

# Index

## A

A/B testing of models, using TensorFlow Serving, [Model A/B Testing with TensorFlow Serving](#)

access scope for Kubernetes cluster nodes, [Pipeline Setup](#)

accuracy, [Classification Metrics](#)

- equal accuracy for all groups, [Model Analysis for Fairness](#)

AI Platform (see Google Cloud AI Platform)

AirflowDagRunner, [Pipeline Setup](#), [Pipeline Setup](#)

AirflowPipelineConfig, [Pipeline Setup](#)

ai\_platform\_trainer\_executor.GenericExecutor, [Running the Trainer Component](#)

analysis, [Model Analysis and Validation](#)

- (see also model analysis and validation)
- analysis step in TensorFlow Transform execution, [Preprocessing Strategies](#)
- model explainability versus, [Model Explainability](#)

annotations

- crowdsourcing, [Crowdsourcing the Annotations](#)
- expert, [Expert Annotations](#)

anonymization, [Other Methods for Data Privacy](#)

Apache Airflow, [Apache Airflow](#), [Introduction to Apache Airflow-Pipeline Execution](#)

- installation and initial setup, [Installation and Initial Setup-Installation and Initial Setup](#)
- orchestrating TFX pipelines with, [Orchestrating TFX Pipelines with Apache Airflow-Pipeline Execution](#)

  - pipeline execution, [Pipeline Execution](#)
  - setting up TFX pipeline, [Pipeline Setup](#)

- setting up basic pipeline, [Basic Airflow Example-Putting it all together](#)
- project-specific configurations, [Project-specific configurations](#)

putting it all together, [Putting it all together-Putting it all together](#)

task definitions, [Task definitions](#)

task dependencies, [Task dependencies](#)

simple interactive pipeline conversion for, [Simple Interactive Pipeline Conversion for Beam and Airflow](#)

Apache Beam, [Introduction to TensorFlow Extended](#), [Introduction to Apache Beam-Summary](#), [TFDV](#), [Introduction to Apache Beam-Orchestrating TFX Pipelines with Apache Beam](#)

basic data pipeline, [Basic Data Pipeline](#)

classes and function decorators for data ingestion via Apache Beam pipelines, [Reusing Existing Components](#)

example ML pipeline, [Putting it all together](#)

executing with Google Cloud's Dataflow, [Scaling with Google's Dataflow](#)

executing your basic pipeline, [Executing Your Basic Pipeline](#)

installation and setup, [Setup](#)

orchestrating TFX pipelines with, [Orchestrating TFX Pipelines with Apache Beam-Orchestrating TFX Pipelines with Apache Beam](#)

as pipeline orchestrator, [Apache Beam](#)

simple interactive pipeline conversion for, [Simple Interactive Pipeline Conversion for Beam and Airflow](#)

use with TensorFlow Transform, [Scaling the Preprocessing Steps](#)

using TensorFlow Transform on local setup, [Standalone Execution of TFT](#)

Apache Flink, [Orchestrating TFX Pipelines with Apache Beam](#)

Apache Spark, [Orchestrating TFX Pipelines with Apache Beam](#)

APIs

creating for model deployment in Python web app, [A Simple Model Server](#)

gRPC versus REST, [gRPC](#)

lack of code separation between model and API, [Lack of Code Separation](#)

Argo

configuration for output from Kubeflow Pipelines orchestrating TFX pipeline, [Pipeline Setup](#)

conversion of Python TFX scripts to Argo instructions, [Orchestrating TFX Pipelines with Kubeflow Pipelines](#)  
in Kubeflow Pipelines, [Introduction to Kubeflow Pipelines](#)  
artifacts  
defined, [Overview of TFX Components](#)  
input artifacts, registration in metadata store, [Component drivers](#)  
inspecting for split dataset, [Splitting one dataset into subsets](#)  
references in, [Component executors](#)  
artifact\_utils function, [Component executors](#)  
AUC (area under the curve), [Classification Metrics](#), [Validation in the Evaluator Component](#)  
Avro-serialized data, converting to tf.Example, [Converting Avro-serialized data to tf.Example](#)  
AWS (Amazon Web Services), [Deploying with Cloud Providers](#)  
accessing private models from S3 storage buckets, [Accessing private models from AWS S3](#)  
adding AWS credentials to TFX component containers, [Pipeline Setup](#)  
S3 storage buckets, [Ingesting Remote Data Files](#)

## B

BaseDriver class, [Component drivers](#)  
bash commands for Kubeflow Pipelines installation, [Installation and Initial Setup](#)  
batches  
batch size requirement using MirroredStrategy, [Distribution Strategies](#)  
batching inference requests, [Batching Inference Requests-Configuring Batch Predictions](#)  
configuring batch predictions, [Configuring Batch Predictions](#)  
preprocessing in, [Best Practices](#)  
Bayesian search, [Strategies for Hyperparameter Tuning](#)  
Beam (see Apache Beam)  
BentoML, [BentoML](#)  
bfloat16 floating point precision, [Quantization](#)  
bias

biased datasets, detecting with TFDV, [Biased Datasets](#)

selection bias, [Biased Datasets](#)

BigQuery database (see Google Cloud BigQuery)

bit shift operator (Airflow), [Task dependencies](#)

blessing (in TFX), [Analysis and Validation in TFX](#), [Human in the Loop](#)

model blessing provided by Slack component, [How to Use the Slack Component](#)

bucket features, processing with TFT, [Integrate TFT into Your Machine Learning Pipeline](#)

business case for pipelines, [Why Machine Learning Pipelines?](#)

bytes\_feature helper function, [Using the SavedModel outside a pipeline](#)

## C

categorical features, statistics generated about in TFDV, [Generating Statistics from Your Data](#)

CentralStorageStrategy, [Distribution Strategies](#)

channels

gRPC, establishing, [Using TensorFlow Serving via gRPC](#)

TFX components communicating through, [Component specifications](#)

types available in TFX, [Component channels](#)

CI/CD systems for machine learning, [CI/CD Systems for Machine Learning](#)

classification metrics, [Classification Metrics-Regression Metrics](#)

classification models, getting predictions via gRPC API, [Getting predictions from classification and regression models](#)

classify method, [Signature Methods](#)

CLI (see command-line interface in TFX)

cloud

deployments of models on, [Deploying with Cloud Providers-Model inference](#)

managed Kubernetes services from providers, [Installation and Initial Setup](#)

model deployments on

example deployment with GCP, [Example Deployment with GCP-Model inference](#)

use cases, [Use Cases](#)

models stored in cloud buckets, [Workflow Overview](#)

accessing models from AWS S3, [Accessing private models from AWS S3](#)

remote loading optimizations, [Optimization of Remote Model Loading](#)

storage buckets, [Ingesting Remote Data Files](#)

using cloud storage buckets for data exchange, [Use Cloud Storage buckets for data exchange](#)

watching infrastructure costs, [Before You Get Started](#)

collections (Apache Beam), [Basic Data Pipeline](#)

basic example, [Basic collection example](#)

command-line interface (CLI) in TFX, [TFX Command-Line Interface-Publishing Your Pipeline with TFX CLI](#)

publishing your pipeline with, [Publishing Your Pipeline with TFX CLI](#)

TFX and its dependencies, [TFX and Its Dependencies](#)

TFX project templates, [TFX Templates](#)

COMPAS algorithm, [Model Analysis for Fairness](#)

computer vision problems

image data, preprocessing with TFT, [Image data for computer vision problems](#)

image data, strategies for ingesting, [Image Data for Computer Vision Problems](#)

concrete functions, [Using the SavedModel outside a pipeline](#)

constrained optimization (for model fairness), [Going Deeper with the What-If Tool](#)

counterfactuals, [Going Deeper with the What-If Tool](#)

crowdsourcing annotations, [Crowdsourcing the Annotations](#)

CSV files, [Ingesting Local Data Files](#)

converting to tf.Example, [Converting comma-separated data to tf.Example](#)

use in TFDV, [TFDV](#)

custom TFX components, [Custom TFX Components-Using our custom executor](#)

reusing existing components, [Reusing Existing Components](#)  
use cases, [Use Cases of Custom Components](#)  
writing from scratch, [Writing a Custom Component from Scratch-Implementation review](#)  
assembling the custom component, [Assembling the custom component](#)  
component drivers, [Component drivers](#)  
component executor, [Component executors](#)  
component specifications (ComponentSpec), [Component specifications](#)  
implementation review, [Implementation review](#)  
parts of custom component, [Writing a Custom Component from Scratch](#)  
using the custom component, [Using our basic custom component](#)

## D

DAG objects

Airflow configuration options for, [Project-specific configurations](#)  
KubeflowDagRunnerConfig, [Pipeline Setup](#)  
turning on a DAG in Airflow, [Pipeline Execution](#)

DAGs (see directed acyclic graphs)

data flywheel, [The Data Flywheel](#)

data ingestion, [Data Ingestion and Data Versioning](#), [Data Ingestion-Summary](#)

concepts, [Concepts for Data Ingestion](#)

directly from databases, [Ingesting Data Directly from Databases](#)

Google Cloud BigQuery, [Google Cloud BigQuery](#)

Presto databases, [Presto databases](#)

ingesting local data files, [Ingesting Local Data Files-Converting your custom data to TFRecord data structures](#)

converting Avro-serialized data to tf.Example, [Converting Avro-serialized data to tf.Example](#)

converting CSV data to tf.Example, [Converting comma-separated data to tf.Example](#)

converting Parquet-serialized data to tf.Example, [Converting Parquet-serialized data to tf.Example](#)

converting your custom data to TFRecord data structures, [Converting your custom data to TFRecord data structures](#)

importing existing TFRecord files, [Importing existing TFRecord Files](#)

ingesting remote data files, [Ingesting Remote Data Files](#)

preparing data for, [Data Preparation-Versioning Datasets](#)

spanning datasets, [Spanning Datasets](#)

splitting datasets, [Splitting Datasets](#)

versioning datasets, [Versioning Datasets](#)

reusing existing components, [Reusing Existing Components](#)

strategies for, [Ingestion Strategies-Summary](#)

image data for computer vision problems, [Image Data for Computer Vision Problems](#)

structured data, [Structured Data](#)

text data for natural language problems, [Text Data for Natural Language Problems](#)

data preprocessing, [Data Preprocessing](#), [Data Preprocessing-Summary](#)

image data for computer vision problems, [Image Data for Computer Vision Problems](#)

in machine learning pipelines, [Data Preprocessing](#)

reasons for performing, [Why Data Preprocessing?-Checking Your Preprocessing Results in Your Pipeline](#)

avoiding training-serving skew, [Avoiding a Training-Serving Skew](#)

checking preprocessing results in ML pipeline, [Checking Your Preprocessing Results in Your Pipeline](#)

deploying preprocessing steps and model as one artifact, [Deploying Preprocessing Steps and the ML Model as One Artifact](#)

processing data in context of entire dataset, [Preprocessing the Data in the Context of the Entire Dataset](#)

scaling preprocessing steps, [Scaling the Preprocessing Steps](#)

using TensorFlow Transform, [Data Preprocessing with TFT-Summary](#)

best practices, [Best Practices](#)



functions in TFT, [TFT Functions](#)  
installing TFT, [Installation](#)  
preprocessing strategies, [Preprocessing Strategies](#)  
standalone execution of TFT, [Standalone Execution of TFT](#)  
data privacy, [Data Privacy](#), [Data Privacy for Machine Learning-Summary](#)  
differential privacy, [Differential Privacy-Differential Privacy for Machine Learning](#)  
epsilon, delta, and the privacy budget, [Epsilon, Delta, and the Privacy Budget](#)  
local and global, [Local and Global Differential Privacy](#)  
for machine learning, [Differential Privacy for Machine Learning](#)  
encrypted machine learning, [Encrypted Machine Learning](#)  
converting trained model to serve encrypted predictions, [Converting a Trained Model to Serve Encrypted Predictions](#)  
encrypted model training, [Encrypted Model Training](#)  
federated learning, [Federated Learning-Federated Learning in TensorFlow](#)  
in TensorFlow, [Federated Learning in TensorFlow](#)  
issues, [Data Privacy Issues-What Data Needs to Be Kept Private?](#)  
simplest way to increase privacy, [The Simplest Way to Increase Privacy](#)  
what data should be kept private, [What Data Needs to Be Kept Private?](#)  
why care about data privacy, [Why Do We Care About Data Privacy?](#)  
model deployment and, [Model Deployment with TensorFlow Serving](#)  
other methods for, [Other Methods for Data Privacy](#)  
TensorFlow Privacy, [Introduction to TensorFlow Privacy-Calculating Epsilon](#)  
calculating epsilon, [Calculating Epsilon](#)  
training with differentially private optimizer, [Training with a Differentially Private Optimizer](#)  
data types  
considering in TFT, [Best Practices](#)  
in tf.Estimator model, [Using the SavedModel outside a pipeline](#)



neural network weights stored as float 32-bit data types,

[Quantization](#)

data validation, [Data Validation](#), [Data Validation-Summary](#)

benefits of, [Why Data Validation?](#)

integrating TFDV into machine learning pipeline, [Integrating TFDV into Your Machine Learning Pipeline](#)

processing large datasets with GCP, [Processing Large Datasets with GCP-Processing Large Datasets with GCP](#)

recognizing problems in your data, [Recognizing Problems in Your Data-Slicing Data in TFDV](#)

biased datasets, [Biased Datasets](#)

comparing datasets, [Comparing Datasets](#)

data skew and drift, [Data Skew and Drift](#)

slicing data, [Slicing Data in TFDV](#)

updating the schema, [Updating the Schema](#)

TensorFlow Data Validation (TFDV), [TFDV-Generating Schema from Your Data](#)

Data Version Control (DVC), [Versioning Datasets](#)

databases

ingesting data directly from, [Ingesting Data Directly from Databases](#)

Presto databases, [Presto databases](#)

initializing Airflow database, [Installation and Initial Setup](#)

Kubeflow Pipelines using, [Pipeline Setup](#)

MySQL database with Kubeflow Pipelines, [Introduction to Kubeflow Pipelines](#)

Dataflow (Google Cloud), [Processing Large Datasets with GCP](#)

executing Apache Beam with, [Scaling with Google's Dataflow](#)  
job details, [Pipeline Execution](#)

TensorFlow Transform on, [Standalone Execution of TFT](#)  
using Apache Beam with, [Setup](#)

decision thresholds, checking with Fairness Indicators, [Checking Decision Thresholds with Fairness Indicators](#)

delta, [Epsilon, Delta, and the Privacy Budget](#)

demographic parity, [Model Analysis for Fairness](#)

deployment cycles, decoupling, [Decoupling Deployment Cycles-Optimization of Remote Model Loading](#)

workflow overview, [Workflow Overview](#)

DevOps deployment cycles, decoupling from data scientists, [Workflow Overview](#)

differential privacy, [Differential Privacy](#)-[Differential Privacy for Machine Learning](#)

epsilon, delta, and the privacy budget, [Epsilon, Delta, and the Privacy Budget](#)

local and global, [Local and Global Differential Privacy](#)

for machine learning, [Differential Privacy for Machine Learning](#)

training with a differentially private optimizer, [Training with a Differentially Private Optimizer](#)

directed acyclic graphs (DAGs), [Directed Acyclic Graphs](#)

distillation (model), [Distillation](#)

distribution strategies, [Distribution Strategies](#)-[Distribution Strategies](#)

Docker, [What Is a Container?](#)

building your first Docker image, [Building Your First Docker Image](#)

command to start TensorFlow Serving container with batching, [Configuring Batch Predictions](#)

command-line interface (CLI), [Diving into the Docker CLI](#)

docker run command, [Diving into the Docker CLI](#)

other useful commands, [Diving into the Docker CLI](#)

configuration for TensorFlow Serving with multiple models, [Multiple Model Configuration](#)

custom TFX images, [Custom TFX Images](#)

deploying via Kubernetes without specific Docker containers, [Simple Scaling with TensorFlow Serving and Kubernetes](#)

installing TensorFlow Serving, [Docker Installation](#)

introduction to, [Introduction to Docker](#)

Docker images, [Introduction to Docker Images](#)

Prometheus Docker setup, [Prometheus Setup](#)

running TensorFlow Serving with image prebuilt for GPU images, [Single Model Configuration](#)

docker run command, [Diving into the Docker CLI](#)

Docker Swarm, [Simple Scaling with TensorFlow Serving and Kubernetes](#)

Dockerfiles, [Introduction to Docker Images](#), [Building Your First Docker Image](#), [Publishing Your Pipeline with TFX CLI](#)

DP (see differential privacy)

drift in data, [Why Data Validation?](#)

detecting with TFDV, [Data Skew and Drift](#)

drift\_comparator (TFDV), [Data Skew and Drift](#)

driver (TFX components), [Overview of TFX Components](#)

for custom TFX component, [Component drivers](#)

## E

early stopping, [Strategies for Hyperparameter Tuning](#)

edge devices, deploying models to, [Steps to Optimize Your Model with TFLite](#)

encrypted machine learning, [Encrypted Machine Learning](#)

converting trained model to serve encrypted predictions,

[Converting a Trained Model to Serve Encrypted Predictions](#)

encrypted model training, [Encrypted Model Training](#)

epsilon, [Epsilon, Delta, and the Privacy Budget](#)

equal accuracy, [Model Analysis for Fairness](#)

equal opportunity, [Model Analysis for Fairness](#)

checking false positive rate for each group, [Slicing Model](#)

[Predictions in TFMA](#)

Estimator models, [Model Training](#)

exporting for TensorFlow Serving, [Exporting Models for](#)

[TensorFlow Serving](#), [Exporting Models for TensorFlow Serving](#)

using Trainer component with, [Using the Trainer component with an Estimator model](#)

EvalConfig, [Analyzing a Single Model in TFMA](#), [Analyzing Multiple Models in TFMA](#), [Evaluator Component](#)

metrics\_specs argument, [Analyzing a Single Model in TFMA](#)

EvalSharedModel, [Analyzing a Single Model in TFMA](#)

evaluation datasets, [Splitting one dataset into subsets](#)

as input for TFMA, [Analyzing a Single Model in TFMA](#)

Evaluator component (TFX), [Evaluator Component](#), [Exporting TFLite Models](#), [Converting Your Interactive TFX Pipeline to a Production Pipeline](#)

model validation in, [Validation in the Evaluator Component](#)

ExampleGen component, [Concepts for Data Ingestion](#), [Converting Your Interactive TFX Pipeline to a Production Pipeline](#)

Examples channel, [Component channels](#)

ExampleValidator component, [Converting Your Interactive TFX Pipeline to a Production Pipeline](#)

executors (TFX component), [Overview of TFX Components](#),  
[Component executors](#)

- executor classes for loading different file types, [Converting Parquet-serialized data to tf.Example](#)

- GenericExecutor, [Running the Trainer Component](#)

- using our custom executor, [Using our custom executor](#)

experiment tracking (model), [Model Experiment Tracking](#)

expert annotations, [Expert Annotations](#)

explainability (model), [Model Explainability-Other Explainability Techniques](#)

- caution with, [Other Explainability Techniques](#)

- generating explanations with What-If Tool, [Generating Explanations with the WIT](#)

- local and global explanations, [Model Explainability](#)

- other explainability techniques, [Other Explainability Techniques](#)

explicit feedback, [Explicit and Implicit Feedback](#)

- tracking, [Tracking Explicit Feedback](#)

exporting a pipeline, [Simple Interactive Pipeline Conversion for Beam and Airflow](#)

exporting models

- exported model folder, [Exporting Models for TensorFlow Serving](#)

- for deployment in mobile apps, [Exporting TFLite Models](#)

- for TensorFlow Serving, [Exporting Models for TensorFlow Serving](#)

export\_saved\_model method, [Exporting Models for TensorFlow Serving](#)

ExternalArtifact channel, [Component specifications](#)

## F

Fairness Indicators project

- checking false positives for equal opportunity, [Slicing Model Predictions in TFMA](#)

documentation, [Model Analysis for Fairness](#)

loading Fairness Indicators in TFX Evaluator, [Evaluator Component](#)  
fairness, model analysis for, [Model Analysis for Fairness-Going Deeper with the What-If Tool](#)

checking decision thresholds with Fairness Indicators, [Checking Decision Thresholds with Fairness Indicators](#)

defining numerically the meaning of fairness, [Model Analysis for Fairness](#)

going deeper with What-If Tool, [Going Deeper with the What-If Tool-Going Deeper with the What-If Tool](#)

slicing model predictions in TFMA, [Slicing Model Predictions in TFMA](#)

false negatives, [Classification Metrics](#)

false positives, [Classification Metrics](#)

false positive rate (FPR), [Classification Metrics](#)

feature attributions, [Generating Explanations with the WIT](#)

features

converting to one-hot, [Integrate TFT into Your Machine Learning Pipeline](#)

feature names in TFT, [Best Practices](#)

slicing datasets on, [Slicing Data in TFDV](#)

tf.train.Feature object, [Converting your custom data to TFRecord data structures](#)

federated learning, [Federated Learning-Federated Learning in TensorFlow](#)

in TensorFlow, [Federated Learning in TensorFlow](#)

feedback loops, [Feedback Loops](#), [Feedback Loops-Summary](#)

design patterns for collecting feedback, [Design Patterns for Collecting Feedback-How to Track Feedback Loops](#)

crowdsourcing annotations, [Crowdsourcing the Annotations](#)

expert annotations, [Expert Annotations](#)

feedback produced automatically by the system, [Producing Feedback Automatically](#)

users correct the prediction, [Users Correct the Prediction](#)

users rate quality of the prediction, [Users Rate the Quality of the Prediction](#)

users take an action as result of prediction, [Users Take Some Action as a Result of the Prediction](#)

explicit and implicit feedback, [Explicit and Implicit Feedback-Feedback Loops in the Real World](#)

data flywheel, [The Data Flywheel](#)

feedback loops in the real world, [Feedback Loops in the Real World](#)

monitoring for unexpected consequences, [Feedback Loops in the Real World](#)

negative consequences of, [Feedback Loops](#)

tracking, [How to Track Feedback Loops](#)

explicit feedback, [Tracking Explicit Feedback](#)

implicit feedback, [Tracking Implicit Feedback](#)

FileBasedExampleGen, [Reusing Existing Components](#), [Using our custom executor](#)

floating point numbers, quantization of, [Quantization](#)

function decorators, [Reusing Existing Components](#)

future of pipelines and next steps, [The Future of Pipelines and Next Steps-Summary](#)

CI/CD systems for machine learning, [CI/CD Systems for Machine Learning](#)

future pipeline capabilities, [Future Pipeline Capabilities](#)

machine learning engineering community, [Machine Learning Engineering Community](#)

model experiment tracking, [Model Experiment Tracking](#)

model release management, [Thoughts on Model Release Management](#)

testing of machine learning models, [Testing Machine Learning Models](#)

TFX with other ML frameworks, [TFX with Other Machine Learning Frameworks](#)

## G

GCP (Google Cloud Platform), [Before You Get Started](#)

accessing models from GCP buckets, [Accessing private models from GCP Buckets](#)

adding GCP credentials of TFX component containers, [Pipeline Setup](#)

AI Platform (see Google Cloud AI Platform)

container registry, [Custom TFX Images](#)

Google Cloud Credentials, [Google Cloud BigQuery](#)

Kubernetes on, [Kubernetes on Google Cloud-Persistent Volume Setups for Kubeflow Pipelines](#)

accessing your cluster with kubectl, [Accessing Your Kubernetes Cluster with kubectl](#)

creating a Kubernetes cluster, [Creating a Kubernetes Cluster](#)

selecting Google Cloud project, [Selecting a Google Cloud Project](#)

setting up Google Cloud project, [Setting Up Your Google Cloud Project](#)

using your Kubernetes cluster with kubectl, [Using Your Kubernetes Cluster with kubectl](#)

persistent volume setup for Kubeflow Pipelines, [Persistent Volume Setups for Kubeflow Pipelines](#)

processing large datasets with, [Processing Large Datasets with GCP-Processing Large Datasets with GCP](#)

storage buckets, [Ingesting Remote Data Files](#)

TensorFlow Serving deployment on Kubernetes, [Simple Scaling with TensorFlow Serving and Kubernetes](#)

training models on AI Platform, [Running the Trainer Component](#)

GenericExecutor, [Running the Trainer Component](#)

get\_model function, [Compile and train the model](#)

get\_serve\_tf\_examples\_fn function, [Model export](#), [Running the Trainer Component](#)

global differential privacy, [Local and Global Differential Privacy](#)

global explanations, [Model Explainability](#)

Google Cloud AI Platform

example model deployment with, [Example Deployment with GCP-Model inference](#)

model deployment, [Model deployment](#)

model inference, [Model inference](#)

feature attributions, [Generating Explanations with the WIT](#)

Kubeflow Pipelines on, [Kubeflow Pipelines on AI Platform](#)



managed Kubeflow Pipelines installations, [Installation and Initial Setup](#)

pipelines based on, [Pipelines Based on Google Cloud AI Platform-Pipeline Execution](#)

executing the TFX pipeline, [Pipeline Execution](#)

Kubeflow Pipelines setup, [Pipeline Setup](#)

TFX pipeline setup, [TFX Pipeline Setup-Scaling with Google's Dataflow](#)

Google Cloud BigQuery

ingesting data directly from, [Google Cloud BigQuery](#)

using Apache Beam with, [Setup](#)

Google Cloud Marketplace, [Pipeline Setup](#)

Google Colab, [Interactive Pipelines](#)

TFMA in Google Colab notebook, [Analyzing a Single Model in TFMA](#)

Google Remote Procedure Calls (gRPC)

endpoint configuration with TensorFlow Serving, [Single Model Configuration](#)

request for model metadata via model server, [gRPC Requests for Model Metadata](#)

REST versus, [REST Versus gRPC-gRPC](#)

using TensorFlow Serving via, [Using TensorFlow Serving via gRPC-Payloads](#)

getting predictions from classification and regression models, [Getting predictions from classification and regression models](#)

payloads from requests, [Payloads](#)

secure connections, [Using TensorFlow Serving via gRPC](#)

GPUs, [Using TensorRT with TensorFlow Serving](#)

Docker container installation of TensorFlow Serving, [Docker Installation](#)

requesting for execution of TFX component containers, [Pipeline Setup](#)

GraphPipe, model deployment with, [GraphPipe](#)

grid search, [Strategies for Hyperparameter Tuning](#)

gRPC (see Google Remote Procedure Calls)

hash\_buckets, [Splitting one dataset into subsets](#)  
homomorphic encryption (HE), [Encrypted Machine Learning](#)  
HTTP requests to model server, URLs for, [URL structure](#)  
Human in the Loop component, [Human in the Loop-How to Use the Slack Component](#)  
Hyperband algorithm, [Strategies for Hyperparameter Tuning](#)  
hyperparameter tuning for models, [Strategies for Hyperparameter Tuning](#)  
in TFX pipelines, [Hyperparameter Tuning in TFX Pipelines](#)

## I

image data for computer vision problems  
preprocessing with TFT, [Image data for computer vision problems](#)  
strategies for ingesting, [Image Data for Computer Vision Problems](#)  
ImageIngestComponent (example)  
assembling, [Assembling the custom component](#)  
implementation review, [Implementation review](#)  
using, [Using our basic custom component](#)  
ImageIngestComponentSpec (example), [Component specifications](#)  
ImageIngestDriver class (example), [Component drivers](#)  
ImageIngestExecutor class (example), [Component executors](#)  
image\_to\_example function, [Reusing Existing Components](#)  
implementation review for custom component, [Using our basic custom component](#)  
implicit feedback, [Explicit and Implicit Feedback](#)  
tracking, [Tracking Implicit Feedback](#)  
infrastructure for machine learning, [Introduction to Infrastructure for Machine Learning-Deploying Applications to Kubernetes](#)  
containers, [What Is a Container?](#)  
deploying applications to Kubernetes, [Deploying Applications to Kubernetes-Deploying Applications to Kubernetes](#)  
Docker, [Introduction to Docker-Diving into the Docker CLI](#)  
Kubernetes, [Introduction to Kubernetes-Defining a Kubernetes Resource](#)  
ingestion of data (see data ingestion)

init\_beam\_pipeline function, [Orchestrating TFX Pipelines with Apache Beam](#)

init\_components function, [Converting Your Interactive TFX Pipeline to a Production Pipeline](#), [Pipeline Setup](#)

input configuration for split datasets, [Preserving existing splits](#)

inputs

- for classification method, [Signature Methods](#)

- inspecting for an exported model, [Inspecting the Model](#)

- for prediction method, [Signature Methods](#)

- for regression method, [Signature Methods](#)

- testing for exported model, [Testing the Model](#)

input\_fn function, [Load the data](#), [Running the Trainer Component](#)

interactive pipelines, [Interactive Pipelines-Interactive Pipelines](#)

- TFX, converting to production pipeline, [Converting Your Interactive TFX Pipeline to a Production Pipeline-Simple Interactive Pipeline Conversion for Beam and Airflow](#)

- using TensorBoard in, [Using TensorBoard in an Interactive Pipeline](#)

## J

JSON

- data structures submitted to REST, conversion to tf.Example, [gRPC](#)

- payloads of REST client requests, [REST](#)

Jupyter Notebooks, [Interactive Pipelines](#)

- comparing datasets in, [Comparing Datasets](#)

- deploying to Kubernetes, [Deploying Applications to Kubernetes-Deploying Applications to Kubernetes](#)

- displaying schema generated by TFDV, [Generating Schema from Your Data](#)

- installing and enabling TFMA notebook extension, [Analyzing a Single Model in TFMA](#)

- installing and enabling WIT notebook extension, [Going Deeper with the What-If Tool](#)

## K

K-anonymity, [Other Methods for Data Privacy](#)

Katib, [Strategies for Hyperparameter Tuning](#)

Keras Tuner, [Strategies for Hyperparameter Tuning](#)

kubectrl, [Simple Scaling with TensorFlow Serving and Kubernetes](#)

accessing your Kubernetes cluster with, [Accessing Your Kubernetes Cluster with kubectrl](#)

getting started with, [Getting Started with Minikube and kubectrl](#)

installing, [Before You Get Started](#)

interacting with, [Interacting with the Kubernetes CLI](#)

using your Kubernetes cluster with, [Using Your Kubernetes Cluster with kubectrl](#)

Kubeflow, [Simple Scaling with TensorFlow Serving and Kubernetes](#)

versus Kubeflow Pipelines, [Introduction to Kubeflow Pipelines](#)

Kubeflow Pipelines, [Kubeflow Pipelines](#), [Pipelines Part 2: Kubeflow Pipelines-Summary](#)

accessing your installation, [Accessing Your Kubeflow Pipelines Installation](#)

on Google Cloud AI Platform, [Kubeflow Pipelines on AI Platform](#)

installation and initial setup, [Installation and Initial Setup](#)

introduction to, [Introduction to Kubeflow Pipelines](#)

Kubeflow versus, [Introduction to Kubeflow Pipelines](#)

operating tips for, [Tips for Operating Kubeflow Pipelines-Publishing Your Pipeline with TFX CLI](#)

custom TFX images, [Custom TFX Images](#)

exchanging data through persistent volumes, [Exchange Data Through Persistent Volumes](#)

TFX CLI, [TFX Command-Line Interface-Publishing Your Pipeline with TFX CLI](#)

OpFunc functions, [Pipeline Setup](#)

orchestrating TFX pipelines with, [Orchestrating TFX Pipelines with Kubeflow Pipelines-Auditing the pipeline lineage](#)

auditing pipeline lineage, [Auditing the pipeline lineage](#)

collaborating and reviewing pipeline runs, [Collaborating and reviewing pipeline runs](#)

executing the pipeline, [Executing the Pipeline-Useful Features of Kubeflow Pipelines](#)

pipeline setup, [Pipeline Setup-Executing the Pipeline](#)

recurring runs, [Recurring runs](#)

restarting failed pipelines, [Restart failed pipelines](#)  
workflow from TFX script to Kubeflow Pipelines, [Orchestrating TFX Pipelines with Kubeflow Pipelines](#)  
overview of, [Introduction to Kubeflow Pipelines](#)  
persistent volume setups for, [Persistent Volume Setups for Kubeflow Pipelines](#)  
pipelines based on Google Cloud AI Platform, [Pipelines Based on Google Cloud AI Platform-Pipeline Execution](#)  
pipeline setup, [Pipeline Setup](#)  
TFX pipeline setup, [TFX Pipeline Setup-Scaling with Google's Dataflow](#)

KubeflowDagRunnerConfig, [Pipeline Setup](#)

Kubernetes, [Pipelines Part 2: Kubeflow Pipelines](#), [Introduction to Kubernetes-Deploying Applications to Kubernetes](#)

(see also Kubeflow Pipelines)

creating a cluster, [Installation and Initial Setup](#)

creating or choosing a cluster from Google Cloud Marketplace, [Pipeline Setup](#)

defining resources, [Defining a Kubernetes Resource](#)

definitions of terminology, [Some Kubernetes Definitions](#)

deploying applications to, [Deploying Applications to Kubernetes-Deploying Applications to Kubernetes](#)

getting started with Minikube and kubectl, [Getting Started with Minikube and kubectl](#)

information resources, [Simple Scaling with TensorFlow Serving and Kubernetes](#)

interacting with the CLI, [Interacting with the Kubernetes CLI](#)

persistent volume setup for Kubeflow Pipelines on GCP, [Persistent Volume Setups for Kubeflow Pipelines](#)

setting up a cluster, [Setting Up a Kubernetes Cluster on Google Cloud-Persistent Volume Setups for Kubeflow Pipelines](#)

before you get started, [Before You Get Started](#)

Kubernetes on Google Cloud, [Kubernetes on Google Cloud-Persistent Volume Setups for Kubeflow Pipelines](#)

simple scaling with TensorFlow Serving and, [Simple Scaling with TensorFlow Serving and Kubernetes-Summary](#)

## L

L-infinity norm, [Data Skew and Drift](#)

labels, version labels for deployed models, [Multiple Model Configuration](#)

latest\_blessed\_model\_resolver, [ResolverNode](#)

layered filesystems, [Introduction to Docker Images](#)

learning rate schedulers, [Strategies for Hyperparameter Tuning](#)

legal and regulatory requirements for data privacy, [Why Do We Care About Data Privacy?](#)

LIME (local interpretable model-agnostic explanations), [Other Explainability Techniques](#)

local differential privacy, [Local and Global Differential Privacy](#)

local explanations, [Model Explainability](#)

## M

machine learning engineering community, [Machine Learning Engineering Community](#)

machine learning pipelines

- architecture, [Summary](#)

- benefits of, [Why Machine Learning Pipelines?](#)

- overview of steps in, [Overview of the Steps in a Machine Learning Pipeline-Data Privacy](#)

- when to consider, [When to Think About Machine Learning Pipelines](#)

MAE (see mean absolute error)

MAPE (see mean absolute percentage error)

mean absolute error (MAE), [Regression Metrics](#)

mean absolute percentage error (MAPE), [Regression Metrics](#)

mean squared error (MSE), [Regression Metrics](#)

metadata

- artifact, [Overview of TFX Components](#)

- request for model metadata via model server

  - on gRPC API, [gRPC Requests for Model Metadata](#)

  - on REST API, [REST Requests for Model Metadata](#)

requesting model metadata from model server, [Requesting Model Metadata from the Model Server-gRPC Requests for Model Metadata](#)

SQLite database, [Orchestrating TFX Pipelines with Apache Beam](#)

TFX components communicating through, [What Is ML Metadata?](#)  
MetadataStore, [Pipeline Orchestration](#), [Introduction to Kubeflow Pipelines](#)

(see also metadata)

components' interactions with, [What Is ML Metadata?](#)

in-memory, setting up with InteractiveContext, [Interactive Pipelines](#)

publisher managing saving of metadata to, [Overview of TFX Components](#)

methods, signature, [Signature Methods](#)

MetricsSpec, [Validation in the Evaluator Component](#)

Microsoft Azure, [Deploying with Cloud Providers](#)

Microsoft Twitter bot, [Feedback Loops in the Real World](#)

Minikube, [Getting Started with Minikube and kubectl](#)

MirroredStrategy, [Distribution Strategies](#)

batch size requirement, [Distribution Strategies](#)

example use with TFX Trainer, [Distribution Strategies](#)

ML Metadata (MLMD) API (TFX), [What Is ML Metadata?](#)

MLflow, model deployment with, [MLflow](#)

mobile deployments for ML models, [Exporting TFLite Models](#)

model analysis and validation, [Model Analysis and Validation-Summary](#)

analyzing your model, [How to Analyze Your Model-Regression Metrics](#)

using classification metrics, [Classification Metrics-Regression Metrics](#)

using regression metrics, [Regression Metrics](#)

in TFX, [Analysis and Validation in TFX-Summary](#)

model analysis, [Model Analysis](#)

model analysis for fairness, [Model Analysis for Fairness-Going Deeper with the What-If Tool](#)

model explainability, [Model Explainability-Other Explainability Techniques](#)



TensorFlow Model Analysis, [TensorFlow Model Analysis-Analyzing Multiple Models in TFMA](#)

model cards framework, [Other Explainability Techniques](#)

model deployment, [Model Deployment](#)

size restrictions on AI Platform, [Serving models through AI Platform endpoints](#)

model deployment with TensorFlow Serving, [Model Deployment with TensorFlow Serving-Summary](#)

advanced, [Advanced Model Deployments with TensorFlow Serving-Summary](#)

decoupling deployment cycles, [Decoupling Deployment Cycles-Optimization of Remote Model Loading](#)

model optimizations for deployment, [Model Optimizations for Deployments-Using TensorRT with TensorFlow Serving](#)

monitoring TensorFlow Serving instances, [Monitoring Your TensorFlow Serving Instances-Simple Scaling with TensorFlow Serving and Kubernetes](#)

simple scaling with Kubernetes and, [Simple Scaling with TensorFlow Serving and Kubernetes-Summary](#)

alternatives to TensorFlow Serving, [TensorFlow Serving Alternatives](#)

batching inference requests, [Batching Inference Requests-Configuring Batch Predictions](#)

configuring a TensorFlow server, [Configuring a TensorFlow Server-Multiple Model Configuration](#)

deployment with cloud providers, [Deploying with Cloud Providers-Model inference](#)

deployment with TFX pipelines, [Model Deployment with TFX Pipelines](#)

inspecting exported models, [Inspecting Exported Models-Testing the Model](#)

making predictions from the model server, [Making Predictions from the Model Server-Payloads](#)

model A/B testing with TensorFlow Serving, [Model A/B Testing with TensorFlow Serving](#)

model signatures, [Model Signatures-Signature Methods](#)

other optimizations, [Other TensorFlow Serving Optimizations](#)

requesting model metadata from model server, [Requesting Model Metadata from the Model Server-gRPC Requests for Model Metadata](#)

setting up TensorFlow Serving, [Setting Up TensorFlow Serving-Building TensorFlow Serving from Source](#)

simple model server, [A Simple Model Server-Inefficient Model Inference](#)

challenges of deployment with Python apps, [The Downside of Model Deployments with Python-Based APIs](#)

TensorFlow Serving, [TensorFlow Serving](#)

architecture overview, [TensorFlow Architecture Overview](#)

exporting models for, [Exporting Models for TensorFlow Serving-Exporting Models for TensorFlow Serving](#)

model release management, [Thoughts on Model Release Management](#)

model signatures, [Model export](#), [Model Signatures-Signature Methods](#)

inspecting for exported models, [Inspecting Exported Models](#)

signature methods, [Signature Methods](#)

model training, [Model Training and Tuning](#), [Model Training-Summary](#)

defining model for example project, [Defining the Model for Our Example Project-Defining the Model for Our Example Project](#)

distribution strategies, [Distribution Strategies-Distribution Strategies](#)

encrypted, [Encrypted Model Training](#)

in machine learning pipelines, [Model Training](#)

TFX Trainer component, [The TFX Trainer Component-Using the SavedModel outside a pipeline](#)

other considerations, [Other Trainer Component Considerations-Using the SavedModel outside a pipeline](#)

running the Trainer, [Running the Trainer Component-Using the SavedModel outside a pipeline](#)

training multiple models simultaneously, [Training Multiple Models Simultaneously](#)

using differentially private optimizer, [Training with a Differentially Private Optimizer](#)

warm start, [Warm Starting Model Training](#)

with AI Platform job, [Training models with an AI Platform job](#)

model tuning, [Model Training and Tuning](#), [Model Tuning-Hyperparameter Tuning in TFX Pipelines](#)

hyperparameter tuning in TFX pipelines, [Hyperparameter Tuning in TFX Pipelines](#)

strategies for hyperparameter tuning, [Strategies for Hyperparameter Tuning](#)

model versioning, [Model Versioning](#)

model release management, [Thoughts on Model Release Management](#)

ModelSpec object, [Analyzing a Single Model in TFMA](#), [gRPC Requests for Model Metadata](#)

model\_config\_list function, [Multiple Model Configuration](#)

MSE (see mean squared error)

MultiWorkerMirroredStrategy, [Distribution Strategies](#)

MySQL database (Kubeflow Pipelines), [Introduction to Kubeflow Pipelines](#), [Pipeline Setup](#)

## N

natural language problems

text data for, preprocessing with TFT, [Text data for natural language problems](#)

text data for, strategies for ingesting, [Text Data for Natural Language Problems](#)

Netflix's movie recommendation system, [Feedback Loops in the Real World](#)

ngrams, [Text data for natural language problems](#)

node sizes for Kubernetes cluster, [Pipeline Setup](#)

numerical features

statistics generated about in TFDV, [Generating Statistics from Your Data](#)

tft.compute\_and\_apply\_vocabulary function, [TFT Functions](#)

Nvidia's Docker support for GPUs, [Docker Installation](#)

Nvidia, TensorRT, [Using TensorRT with TensorFlow Serving](#)  
(see also TensorRT)

## O

one-hot features, [Integrate TFT into Your Machine Learning Pipeline](#)  
converting TFT index to, [Integrate TFT into Your Machine Learning Pipeline](#)  
OneDeviceStrategy, [Distribution Strategies](#)  
OpFunc functions, [Pipeline Setup](#)  
orchestration (pipeline), [Pipeline Orchestration](#)  
alternatives to TFX, [Alternatives to TFX](#)  
orchestration of container replication, [Simple Scaling with TensorFlow Serving and Kubernetes](#)  
orchestration tools for pipelines, [What Is TFX?](#), [Pipelines Part 1: Apache Beam and Apache Airflow-Summary](#), [Pipelines Part 2: Kubeflow Pipelines](#)  
(see also Kubeflow Pipelines)  
Apache Airflow, [Introduction to Apache Airflow-Pipeline Execution](#)  
example, setting up basic pipeline, [Basic Airflow Example-Putting it all together](#)  
installation and setup, [Installation and Initial Setup](#)  
orchestrating TFX pipelines, [Orchestrating TFX Pipelines with Apache Airflow-Pipeline Execution](#)  
Apache Beam, [Introduction to Apache Beam-Orchestrating TFX Pipelines with Apache Beam](#)  
converting interactive TFX pipeline to production, [Converting Your Interactive TFX Pipeline to a Production Pipeline-Simple Interactive Pipeline Conversion for Beam and Airflow](#)  
simple interactive pipeline conversion for Apache Beam and Airflow, [Simple Interactive Pipeline Conversion for Beam and Airflow](#)  
deciding which tool to use, [Which Orchestration Tool to Choose?-Kubeflow Pipelines on AI Platform](#)  
Apache Airflow, [Apache Airflow](#)  
Apache Beam, [Apache Beam](#)  
Kubeflow Pipelines, [Kubeflow Pipelines](#)  
output configuration for split datasets, [Splitting one dataset into subsets](#)  
overall slice, [Analyzing a Single Model in TFMA](#)

Pachyderm, [Versioning Datasets](#)

ParameterServerStrategy, [Distribution Strategies](#)

Parquet-serialized data, converting to tf.Example, [Converting Parquet-serialized data to tf.Example](#)

partial dependence plots (PDPs), [Going Deeper with the What-If Tool](#)

assumption that all features are independent of each other,

[Generating Explanations with the WIT](#)

local, [Generating Explanations with the WIT](#)

payloads

for requests to TensorFlow Serving, [Payloads](#)

from TensorFlow Server on gRPC, [Payloads](#)

persistent volumes

exchanging data through in Kubeflow Pipelines, [Exchange Data](#)

[Through Persistent Volumes](#)

mounting with TFX, [Pipeline Setup](#)

output of TFX and Kubernetes Pipelines stored on, [Introduction to](#)

[Kubeflow Pipelines](#)

setups for Kubeflow Pipelines, [Persistent Volume Setups for](#)

[Kubeflow Pipelines](#)

personally identifying information (PII), [What Data Needs to Be Kept Private?](#)

Pipeline function (Apache Beam), [Orchestrating TFX Pipelines with Apache Beam](#)

pipeline object, setup for use on GCP, [Processing Large Datasets with GCP](#)

pipeline orchestration, [Pipeline Orchestration](#)

(see also orchestration (pipeline); orchestration tools for pipelines)

pipeline orchestrators, [Pipelines Part 1: Apache Beam and Apache Airflow](#)

(see also orchestration tools for pipelines)

port forwarding, creating to access Kubeflow Pipelines installation,

[Accessing Your Kubeflow Pipelines Installation](#)

POST requests, [Getting Model Predictions via REST](#)

precision, [Classification Metrics](#)

predict method, [Signature Methods](#)

preprocessing\_fn function (TFT), [Data Preprocessing with TFT](#),

[Preprocessing Strategies](#), [Integrate TFT into Your Machine Learning](#)

## Pipeline

best practices with, [Best Practices](#)

Presto databases, ingesting data from, [Presto databases](#)

principal component analysis (PCA), [TFT Functions](#)

privacy (see data privacy)

Private Aggregation of Teacher Ensembles (PATE), [Differential Privacy for Machine Learning](#)

project-specific configurations (Airflow), [Project-specific configurations](#)

Prometheus, [Monitoring Your TensorFlow Serving Instances](#)

configuring TensorFlow Server to provide metrics for, [TensorFlow Serving Configuration](#)

setting up and configuring, [Prometheus Setup](#)

protected groups (in US law), [Model Analysis for Fairness](#)

Protocol Buffers, use by gRPC APIs, [Payloads](#)

pruning models, [Pruning](#)

Publisher (TFX components), [Overview of TFX Components](#)

publish\_artifacts function, [Component drivers](#), [Component drivers](#)

Pusher component (TFX), [TFX Pusher Component](#), [Model Deployment with TFX Pipelines](#), [Exporting TFLite Models](#), [Converting Your Interactive TFX Pipeline to a Production Pipeline](#)

observing AI Platform configuration, [Serving models through AI Platform endpoints](#)

push destination, [Component specifications](#)

PySyft platform, [Federated Learning in TensorFlow](#)

## Q

quantization (model), [Quantization](#)

quasi-identifying data, [What Data Needs to Be Kept Private?](#), [Other Methods for Data Privacy](#)

## R

random search, [Strategies for Hyperparameter Tuning](#)

randomized response, [Differential Privacy](#)

ratio, defining when splitting datasets, [Splitting one dataset into subsets](#)

Real Time Messaging (RTM) protocol, [How to Use the Slack Component](#)  
recall, [Classification Metrics](#)

receiver function, [Exporting Models for TensorFlow Serving](#)

receiver operating characteristic (ROC), [Classification Metrics](#)

regress method, [Signature Methods](#)

regression metrics, [Regression Metrics](#)

regression models, getting predictions via gRPC API, [Getting predictions from classification and regression models](#)

regulatory requirements for data privacy, [Why Do We Care About Data Privacy?](#)

release management for models, [Thoughts on Model Release Management](#)

remote model loading, optimization of, [Optimization of Remote Model Loading](#)

Remote Procedure Calls (RPC) (see Google Remote Procedure Calls)

render\_slicing\_metrics function, [Analyzing a Single Model in TFMA](#)

requests Python library, [Getting Model Predictions via REST](#)

Resolver component, [Converting Your Interactive TFX Pipeline to a Production Pipeline](#)

ResolverNodes, [ResolverNode](#)

required for warm start model training, [Warm Starting Model Training](#)

resolve\_input\_artifacts method, [Component drivers](#)

REST APIs

endpoint configuration with TensorFlow Serving, [Single Model Configuration](#)

endpoint for TensorFlow Serving metrics provided to Prometheus, [TensorFlow Serving Configuration](#)

getting model predictions via, [Getting Model Predictions via REST](#)

randomly changing request URL for model inference in API for A/B testing, [Model A/B Testing with TensorFlow Serving](#)

request for model metadata via model server, [REST Requests for Model Metadata](#)

REST versus gRPC, [REST Versus gRPC-gRPC](#)



reusing existing TFX components, [Writing a Custom Component from Scratch](#)

ROC (see receiver operating characteristic)

routing tools for random routing of model inference on server side,

[Model A/B Testing with TensorFlow Serving](#)

RTM (Real Time Messaging) protocol, [How to Use the Slack Component](#)

run\_fn function, [run\\_fn\(\) Function-Model export](#)

adding callbacks to use TensorBoard, [Using TensorBoard in an Interactive Pipeline](#)

compiling and training the model, [Compile and train the model](#)

loading the data, [Load the data](#)

model export, [Model export](#)

## S

SavedModel format, [Model export](#), [Running the Trainer Component](#)

saved model as input for TFMA, [Analyzing a Single Model in TFMA](#)

using outside a pipeline, [Using the SavedModel outside a pipeline](#)

saved\_model\_cli, [Using TensorRT with TensorFlow Serving](#)

scheduler (Airflow), [Installation and Initial Setup](#)

schema

generating from data with TFDV, [Generating Schema from Your Data](#)

updating in TFDV, [Updating the Schema](#)

SchemaGen component, [Converting Your Interactive TFX Pipeline to a Production Pipeline](#)

schema generation by, [Integrating TFDV into Your Machine Learning Pipeline](#)

scores (output tensor), [Signature Methods](#)

SDK (Kubeflow Pipelines), [Introduction to Kubeflow Pipelines](#)

OpFunc functions, [Pipeline Setup](#)

search approaches (hyperparameter), [Strategies for Hyperparameter Tuning](#)

secure connections on gRPC, [Using TensorFlow Serving via gRPC](#)

secure multiparty computation (SMPC), [Encrypted Machine Learning](#)

Seldon, model deployment with, [Seldon](#)

selection bias, [Biased Datasets](#)

sensitive data, [What Data Needs to Be Kept Private?](#)

SerializeToString function, [gRPC Requests for Model Metadata](#)

serve\_tf\_examples\_fn function, [Model export](#)

service object (GCP), [Model inference](#)

SHAP (Shapley Additive Explanations) library, [Other Explainability Techniques](#)

Shapley values, [Other Explainability Techniques](#)

Simple Storage Service (S3), [Ingesting Remote Data Files](#)  
(see also AWS)

Simple TensorFlow Serving, [Simple TensorFlow Serving](#)

Scaffold, [TFX and Its Dependencies](#)  
configuration with build.yaml file, [Publishing Your Pipeline with TFX CLI](#)

skew and drift in data, [Data Skew and Drift](#)  
avoiding training-serving skew, [Avoiding a Training-Serving Skew](#)

Slack messenger and Human in the Loop component, [Human in the Loop-How to Use the Slack Component](#)

slicing data  
Fairness Indicators slicing visualization, [Checking Decision Thresholds with Fairness Indicators](#)  
render\_slicing\_metrics function, [Analyzing a Single Model in TFMA](#)  
slicing model predictions in TFMA, [Slicing Model Predictions in TFMA](#)  
in TensorFlow Data Validation, [Slicing Data in TFDV](#)

spanning datasets, [Spanning Datasets](#)

sparse features, converting to dense features for TFT, [Integrate TFT into Your Machine Learning Pipeline](#)

SparseTensors, [Data Preprocessing with TFT](#)

splitting datasets, [Splitting Datasets](#)  
default splits, [Splitting one dataset into subsets](#)  
in custom TFX component, [Component executors](#)  
preserving existing splits, [Preserving existing splits](#)

SQLite databases  
Airflow database, [Installation and Initial Setup](#)  
TFX database storing ML metadata, [Orchestrating TFX Pipelines with Apache Beam](#)

SSL, using for TensorFlow Serving on gRPC, [Using TensorFlow Serving via gRPC](#)

statistics

data validation for, [Why Data Validation?](#)

generating from data with TFDV, [Generating Statistics from Your Data](#)

StatisticsGen component, [Using our basic custom component](#),  
[Converting Your Interactive TFX Pipeline to a Production Pipeline](#)

structured data

converting to TFRecord data structures, [Converting your custom data to TFRecord data structures](#)

data ingestion strategies for, [Structured Data](#)

datasets stored in CSV files, [Converting comma-separated data to tf.Example](#)

ingesting from Google Cloud BigQuery, [Google Cloud BigQuery stubs](#), [Using TensorFlow Serving via gRPC](#)

## T

tag-sets (model), [Inspecting the Model](#)

tasks

Airflow task definitions, [Task definitions](#)

task dependencies in Airflow, [Task dependencies](#)

TensorBoard, [TensorFlow Model Analysis](#)

using Fairness Indicators via, [Checking Decision Thresholds with Fairness Indicators](#)

using in interactive pipeline, [Using TensorBoard in an Interactive Pipeline](#)

TensorFlow Data Validation (TFDV), [Data Validation](#), [TFDV-Generating Schema from Your Data](#)

generating schema from data, [Generating Schema from Your Data](#)

generating statistics from data, [Generating Statistics from Your Data](#)

installation, [Installation](#)

integrating into machine learning pipeline, [Integrating TFDV into Your Machine Learning Pipeline](#)

# TensorFlow Extended (TFX), [Introduction to TensorFlow Extended-Summary](#)

about, [What Is TFX?](#)

advanced, [Advanced TensorFlow Extended-Summary](#)

exporting TFLite models, [Exporting TFLite Models](#)

Human in the Loop component, [Human in the Loop-How to Use the Slack Component](#)

ongoing developments, [Advanced TensorFlow Extended](#)

training multiple models simultaneously, [Training Multiple Models Simultaneously](#)

warm start model training, [Warm Starting Model Training](#)

alternatives to, [Alternatives to TFX](#)

base pipeline, [Converting Your Interactive TFX Pipeline to a Production Pipeline](#)

command-line interface (CLI), [TFX Command-Line Interface](#)

publishing pipeline to Kubeflow Pipelines, [Publishing Your Pipeline with TFX CLI](#)

TFX and its dependencies, [TFX and Its Dependencies](#)

TFX templates, [TFX Templates](#)

custom components, [Custom TFX Components-Using our custom executor](#)

use cases for, [Use Cases of Custom Components](#)

writing from scratch, [Writing a Custom Component from Scratch-Implementation review](#)

custom Docker image for, [Custom TFX Images](#)

hyperparameter tuning in TFX pipelines, [Hyperparameter Tuning in TFX Pipelines](#)

installing, [Installing TFX](#)

interactive pipelines, [Interactive Pipelines-Interactive Pipelines](#)

ML Metadata, [What Is ML Metadata?](#)

model analysis and validation in, [Analysis and Validation in TFX-Summary](#)

Evaluator component, [Evaluator Component](#)

Pusher component, [TFX Pusher Component](#)

ResolverNode component, [ResolverNode](#)

validation in Evaluator component, [Validation in the Evaluator Component](#)

model deployment with TFX pipelines, [Model Deployment with TFX Pipelines](#)

overview of components, [What Is TFX?](#), [Overview of TFX Components](#)

use with other ML frameworks, [TFX with Other Machine Learning Frameworks](#)

TensorFlow Federated (TFF), [Federated Learning in TensorFlow](#)  
TensorFlow Lite

analysis of models in TFMA, [Analyzing a Single Model in TFMA](#)

exporting TFLite models, [Exporting TFLite Models](#)

limitations of, [Exporting TFLite Models](#)

models, serving with TensorFlow Serving, [Serving TFLite Models with TensorFlow Serving](#)

steps to optimizing your models, [Steps to Optimize Your Model with TFLite](#)

using to optimize machine learning models, [TFLite](#)

TensorFlow Model Analysis (TFMA), [Model Analysis and Validation](#),  
[TensorFlow Model Analysis-Analyzing Multiple Models in TFMA](#)

analyzing a single model, [Analyzing a Single Model in TFMA](#)

analyzing multiple models, [Analyzing Multiple Models in TFMA](#)

calculating metrics for set of decision thresholds, [Checking Decision Thresholds with Fairness Indicators](#)

slicing model predictions in, [Slicing Model Predictions in TFMA](#)

use of library in TFX Evaluator component, [Evaluator Component](#)

TensorFlow Privacy, [Introduction to TensorFlow Privacy-Calculating Epsilon](#)

calculating epsilon, [Calculating Epsilon](#)

training with a differentially private optimizer, [Training with a Differentially Private Optimizer](#)

TensorFlow Serving, [TensorFlow Serving](#)

alternatives to, [TensorFlow Serving Alternatives](#)

architecture overview, [TensorFlow Architecture Overview](#)

batching inference requests, [Batching Inference Requests-Configuring Batch Predictions](#)

configuring batch predictions, [Configuring Batch Predictions](#)

configuring a server, [Configuring a TensorFlow Server-Multiple Model Configuration](#)

multiple model configuration, [Multiple Model Configuration](#)  
single model configuration, [Single Model Configuration](#)  
decoupling deployment cycles, [Decoupling Deployment Cycles-  
Optimization of Remote Model Loading](#)  
accessing models from AWS S3 buckets, [Accessing private models from AWS S3](#)  
accessing models from GCP buckets, [Accessing private models from GCP Buckets](#)  
optimizing remote model loading, [Optimization of Remote Model Loading](#)  
workflow overview, [Workflow Overview](#)  
deploying models produced by TFX pipelines, [Model Deployment with TFX Pipelines](#)  
exporting models for, [Exporting Models for TensorFlow Serving-  
Exporting Models for TensorFlow Serving](#)  
loading models from remote storage drives, [Decoupling Deployment Cycles](#)  
making predictions from the model server, [Making Predictions from the Model Server-Payloads](#)  
using TensorFlow Serving via gRPC, [Using TensorFlow Serving via gRPC-Payloads](#)  
model A/B testing with, [Model A/B Testing with TensorFlow Serving](#)  
model hot swap capability, [Single Model Configuration](#)  
model optimizations for deployment, [Model Optimizations for Deployments-Using TensorRT with TensorFlow Serving](#)  
distillation, [Distillation](#)  
pruning, [Pruning](#)  
quantization, [Quantization](#)  
monitoring instances, [Monitoring Your TensorFlow Serving Instances-Simple Scaling with TensorFlow Serving and Kubernetes](#)  
configuring TensorFlow Serving to provide metrics to Prometheus, [TensorFlow Serving Configuration](#)  
Prometheus setup, [Prometheus Setup](#)  
other optimizations, [Other TensorFlow Serving Optimizations](#)  
setting up, [Setting Up TensorFlow Serving-Building TensorFlow Serving from Source](#)  
building from source, [Building TensorFlow Serving from Source](#)

Docker installation, [Docker Installation](#)  
native Ubuntu installation, [Native Ubuntu Installation](#)  
simple scaling with Kubernetes and, [Simple Scaling with TensorFlow Serving and Kubernetes-Summary](#)  
using TensorRT with, [Using TensorRT with TensorFlow Serving-Serving TFLite Models with TensorFlow Serving](#)  
TensorFlow Lite, [TFLite](#)  
TensorFlow Tensors, [Data Preprocessing with TFT](#)  
TensorFlow Text library, [Text data for natural language problems](#)  
TensorFlow Transform (TFT), [Data Preprocessing](#)  
(see also data preprocessing)  
avoiding training-serving skew with, [Avoiding a Training-Serving Skew](#)  
best practices with, [Best Practices](#)  
eager execution prohibited in preprocessing, [Best Practices](#)  
functions in, [Data Preprocessing with TFT](#), [TFT Functions](#)  
installation, [Installation](#)  
integrating into machine learning pipeline, [Integrate TFT into Your Machine Learning Pipeline](#)  
standalone execution, [Standalone Execution of TFT](#)  
tensorflow\_hub library, [Custom TFX Images](#)  
tensorflow\_model\_server command, [Single Model Configuration](#)  
TensorRT, using with TensorFlow Serving, [Using TensorRT with TensorFlow Serving-Serving TFLite Models with TensorFlow Serving](#)  
testing  
for exported models, [Testing the Model](#)  
of machine learning models, [Testing Machine Learning Models](#)  
model A/B testing with TensorFlow Serving, [Model A/B Testing with TensorFlow Serving](#)  
text data for natural language problems  
converting in TFT, [Text data for natural language problems](#)  
strategies for ingesting, [Text Data for Natural Language Problems](#)  
text input features, preprocessing in TFT, [Integrate TFT into Your Machine Learning Pipeline](#)  
tf. prefix for TensorFlow operations, [Data Preprocessing with TFT](#)  
tf.data, difference from tf.transform, [Checking Your Preprocessing Results in Your Pipeline](#)



tf.Example, [Model export](#), [Inspecting the Model](#)  
conversion of JSON data structures submitted to REST to, [gRPC](#)  
converting Avro-serialized data to, [Converting Avro-serialized data to tf.Example](#)  
converting CSV data to, [Converting comma-separated data to tf.Example](#)  
converting Parquet-serialized data to, [Converting Parquet-serialized data to tf.Example](#)

tf.Features object, [Converting your custom data to TFRecord data structures](#)

tf.float32 data type, [Standalone Execution of TFT](#)

tf.images API, [Image data for computer vision problems](#)

tf.io API, [Image data for computer vision problems](#)

tf.Keras models, [Model Training](#)

tf.make\_tensor\_proto function, [Payloads](#)

tf.reshape function, [Image data for computer vision problems](#)

tf.transform, difference from tf.data, [Checking Your Preprocessing Results in Your Pipeline](#)

TFIDF (term frequency inverse document frequency), [Text data for natural language problems](#)

TFLite (see TensorFlow Lite)

TFRecord, [Concepts for Data Ingestion](#)

TFRecord data structures  
converting your custom data to, [Converting your custom data to TFRecord data structures](#)

TFRecord files  
existing, importing, [Importing existing TFRecord Files](#)

TFRecords  
benefits of storing data as, [Concepts for Data Ingestion](#)  
evaluation dataset in, [Analyzing a Single Model in TFMA](#)  
use in TFDV, [TFDV](#)

tft.apply\_saved\_model function, [TFT Functions](#)

tft.bag\_of\_words function, [Text data for natural language problems](#)

tft.bucketize function, [TFT Functions](#)

tft.compute\_and\_apply\_vocabulary function, [TFT Functions](#)

tft.ngrams function, [Text data for natural language problems](#)

tft.pca function, [TFT Functions](#)

tft.scale\_to\_z\_score function, [TFT Functions](#)

tft.tfidf function, [Text data for natural language problems](#)

tft\_beam.Context, [Standalone Execution of TFT](#)

TFX Trainer component (see Trainer component)

TPUStrategy, [Distribution Strategies](#)

Trainer component, [The TFX Trainer Component-Using the SavedModel outside a pipeline](#), [Converting Your Interactive TFX Pipeline to a Production Pipeline](#)

distribution strategies and, [Distribution Strategies](#)

observing AI Platform configuration, [Training models with an AI Platform job](#)

other considerations, [Other Trainer Component Considerations-Using the SavedModel outside a pipeline](#)

using SavedModel outside a pipeline, [Using the SavedModel outside a pipeline](#)

using Trainer with Estimator model, [Using the Trainer component with an Estimator model](#)

required input in our project, [The TFX Trainer Component Resolver passing latest trained model to, Warm Starting Model Training](#)

running, [Running the Trainer Component-Using the SavedModel outside a pipeline](#)

run\_fn function, [run\\_fn\(\) Function-Model export](#)

compiling and training the model, [Compile and train the model](#)

loading the data, [Load the data](#)

model export, [Model export](#)

train\_args, [Component specifications](#)

training datasets, [Splitting one dataset into subsets](#)

comparing for data drift in TFDV, [Data Skew and Drift](#)

comparing to validation dataset in TFDV, [Comparing Datasets](#)

training steps (TFX Trainer), [Compile and train the model](#)

training-serving skew, [Avoiding a Training-Serving Skew](#)

transfer learning, [The Data Flywheel](#)

Transform component, [Converting Your Interactive TFX Pipeline to a Production Pipeline](#)

transformations (Apache Beam), [Basic Data Pipeline](#)

true negatives, [Classification Metrics](#)

true positives, [Classification Metrics](#)

true positive rate (TPR), [Classification Metrics](#)

Tuner component, [Hyperparameter Tuning in TFX Pipelines](#)

tuning models (see model tuning)

## U

Ubuntu

installing TensorFlow Serving on, [Native Ubuntu Installation](#)

packages for TensorFlow Serving, [Native Ubuntu Installation](#)

URLs

for HTTP requests to model server, [URL structure](#)

randomly changing request URL for model inference in REST API,

[Model A/B Testing with TensorFlow Serving](#)

## V

valency, [Generating Schema from Your Data](#)

vector representations

bag-of-words vector, [Text data for natural language problems](#)

defining difference between two vectors, [Data Skew and Drift](#)

features embedded from string to vector representation, [Integrate](#)

[TFT into Your Machine Learning Pipeline](#)

TFIDF, [Text data for natural language problems](#)

version labels, [Multiple Model Configuration](#)

versioning

configuring specific model versions for TensorFlow Serving,

[Multiple Model Configuration](#)

lack of model version control in deployment with Python app, [Lack](#)

[of Model Version Control](#)

of datasets, [Versioning Datasets](#)

TensorFlow Serving loading model with highest version, [Single](#)

[Model Configuration](#)

visualizations

basic TensorBoard visualization, [Using TensorBoard in an](#)

[Interactive Pipeline](#)

offered in TensorFlow Data Validation, [TFDV](#)

TFMA visualization comparing two models, [Analyzing Multiple Models in TFMA](#)

## W

warm start model training, [Warm Starting Model Training](#)

Weights and Biases, experiment tracking in, [Model Experiment Tracking](#)

What-If Tool (WIT), [Going Deeper with the What-If Tool-Going Deeper with the What-If Tool](#)

counterfactuals, [Going Deeper with the What-If Tool](#)

decision thresholds, [Going Deeper with the What-If Tool](#)

extra steps for running in standalone notebook, [Going Deeper with the What-If Tool](#)

generating model explanations with, [Generating Explanations with the WIT-Generating Explanations with the WIT](#)

partial dependence plots, [Going Deeper with the What-If Tool](#)

## Y

YouTube's recommendation system, [Feedback Loops in the Real World](#)

## Z

zip codes represented as strings, converting to float values, [Integrate TFT into Your Machine Learning Pipeline](#)

[Support](#)   [Sign Out](#)