DeUmbra

Moving Your Machine Learning Models to Production with TensorFlow Extended

Jonathan Mugan

Moving From Our Hut to the Production Floor

Your model is going to live for a long time. Not just for a demo.

You must know when to update it. The world changes.

You must ensure production data matches training data. Data reflects its origins.

You may need to track multiple model versions. E.g., for different states.

You need to batch the input to serving. One-at-a-time is slow.

Interchangeable Parts and the ML Revolution

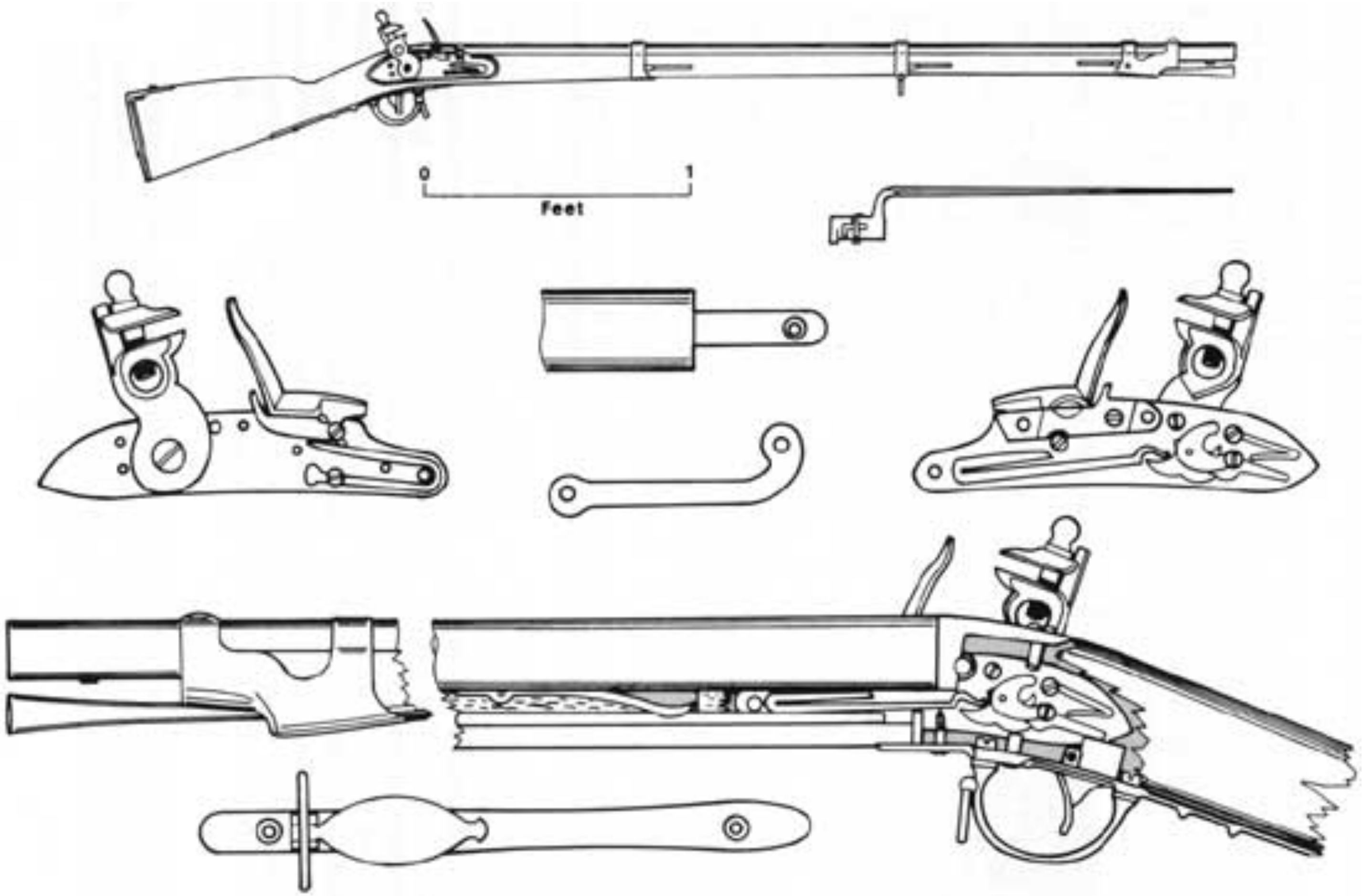


Image by https://www.flickr.com/photos/36224933@N07
https://creativecommons.org/licenses/by-sa/2.0/deed.en

Data Ingestion
(ExampleGen)

↓

TensorFlow Data Validation
(StatisticsGen, SchemaGen, Example Validator)

↓

TensorFlow Transform
(Transform)

↓

Estimator or Keras Model
(Trainer)

↓

TensorFlow Model Analysis
(Evaluator, Model Validator)

↓

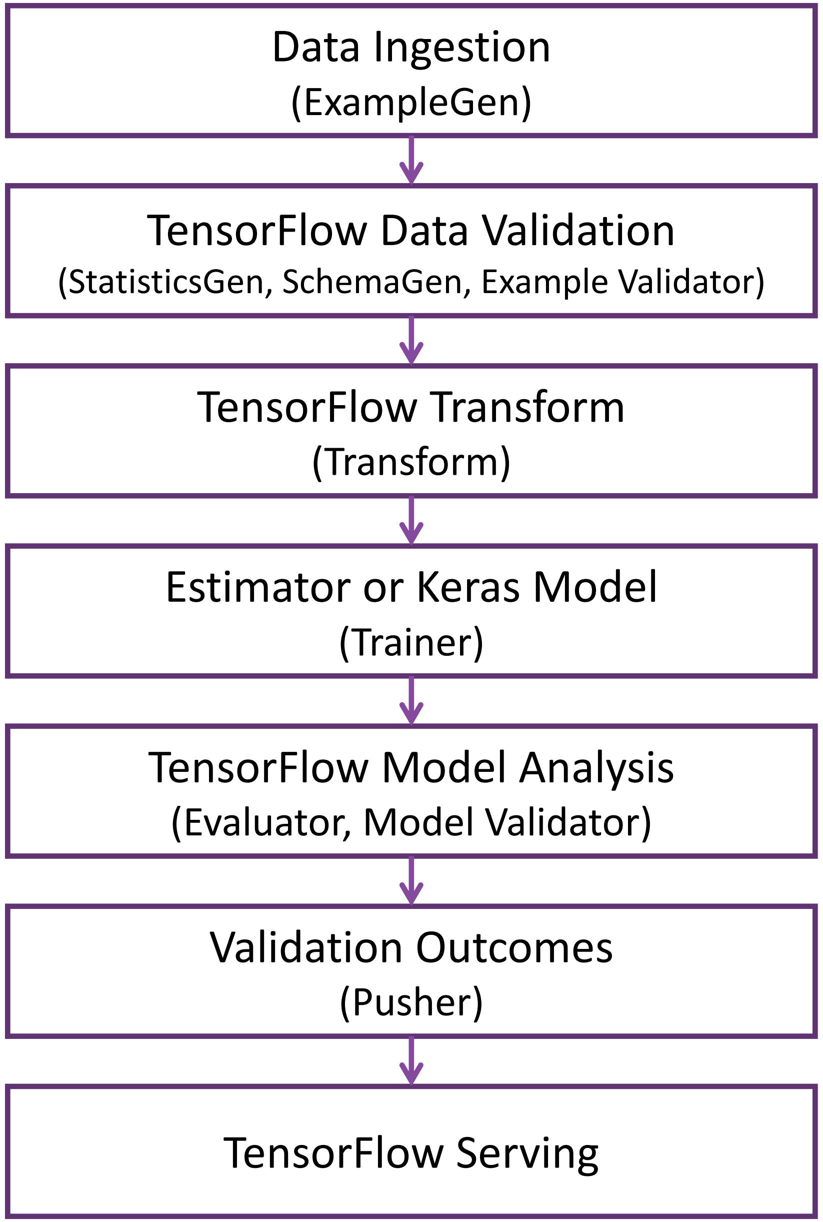
Validation Outcomes
(Pusher)

↓

TensorFlow Serving

Interchangeable Parts and the ML Revolution

- TensorFlow Extended (TFX)
- TFX used internally by Google and recently open sourced
- Represents your pipeline to production as a sequence of components
- Building any one model is more work, but for large endeavors, TFX helps to keep you organized



Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

TensorFlow Extended

- Individual components talk to the metadata store (MLMD).
- MLMD doesn't store data itself. It stores data about data.

Data Ingestion
(ExampleGen)

TensorFlow Data Validation
(StatisticsGen, SchemaGen, Example Validator)

TensorFlow Transform
(Transform)

Estimator or Keras Model
(Trainer)

TensorFlow Model Analysis
(Evaluator, Model Validator)

Validation Outcomes
(Pusher)

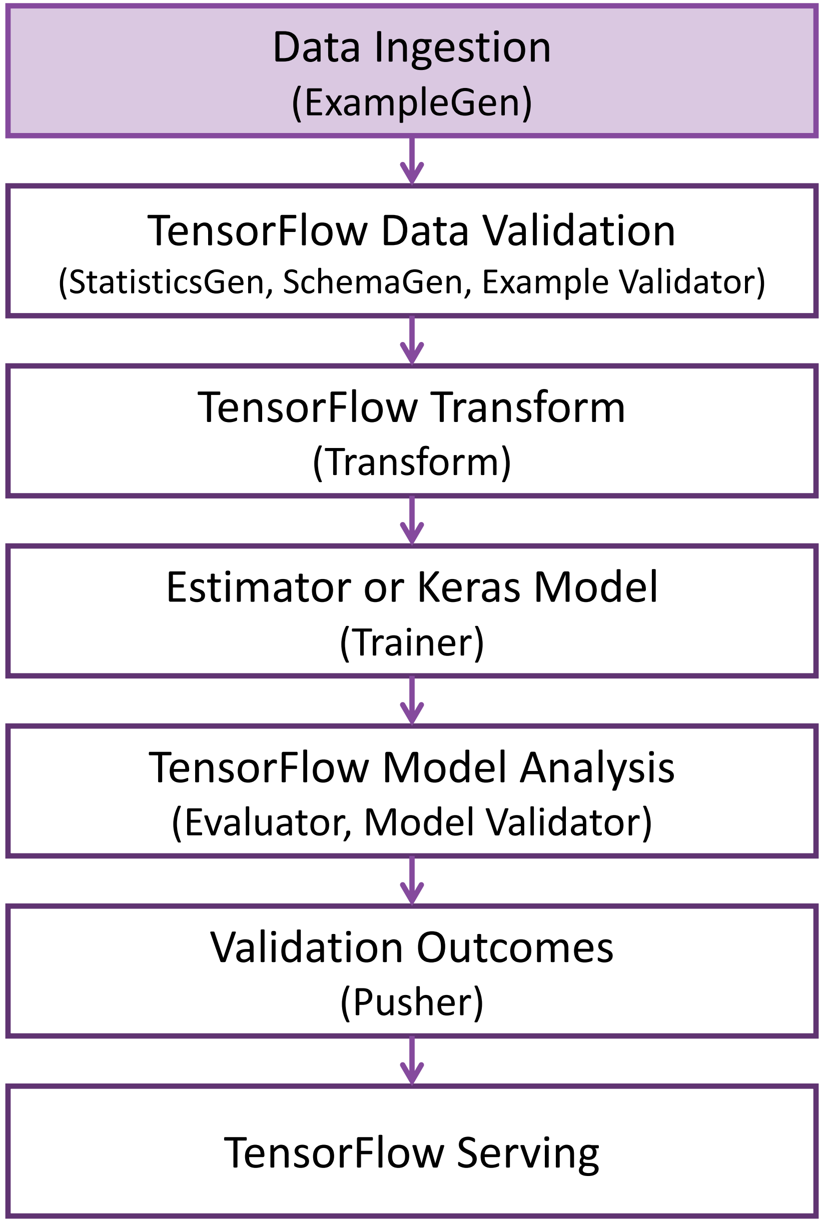
TensorFlow Serving

ML Metadata (MLMD)

Organized by library (components in parenthesis)

Each component has three pieces
1. Driver (gets information from MLMD)
2. Executor (does what the component does)
3. Publisher (writes results to MLMD)

https://www.tensorflow.org/tfx/guide
https://www.tensorflow.org/tfx/guide/mlmd

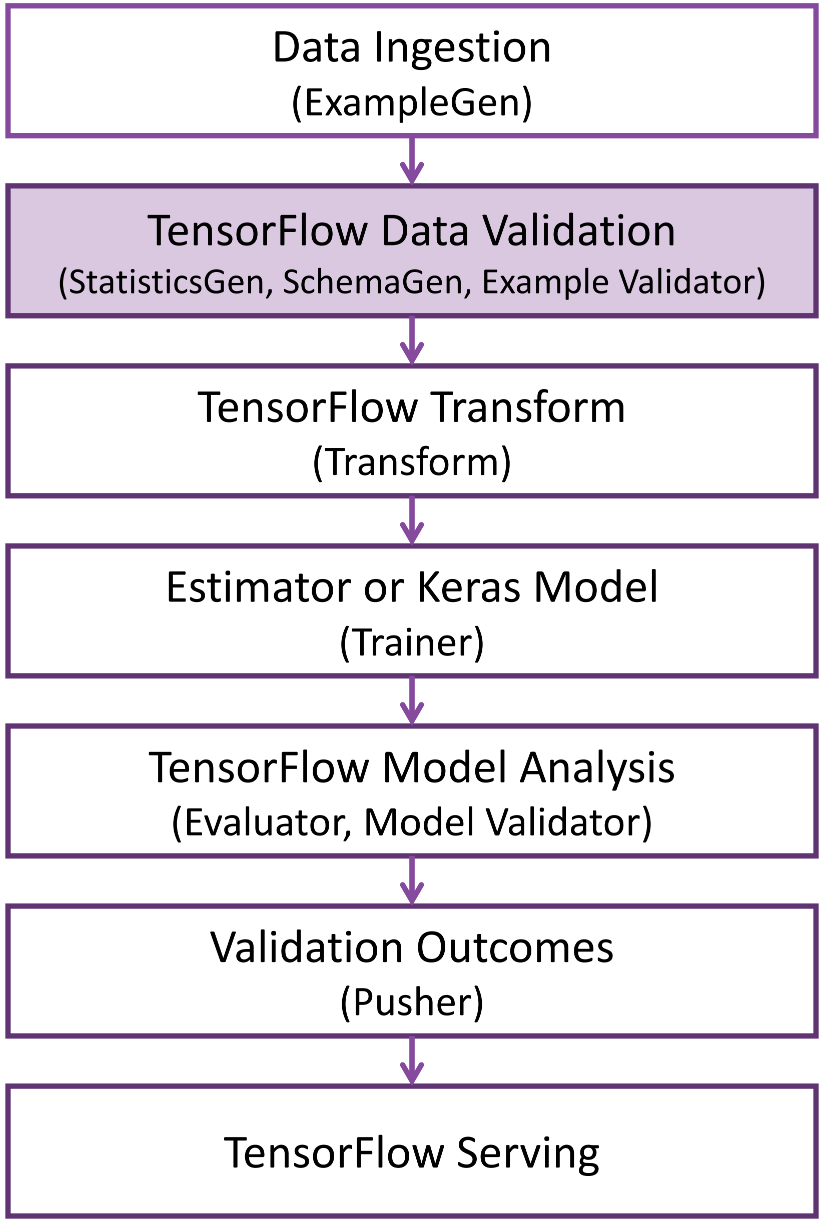


Data Ingestion: ExampleGen

- Pulls in your data and put it into binary format
- Also splits it into train and test

- Protocol Buffers
- `tf.Example` into a `TFRecord` file

https://www.tensorflow.org/tutorials/load_data/tfrecord
https://www.tensorflow.org/tfx/guide/examplegen



TensorFlow Data Validation: StatisticsGen, SchemaGen, Example Validator

Looks at your data and generates a schema, which you manually update.

It makes sure the data you pass in later during serving is still in the same format and hasn’t drifted.

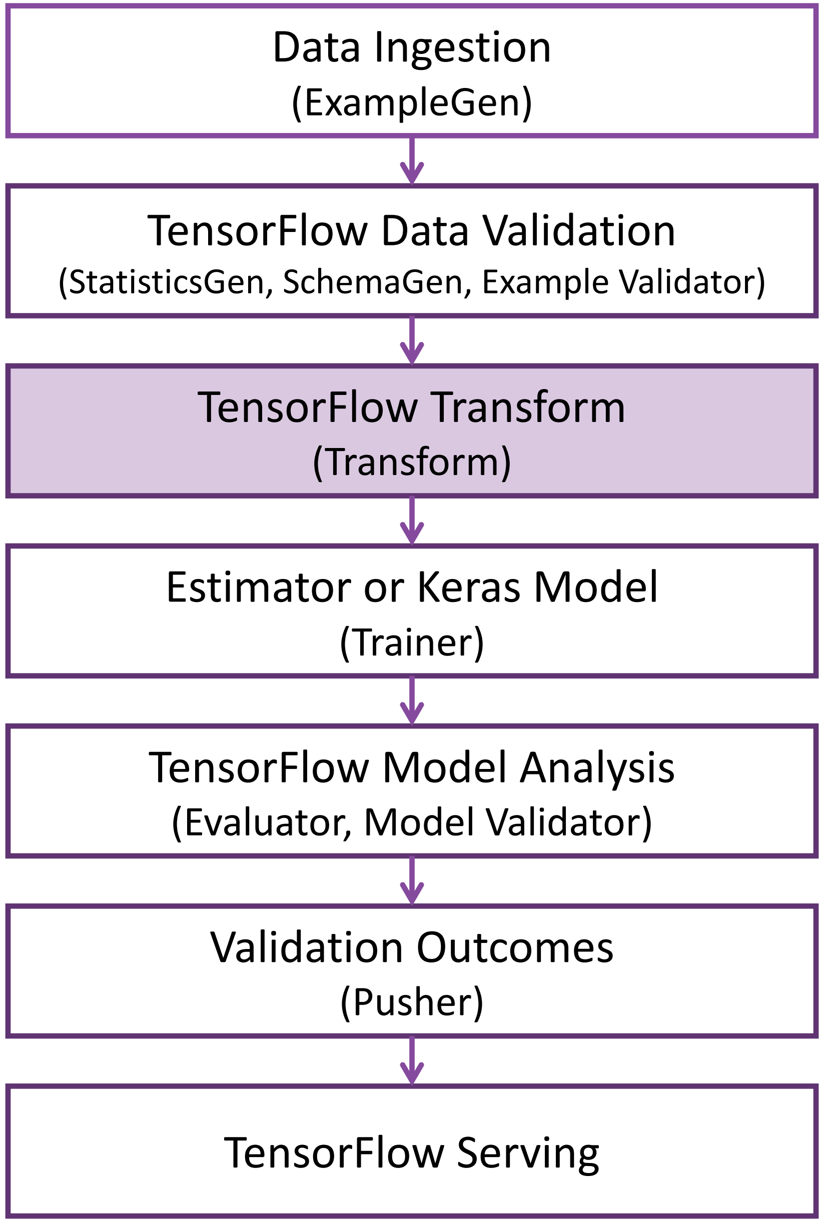
Also has a great way to visualize data, FACETS, we will see later.

https://www.tensorflow.org/tfx/guide/tfdv

- Data Ingestion (ExampleGen)
- **TensorFlow Data Validation (StatisticsGen, SchemaGen, Example Validator)**
- TensorFlow Transform (Transform)
- Estimator or Keras Model (Trainer)
- TensorFlow Model Analysis (Evaluator, Model Validator)
- Validation Outcomes (Pusher)
- TensorFlow Serving

Example Schema

	Type	**Presence**	**Valency**	**Domain**
Feature name				
'race'	STRING	required	single	'race'
'gender'	STRING	required	single	'gender'
'diabetes_comp'	INT	required	single	-
'metastic_cancer'	INT	required	single	-
'perc_hs_grad'	FLOAT	required	single	-
'mh_30d_before_opioids'	INT	required	single	-
'recent_mme'	FLOAT	required	single	-
'drug_screen_gt90d_lt120mme'	INT	required	single	-
'weight_loss'	INT	required	single	-



TensorFlow Transform: Transform

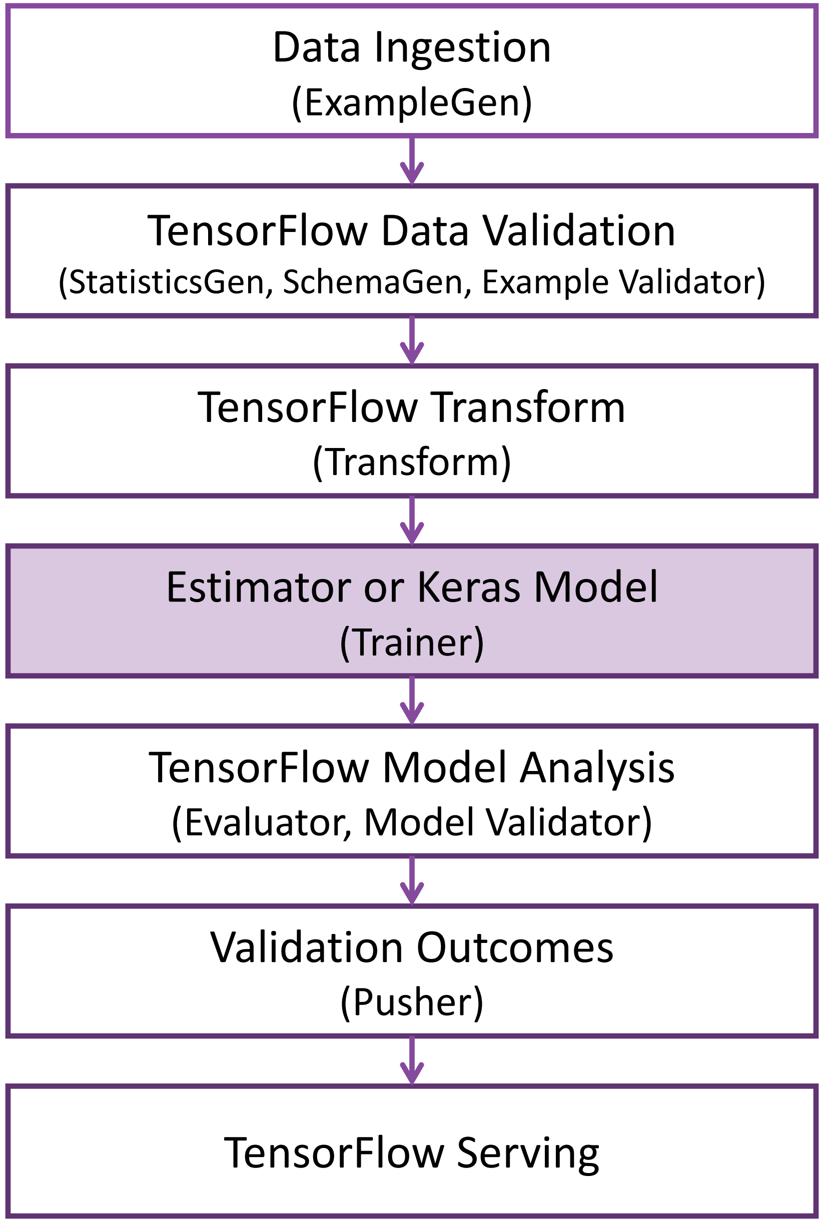
Converts your data

- E.g., One-hot encoding, categorical with a vocab
- Part of TensorFlow graph, for better or worse
- Good for transformations that require looking at all values

Example: `tft.scale_to_z_score`
subtracts mean and divides by standard deviation

Features come in many types, and TensorFlow Transform converts them into a format that can be ingested by a machine learning model.

Nice to have this explicit.
https://www.tensorflow.org/tfx/guide/transform

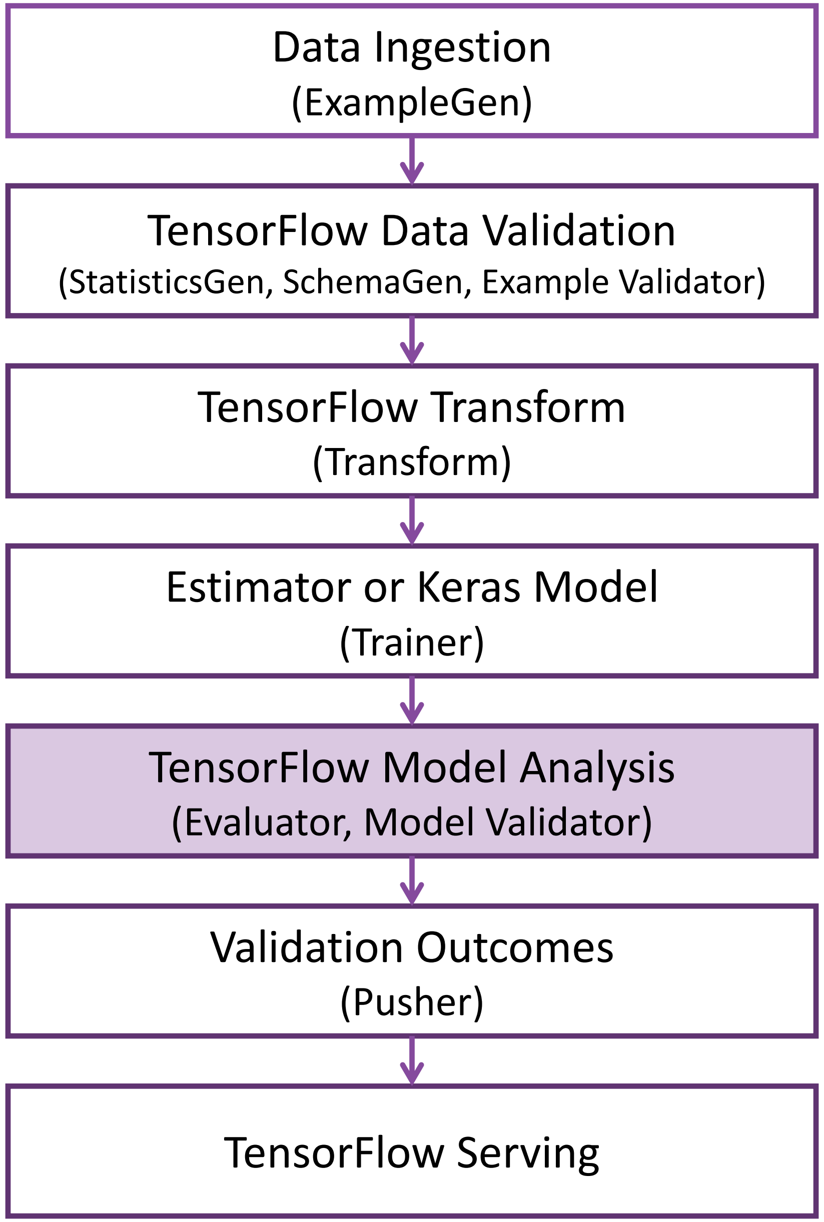


Estimator or Keras Model: Trainer

- Trains the model: Part we are all familiar with
- Except uses an Estimator
- Can use KERAS

```
tf.keras.estimator.model_to_estimator()
```

https://www.tensorflow.org/tfx/guide/trainer



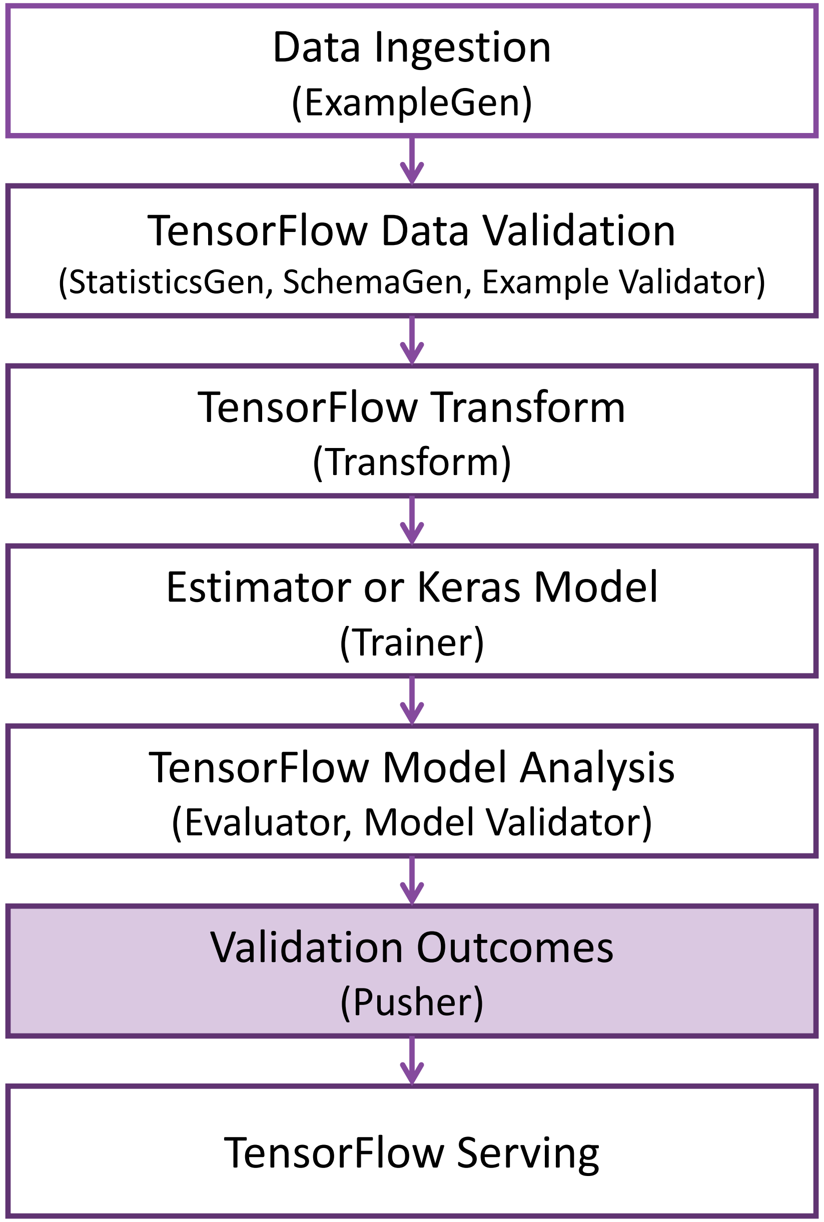
TensorFlow Model Analysis: Evaluator, Model Validator

Evaluator Component

- Evaluates the model.
- Uses TensorFlow Model Analysis (TFMA), which we will see shortly.
- https://www.tensorflow.org/tfx/guide/evaluator

Model Validator Component

- You set a baseline (such as the current serving model) and a metric (such as AUC)
- Marks in the metadata if the model passes the baseline.
- https://www.tensorflow.org/tfx/guide/modelval

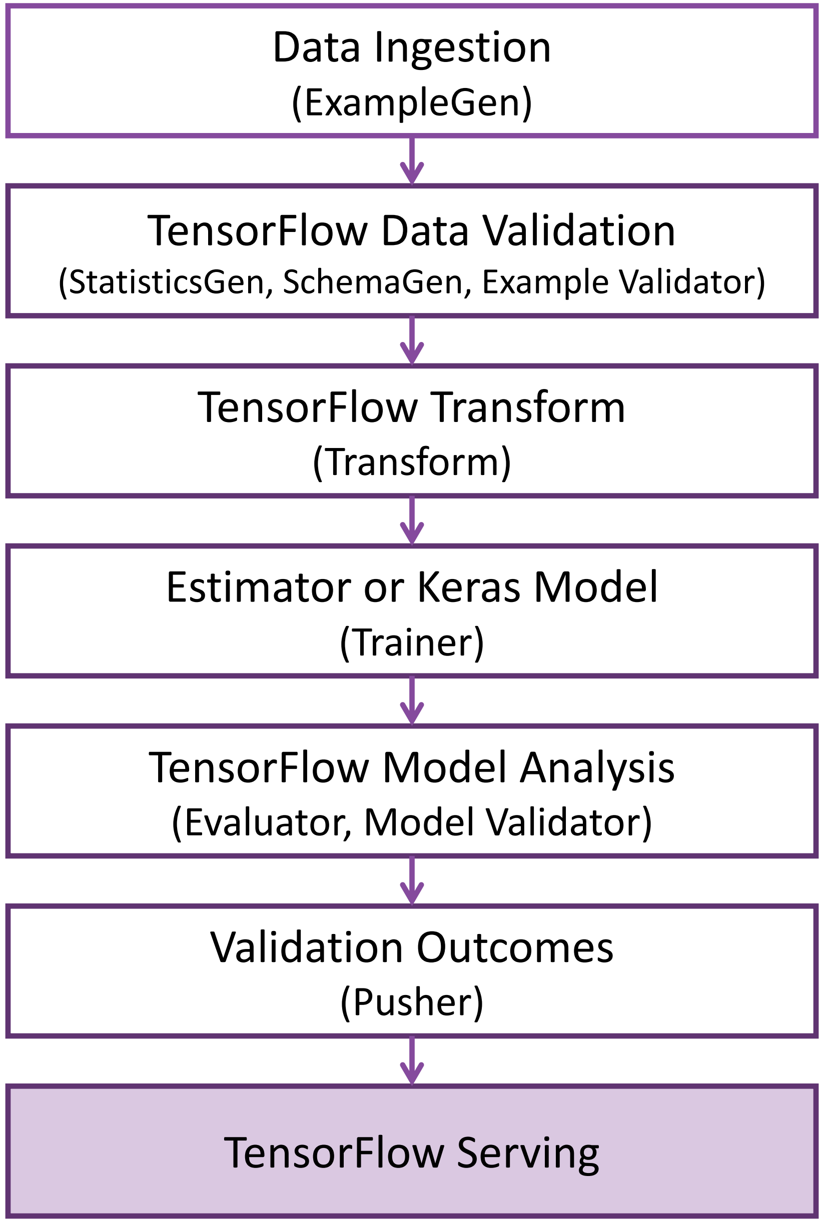


Validation Outcomes: Pusher

- Pushes the model to Serving if it is validated
- I.e., if your new model is better than the existing model, push it to the model server.

For a deeper understanding, see Ice-T's 1988 hit song, "I'm Your Pusher"

https://www.tensorflow.org/tfx/guide/pusher



TensorFlow Serving

- Uses the model to perform inference
- Called via gRPC APIs or RESTFUL APIs
- Easy to get running with Docker
- You can call a particular version of a model
- Takes care of batching

https://www.tensorflow.org/tfx/guide/serving

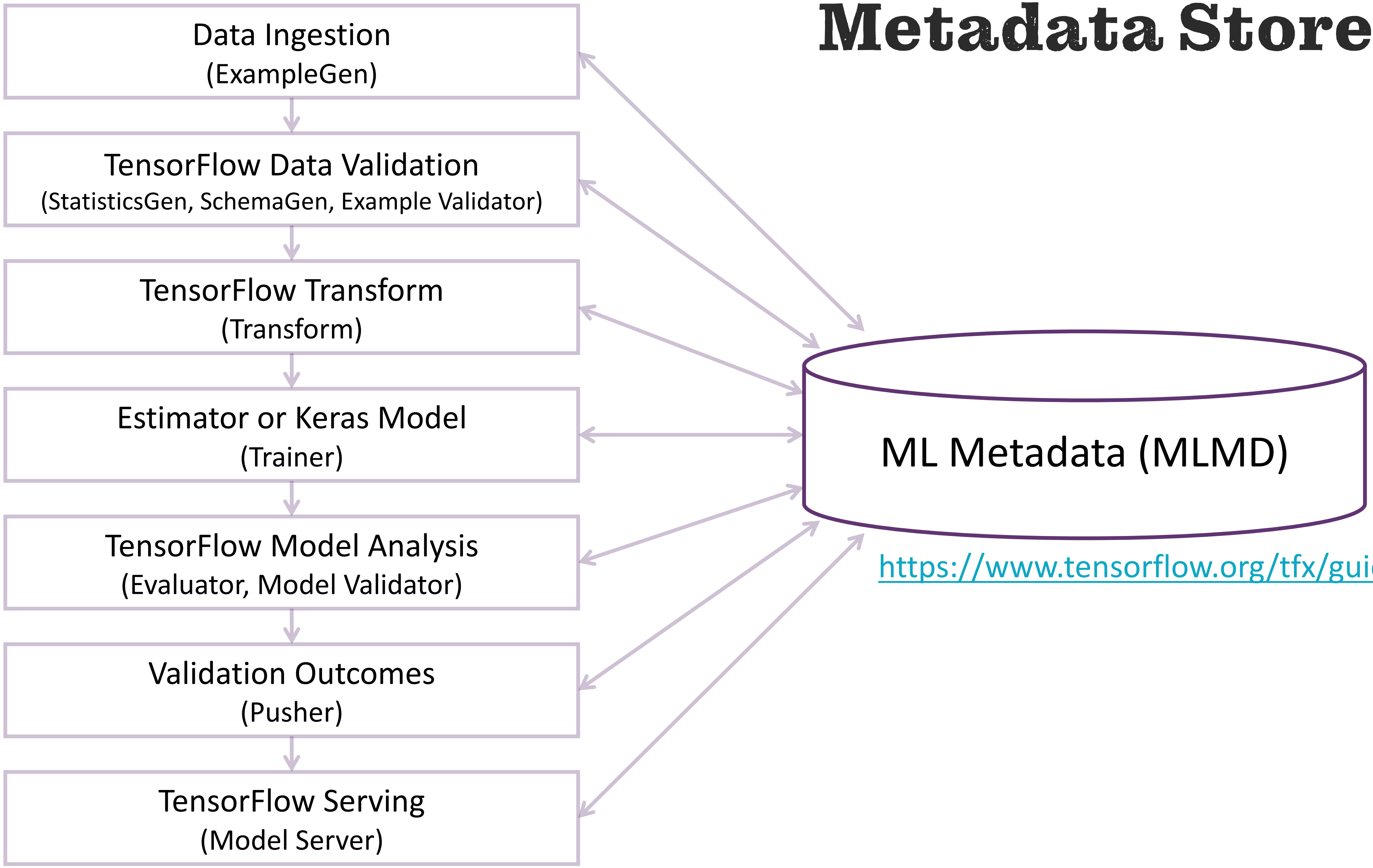
Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
  - ML Metadata
  - Apache Airflow
- TensorFlow and TensorFlow Tools
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

Metadata Store



https://www.tensorflow.org/tfx/guide/mlmd

Table: Artifact

	id	type_id	uri
	Filter	Filter	Filter
1	1	1	/home/jmugan/data/otf/tfx/data_rx_od_chronic
2	2	3	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/CsvExampleGen/examples/1/train/
3	3	3	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/CsvExampleGen/examples/1/eval/
4	4	5	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/StatisticsGen/output/2/train/
5	5	5	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/StatisticsGen/output/2/eval/
6	6	7	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/SchemaGen/output/3/
7	7	10	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/ExampleValidator/output/5/
8	8	11	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/Transform/transform_output/6/
9	9	3	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/Transform/transformed_examples/6/train/
10	10	3	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/Transform/transformed_examples/6/eval/
11	11	13	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/Trainer/output/7/
12	12	15	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/ModelValidator/blessing/8/
13	13	16	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/ModelValidator/results/8/
14	14	18	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/Evaluator/output/9/
15	15	20	/home/jmugan/airflow/tfx/pipelines/otf_[illegible]d_chronic/Pusher/model_push/10/

Looking at the table Artifact using the DB Browser for SQLite

Table: Type

	id	name	is_artifact_type	input_type	output_type
	Filter	Filter	Filter	Filter	Filter
1	1	ExternalPath	1	NULL	NULL
2	2	CsvExampleGen	0	NULL	NULL
3	3	ExamplesPath	1	NULL	NULL
4	4	StatisticsGen	0	NULL	NULL
5	5	ExampleStatisticsPath	1	NULL	NULL
6	6	SchemaGen	0	NULL	NULL
7	7	SchemaPath	1	NULL	NULL
8	8	Transform	0	NULL	NULL
9	9	ExampleValidator	0	NULL	NULL
10	10	ExampleValidationPath	1	NULL	NULL
11	11	TransformPath	1	NULL	NULL
12	12	Trainer	0	NULL	NULL
13	13	ModelExportPath	1	NULL	NULL
14	14	ModelValidator	0	NULL	NULL
15	15	ModelBlessingPath	1	NULL	NULL
16	16	ModelValidationPath	1	NULL	NULL
17	17	Evaluator	0	NULL	NULL
18	18	ModelEvalPath	1	NULL	NULL
19	19	Pusher	0	NULL	NULL
20	20	ModelPushPath	1	NULL	NULL

The Types of Artifacts

The type_id field from the previous slide maps here

Table: ArtifactProperty

	artifact_id	name	is_custom_property	int_value	double_value	string_value
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	type_name	0	NULL	NULL	ExternalPath
2	1	split	0	NULL	NULL	
3	1	state	0	NULL	NULL	published
4	1	span	0	1	NULL	NULL
5	2	type_name	0	NULL	NULL	ExamplesPath
6	2	split	0	NULL	NULL	train
7	2	state	0	NULL	NULL	published
8	2	span	0	1	NULL	NULL
9	3	split	0	NULL	NULL	eval
10	3	state	0	NULL	NULL	published
11	3	span	0	1	NULL	NULL
12	3	type_name	0	NULL	NULL	ExamplesPath
13	4	split	0	NULL	NULL	train

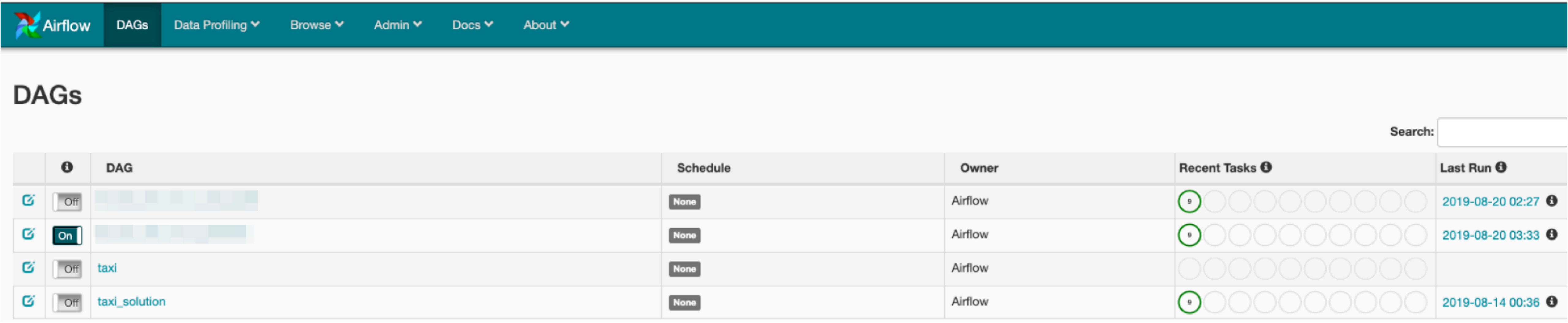
You can see the properties of the artifacts in the ArtifactProperty table

Outline

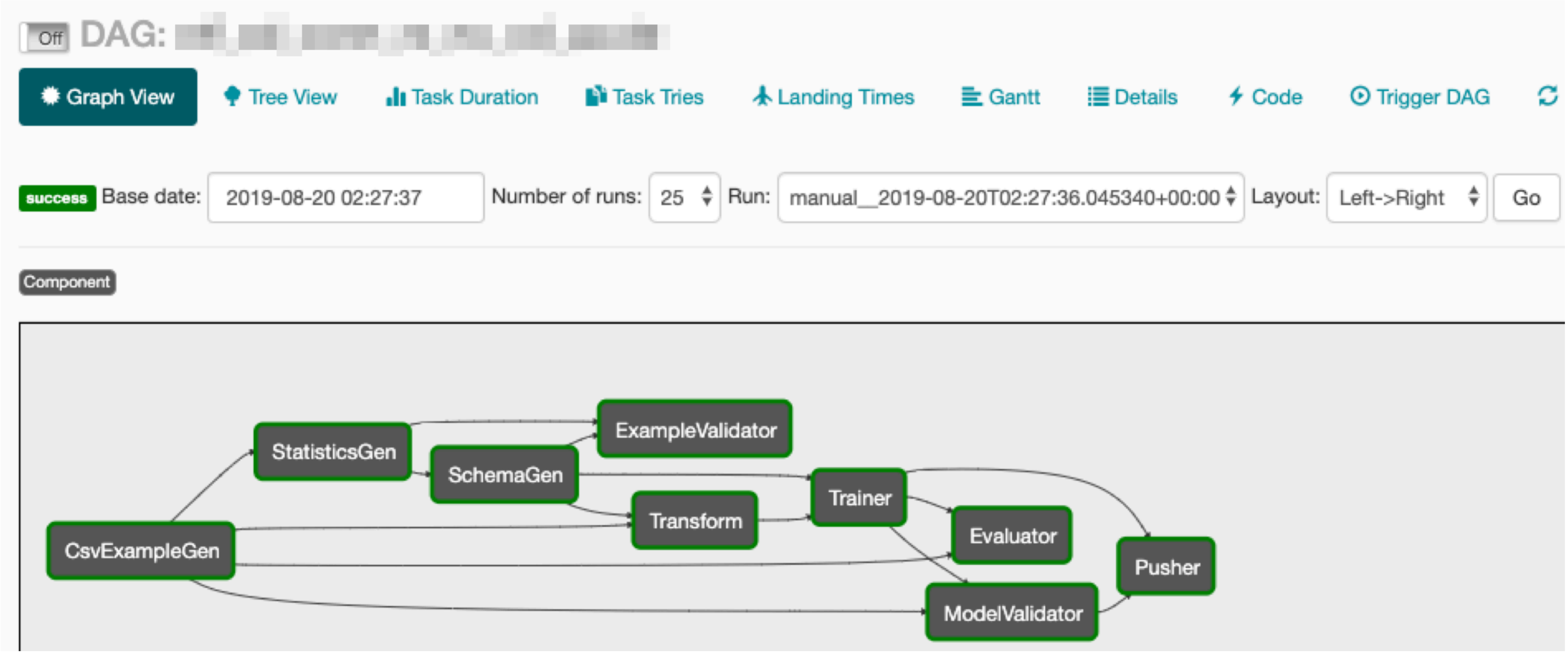
- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
    - ML Metadata
    - Apache Airflow
- TensorFlow and TensorFlow Tools
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

Pipeline Management with Apache Airflow

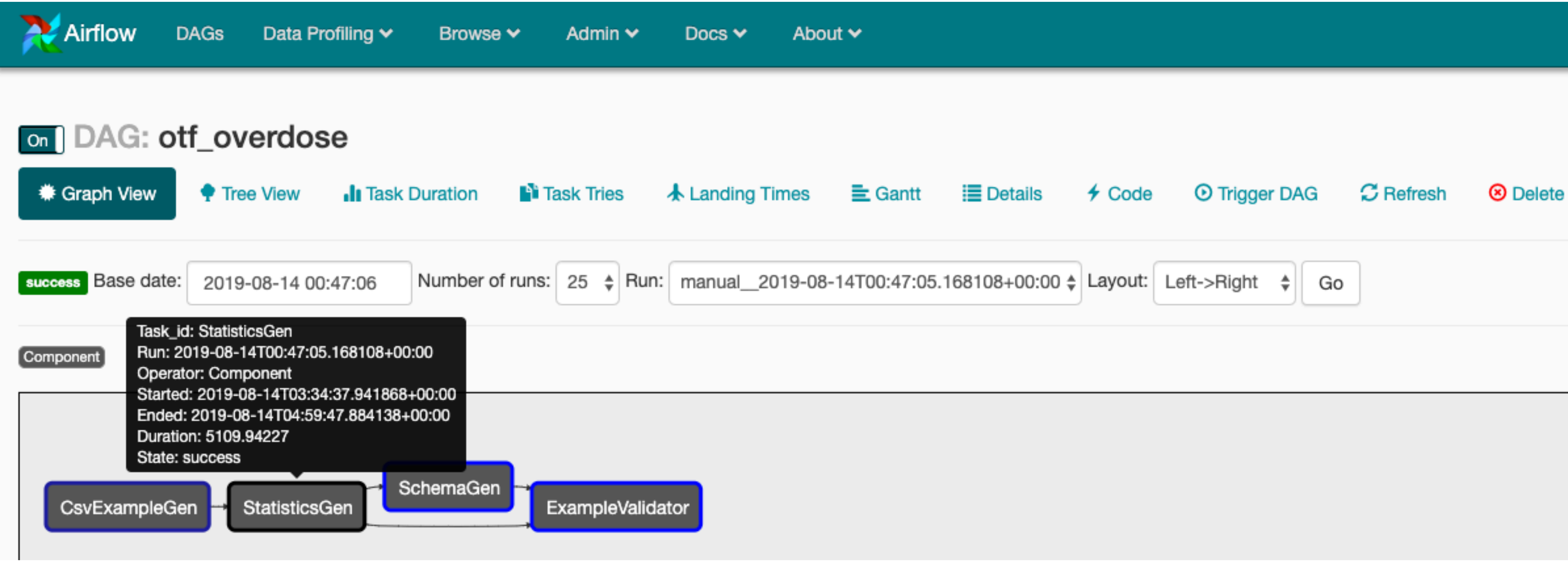
Allows you to trigger and keep track of pipelines.



Pipeline Management with Apache Airflow



Pipeline Management with Apache Airflow



- You can also use Kubeflow https://github.com/tensorflow/tfx/blob/master/tfx/examples/chicago_taxi_pipeline/taxi_pipeline_kubeflow_gcp.py
- And Apache Beam https://github.com/tensorflow/tfx/blob/master/tfx/examples/chicago_taxi_pipeline/taxi_pipeline_beam.py

Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
  - TensorFlow 2.0
  - TensorFlow Data and Features
  - TensorFlow Estimators
  - TensorBoard
  - TensorFlow Data Visualization (TFDV) [Facets]
  - TensorFlow Model Analysis (TFMA)
  - What-If Tool
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

TensorFlow 2.0

- Don't have to define the graph separately
  - More like PyTorch
- There are two ways you can do computation:
  - Eager: like PyTorch, just compute
  - tf.function: You decorate a function and call it

TensorFlow 1.x session

```python
import tensorflow as tf

x = tf.Variable(13)
y = tf.placeholder(tf.int32)

bob = tf.cond(y > 55,
              lambda: tf.add(y, 3),
              lambda: tf.subtract(y,4))

z = x + bob
init = tf.global_variables_initializer()

with tf.Session() as sess:
    sess.run(init)
    z_out = sess.run(z,{y:24})

print(z_out)
```

output 33

TensorFlow 2.x function

```python
import tensorflow as tf

x = tf.Variable(13)

@tf.function
def bob(y) -> tf.Variable:
    print(tf.executing_eagerly())
    if y > 55:
        return y + 3
    else:
        return y - 4

z = x + bob(24)
print(z.numpy())
```

output
```
False
33
```

Still get performance of Session
https://www.tensorflow.org/guide/function

TensorFlow 2.x eager

```python
import tensorflow as tf

x = tf.Variable(13)

def bob(y) -> tf.Variable:
    print(tf.executing_eagerly())
    if y > 55:
        return y + 3
    else:
        return y - 4

z = x + bob(24)
print(z.numpy())
```

output
```
True
33
```

Debug like a civilized person
https://www.tensorflow.org/guide/eager

Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
    - TensorFlow 2.0
    - TensorFlow Data and Features
    - TensorFlow Estimators
    - TensorBoard
    - TensorFlow Data Visualization (TFDV) [Facets]
    - TensorFlow Model Analysis (TFMA)
    - What-If Tool
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

Data

- tf.train.Example is tf.train.Feature protobuf message, where each value has a name and a type (tf.train.BytesList, tf.train.FloatList, tf.train.Int64List)
- TFRecord is a format for storing sequences of binary records, each record is tf.train.Example
- tf.data.Dataset can take in TFRecord and create an iterator for batching
- tf.parse_example unpacks tf.Example into standard tensors.

https://www.tensorflow.org/tutorials/load_data/tf_records

Features

- tf.feature_column, where you further specify what it is, such as one-hot, vocabulary, and embeddings and such.

tf.train.Example specifies what it is for storage, and tf.feature_column is for the input to a model.

https://www.tensorflow.org/guide/feature_columns

Outline

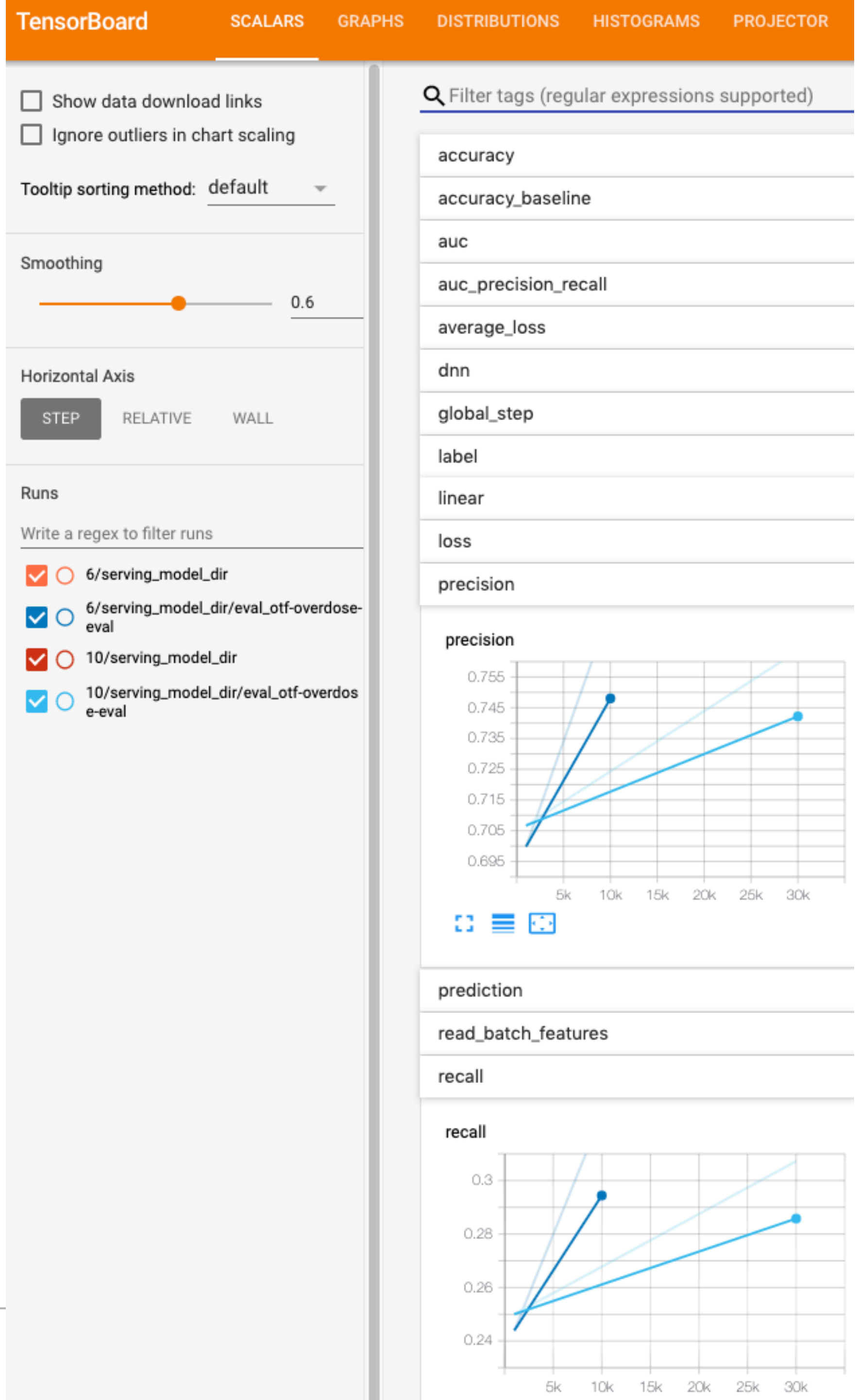
- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
  - TensorFlow 2.0
  - TensorFlow Data and Features
  - TensorFlow Estimators
  - TensorBoard
  - TensorFlow Data Visualization (TFDV) [Facets]
  - TensorFlow Model Analysis (TFMA)
  - What-If Tool
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

To build a model you need

- format of model input
  - `tf.feature_column`
- model architecture and hyperparameters
  - `tf.estimator`
  - `(or KERAS with tf.keras.estimator.model_to_estimator)`
- function to deliver training data
  - `tf.estimator.TrainSpec from tf.data`
- function to deliver eval data
  - `tf.estimator.EvalSpec from tf.data`
- function to deliver serving data
  - `tf.estimator.FinalExporter`

TensorFlow Estimator

- Estimator is a wrapper for regular TensorFlow that automatically scales to multiple machines and automatically outputs results to TensorBoard



Shout out to model explainability using estimator using boosted trees

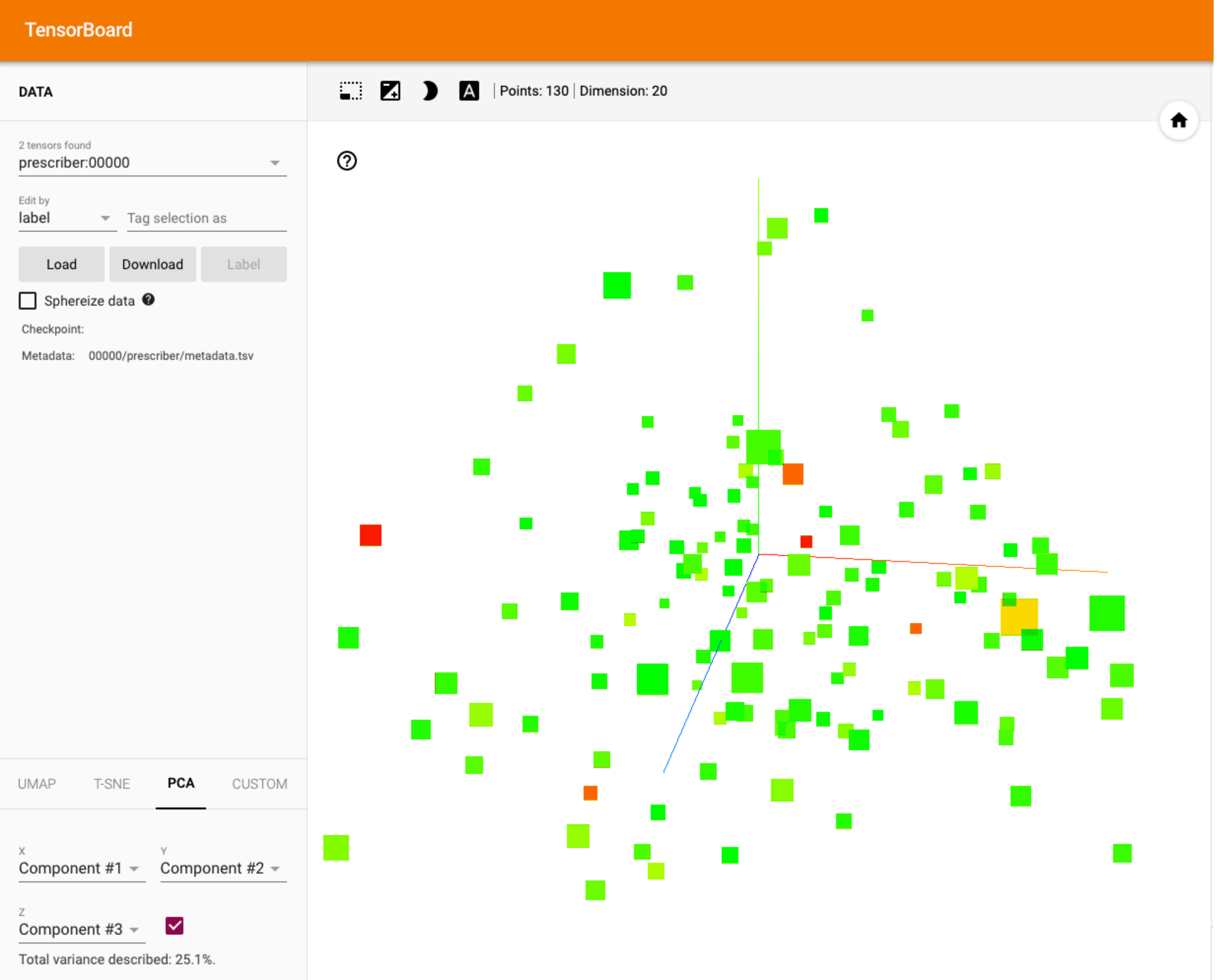
https://www.tensorflow.org/tutorials/estimator/boosted_trees_model_understanding

https://www.tensorflow.org/tutorials/estimator/boosted_trees

DeUmbra

Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
    - TensorFlow 2.0
    - TensorFlow Data and Features
    - TensorFlow Estimators
    - TensorBoard
    - TensorFlow Data Visualization (TFDV) [Facets]
    - TensorFlow Model Analysis (TFMA)
    - What-If Tool
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion



TensorBoard

Plotting of prescribers
Red has more overdose
Green has fewer

3D plots not that useful, but they look cool

You can even use TensorBoard from PyTorch
https://pytorch.org/docs/stable/tensorboard.html

Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
    - TensorFlow 2.0
    - TensorFlow Data and Features
    - TensorFlow Estimators
    - TensorBoard
    - TensorFlow Data Visualization (TFDV) [Facets]
    - TensorFlow Model Analysis (TFMA)
    - What-If Tool
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

TensorFlow Data Validation (TFDV)

- We need to understand our data as well as possible.
- TFDV provides tools that make that less difficult.
- Helps to identify bugs in the data by showing you pictures that don't look right.
- https://www.tensorflow.org/tfx/data_validation/get_started

Sort by

Feature order

☐ Reverse order

Feature search (regex enabled)

Features: ☑ int(70) ☑ float(12) ☑ string(2)

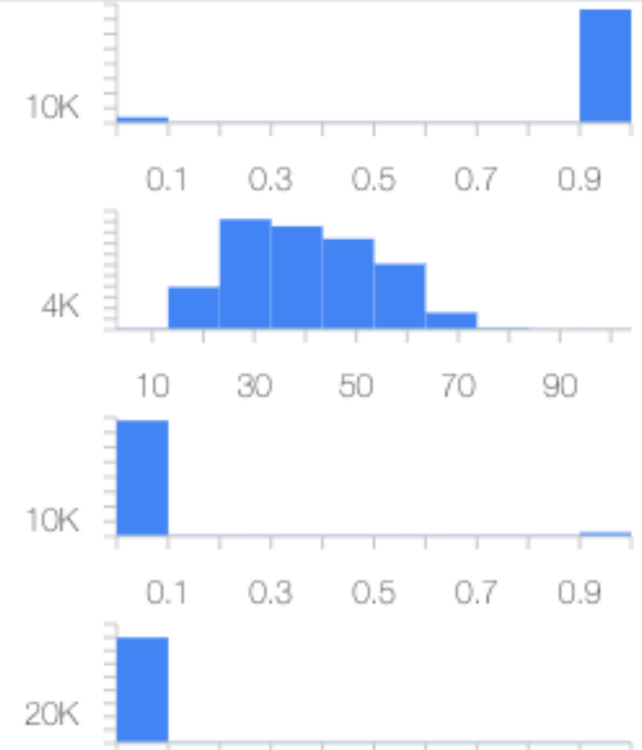
Numeric Features (82)

	count	missing	mean	std dev	zeros	min	median	max
not_weaned	80.5k	0%	0.95	0.22	4.88%	0	1	1
age	80.5k	0%	40.51	13.62	0%	3	39	104
rapid_dose_escalation	80.5k	0%	0.03	0.17	96.87%	0	0	1
osteoporosis	80.5k	0%	0.01	0.09	99.22%	0	0	1

Chart to show

Standard

☐ log ☐ expand



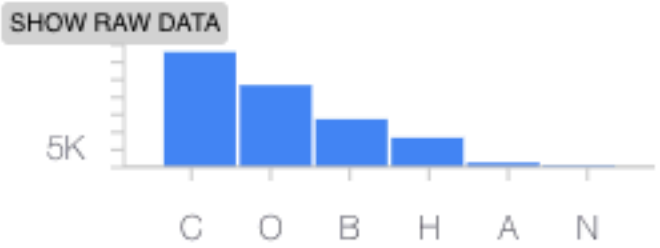
Categorical Features (2)

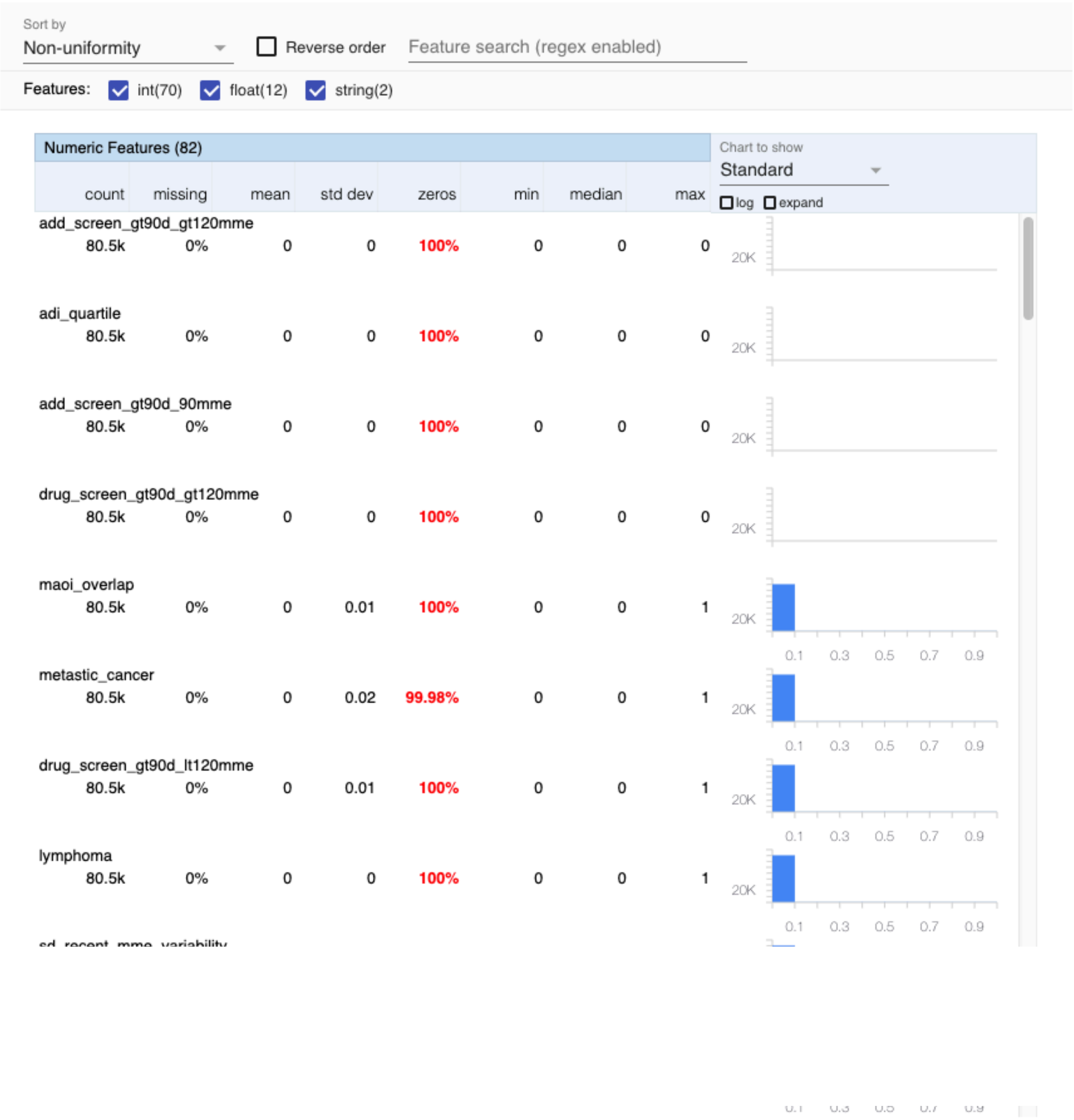
Chart to show

Standard

log expand

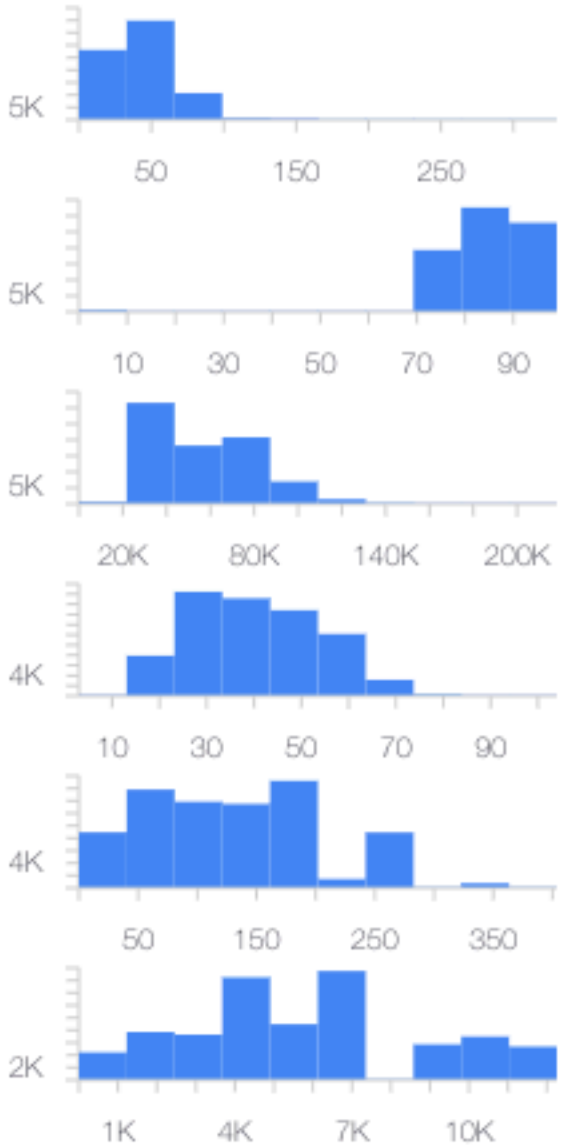
	count	missing	unique	top	freq top	avg str len
race	80.5k	0%	6	C	32.9k	1
gender	80.5k	0%	3	F	48.7k	1





By sorting by non-uniformity, we can debug features.

naloxonerx_100k	80.5k	0%	42.9	27.69	5.57%	0	43.43	330.03
perc_hs_grad	80.5k	0%	84.69	10.84	0.85%	0	86.5	99
median_income	80.5k	0%	59.7k	23.3k	0.85%	0	57.3k	218k
age	80.5k	0%	40.51	13.62	0%	3	39	104
op_death_100k	80.5k	0%	130.67	74.51	1.61%	0	133.46	403.67
crimes_100k	80.5k	0%	5,750.99	3,042.01	0.85%	0	5,652.89	12.3k



In general, we can make sure the distributions are what we would expect.

Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
  - TensorFlow 2.0
  - TensorFlow Data and Features
  - TensorFlow Estimators
  - TensorBoard
  - TensorFlow Data Visualization (TFDV) [Facets]
  - TensorFlow Model Analysis (TFMA)
  - What-If Tool
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

TensorFlow Model Analysis (TFMA)

Examples (Weighted) Threshold

10

Visualization

Slices Overview

Show

accuracy

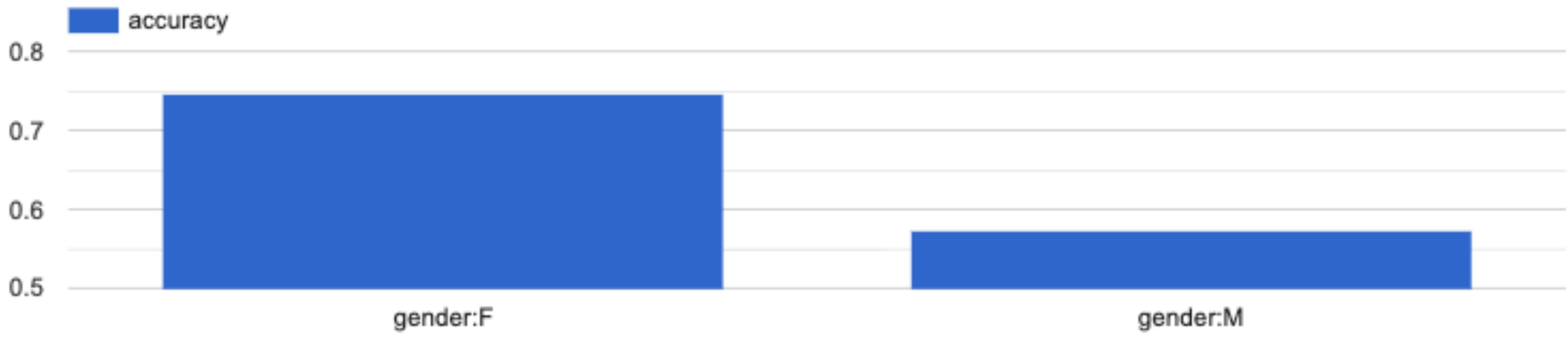
Sort by

Slice

We can see how well our model does by each slice.

We see that this model does much better for females than males.

https://www.tensorflow.org/tfx/guide/tfma



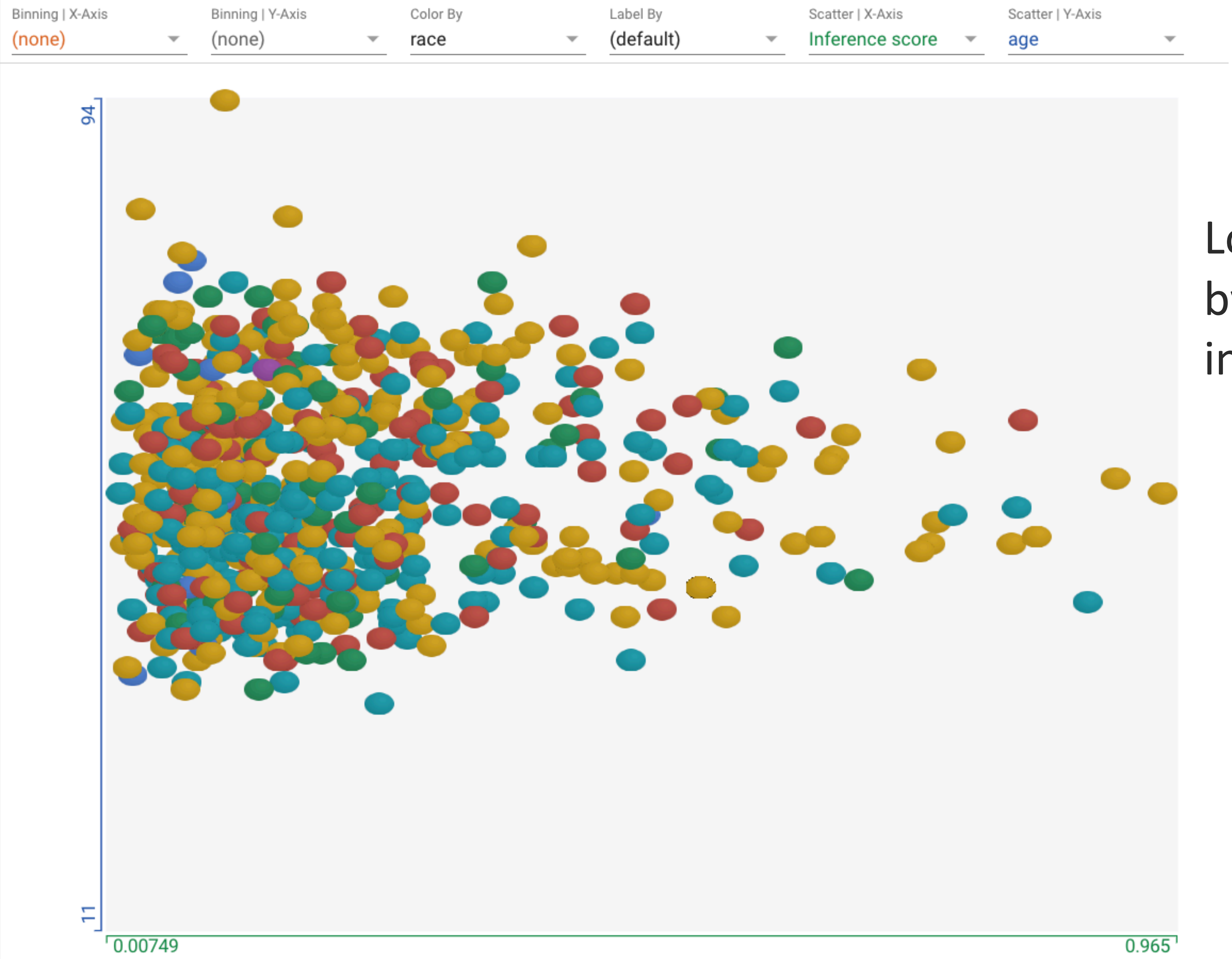
feature	accuracy	accuracy_baseline	auc	auc_precision_recall	average_loss
gender:M	0.57293	0.53354	0.70861	0.71211	0.90811
gender:F	0.74527	0.67612	0.80165	0.64202	0.54457

Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
    - TensorFlow 2.0
    - TensorFlow Data and Features
    - TensorFlow Estimators
    - TensorBoard
    - TensorFlow Data Visualization (TFDV) [Facets]
    - TensorFlow Model Analysis (TFMA)
    - What-If Tool
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

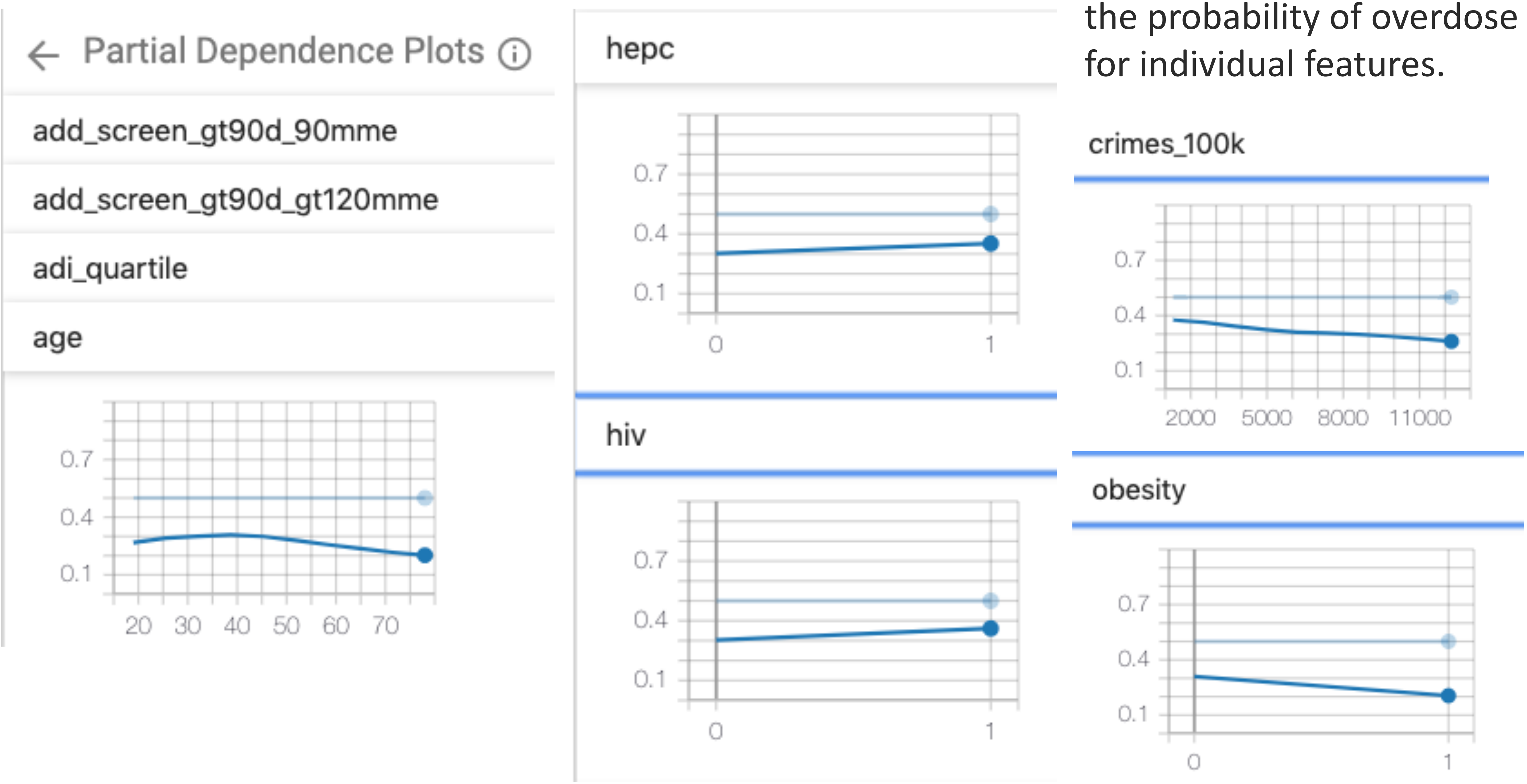
What-IF Tool

- The What-If Tool applies a model from TensorFlow Serving to any data you give it.
  - https://pair-code.github.io/what-if-tool/index.html
- Change a record and see what the model does
- Find the most similar record with a different classification
- Can be used for fairness. Adjust the model so it is equally likely to predict "yes" for each group
  - https://www.coursera.org/lecture/machine-learning-business-professionals/activity-applying-fairness-concerns-with-the-what-if-tool-review-0mYda



Looking at the data by race, age, and inference score

What-If Tool showing the the probability of overdose for individual features.



Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

Alternatives (kind of)

They all do something a little different,
with pieces straddling different sides of the data science/production divide

- MLflow https://mlflow.org/docs/latest/index.html
- Netflix Metaflow https://github.com/Netflix/metaflow
- Sacred https://github.com/IDSIA/sacred
- Dataiku DSS https://www.dataiku.com/product/
- Polyaxon https://polyaxon.com/
- Facebook Ax https://www.ax.dev/

Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
- Alternatives to TensorFlow Extended
- Other Useful Tools
  - Streamlit Dashboard
  - Python Typing, Dataclasses, and Enum
- Conclusion

Streamlit Dashboard

```python
# streamlit run jwm_streamlit.py
import streamlit as st
import numpy
import pandas

st.title('Not my circus, not my monkeys')
st.write('frightened cats prefer a little alcohol in their milk')
st.markdown('## Free bratwurst!')

#*****************************************************************
st.write("Table of Thermopylae")
df = pandas.DataFrame({
    'chickens': [2, 1, 33, 4],
    'misunderstandings': [6, -2, 5, 74],
    'resolutions': [9, 9, -1, 3]
})
st.write(df)

#*****************************************************************
st.write("El Hombre de la Triste Figura ")
chart_data = pandas.DataFrame(
    numpy.random.randn(15, 3),
    columns=['cats', 'dogs', 'sasquatch'])
st.line_chart(chart_data)
```

Not my circus, not my monkeys

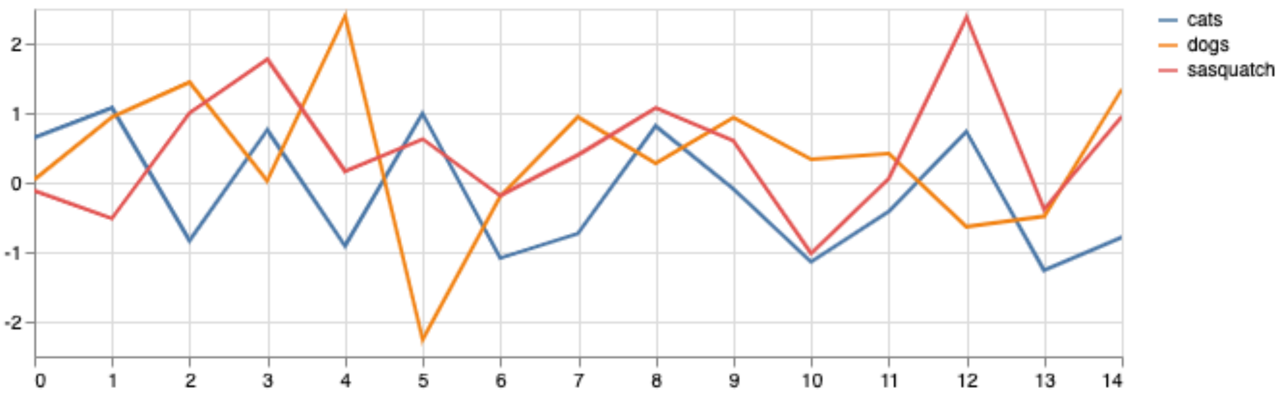
frightened cats prefer a little alcohol in their milk

Free bratwurst!

Table of Thermopylae

	chickens	misunderstandings	resolutions
0	2	6	9
1	1	-2	9
2	33	5	-1
3	4	74	3

El Hombre de la Triste Figura



- Writes to the browser
- Works well for artifacts in the ML pipeline.

https://github.com/streamlit/streamlit/
https://streamlit.io/docs/getting_started.html

Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
- Alternatives to TensorFlow Extended
- Other Useful Tools
  - Streamlit Dashboard
  - Python Typing, Dataclasses, and Enum
- Conclusion

```python
from typing import Dict, List
import enum
from dataclasses import dataclass

class EvalCondition(enum.Enum):
    BASIC='basic'
    SAMPLE_WEIGHT= 'sample_weight'
    AUGMENT_PREDICTION='augment_prediction'
    AUGMENT_GAN='augment_gan'
    AUGMENT_PREDICTION_AND_GAN= 'augment_pred_gan'

@dataclass
class Turtle:
    size: float
    name: str

@dataclass
class EvalArtifact:
    name: str
    eval_condition: EvalCondition
    num_chickens: Dict[str,int]

def my_function(ea: EvalArtifact) -> List[Turtle]:
    t1 = Turtle(6,ea.name)
    t2 = Turtle(2,ea.name)
    return [t1,t2]

ea = EvalArtifact('Anita',EvalCondition.BASIC,{'man':45})
print(my_function(ea))
```

Typing, Dataclasses, and Enum

You can build interchangeable parts right in Python.

Not new of course, but they make Python a little less wild west.

Output:

```
[Turtle(size=6, name='Anita'),
Turtle(size=2, name='Anita')]
```

Outline

- Introduction to TensorFlow Extended (TFX)
- TensorFlow Extended Pipeline Components
- Running the Pipeline
- TensorFlow and TensorFlow Tools
- Alternatives to TensorFlow Extended
- Other Useful Tools
- Conclusion

TFX Disadvantages

- Steep learning curve
- Changes constantly (but not while you are watching it)
- Somewhat inflexible, you can create your own components, but steep learning curve
- No hyperparameter search (yet, https://github.com/tensorflow/tfx/issues/182)

TFX Advantages

- Set up to scale
- Documents your process through artifacts
- Warm-starting: as new data comes in, you don’t have to start training over. Keeps models fresh
- Tools to see data and debug problems
- Don’t have to rerun what is already run

Where to Start

- Jupyter notebook tutorial
  https://www.tensorflow.org/tfx/tutorials/tfx/components
- Airflow tutorial
  https://www.tensorflow.org/tfx/tutorials/tfx/airflow_workshop

Happy Hour!

DeUmbra

6500 River Place Blvd.
Bldg. 3, Suite 120
Austin, TX. 78730

Jonathan Mugan, Ph. D.
Email: jmugan@deumbra.com

Appendix

- Original TFX paper
  https://ai.google/research/pubs/pub46484
- Documentation
  - https://www.tensorflow.org/tfx
  - https://www.tensorflow.org/tfx/tutorials
  - https://www.tensorflow.org/tfx/guide
  - https://www.tensorflow.org/tfx/api_docs