Implementation using TF and TFX

The testing and debugging guidelines in this course can be complex to implement. You can implement some of the guidelines using TensorFlow and **TensorFlow Extended (TFX)**. TFX is an end-to-end ML pipeline based on TensorFlow. For a demo, view this end-to-end TFX example (https://github.com/tensorflow/tfx/blob/master/tfx/examples/chicago_taxi_pipeline/README.md). To complement the end-to-end example, the following table lists available resources in TF and TFX by guideline. Only guidelines supported by TF or TFX are listed.

Guideline	TF/TFX Implementation
Guidelines for debugging your ML model	
Exploring your data to understand it	Explore your data using Pandas or Facets. • Pandas: Data exploration example (/machine-learning/data-prep/programming-exercise). • Facets: See the Facets GitHub page (https://github.com/PAIR-code/facets).
Validating input data using a data schema	Use TensorFlow Data Validation (https://www.tensorflow.org/tfx/data_validation/).
Implementing tests for ML code	First, debug your TF models with Eager Execution (https://www.tensorflow.org/guide/eager). Then write tests with Tensorflow Testing (https://www.tensorflow.org/api_guides/python/test).
Metrics	
Generating model metrics	TensorBoard visualizes your TF graph and plots metrics. See Tensorboard: Graph Visualization (https://www.tensorflow.org/guide/graph_viz).
Deployment to Pipeline	
Testing model quality in production	Use Tensorflow Model Analysis (https://github.com/tensorflow/model-analysis).
Checking for training-serving skew	Avoid feature skew by sharing feature engineering code across training and serving by using TFX Transform (https://www.tensorflow.org/tfx/transform/get_started).
Tracking model staleness	--

Previous
← Check Your Understanding
(/machine-learning/testing-debugging/pipeline/check-your-understanding)

Next
Summary and Next Steps (/machine-learning/testing-debugging/summary) →

Last updated 2020-02-05 UTC.