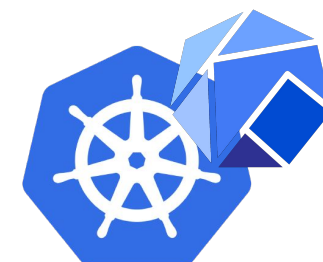
Cloud TPUs



Compute Engine



Cloud Dataproc



Kubernetes Engine



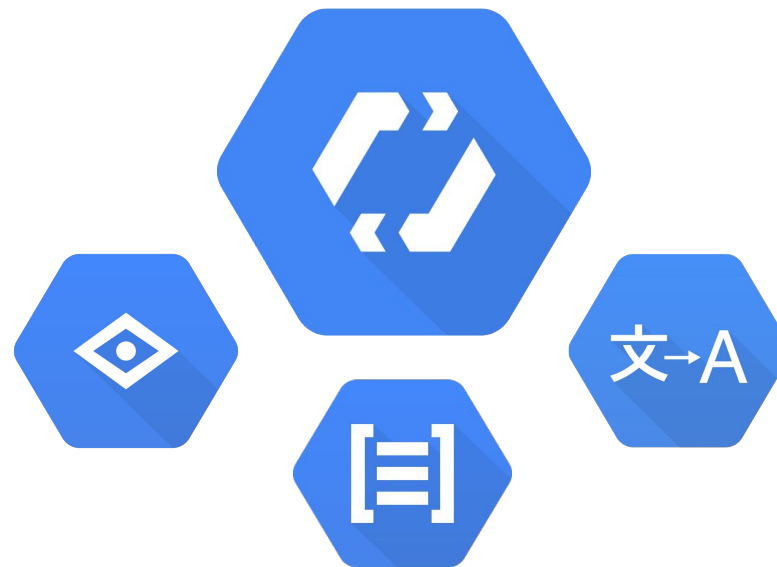
Cloud AI Platform



BigQuery ML

## Build Custom Model (codeless)

AutoML



## Call a Pretrained Model



Cloud Translation API



Cloud Vision API



Cloud Speech API



Cloud Video Intelligence API



Data Loss Prevention API



Cloud Speech Synthesis API

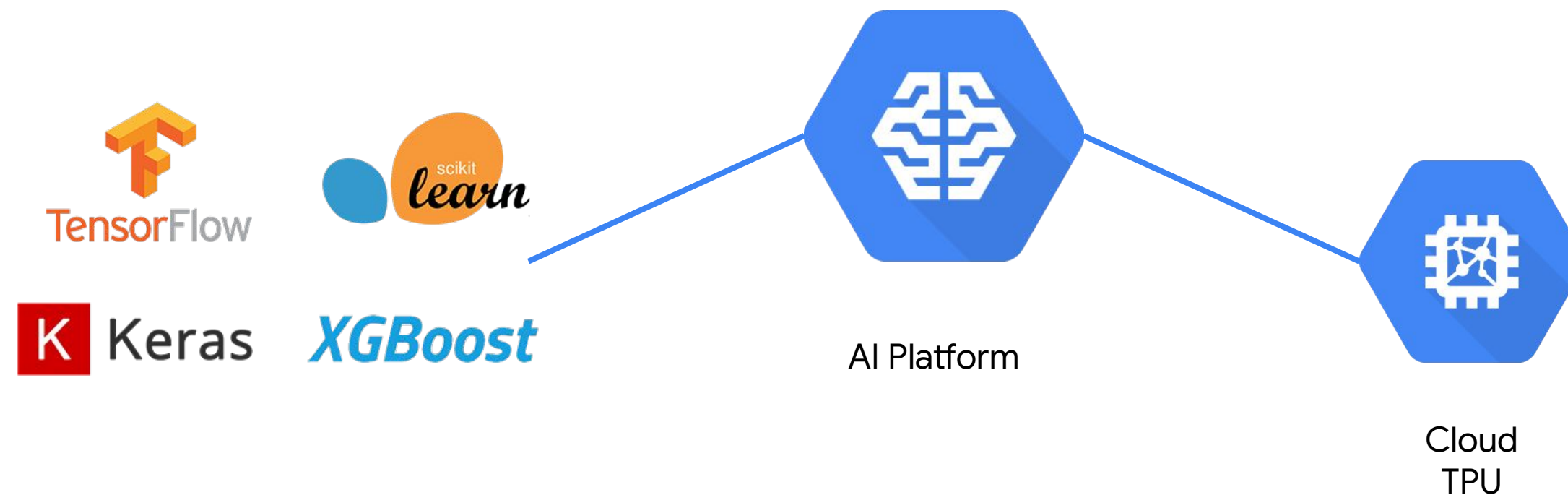


Cloud Natural Language API



Dialogflow

# Cloud AI Platform is a fully managed service for custom machine learning models



- Scales to production
- Batching and distribution of model training
- Performs transformations on input data
- Hyper-parameter tuning
- Host and autoscale predictions
- Serverless - self-tuning - manages overhead

In this course, we don't cover writing TensorFlow models, only ways to operationalize them

- [Intro to ML on GCP Specialization on Coursera](#)

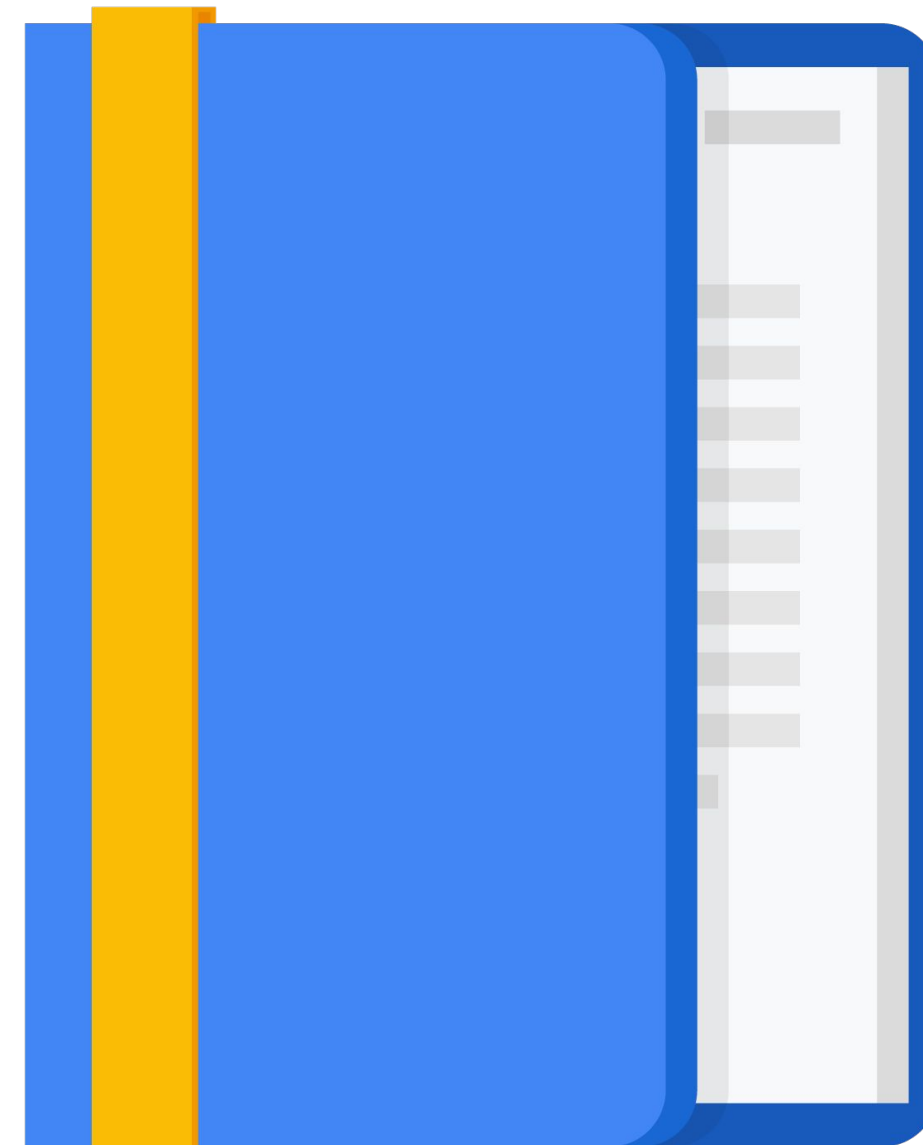
# Agenda

---

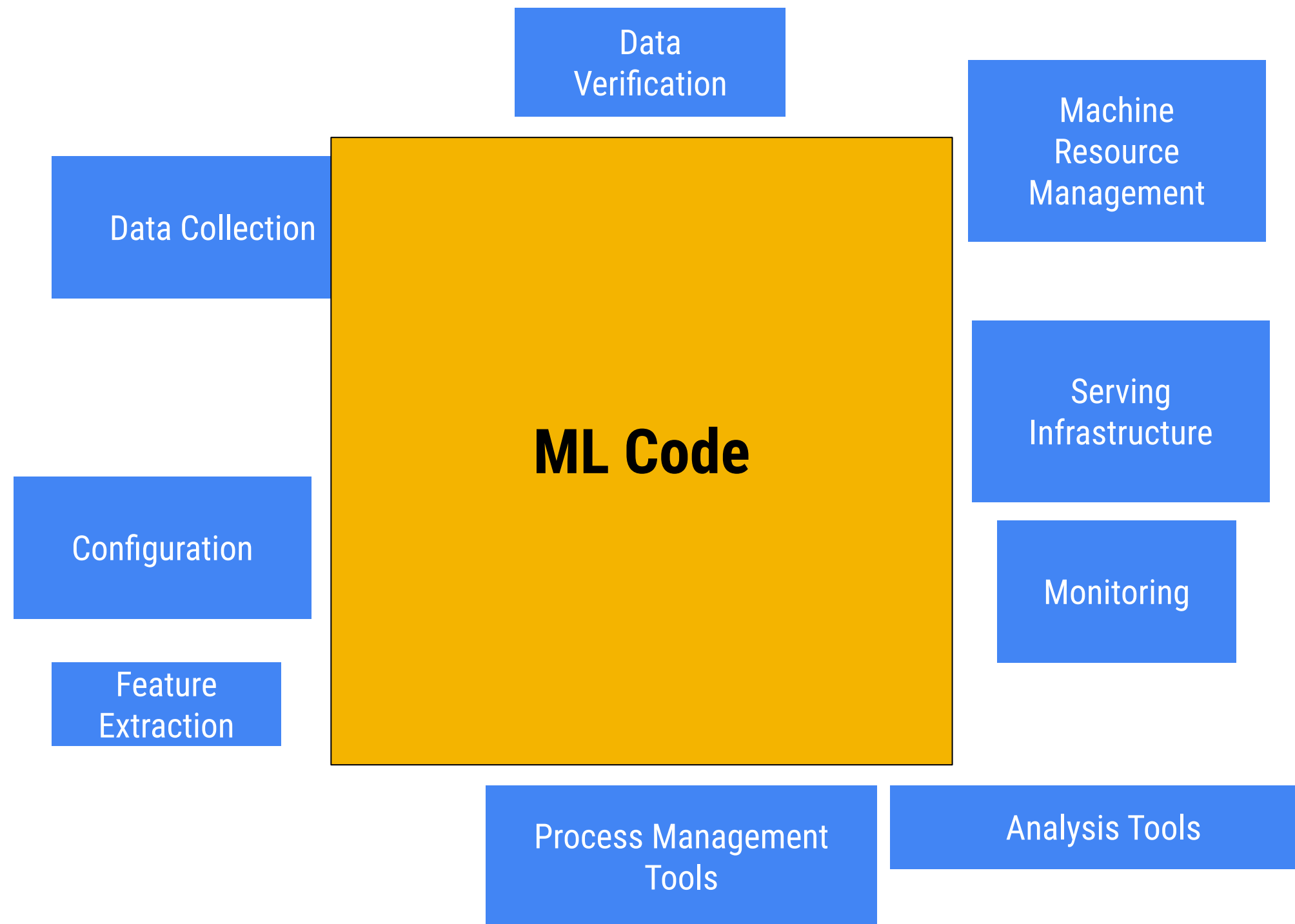
Ways to do ML on GCP

Kubeflow

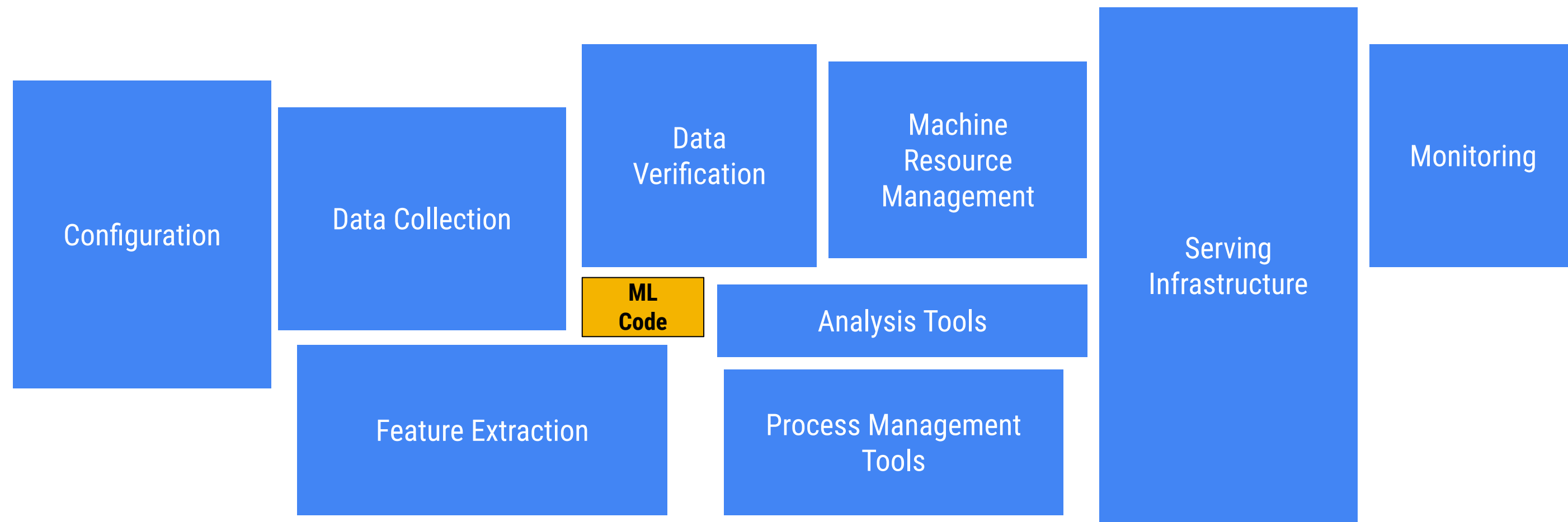
AI Hub



# Perception: ML products are mostly about ML



# Reality: ML Requires lots of DevOps



Source: [Sculley et al.: Hidden Technical Debt in Machine Learning Systems](#)



# Kubeflow provides a platform for building ML products

- Leverage containers and Kubernetes to solve the challenges of building ML products
- Kubeflow = Cloud Native, multi-cloud solution for ML.
- Kubeflow provides a platform for composable, portable and scalable ML pipelines
- If you have a Kubernetes conformant cluster, you can run Kubeflow

# Kubernetes is a great platform for ML

- Containers
- Scaling built in
- Unified architecture
- Easy to integrate building blocks
  - ML APIs
  - Dataflow
- Lots of options for CI/CD
- Portability
  - Dev, On-Prem, Multi-cloud: same stack



# Kubeflow Pipelines enable:



**ML Workflow  
Orchestration**



**Share, Re-use  
& Compose**



**Rapid Reliable  
Experimentation**

# What constitutes a Kubeflow Pipeline

## Containerized implementations of ML Tasks

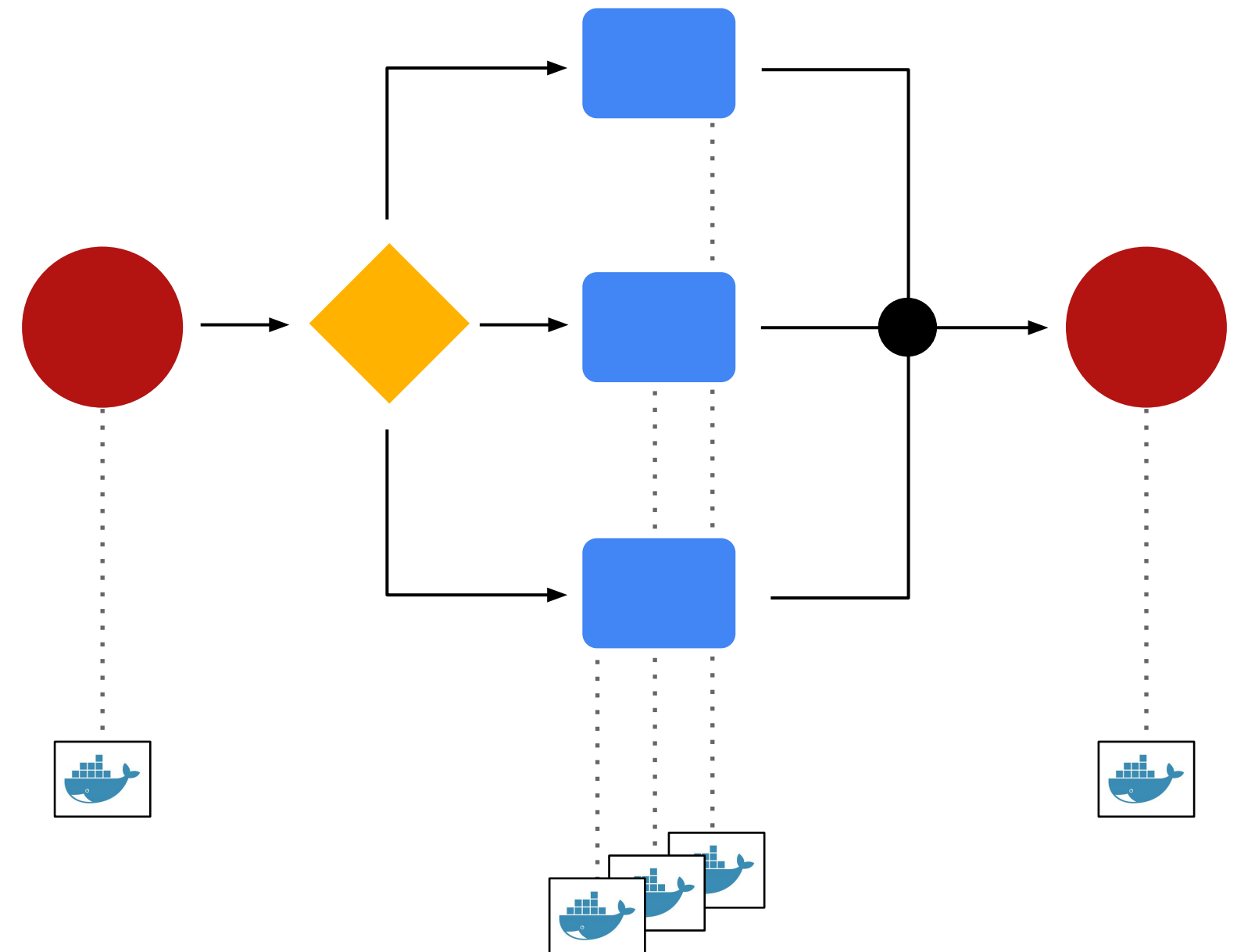
- Containers provide portability, repeatability and encapsulation
- A task can be single node or \*distributed\*
- A containerized task can invoke other services like CMLE, Dataflow or Dataproc

## Specification of the sequence of steps

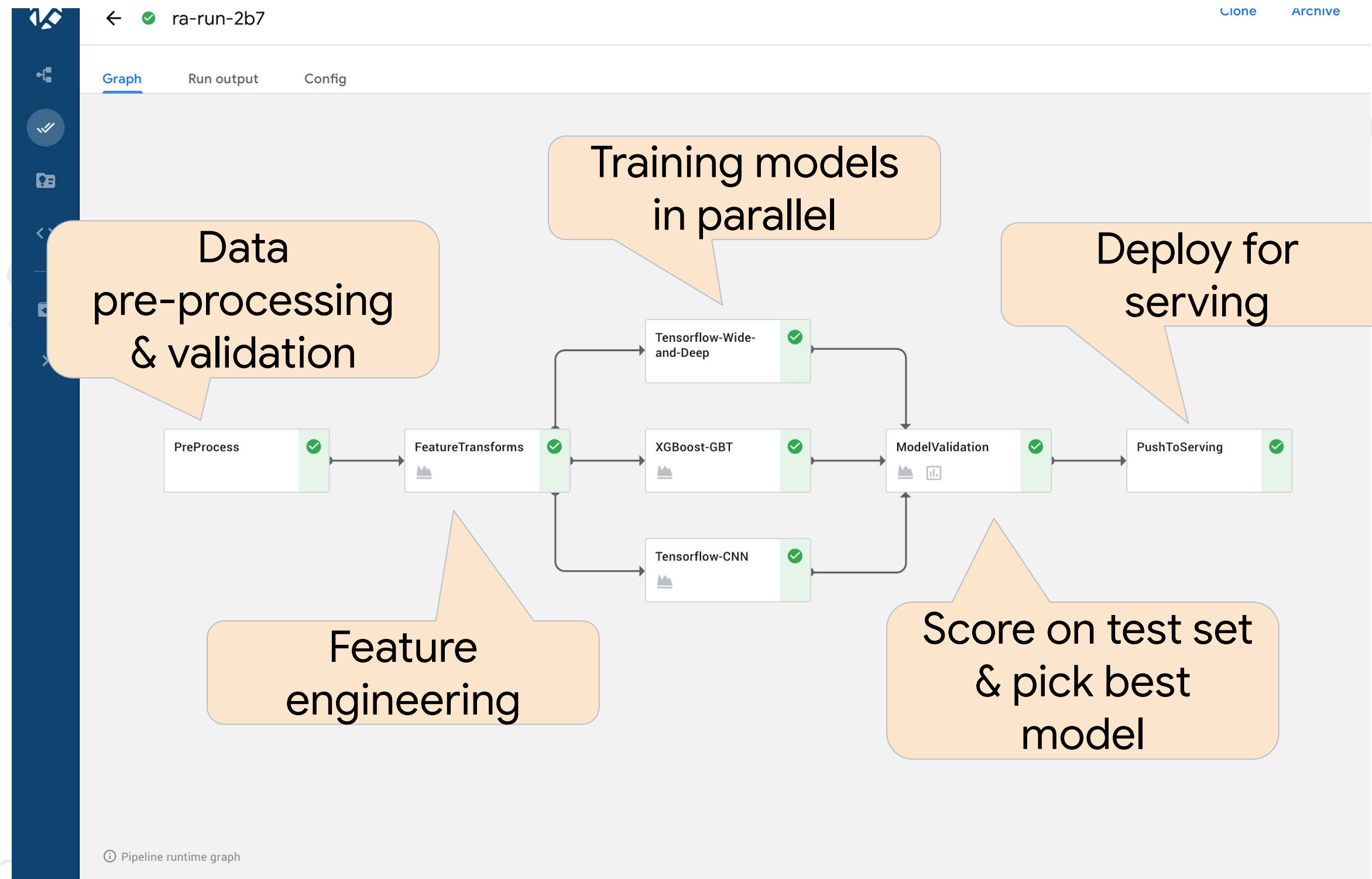
- Specified via Python SDK

## Input Parameters

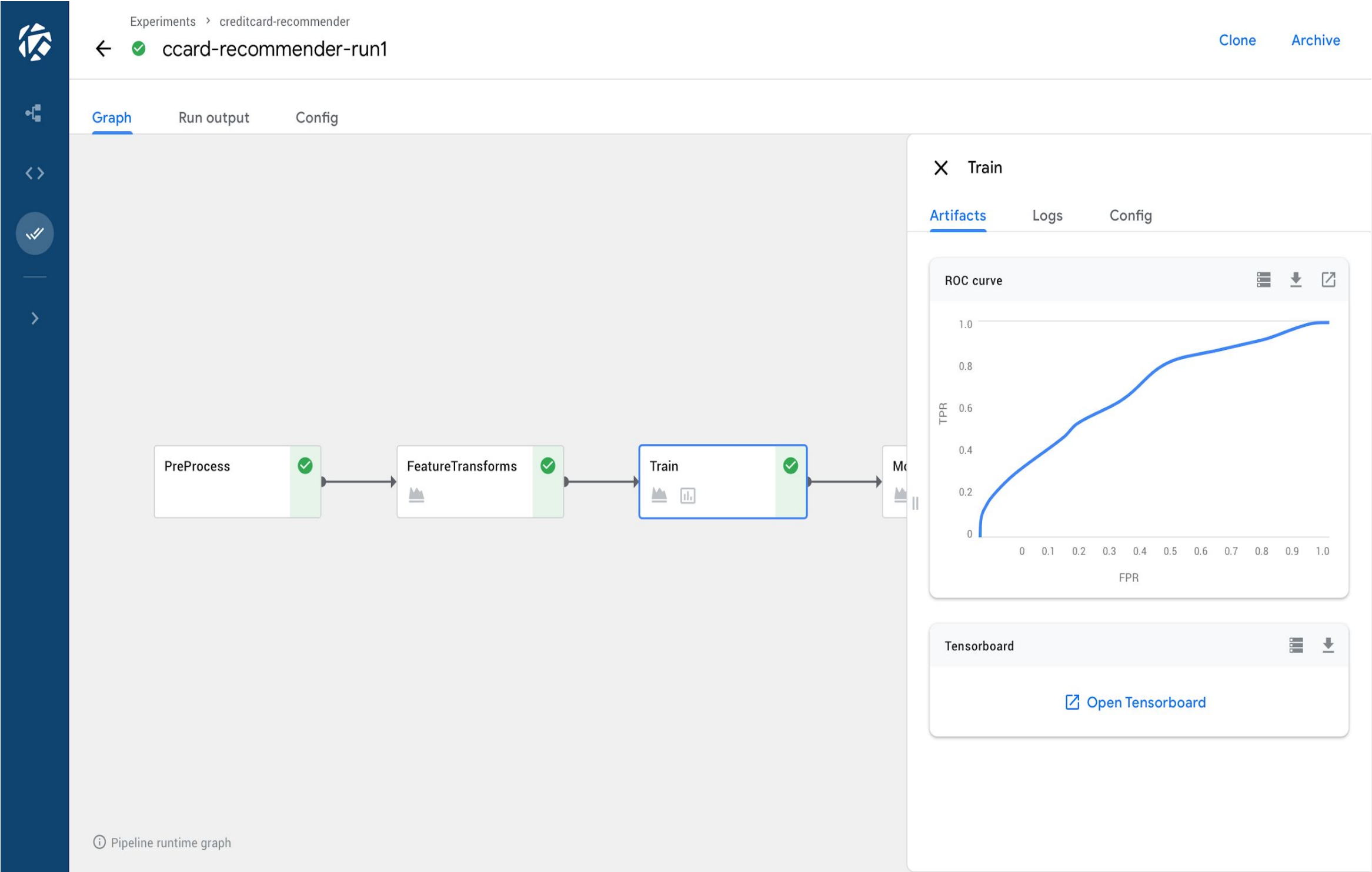
- A “Job” = Pipeline invoked w/ specific parameters



# Visual depiction of pipeline topology



# Rich visualization of metrics



# View all configs, inputs and outputs

Experiments > Product Image Classification

←

✔

Simple XGBoost Classifier

Graph

Config

Run details

Status	Succeeded
Description	
Created at	11/25/2018, 12:56:44 PM
Started at	11/25/2018, 12:56:44 PM
Finished at	11/25/2018, 1:16:37 PM
Duration	0:19:53

Run parameters

output	gs://mlpipelines
project	foo2thebar
region	us-central1
train-data	gs://ml-pipeline-playground/sfpd/train.csv
eval-data	gs://ml-pipeline-playground/sfpd/eval.csv
schema	gs://ml-pipeline-playground/sfpd/schema.json
target	resolution
rounds	200
workers	2
true-label	ACTION



# Author pipelines with an intuitive Python SDK

jupyter TFX Pipeline Notebook Last Checkpoint: 10 minutes ago (unsaved changes)

LogoutControl Panel

FileEditViewInsertCellKernelWidgetsHelp

Not TrustedPython 3

+

⌕

📄

⬆️

⬆️

▶️ Run

⬛

↺️

▶️

Code

⌨️

## Define a multi-step Pipeline

```
In [3]: import kfp.dsl as dsl

@dsl.pipeline(
    name='TFX Taxi Cab Classification Pipeline Example',
    description='Example pipeline that does classification with model analysis based on a public BigQuery dataset.'
)
def taxi_cab_classification(
    output,
    column_names=dsl.PipelineParam(
        name='column-names',
        value='gs://ml-pipeline-playground/tfx/taxi-cab-classification/column-names.json'),
    key_columns=dsl.PipelineParam(name='key-columns', value='trip_start_timestamp'),
    ...
    analyze_slice_column=dsl.PipelineParam(name='analyze-slice-column', value='trip_start_hour')):

    ...


    preprocess = dataflow_tf_transform_op(train, evaluation, schema, project, preprocess_mode, preprocess_module, trans
    training = tf_train_op(preprocess.output, schema, learning_rate, hidden_layer_size, steps, target, preprocess_modul
    analysis = dataflow_tf_model_analyze_op(training.output, evaluation, schema, project, analyze_mode, analyze_slice_c
    prediction = dataflow_tf_predict_op(evaluation, schema, target, training.output, predict_mode, project, prediction_
    deploy = kubeflow_deploy_op(training.output, tf_server_name)
```

## Submit the run

```
In [16]: import kfp
from kfp import compiler

compiler.Compiler().compile(taxi_cab_classification, 'tfx.tar.gz')

run = client.run_pipeline(exp.id, 'tfx', 'tfx.tar.gz',
                          params={'output': 'gs://bradley-playground',
                                  'project': 'bradley-playground'})
```

 Google Cloud



# Package & share pipelines as zip files

- Upload and execute pipelines via UI (in addition to API/SDK)
- Pipeline steps can be authored as reusable components

Run details

Pipeline\*  
xgboost training - confusion matrix [Choose](#)

Run name\*  
product-recommender-model

Description (optional)  
Train XGB model for product recommendation application.

Run parameters

Specify parameters required by the pipeline

output

project

region  
us-central1

train-data  
gs://ml-pipeline-playground/sfpd/train.csv

eval-data  
gs://ml-pipeline-playground/sfpd/eval.csv

schema  
gs://ml-pipeline-playground/sfpd/schema.json

target  
resolution

rounds  
200

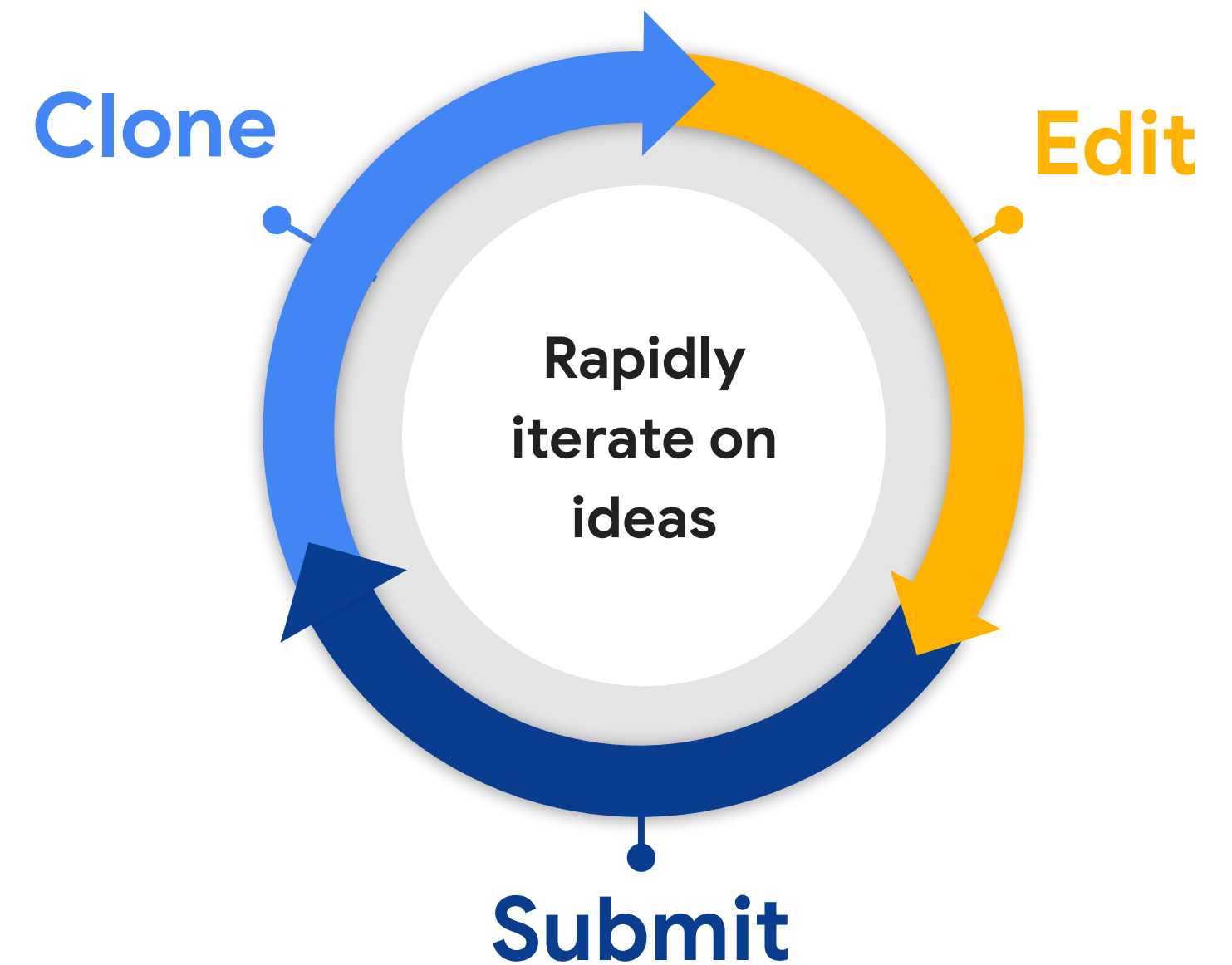
workers  
2

true-label  
ACTION

[Create](#) [Cancel](#)

# Rapid, Reliable, Experimentation

- Every run logged with all config params, inputs, outputs & metrics
- Easily search and find old runs
- Clone and re-run or modify



View all current  
and past runs in  
one place

Experiments

+ Create experiment

Compare runs

Archive

All experiments

All runs

Filter experiments

<input type="checkbox"/>	Experiment name	Last 5 runs	Created on ↑	Created by
<input type="checkbox"/>	▶ tfma-experiment	<div></div>	6:17 PM, Aug 24, 2018	John Doe
<input type="checkbox"/>	▶ xgboost-train	<div><div></div><div></div><div></div></div>	6:17 PM, Aug 24, 2018	John Doe
<input type="checkbox"/>	▶ promo-email	<div><div></div><div></div><div></div><div></div><div></div><div></div></div>	6:17 PM, Aug 24, 2018	Walter Fisher
<input type="checkbox"/>	▶ data-prep	<div><div></div></div>	6:17 PM, Aug 24, 2018	Walter Fisher
<input type="checkbox"/>	▶ tf-preprocessing	<div><div></div><div></div><div></div></div>	6:17 PM, Aug 24, 2018	John Doe
<input type="checkbox"/>	▶ tf-training	<div><div></div><div></div><div></div><div></div><div></div><div></div></div>	6:17 PM, Aug 24, 2018	Walter Fisher
<input type="checkbox"/>	▶ tf-serving	<div><div></div><div></div><div></div></div>	6:17 PM, Aug 24, 2018	Walter Fisher

Rows per page: 10

1–10 of 241

<

>

# Easy comparison and analysis of runs

Experiments

← image-classifier

Edit

Archive

Fastest run time

Slowest run time

1m 59s

3m 20s

View run

View run

All runs

Start new run

Start recurring run

Compare runs

Stop

Archive

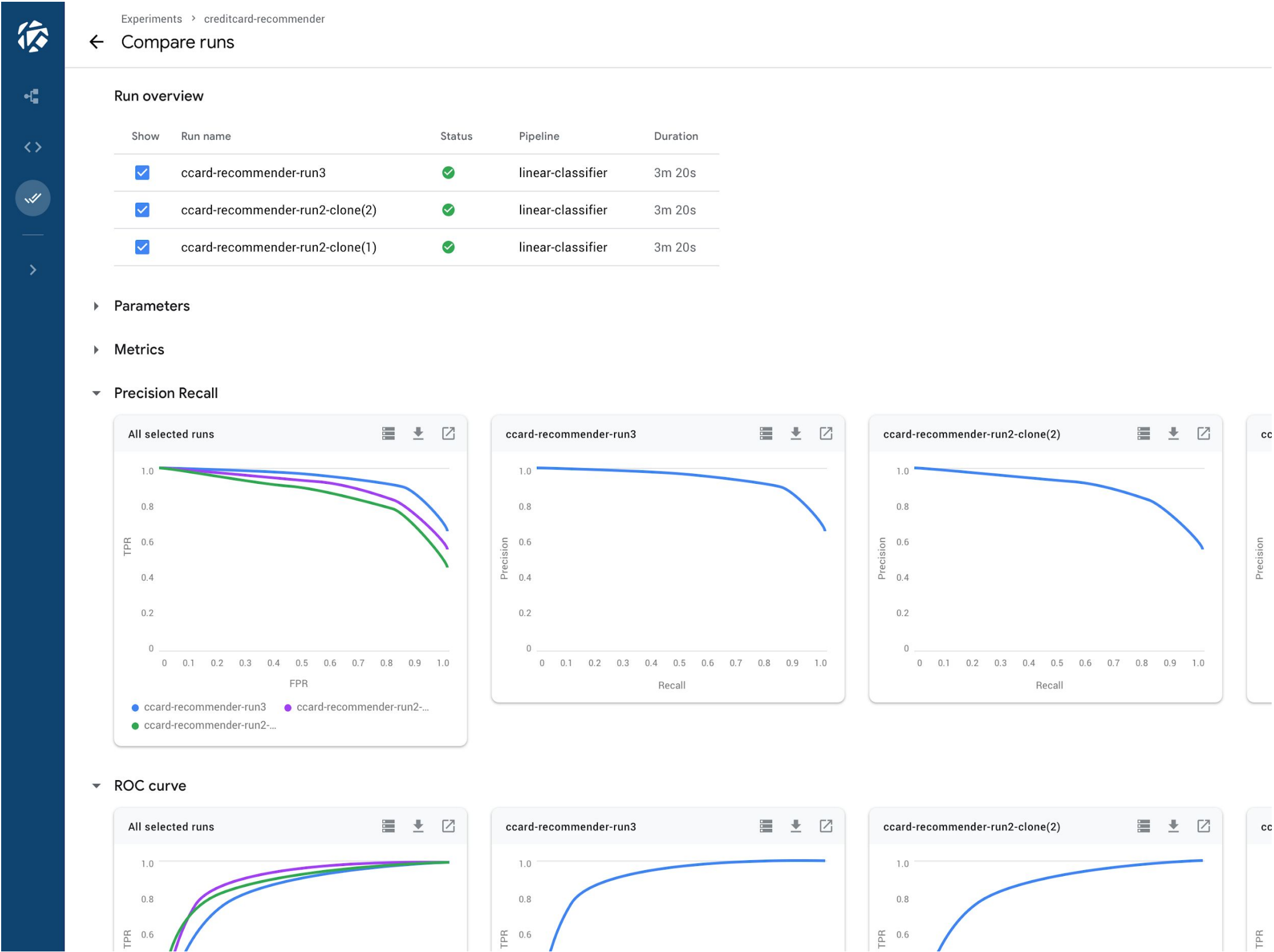
Metrics

Filter runs

<input type="checkbox"/> Runs	Status	Duration	Pipeline	Recurring run config.	Start time ↑	rmse	eta
<input type="checkbox"/> ccard-recommender-run3	✓	1m 59s	linear-classifier		9:32 AM, Aug 26, 2018	0.88	0.92
<input type="checkbox"/> ccard-recommender-run2-clone(2)	✓	2m 12s	linear-classifier		11:42 AM, Aug 25, 2018	0.72	0.86
<input type="checkbox"/> ccard-recommender-run2-clone(1)	✓	2m 44s	linear-classifier		10:48 AM, Aug 25, 2018	0.74	0.84
<input type="checkbox"/> ccard-recommender-run2	✓	2m 18s	linear-classifier		10:22 PM, Aug 25, 2018	0.82	0.76
<input type="checkbox"/> ccard-recommender-run1-clone(1)	✓	2m 20s	linear-classifier		10:10 AM, Aug 25, 2018	0.80	0.84
<input type="checkbox"/> ccard-recommender-run1	✓	3m 20s	linear-classifier		6:17 PM, Aug 24, 2018	0.72	0.76

Rows per page: 10 1–10 of 241

# Easy comparison and analysis of runs



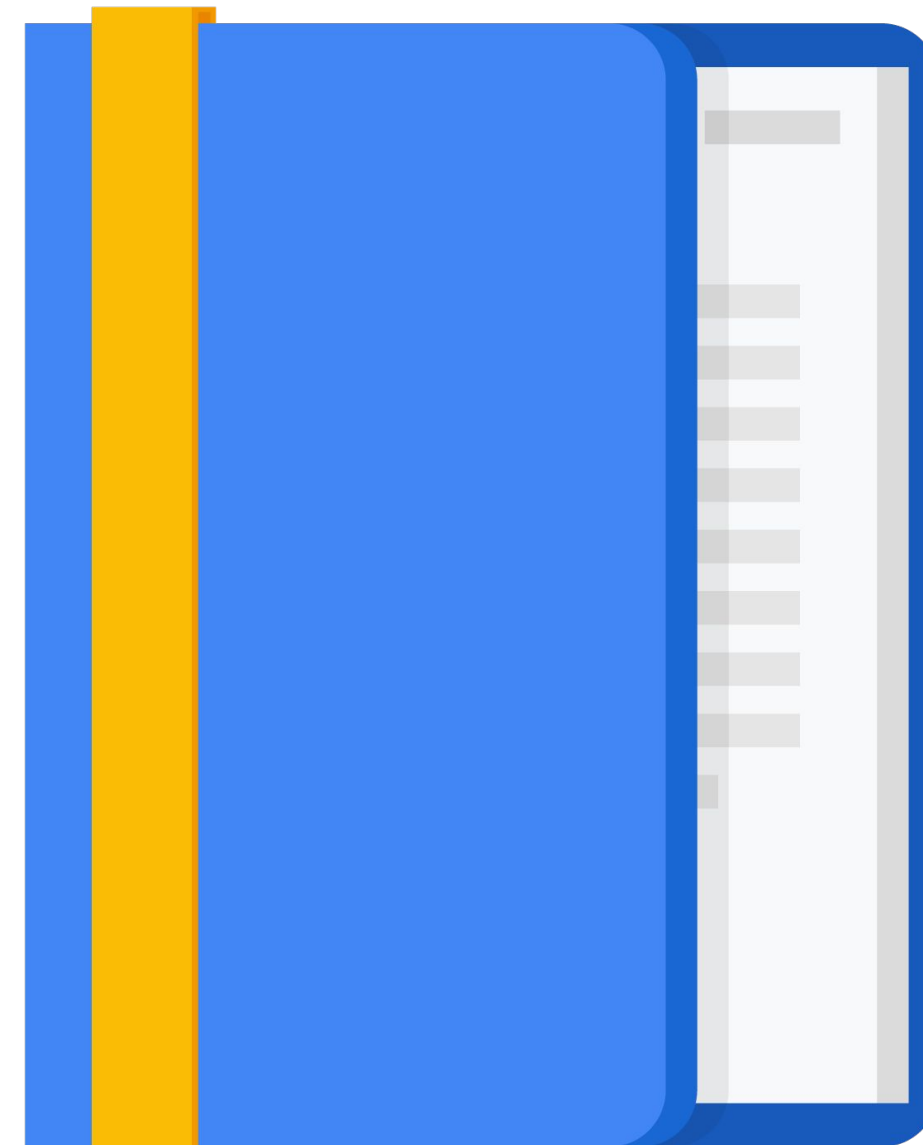
# Agenda

---

Ways to do ML on GCP

Kubeflow

AI Hub



# AI Hub is a repository for AI assets

- Don't reinvent the wheel! Find and deploy ML pipelines

AI Hub

Search

Upload

Home

My assets

Scope

Public

Private

Category

Kubeflow pipeline

Notebook

Service

TensorFlow module

VM image

Trained model

Technical guide

Data type

Image

Text

Video

Other

ML Workflow

Data gathering

Data preparation

Training

Deploying

Documentation

Feedback

Google Site Terms

Terms of services

Privacy

TensorFlow modules

Modules let you train your models with smaller datasets, train faster, and improve your model's generalization

Explore modules

Learn more

Kubeflow pipelines

Submitting a SparkSql Job to Cloud Dataproc

By Google

Public Kubeflow pipeline Text data GCP Dataproc Kubeflow Pipeline

A Kubeflow Pipeline component to submit a SparkSql job to Google Cloud Dataproc service.

Data preparation by using the General Purpose Preprocessing component

By Google

Public Kubeflow pipeline Other data Preprocessing Data transformation Cloud Dataflow TFT TFRecord CSV TFDV

The component gives you a standard way of preprocessing datasets. Use it to read datasets and serve raw data serving using a standard process. The output of this component is in the TFRecord format.

Batch predicting using Cloud Machine Learning Engine

By Google

Public Kubeflow pipeline Text data GCP ML Engine Kubeflow Pipeline

A Kubeflow Pipeline component to submit a batch prediction job against a trained model to Cloud ML Engine service.

View more

Notebooks

Text generation using a RNN with eager execution

By Google

Public Notebook Text data recurrent text charnn gru Seedbank

Given a sequence of characters from this data ("Shakespear"), train a model to predict the next character in the sequence ("e"). Longer sequences of text can be generated by calling the model repeatedly.

Piano Transcription

By Google

Public Notebook Other data sound transcription recurrent magenta music Seedbank

Onsets and Frames is an automatic piano music transcription model. This notebook demonstrates running the model on user-supplied recordings.

Training and Prediction with XGBoost

By Google

Public Notebook Other data XGBoost Training Prediction

This notebook uses the Census Income Data Set to demonstrate how to train a model and generate local predictions using XGBoost.


View more

Services

Cloud Text-to-Speech

By Google

Public Service Text data enaach enthaic enthaiza unina text-to-enaach

 Google Cloud

# AI Hub stores various asset types

- Kubeflow pipelines and components
- Jupyter notebooks
- TensorFlow modules
- Trained models
- Services
- VM images



# This is what a typical asset looks like...

☰

AI Hub

🔍

Search

📄

Feedback

👤

←

Deploying a trained model to Cloud Machine Learning Engine

↔

Scope

Public

Version

1

Category

Kubeflow pipeline

Publisher

Google

Data type

Text

Labels

GCP

ML Engine

Kubeflow

Pipeline

📘

Pipelines are standalone solutions that can integrate into your existing workflow or be used as end-to-end solutions

Learn more

Documentation

Deploying a trained model to Cloud Machine Learning Engine

A Kubeflow Pipeline component to deploy a trained model from a Cloud Storage path to a Cloud Machine Learning Engine service.

Intended use

Use the component to deploy a trained model to Cloud Machine Learning Engine service. The deployed model can serve online or batch predictions in a KFP pipeline.

Runtime arguments:

Name	Description	Type	Optional	Default
model_uri	The Cloud Storage URI which contains a model file. The commonly used TF model search path (export/exporter) will be used.	GCSPath	No	
project_id	The ID of the parent project of the serving model.	GCPProjectID	No	
model_id	The user-specified name of the model. If it is not provided, the operation uses a random name.	String	Yes	
version_id	The user-specified name of the version. If it is not provided, the operation uses a random name.	String	Yes	
runtime_version	The Cloud Machine Learning Engine's runtime version to use for this deployment. If it is not set, the Cloud ML Engine uses the default stable version, 1.0.	String	Yes	
python_version	The version of Python used in the prediction. If it is not set, the default version is 2.7. Python 3.5 is available when the runtime_version is set to 1.4 and above. Python 2.7 works with all supported runtime versions.	String	Yes	
version	The JSON payload of the new <a href="#">Version</a> .	Dict	Yes	
replace_existing_version	A Boolean flag indicates whether to replace existing version in case of conflict.	Bool	Yes	False
set_default	A Boolean flag indicates whether to set the new version as default version in the model.	Bool	Yes	False
wait_interval	A time-interval to wait for in case the operation has a long run time.	Integer	Yes	30

Output:

📄

Download

Create a [Kubeflow Cluster](#) to use this pipeline

[Learn more](#) about how to use pipelines

📄

Feedback

🐦

f

in

One-click deployment of ML pipelines via Kubeflow on GCP as platform for AI, or on premise.

Google Cloud

# Assets on AI Hub are collected in two scopes: public assets and restricted assets

- Public scope are available to all AI Hub users
- Restricted scope contains AI components that you have uploaded and assets that have been shared with you



---

# Running ML Pipelines on Kubeflow

## Objectives

- Create a Kubernetes cluster and configure AI Platform pipelines
- Launch the pipelines dashboard
- Create and run an experiment from an example end-to-end ML Pipeline
- Examine and verify the output of each step
- Inspect the pipeline graph, various metrics, logs, charts and parameters

# Module Summary

- Use ML on GCP using either
  - AI Platform (your model, your data)
  - AutoML (our models, your data)
  - Perception API (our models, our data)
- Use Kubeflow to deploy end-to-end ML pipelines
- Don't reinvent the wheel for your ML pipeline! Leverage pipelines on AI Hub