Ungraded Lab: Model Analysis with TFX Evaluator

Now that you've used TFMA as a standalone library in the previous lab, you will now see how it is used by TFX with its Evaluator component. This component comes after your `Trainer` run and it checks if your trained model meets the minimum required metrics and also compares it with previously generated models.

You will go through a TFX pipeline that prepares and trains the same model architecture you used in the previous lab. As a reminder, this is a binary classifier to be trained on the Census Income dataset. Since you're already familiar with the earlier TFX components, we will just go over them quickly but we've placed notes on where you can modify code if you want to practice or produce a better result.

Let's begin!

Credits: Some of the code and discussions are based on the TensorFlow team's official tutorial.

Setup

Install TFX

```
In [ ]:    !pip install tfx==1.2
```

Note: In Google Colab, you need to restart the runtime at this point to finalize updating the packages you just installed. You can do so by clicking the `Restart Runtime` at the end of the output cell above (after installation), or by selecting `Runtime > Restart Runtime` in the Menu bar. **Please do not proceed to the next section without restarting.** *You can also ignore the errors about version incompatibility of some of the bundled packages because we won't be using those in this notebook.*

Imports

```
In [1]:   import os
          import pprint

          from tfx.components import CsvExampleGen
          from tfx.components import ExampleValidator
          from tfx.components import SchemaGen
          from tfx.components import StatisticsGen
          from tfx.components import Transform
          from tfx.components import Tuner
          from tfx.components import Trainer
          from tfx.components import Evaluator

          import tensorflow as tf
          import tensorflow_model_analysis as tfma

          from tfx.orchestration.experimental.interactive.interactive_context import

          tf.get_logger().propagate = False
          tf.get_logger().setLevel('ERROR')
          pp = pprint.PrettyPrinter()
```

Set up pipeline paths

```
In [2]:   # Location of the pipeline metadata store
          _pipeline_root = './pipeline/'

          # Directory of the raw data files
          _data_root = './data/census'

          _data_filepath = os.path.join(_data_root, "data.csv")
```

```
In [3]:   # Create the TFX pipeline files directory
          !mkdir {_pipeline_root}

          # Create the dataset directory
          !mkdir -p {_data_root}
```

```
mkdir: cannot create directory './pipeline/': File exists
```

Download and prepare the dataset

Here, you will download the training split of the Census Income Dataset. This is twice as large as the test dataset you used in the previous lab.

```
In [4]:   # Define filename and URL
          TAR_NAME = 'lab_2_data.tar.gz'
          DATA_PATH = f'https://github.com/https-deeplearning-ai/MLEP-public/raw/main

          # Download dataset
          !wget -nc {DATA_PATH}

          # Extract archive
          !tar xvzf {TAR_NAME}

          # Delete archive
          !rm {TAR_NAME}
```

```
--2021-09-03 08:51:13--  https://github.com/https-deeplearning-ai/MLEP-publ
ic/raw/main/course3/week4-ungraded-lab/lab_2_data.tar.gz
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
```

```
Location: https://github.com/https-deeplearning-ai/machine-learning-enginee
ring-for-production-public/raw/main/course3/week4-ungraded-lab/lab_2_data.t
ar.gz [following]
--2021-09-03 08:51:13--  https://github.com/https-deeplearning-ai/machine-l
earning-engineering-for-production-public/raw/main/course3/week4-ungraded-l
ab/lab_2_data.tar.gz
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/https-deeplearning-ai/machine-l
earning-engineering-for-production-public/main/course3/week4-ungraded-lab/l
ab_2_data.tar.gz [following]
--2021-09-03 08:51:13--  https://raw.githubusercontent.com/https-deeplearni
ng-ai/machine-learning-engineering-for-production-public/main/course3/week4
-ungraded-lab/lab_2_data.tar.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.
111.133, 185.199.108.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.19
9.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 418898 (409K) [application/octet-stream]
Saving to: 'lab_2_data.tar.gz'

lab_2_data.tar.gz   100%[===================>] 409.08K  --.-KB/s    in 0.01
s

2021-09-03 08:51:13 (31.2 MB/s) - 'lab_2_data.tar.gz' saved [418898/418898]

  /data/census/data.csv
```

Take a quick look at the first few rows.

In [5]:
```python
# Preview dataset
!head {_data_filepath}
```

```
age,workclass,fnlwgt,education,education-num,marital-status,occupation,rela
tionship,race,sex,capital-gain,capital-loss,hours-per-week,native-country,l
abel
39,State-gov,77516,Bachelors,13,Never-married,Adm-clerical,Not-in-family,Wh
ite,Male,2174,0,40,United-States,0
50,Self-emp-not-inc,83311,Bachelors,13,Married-civ-spouse,Exec-managerial,H
usband,White,Male,0,0,13,United-States,0
38,Private,215646,HS-grad,9,Divorced,Handlers-cleaners,Not-in-family,White,
Male,0,0,40,United-States,0
53,Private,234721,11th,7,Married-civ-spouse,Handlers-cleaners,Husband,Blac
k,Male,0,0,40,United-States,0
28,Private,338409,Bachelors,13,Married-civ-spouse,Prof-specialty,Wife,Blac
k,Female,0,0,40,Cuba,0
37,Private,284582,Masters,14,Married-civ-spouse,Exec-managerial,Wife,White,
Female,0,0,40,United-States,0
49,Private,160187,9th,5,Married-spouse-absent,Other-service,Not-in-family,B
lack,Female,0,0,16,Jamaica,0
52,Self-emp-not-inc,209642,HS-grad,9,Married-civ-spouse,Exec-managerial,Hus
band,White,Male,0,0,45,United-States,1
31,Private,45781,Masters,14,Never-married,Prof-specialty,Not-in-family,Whit
e,Female,14084,0,50,United-States,1
```

TFX Pipeline

Create the InteractiveContext

As usual, you will initialize the pipeline and use a local SQLite file for the metadata store.

In [6]:
```python
# Initialize InteractiveContext
context = InteractiveContext(pipeline_root=_pipeline_root)
```

```
WARNING:absl:InteractiveContext metadata_connection_config not provided: us
ing SQLite ML Metadata database at ./pipeline/metadata.sqlite.
```

ExampleGen

You will start by ingesting the data through `CsvExampleGen`. The code below uses the default 2:1 train-eval split (i.e. 33% of the data goes to eval) but feel free to modify if you want. You can review splitting techniques here.

In [7]:
```python
# Run CsvExampleGen
example_gen = CsvExampleGen(input_base=_data_root)
context.run(example_gen)
```

```
WARNING:apache_beam.runners.interactive.interactive_environment:Dependencie
s required for Interactive Beam PCollection visualization are not availabl
e, please use: `pip install apache-beam[interactive]` to install necessary
dependencies to enable all data visualization features.
```

```
WARNING:root:Make sure that locally built Python SDK docker image has Pytho
n 3.7 interpreter.
WARNING:apache_beam.io.tfrecordio:Couldn't find python-snappy so the implem
entation of _TFRecordUtil._masked_crc32c is not as fast as it could be.
```

Out[7]:
▼**ExecutionResult** at 0x7f0b4a161310

.execution_id	13
.component	▶**CsvExampleGen** at 0x7f0ac9d8da50
.component.inputs	{}
.component.outputs	['examples'] ▶**Channel** of type **'Examples'** (1 artifact) at 0x7f0acd962fd0

In [8]:
```python
# Print split names and URI
artifact = example_gen.outputs['examples'].get()[0]
print(artifact.split_names, artifact.uri)
```

```
["train", "eval"] ./pipeline/CsvExampleGen/examples/13
```

StatisticsGen

You will then compute the statistics so it can be used by the next components.

In [9]:
```python
# Run StatisticsGen
statistics_gen = StatisticsGen(
    examples=example_gen.outputs['examples'])
context.run(statistics_gen)
```

```
WARNING:root:Make sure that locally built Python SDK docker image has Pytho
n 3.7 interpreter.
```

Out[9]:
▼**ExecutionResult** at 0x7f0ac7cc6110

.execution_id	14
.component	▶**StatisticsGen** at 0x7f0ac7cc6dd0
.component.inputs	['examples'] ▶**Channel** of type **'Examples'** (1 artifact) at 0x7f0acd962fd0

.component.outputs

['statistics'] ▶ **Channel** of type **'ExampleStatistics'** (1 artifact) at 0x7f0ac7cc6f10

You can look at the visualizations below if you want to explore the data some more.

In [43]:
```python
# Visualize statistics
context.show(statistics_gen.outputs['statistics'])
```

Artifact at ./pipeline/StatisticsGen/statistics/14

'train' split:

Sort by

Feature order ▾   ☐ Reverse order   Feature search (regex enabled)

Features: ☑ int(7)   ☑ string(8)

Numeric Features (7)						
count	missing	mean	std dev	zeros	min	median
age						
21.8k	0%	38.61	13.62	0%	17	37
capital-gain						
21.8k	0%	1,114.65	7,616.76	**91.63%**	0	0
capital-loss						
21.8k	0%	89.15	406.88	**95.22%**	0	0
education-num						
21.8k	0%	10.08	2.58	0%	1	10

'eval' split:

Numeric Features (7)						
count	missing	mean	std dev	zeros	min	median
age						
10.8k	0%	38.53	13.68	0%	17	37

SchemaGen

You can then infer the dataset schema with SchemaGen. This will be used to validate incoming data to ensure that it is formatted correctly.

In [11]:
```python
# Run SchemaGen
schema_gen = SchemaGen(
    statistics=statistics_gen.outputs['statistics'])
context.run(schema_gen)
```

Out[11]:

▼**ExecutionResult** at 0x7f0ac85bc810

.execution_id 15

.component ▶SchemaGen at 0x7f0ac85c4d90

.component.inputs

['statistics'] ▶**Channel** of type **'ExampleStatistics'** (1 artifact) at 0x7f0ac7cc6f10

.component.outputs

['schema'] ▶**Channel** of type **'Schema'** (1 artifact) at 0x7f0ac85a2f50

For simplicity, you will just accept the inferred schema but feel free to modify with the TFDV API if you want.

In [12]:
```python
# Visualize the inferred Schema
context.show(schema_gen.outputs['schema'])
```

Artifact at ./pipeline/SchemaGen/schema/15

Feature name	Type	Presence	Valency	Domain
'education'	STRING	required		'education'
'marital-status'	STRING	required		'marital-status'
'native-country'	STRING	required		'native-country'

	Type	Presence	Valency	Domain
Feature name				
'occupation'	STRING	required		'occupation'
'race'	STRING	required		'race'
'relationship'	STRING	required		'relationship'
'sex'	STRING	required		'sex'
'workclass'	STRING	required		'workclass'
'age'	INT	required		-
'capital-gain'	INT	required		-
'capital-loss'	INT	required		-
'education-num'	INT	required		-
'fnlwgt'	INT	required		-

Domain	Values
'education'	'10th', '11th', '12th', '1st-4th', '5th-6th', '7th-8th', '9th', 'Assoc-acdm', 'Assoc-voc', 'Bachelors', 'Doctorate', 'HS-grad', 'Masters', 'Preschool', 'Prof-school', 'Some-college'
'marital-status'	'Divorced', 'Married-AF-spouse', 'Married-civ-spouse', 'Married-spouse-absent', 'Never-married', 'Separated', 'Widowed'
'native-country'	'?', 'Cambodia', 'Canada', 'China', 'Columbia', 'Cuba', 'Dominican-Republic', 'Ecuador', 'El-Salvador', 'England', 'France', 'Germany', 'Greece', 'Guatemala', 'Haiti', 'Holand-Netherlands', 'Honduras', 'Hong', 'Hungary', 'India', 'Iran', 'Ireland', 'Italy', 'Jamaica', 'Japan', 'Laos', 'Mexico', 'Nicaragua', 'Outlying-US(Guam-USVI-etc)', 'Peru', 'Philippines', 'Poland', 'Portugal', 'Puerto-Rico', 'Scotland', 'South', 'Taiwan', 'Thailand', 'Trinadad&Tobago', 'United-States', 'Vietnam', 'Yugoslavia'
'occupation'	'?', 'Adm-clerical', 'Armed-Forces', 'Craft-repair', 'Exec-managerial', 'Farming-fishing', 'Handlers-cleaners', 'Machine-op-inspct', 'Other-service', 'Priv-house-serv', 'Prof-specialty', 'Protective-serv', 'Sales', 'Tech-support', 'Transport-moving'
'race'	'Amer-Indian-Eskimo', 'Asian-Pac-Islander', 'Black', 'Other', 'White'
'relationship'	'Husband', 'Not-in-family', 'Other-relative', 'Own-child', 'Unmarried', 'Wife'
'sex'	'Female', 'Male'
'workclass'	'?', 'Federal-gov', 'Local-gov', 'Never-worked', 'Private', 'Self-emp-inc', 'Self-emp-not-inc', 'State-gov', 'Without-pay'

ExampleValidator

Next, run `ExampleValidator` to check if there are anomalies in the data.

In [13]:
```python
# Run ExampleValidator
example_validator = ExampleValidator(
    statistics=statistics_gen.outputs['statistics'],
    schema=schema_gen.outputs['schema'])
context.run(example_validator)
```

```
Out[13]:  ▼ExecutionResult at 0x7f0ac85abf50

              .execution_id   16

                .component   ▶ExampleValidator at 0x7f0ac85bc610

          .component.inputs
                              ['statistics']   ▶Channel of type 'ExampleStatistics'
                                               (1 artifact) at 0x7f0ac7cc6f10

                                 ['schema']   ▶Channel of type 'Schema' (1 artifact)
                                               at 0x7f0ac85a2f50

         .component.outputs
                              ['anomalies']   ▶Channel of type
                                               'ExampleAnomalies' (1 artifact) at
                                               0x7f0ac85bc990
```

If you just used the inferred schema, then there should not be any anomalies detected. If you modified the schema, then there might be some results here and you can again use TFDV to modify or relax constraints as needed.

In actual deployments, this component will also help you understand how your data evolves over time and identify data errors. For example, the first batches of data that you get from your users might conform to the schema but it might not be the case after several months. This component will detect that and let you know that your model might need to be updated.

```
In [14]:   # Check results
           context.show(example_validator.outputs['anomalies'])
```

Artifact at ./pipeline/ExampleValidator/anomalies/16

'train' split:

```
/usr/local/lib/python3.7/dist-packages/tensorflow_data_validation/utils/dis
play_util.py:217: FutureWarning: Passing a negative integer is deprecated i
n version 1.0 and will not be supported in future version. Instead, use Non
e to not limit the column width.
  pd.set_option('max_colwidth', -1)
```

No anomalies found.

'eval' split:

No anomalies found.

Transform

Now you will perform feature engineering on the training data. As shown when you previewed the CSV earlier, the data is still in raw format and cannot be consumed by the model just yet. The transform code in the following cells will take care of scaling your numeric features and one-hot encoding your categorical variables.

Note: If you're running this exercise for the first time, we advise that you leave the transformation code as is. After you've gone through the entire notebook, then you can modify these for practice and see what results you get. Just make sure that your model builder code in the `Trainer` component will also reflect those changes if needed. For example, removing a feature here should also remove an input layer for that feature in the model.

In [15]:
```python
# Set the constants module filename
_census_constants_module_file = 'census_constants.py'
```

In [16]:
```python
%%writefile {_census_constants_module_file}

# Features with string data types that will be converted to indices
VOCAB_FEATURE_DICT = {
    'education': 16, 'marital-status': 7, 'occupation': 15, 'race': 5,
    'relationship': 6, 'workclass': 9, 'sex': 2, 'native-country': 42
}

# Numerical features that are marked as continuous
NUMERIC_FEATURE_KEYS = ['fnlwgt', 'education-num', 'capital-gain', 'capital

# Feature that can be grouped into buckets
BUCKET_FEATURE_DICT = {'age': 4}

# Number of out-of-vocabulary buckets
NUM_OOV_BUCKETS = 5

# Feature that the model will predict
LABEL_KEY = 'label'

# Utility function for renaming the feature
def transformed_name(key):
    return key + '_xf'
```

Overwriting census_constants.py

In [17]:
```python
# Set the transform module filename
_census_transform_module_file = 'census_transform.py'
```

```
In [18]:   %%writefile {_census_transform_module_file}

           import tensorflow as tf
           import tensorflow_transform as tft

           # import constants from cells above
           import census_constants

           # Unpack the contents of the constants module
           _NUMERIC_FEATURE_KEYS = census_constants.NUMERIC_FEATURE_KEYS
           _VOCAB_FEATURE_DICT = census_constants.VOCAB_FEATURE_DICT
           _BUCKET_FEATURE_DICT = census_constants.BUCKET_FEATURE_DICT
           _NUM_OOV_BUCKETS = census_constants.NUM_OOV_BUCKETS
           _LABEL_KEY = census_constants.LABEL_KEY
           _transformed_name = census_constants.transformed_name

           # Define the transformations
           def preprocessing_fn(inputs):
               """tf.transform's callback function for preprocessing inputs.
               Args:
                   inputs: map from feature keys to raw not-yet-transformed features.
               Returns:
                   Map from string feature key to transformed feature operations.
               """

               # Initialize outputs dictionary
               outputs = {}

               # Scale these features to the range [0,1]
               for key in _NUMERIC_FEATURE_KEYS:
                   scaled = tft.scale_to_0_1(inputs[key])
                   outputs[_transformed_name(key)] = tf.reshape(scaled, [-1])

               # Convert strings to indices and convert to one-hot vectors
               for key, vocab_size in _VOCAB_FEATURE_DICT.items():
                   indices = tft.compute_and_apply_vocabulary(inputs[key], num_oov_bud
                   one_hot = tf.one_hot(indices, vocab_size + _NUM_OOV_BUCKETS)
                   outputs[_transformed_name(key)] = tf.reshape(one_hot, [-1, vocab_si

               # Bucketize this feature and convert to one-hot vectors
               for key, num_buckets in _BUCKET_FEATURE_DICT.items():
                   indices = tft.bucketize(inputs[key], num_buckets)
                   one_hot = tf.one_hot(indices, num_buckets)
                   outputs[_transformed_name(key)] = tf.reshape(one_hot, [-1, num_buck

               # Cast label to float
               outputs[_transformed_name(_LABEL_KEY)] = tf.cast(inputs[_LABEL_KEY], tf

               return outputs
```

```
Overwriting census_transform.py
```

Now, we pass in this feature engineering code to the `Transform` component and run it to transform your data.

```
In [19]:   # Run the Transform component
           transform = Transform(
               examples=example_gen.outputs['examples'],
               schema=schema_gen.outputs['schema'],
               module_file=os.path.abspath(_census_transform_module_file))
           context.run(transform, enable_cache=False)
```

```
WARNING:root:This output type hint will be ignored and not used for type-ch
ecking purposes. Typically, output type hints for a PTransform are single
(or nested) types wrapped by a PCollection, PDone, or None. Got: Tuple[Dict
```

```
[str, Union[NoneType, _Dataset]], Union[Dict[str, Dict[str, PCollection]],
NoneType]] instead.
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary/apply_vocab/text
_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_1/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_2/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_3/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_4/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_5/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_6/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_7/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary/apply_vocab/text
_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_1/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_2/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_3/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_4/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
text using  `tf.init_scope`.: compute_and_apply_vocabulary_5/apply_vocab/te
xt_file_init/InitializeTableFromTextFileV2
WARNING:absl:Tables initialized inside a tf.function  will be re-initialize
d on every invocation of the function. This  re-initialization can have sig
nificant impact on performance. Consider lifting  them out of the graph con
```

Out[19]:

▼**ExecutionResult** at 0x7f0ac8690d10

.execution_id   17

.component   ▶**Transform** at 0x7f0ac8615150

.component.inputs

['examples']   ▶**Channel** of type **'Examples'** (1 artifact) at 0x7f0acd962fd0

['schema']   ▶**Channel** of type **'Schema'** (1 artifact) at 0x7f0ac85a2f50

.component.outputs

['transform_graph']   ▶**Channel** of type **'TransformGraph'** (1 artifact) at 0x7f0ac86151d0

['transformed_examples']   ▶**Channel** of type **'Examples'** (1 artifact) at 0x7f0ac8615510

['updated_analyzer_cache']   ▶**Channel** of type **'TransformCache'** (1 artifact) at 0x7f0ac8615250

['pre_transform_schema']   ▶**Channel** of type **'Schema'** (1 artifact) at 0x7f0ac8615310

['pre_transform_stats']   ▶**Channel** of type **'ExampleStatistics'** (1 artifact) at 0x7f0ac8615dd0

['post_transform_schema']   ▶**Channel** of type **'Schema'** (1 artifact) at 0x7f0ac8615790

['post_transform_stats']   ▶**Channel** of type **'ExampleStatistics'** (1 artifact) at 0x7f0ac8615f10

	['post_transform_anomalies']	▶ **Channel** of type **'ExampleAnomalies'** (1 artifact) at 0x7f0ac8615390

You can see a sample result for one row with the code below. Notice that the numeric features are indeed scaled and the categorical features are now one-hot encoded.

In [20]:
```python
# Get the URI of the output artifact representing the transformed examples
train_uri = os.path.join(transform.outputs['transformed_examples'].get()[0]

# Get the list of files in this directory (all compressed TFRecord files)
tfrecord_filenames = [os.path.join(train_uri, name)
                      for name in os.listdir(train_uri)]

# Create a `TFRecordDataset` to read these files
dataset = tf.data.TFRecordDataset(tfrecord_filenames, compression_type="GZI

# Decode the first record and print output
for tfrecord in dataset.take(1):
  serialized_example = tfrecord.numpy()
  example = tf.train.Example()
  example.ParseFromString(serialized_example)
  pp.pprint(example)
```

```
features {
  feature {
    key: "age_xf"
    value {
      float_list {
        value: 0.0
        value: 0.0
        value: 1.0
        value: 0.0
      }
    }
  }
  feature {
    key: "capital-gain_xf"
    value {
      float_list {
        value: 0.02174021676182747
      }
    }
  }
  feature {
    key: "capital-loss_xf"
    value {
      float_list {
        value: 0.0
      }
    }
  }
  feature {
    key: "education-num_xf"
    value {
      float_list {
        value: 0.800000011920929
      }
    }
  }
  feature {
    key: "education_xf"
    value {
      float_list {
        value: 0.0
        value: 0.0
        value: 1.0
        value: 0.0
```

```
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                  }
                }
              }
              feature {
                key: "fnlwgt_xf"
                value {
                  float_list {
                    value: 0.044301897287368774
                  }
                }
              }
              feature {
                key: "hours-per-week_xf"
                value {
                  float_list {
                    value: 0.3979591727256775
                  }
                }
              }
              feature {
                key: "label_xf"
                value {
                  float_list {
                    value: 0.0
                  }
                }
              }
              feature {
                key: "marital-status_xf"
                value {
                  float_list {
                    value: 0.0
                    value: 1.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                  }
                }
              }
              feature {
                key: "native-country_xf"
                value {
                  float_list {
                    value: 1.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
                    value: 0.0
```

```
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
          }
        }
      }
      feature {
        key: "occupation_xf"
        value {
          float_list {
            value: 0.0
            value: 0.0
            value: 0.0
            value: 1.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
            value: 0.0
          }
        }
      }
      feature {
        key: "race_xf"
        value {
          float_list {
            value: 1.0
            value: 0.0
```

```
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
              }
            }
          }
          feature {
            key: "relationship_xf"
            value {
              float_list {
                value: 0.0
                value: 1.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
              }
            }
          }
          feature {
            key: "sex_xf"
            value {
              float_list {
                value: 1.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
              }
            }
          }
          feature {
            key: "workclass_xf"
            value {
              float_list {
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 1.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
                value: 0.0
              }
            }
          }
        }
      }
```

As you already know, the `Transform` component not only outputs the transformed examples but also the transformation graph. This should be used on all inputs when your model is deployed to ensure that it is transformed the same way as your training data. Else, it can produce training-serving skew which leads to noisy predictions.

The `Transform` component stores related files in its `transform_graph` output and it would be good to quickly review its contents before we move on to the next component. As

shown below, the URI of this output points to a directory containing three subdirectories.

In [21]:
```python
# Get URI and list subdirectories
graph_uri = transform.outputs['transform_graph'].get()[0].uri
os.listdir(graph_uri)
```

Out[21]: `['metadata', 'transformed_metadata', 'transform_fn']`

- The `transformed_metadata` subdirectory contains the schema of the preprocessed data.
- The `transform_fn` subdirectory contains the actual preprocessing graph.
- The `metadata` subdirectory contains the schema of the original data.

Trainer

Next, you will now build the model to make your predictions. As mentioned earlier, this is a binary classifier where the label is 1 if a person earns more than 50k USD and 0 if less than or equal. The model used here uses the wide and deep model as reference but feel free to modify after you've completed the exercise. Also for simplicity, the hyperparameters (e.g. number of hidden units) have been hardcoded but feel free to use a `Tuner` component as you did in Week 1 if you want to get some practice.

As a reminder, it is best to start from `run_fn()` when you're reviewing the module file below. The `Trainer` component looks for that function first. All other functions defined in the module are just utility functions for `run_fn()`.

Another thing you will notice below is the `_get_serve_tf_examples_fn()` function. This is tied to the `serving_default` signature which makes it possible for you to just pass in raw features when the model is served for inference. You have seen this in action in the previous lab. This is done by decorating the enclosing function, `serve_tf_examples_fn()`, with tf.function. This indicates that the computation will be done by first tracing a Tensorflow graph. You will notice that this function uses `model.tft_layer` which comes from `transform_graph` output. Now when you call the `.get_concrete_function()` method on this tf.function in `run_fn()`, you are creating the graph that will be used in later computations. This graph is used whenever you pass in an `examples` argument pointing to a Tensor with `tf.string` dtype. That matches the format of the serialized batches of data you used in the previous lab.

In [22]:
```python
# Declare trainer module file
_census_trainer_module_file = 'census_trainer.py'
```

```python
In [23]:   %%writefile {_census_trainer_module_file}

           from typing import List, Text

           import tensorflow as tf
           import tensorflow_transform as tft
           from tensorflow_transform.tf_metadata import schema_utils

           from tfx.components.trainer.fn_args_utils import DataAccessor, FnArgs
           from tfx_bsl.public.tfxio import TensorFlowDatasetOptions

           # import same constants from transform module
           import census_constants

           # Unpack the contents of the constants module
           _NUMERIC_FEATURE_KEYS = census_constants.NUMERIC_FEATURE_KEYS
           _VOCAB_FEATURE_DICT = census_constants.VOCAB_FEATURE_DICT
           _BUCKET_FEATURE_DICT = census_constants.BUCKET_FEATURE_DICT
           _NUM_OOV_BUCKETS = census_constants.NUM_OOV_BUCKETS
           _LABEL_KEY = census_constants.LABEL_KEY
           _transformed_name = census_constants.transformed_name


           def _gzip_reader_fn(filenames):
             '''Load compressed dataset

             Args:
               filenames - filenames of TFRecords to load

             Returns:
               TFRecordDataset loaded from the filenames
             '''

             # Load the dataset. Specify the compression type since it is saved as `.g
             return tf.data.TFRecordDataset(filenames, compression_type='GZIP')


           def _input_fn(file_pattern,
                         tf_transform_output,
                         num_epochs=None,
                         batch_size=32) -> tf.data.Dataset:
             '''Create batches of features and labels from TF Records

             Args:
               file_pattern - List of files or patterns of file paths containing Examp
               tf_transform_output - transform output graph
               num_epochs - Integer specifying the number of times to read through the
                         If None, cycles through the dataset forever.
               batch_size - An int representing the number of records to combine in a

             Returns:
               A dataset of dict elements, (or a tuple of dict elements and label).
               Each dict maps feature keys to Tensor or SparseTensor objects.
             '''

             # Get post-transfrom feature spec
             transformed_feature_spec = (
                 tf_transform_output.transformed_feature_spec().copy())

             # Create batches of data
             dataset = tf.data.experimental.make_batched_features_dataset(
                 file_pattern=file_pattern,
                 batch_size=batch_size,
                 features=transformed_feature_spec,
                 reader=_gzip_reader_fn,
                 num_epochs=num_epochs,
                 label_key=_transformed_name(_LABEL_KEY)
```

```python
    return dataset


def _get_serve_tf_examples_fn(model, tf_transform_output):
  """Returns a function that parses a serialized tf.Example and applies TFT

  # Get transformation graph
  model.tft_layer = tf_transform_output.transform_features_layer()

  @tf.function
  def serve_tf_examples_fn(serialized_tf_examples):
    """Returns the output to be used in the serving signature."""
    # Get pre-transform feature spec
    feature_spec = tf_transform_output.raw_feature_spec()

    # Pop label since serving inputs do not include the label
    feature_spec.pop(_LABEL_KEY)

    # Parse raw examples into a dictionary of tensors matching the feature
    parsed_features = tf.io.parse_example(serialized_tf_examples, feature_s

    # Transform the raw examples using the transform graph
    transformed_features = model.tft_layer(parsed_features)

    # Get predictions using the transformed features
    return model(transformed_features)

  return serve_tf_examples_fn


def _build_keras_model(hidden_units: List[int] = None) -> tf.keras.Model:
  """Creates a DNN Keras model for classifying income data.

  Args:
    hidden_units: [int], the layer sizes of the DNN (input layer first).

  Returns:
    A keras Model.
  """

  # Use helper function to create the model
  model = _wide_and_deep_classifier(
      dnn_hidden_units=hidden_units or [100, 70, 50, 25])

  return model


def _wide_and_deep_classifier(dnn_hidden_units):
  """Build a simple keras wide and deep model using the Functional API.

  Args:
    wide_columns: Feature columns wrapped in indicator_column for wide (lin
      part of the model.
    deep_columns: Feature columns for deep part of the model.
    dnn_hidden_units: [int], the layer sizes of the hidden DNN.

  Returns:
    A Wide and Deep Keras model
  """

  # Define input layers for numeric keys
  input_numeric = [
      tf.keras.layers.Input(name=_transformed_name(colname), shape=(1,), dt
      for colname in _NUMERIC_FEATURE_KEYS
  ]
```

```python
    # Define input layers for bucket key
    input_categorical += [
        tf.keras.layers.Input(name=_transformed_name(colname), shape=(num_buc
        for colname, num_buckets in _BUCKET_FEATURE_DICT.items()
    ]

    # Concatenate numeric inputs
    deep = tf.keras.layers.concatenate(input_numeric)

    # Create deep dense network for numeric inputs
    for numnodes in dnn_hidden_units:
        deep = tf.keras.layers.Dense(numnodes)(deep)

    # Concatenate categorical inputs
    wide = tf.keras.layers.concatenate(input_categorical)

    # Create shallow dense network for categorical inputs
    wide = tf.keras.layers.Dense(128, activation='relu')(wide)

    # Combine wide and deep networks
    combined = tf.keras.layers.concatenate([deep, wide])

    # Define output of binary classifier
    output = tf.keras.layers.Dense(
        1, activation='sigmoid')(combined)

    # Setup combined input
    input_layers = input_numeric + input_categorical

    # Create the Keras model
    model = tf.keras.Model(input_layers, output)

    # Define training parameters
    model.compile(
        loss='binary_crossentropy',
        optimizer=tf.keras.optimizers.Adam(lr=0.001),
        metrics='binary_accuracy')

    # Print model summary
    model.summary()

    return model


# TFX Trainer will call this function.
def run_fn(fn_args: FnArgs):
    """Defines and trains the model.

    Args:
        fn_args: Holds args as name/value pairs. Refer here for the complete at
        https://www.tensorflow.org/tfx/api_docs/python/tfx/components/trainer/f
    """

    # Number of nodes in the first layer of the DNN
    first_dnn_layer_size = 100
    num_dnn_layers = 4
    dnn_decay_factor = 0.7

    # Get transform output (i.e. transform graph) wrapper
    tf_transform_output = tft.TFTransformOutput(fn_args.transform_output)

    # Create batches of train and eval sets
```

```python
                            max(2, int(first_dnn_layer_size * dnn_decay_factor ** i))
            for i in range(num_dnn_layers)
        ])

    # Callback for TensorBoard
    tensorboard_callback = tf.keras.callbacks.TensorBoard(
        log_dir=fn_args.model_run_dir, update_freq='batch')


    # Train the model
    model.fit(
        train_dataset,
        steps_per_epoch=fn_args.train_steps,
        validation_data=eval_dataset,
        validation_steps=fn_args.eval_steps,
        callbacks=[tensorboard_callback])


    # Define default serving signature
    signatures = {
        'serving_default':
            _get_serve_tf_examples_fn(model,
                                      tf_transform_output).get_concrete_funct
                                          tf.TensorSpec(
                                              shape=[None],
                                              dtype=tf.string,
                                              name='examples')),
    }


    # Save model with signature
    model.save(fn_args.serving_model_dir, save_format='tf', signatures=signat
```

```
Overwriting census_trainer.py
```

Now, we pass in this model code to the `Trainer` component and run it to train the model.

Note: You can ignore the `Exception ignored in: <function CapturableResource.__del__>` prompt which generates a long traceback. This might be an issue with the underlying TFMA version used and has been flagged to the TFX team so it can be suppressed. This might pop up here and in the Evaluator component.

In [24]:

```python
from tfx.proto import trainer_pb2

trainer = Trainer(
    module_file=os.path.abspath(_census_trainer_module_file),
    examples=transform.outputs['transformed_examples'],
    transform_graph=transform.outputs['transform_graph'],
    schema=schema_gen.outputs['schema'],
    train_args=trainer_pb2.TrainArgs(num_steps=50),
    eval_args=trainer_pb2.EvalArgs(num_steps=50))
context.run(trainer, enable_cache=False)
```

```
WARNING:absl:Examples artifact does not have payload_format custom propert
y. Falling back to FORMAT_TF_EXAMPLE
WARNING:absl:Examples artifact does not have payload_format custom propert
y. Falling back to FORMAT_TF_EXAMPLE
WARNING:absl:Examples artifact does not have payload_format custom propert
y. Falling back to FORMAT_TF_EXAMPLE
```

Model: "model"

_____

Layer (type)	Output Shape	Param #	Connected to
fnlwgt_xf (InputLayer)	[(None, 1)]	0	
education-num_xf (InputLayer)	[(None, 1)]	0	
capital-gain_xf (InputLayer)	[(None, 1)]	0	
capital-loss_xf (InputLayer)	[(None, 1)]	0	
hours-per-week_xf (InputLayer)	[(None, 1)]	0	
concatenate (Concatenate)	(None, 5)	0	fnlwgt_xf[0][0]
			education-num_xf[0][0]
			capital-gain_xf[0][0]
			capital-loss_xf[0][0]
			hours-per-week_xf[0][0]
dense (Dense)	(None, 100)	600	concatenate[0][0]
dense_1 (Dense)	(None, 70)	7070	dense[0][0]
education_xf (InputLayer)	[(None, 21)]	0	
marital-status_xf (InputLayer)	[(None, 12)]	0	
occupation_xf (InputLayer)	[(None, 20)]	0	
race_xf (InputLayer)	[(None, 10)]	0	
relationship_xf (InputLayer)	[(None, 11)]	0	
workclass_xf (InputLayer)	[(None, 14)]	0	
sex_xf (InputLayer)	[(None, 7)]	0	
native-country_xf (InputLayer)	[(None, 47)]	0	
age_xf (InputLayer)	[(None, 4)]	0	
dense_2 (Dense)	(None, 48)	3408	dense_1[0][0]

```
_____
concatenate_1 (Concatenate)     (None, 146)          0           education_
xf[0][0]

                                                                 marital-st
atus_xf[0][0]

                                                                 occupation
_xf[0][0]

                                                                 race_xf
[0][0]

                                                                 relationsh
ip_xf[0][0]

                                                                 workclass_
xf[0][0]

                                                                 sex_xf
[0][0]

                                                                 native-cou
ntry_xf[0][0]

                                                                 age_xf
[0][0]
_____
dense_3 (Dense)                 (None, 34)           1666        dense_2
[0][0]
_____
dense_4 (Dense)                 (None, 128)          18816       concatenat
e_1[0][0]
_____
concatenate_2 (Concatenate)     (None, 162)          0           dense_3
[0][0]
                                                                 dense_4
[0][0]
_____
dense_5 (Dense)                 (None, 1)            163         concatenat
e_2[0][0]
================================================================================
========================
Total params: 31,723
Trainable params: 31,723
Non-trainable params: 0

_____
50/50 [==============================] - 2s 15ms/step - loss: 0.5082 - bina
ry accuracy: 0.7625 - val loss: 0.4058 - val binary accuracy: 0.8075
```

Out[24]: ▼**ExecutionResult** at 0x7f0ac845d150

　　　　　　.execution_id　18

　　　　　　.component　▶**Trainer** at 0x7f0ac7731f50

　　.component.inputs

　　　　　　　　['examples']　▶**Channel** of type **'Examples'** (1 artifact) at 0x7f0ac8615510

　　　　　　　　['transform_graph']　▶**Channel** of type **'TransformGraph'** (1 artifact) at 0x7f0ac86151d0

　　　　　　　　['schema']　▶**Channel** of type **'Schema'** (1 artifact) at 0x7f0ac85a2f50

　　.component.outputs

　　　　　　　　['model']　▶**Channel** of type **'Model'** (1 artifact) at 0x7f0ac7731a90

Let's review the outputs of this component. The `model` output points to the model output itself.

In [25]:
```python
# Get `model` output of the component
model_artifact_dir = trainer.outputs['model'].get()[0].uri

# List top-level directory
pp.pprint(os.listdir(model_artifact_dir))

# Get model directory
model_dir = os.path.join(model_artifact_dir, 'Format-Serving')

# List subdirectories
pp.pprint(os.listdir(model_dir))
```

```
['Format-Serving']
['keras_metadata.pb', 'saved_model.pb', 'assets', 'variables']
```

The `model_run` output acts as the working directory and can be used to output non-model related output (e.g., TensorBoard logs).

In [26]:
```python
# Get `model_run` output URI
model_run_artifact_dir = trainer.outputs['model_run'].get()[0].uri

# Load results to Tensorboard
%load_ext tensorboard
%tensorboard --logdir {model_run_artifact_dir}
```

Evaluator

The `Evaluator` component computes model performance metrics over the evaluation set using the TensorFlow Model Analysis library. The `Evaluator` can also optionally validate that a newly trained model is better than the previous model. This is useful in a production pipeline setting where you may automatically train and validate a model every day.

There's a few steps needed to setup this component and you will do it in the next cells.

Define EvalConfig

First, you will define the `EvalConfig` message as you did in the previous lab. You can also set thresholds so you can compare subsequent models to it. The module below should look familiar. One minor difference is you don't have to define the candidate and baseline model names in the `model_specs`. That is automatically detected.

```python
import tensorflow_model_analysis as tfma
from google.protobuf import text_format

eval_config = text_format.Parse("""
  ## Model information
  model_specs {
    # This assumes a serving model with signature 'serving_default'.
    label_key: "label"
  }

  ## Post training metric information
  metrics_specs {
    metrics { class_name: "ExampleCount" }
    metrics {
      class_name: "BinaryAccuracy"
      threshold {
        # Ensure that metric is always > 0.5
        value_threshold {
          lower_bound { value: 0.5 }
        }
        # Ensure that metric does not drop by more than a small epsilon
        # e.g. (candidate - baseline) > -1e-10 or candidate > baseline - 1e
        change_threshold {
          direction: HIGHER_IS_BETTER
          absolute { value: -1e-10 }
        }
      }
    }
    metrics { class_name: "BinaryCrossentropy" }
    metrics { class_name: "AUC" }
    metrics { class_name: "AUCPrecisionRecall" }
    metrics { class_name: "Precision" }
    metrics { class_name: "Recall" }
    metrics { class_name: "MeanLabel" }
    metrics { class_name: "MeanPrediction" }
    metrics { class_name: "Calibration" }
    metrics { class_name: "CalibrationPlot" }
    metrics { class_name: "ConfusionMatrixPlot" }
    # ... add additional metrics and plots ...
  }

  ## Slicing information
  slicing_specs {}  # overall slice
  slicing_specs {
    feature_keys: ["race"]
  }
  slicing_specs {
    feature_keys: ["sex"]
  }
""", tfma.EvalConfig())
```

Resolve latest blessed model

If you remember in the last lab, you were able to validate a candidate model against a baseline by modifying the `EvalConfig` and `EvalSharedModel` definitions. That is also possible using the `Evaluator` component and you will see how it is done in this section.

First thing to note is that the `Evaluator` marks a model as `BLESSED` if it meets the metrics thresholds you set in the eval config module. You can load the latest blessed model by using the `LatestBlessedModelStrategy` with the Resolver component. This component takes care of finding the latest blessed model for you so you don't have to remember it manually. The syntax is shown below.

```
In [28]:    from tfx.dsl.components.common.resolver import Resolver
            from tfx.dsl.input_resolution.strategies.latest_blessed_model_strategy impc
            from tfx.types import Channel
            from tfx.types.standard_artifacts import Model, ModelBlessing

            # Setup the Resolver node to find the latest blessed model
            model_resolver = Resolver(
                    strategy_class=LatestBlessedModelStrategy,
                    model=Channel(type=Model),
                    model_blessing=Channel(
                        type=ModelBlessing)).with_id(
                            'latest_blessed_model_resolver')

            # Run the Resolver node
            context.run(model_resolver)
```

Out[28]:

▼**ExecutionResult** at 0x7f0ac86e80d0

.execution_id	19
.component	<tfx.dsl.components.common.resolver.Resolver object at 0x7f0ac811b550>

.component.inputs

['model']	▶**Channel** of type **'Model'** (0 artifacts) at 0x7f0ac814fad0
['model_blessing']	▶**Channel** of type **'ModelBlessing'** (0 artifacts) at 0x7f0ac817fc90

.component.outputs

['model']	▶**Channel** of type **'Model'** (0 artifacts) at 0x7f0ac40b2a50
['model_blessing']	▶**Channel** of type **'ModelBlessing'** (0 artifacts) at 0x7f0ac075bf50

As expected, the search yielded 0 artifacts because you haven't evaluated any models yet. You will run this component again in later parts of this notebook and you will see a different result.

```
In [29]:    # Load Resolver outputs
            model_resolver.outputs['model']
```

Out[29]:

▼**Channel** of type **'Model'** (0 artifacts) at 0x7f0ac40b2a50

.type_name	Model
._artifacts	[]

Run the Evaluator component

With the `EvalConfig` defined and code to load the latest blessed model available, you can proceed to run the `Evaluator` component.

You will notice that two models are passed into the component. The `Trainer` output will serve as the candidate model while the output of the `Resolver` will be the baseline model. While you can indeed run the `Evaluator` without comparing two models, it would likely be required in production environments so it's best to include it. Since the `Resolver` doesn't have any results yet, the `Evaluator` will just mark the candidate model as `BLESSED` in this run.

Aside from the eval config and models (i.e. Trainer and Resolver output), you will also pass in the *raw* examples from `ExampleGen`. By default, the component will look for the `eval` split of these examples and since you've defined the serving signature, these will be transformed internally before feeding to the model inputs.

Note: You can ignore the `Exception ignored in: <function CapturableResource.__del__>` *prompt which generates a long traceback. This might be an issue with the underlying TFMA version used and has been flagged to the TFX team so it can be suppressed.*

In [30]:

```python
# Setup and run the Evaluator component
evaluator = Evaluator(
    examples=example_gen.outputs['examples'],
    model=trainer.outputs['model'],
    baseline_model=model_resolver.outputs['model'],
    eval_config=eval_config)
context.run(evaluator, enable_cache=False)
```

```
ERROR:absl:There are change thresholds, but the baseline is missing. This i
s allowed only when rubber stamping (first run).
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
```

```
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
```

```
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
```

```
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
```

```
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
```

```
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
```

```
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
```

Out[30]:

▼**ExecutionResult** at 0x7f0ac85d5e50

    .execution_id 20

    .component ▶**Evaluator** at 0x7f0ac02117d0

    .component.inputs

        ['examples'] ▶**Channel** of type **'Examples'** (1 artifact) at 0x7f0acd962fd0

        ['model'] ▶**Channel** of type **'Model'** (1 artifact) at 0x7f0ac7731a90

        ['baseline_model'] ▶**Channel** of type **'Model'** (0 artifacts) at 0x7f0ac40b2a50

    .component.outputs

        ['evaluation'] ▶**Channel** of type **'ModelEvaluation'** (1 artifact) at 0x7f0ac0211d90

        ['blessing'] ▶**Channel** of type **'ModelBlessing'** (1 artifact) at 0x7f0ac0211f50

Now let's examine the output artifacts of `Evaluator` .

```
In [31]:    # Print component output keys
            evaluator.outputs.keys()
```

Out[31]:   dict_keys(['evaluation', 'blessing'])

The `blessing` output simply states if the candidate model was blessed. The artifact URI will have a `BLESSED` or `NOT_BLESSED` file depending on the result. As mentioned earlier, this first run will pass the evaluation because there is no baseline model yet.

```
In [32]:    # Get `Evaluator` blessing output URI
            blessing_uri = evaluator.outputs['blessing'].get()[0].uri

            # List files under URI
            os.listdir(blessing_uri)
```

Out[32]:   ['BLESSED']

The `evaluation` output, on the other hand, contains the evaluation logs and can be used to visualize the global metrics on the entire evaluation set.

```
In [33]:    # Visualize the evaluation results
            context.show(evaluator.outputs['evaluation'])
```

Artifact at ./pipeline/Evaluator/evaluation/20

To see the individual slices, you will need to import TFMA and use the commands you learned in the previous lab.

```
In [34]:    import tensorflow_model_analysis as tfma

            # Get the TFMA output result path and load the result.
            PATH_TO_RESULT = evaluator.outputs['evaluation'].get()[0].uri
            tfma_result = tfma.load_eval_result(PATH_TO_RESULT)

            # Show data sliced along feature column trip_start_hour.
            tfma.view.render_slicing_metrics(
                tfma_result, slicing_column='sex')
```

You can also use TFMA to load the validation results as before by specifying the output URI of the evaluation output. This would be more useful if your model was not blessed because you can see the metric failure prompts. Try to simulate this later by training with fewer epochs (or raising the threshold) and see the results you get here.

```
In [35]:    # Get `evaluation` output URI
            PATH_TO_RESULT = evaluator.outputs['evaluation'].get()[0].uri

            # Print validation result
            print(tfma.load_validation_result(PATH_TO_RESULT))
```

```
validation_ok: true
validation_details {
  slicing_details {
    slicing_spec {
    }
    num_matching_slices: 8
  }
}
```

Now that your `Evaluator` has finished running, the `Resolver` component should be able to detect the latest blessed model. Let's run the component again.

In [36]:
```python
# Re-run the Resolver component
context.run(model_resolver)
```

Out[36]:

▼**ExecutionResult** at 0x7f0abfc93850

.execution_id   21

.component   <tfx.dsl.components.common.resolver.Resolver object at 0x7f0ac811b550>

.component.inputs

['model']   ▶Channel of type **'Model'** (0 artifacts) at 0x7f0ac814fad0

['model_blessing']   ▶Channel of type **'ModelBlessing'** (0 artifacts) at 0x7f0ac817fc90

.component.outputs

['model']   ▶Channel of type **'Model'** (1 artifact) at 0x7f0ab9bb1f50

['model_blessing']   ▶Channel of type **'ModelBlessing'** (1 artifact) at 0x7f0ab9bb14d0

You should now see an artifact in the component outputs. You can also get it programmatically as shown below.

In [37]:
```python
# Get path to latest blessed model
model_resolver.outputs['model'].get()[0].uri
```

Out[37]:   `'./pipeline/Trainer/model/18'`

Comparing two models

Now let's see how `Evaluator` compares two models. You will train the same model with more epochs and this should hopefully result in higher accuracy and overall metrics.

```python
from tfx.proto import trainer_pb2

# Setup trainer to train with more epochs
trainer = Trainer(
    module_file=os.path.abspath(_census_trainer_module_file),
    examples=transform.outputs['transformed_examples'],
    transform_graph=transform.outputs['transform_graph'],
    schema=schema_gen.outputs['schema'],
    train_args=trainer_pb2.TrainArgs(num_steps=500),
    eval_args=trainer_pb2.EvalArgs(num_steps=200))

# Run trainer
context.run(trainer, enable_cache=False)
```

```
WARNING:absl:Examples artifact does not have payload_format custom propert
y. Falling back to FORMAT_TF_EXAMPLE
WARNING:absl:Examples artifact does not have payload_format custom propert
y. Falling back to FORMAT_TF_EXAMPLE
WARNING:absl:Examples artifact does not have payload_format custom propert
y. Falling back to FORMAT_TF_EXAMPLE
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/optimizer_v2
/optimizer_v2.py:375: UserWarning: The `lr` argument is deprecated, use `le
arning_rate` instead.
  "The `lr` argument is deprecated, use `learning_rate` instead.")
```
```
Model: "model_1"
_____
_____
Layer (type)                    Output Shape         Param #     Connected
to
==========================================================================
==========================
fnlwgt_xf (InputLayer)          [(None, 1)]          0
_____
_____
education-num_xf (InputLayer)   [(None, 1)]          0
_____
_____
capital-gain_xf (InputLayer)    [(None, 1)]          0
_____
_____
capital-loss_xf (InputLayer)    [(None, 1)]          0
_____
_____
hours-per-week_xf (InputLayer)  [(None, 1)]          0
_____
_____
concatenate_3 (Concatenate)     (None, 5)            0           fnlwgt_xf
[0][0]
                                                                 education-
num_xf[0][0]
                                                                 capital-ga
in_xf[0][0]
                                                                 capital-lo
ss_xf[0][0]
                                                                 hours-per-
week_xf[0][0]
_____
_____
dense_6 (Dense)                 (None, 100)          600         concatenat
e_3[0][0]
_____
_____
dense_7 (Dense)                 (None, 70)           7070        dense_6
[0][0]
_____
_____
education_xf (InputLayer)       [(None, 21)]         0
_____
_____
marital-status_xf (InputLayer)  [(None, 12)]         0
_____
_____
```

```
_____
occupation_xf (InputLayer)      [(None, 20)]          0
_____
_____
race_xf (InputLayer)            [(None, 10)]          0
_____
_____
relationship_xf (InputLayer)    [(None, 11)]          0
_____
_____
workclass_xf (InputLayer)       [(None, 14)]          0
_____
_____
sex_xf (InputLayer)             [(None, 7)]           0
_____
_____
native-country_xf (InputLayer)  [(None, 47)]          0
_____
_____
age_xf (InputLayer)             [(None, 4)]           0
_____
_____
dense_8 (Dense)                 (None, 48)            3408        dense_7
[0][0]
_____
_____
concatenate_4 (Concatenate)     (None, 146)           0           education_
xf[0][0]
                                                                  marital-st
atus_xf[0][0]
                                                                  occupation
_xf[0][0]
                                                                  race_xf
[0][0]
                                                                  relationsh
ip_xf[0][0]
                                                                  workclass_
xf[0][0]
                                                                  sex_xf
[0][0]
                                                                  native-cou
ntry_xf[0][0]
                                                                  age_xf
[0][0]
_____
_____
dense_9 (Dense)                 (None, 34)            1666        dense_8
[0][0]
_____
_____
dense_10 (Dense)                (None, 128)           18816       concatenat
e_4[0][0]
_____
_____
concatenate_5 (Concatenate)     (None, 162)           0           dense_9
[0][0]
                                                                  dense_10
[0][0]
_____
_____
dense_11 (Dense)                (None, 1)             163         concatenat
e_5[0][0]
================================================================
========================
Total params: 31,723
Trainable params: 31,723
Non-trainable params: 0
_____
_____
500/500 [==============================] - 4s 6ms/step - loss: 0.3535 - bin
```

Out[38]:

▼**ExecutionResult** at 0x7f0abaacfd90

    .execution_id 22

.component ▶ **Trainer** at 0x7f0ac860c910

.component.inputs

['examples'] ▶ **Channel** of type **'Examples'** (1 artifact) at 0x7f0ac8615510

['transform_graph'] ▶ **Channel** of type **'TransformGraph'** (1 artifact) at 0x7f0ac86151d0

['schema'] ▶ **Channel** of type **'Schema'** (1 artifact) at 0x7f0ac85a2f50

.component.outputs

['model'] ▶ **Channel** of type **'Model'** (1 artifact) at 0x7f0ac860ced0

['model_run'] ▶ **Channel** of type **'ModelRun'** (1 artifact) at 0x7f0ac851d550

You will re-run the evaluator but you will specify the latest trainer output as the candidate model. The baseline is automatically found with the Resolver node.

In [39]:

```python
# Re-run the evaluator
evaluator = Evaluator(
    examples=example_gen.outputs['examples'],
    model=trainer.outputs['model'],
    baseline_model=model_resolver.outputs['model'],
    eval_config=eval_config)

context.run(evaluator, enable_cache=False)
```

```
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
```

```
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
```

```
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
```

```
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
```

```
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
```

```
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
```

```
        filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
Exception ignored in: <function CapturableResource.__del__ at 0x7f0af1d10ef
0>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/training/t
racking/tracking.py", line 277, in __del__
    self._destroy_resource()
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 889, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 924, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3022, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3444, in _maybe_define_function
    graph_function = self._create_graph_function(args, kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/func
tion.py", line 3289, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/
func_graph.py", line 999, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/def_
function.py", line 672, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
AttributeError: 'NoneType' object has no attribute '__wrapped__'
WARNING:root:Make sure that locally built Python SDK docker image has Pytho
```

Out[39]: ▼**ExecutionResult** at 0x7f0ab9b32090

    .execution_id 23

    .component ▶**Evaluator** at 0x7f0ab9b3c3d0

.component.inputs

    ['examples'] ▶**Channel** of type **'Examples'** (1 artifact) at 0x7f0acd962fd0

    ['model'] ▶**Channel** of type **'Model'** (1 artifact) at 0x7f0ac860ced0

    ['baseline_model'] ▶**Channel** of type **'Model'** (1 artifact) at 0x7f0ab9bb1f50

.component.outputs

    ['evaluation'] ▶**Channel** of type **'ModelEvaluation'** (1 artifact) at 0x7f0ab9b3c410

    ['blessing'] ▶**Channel** of type **'ModelBlessing'** (1 artifact) at 0x7f0ab9b3c550

Depending on the result, the Resolver should reflect the latest blessed model. Since you trained with more epochs, it is most likely that your candidate model will pass the thresholds you set in the eval config. If so, the artifact URI should be different here compared to your earlier runs.

```
In [40]:    # Re-run the resolver
            context.run(model_resolver, enable_cache=False)
```

Out[40]:

▼**ExecutionResult** at 0x7f0ab9b36910

.execution_id	24
.component	\<tfx.dsl.components.common.resolver.Resolver object at 0x7f0ac811b550\>

.component.inputs

['model']	▶**Channel** of type **'Model'** (0 artifacts) at 0x7f0ac814fad0
['model_blessing']	▶**Channel** of type **'ModelBlessing'** (0 artifacts) at 0x7f0ac817fc90

.component.outputs

['model']	▶**Channel** of type **'Model'** (1 artifact) at 0x7f0ac2155a10
['model_blessing']	▶**Channel** of type **'ModelBlessing'** (1 artifact) at 0x7f0ac20d3bd0

```
In [41]:    # Get path to latest blessed model
            model_resolver.outputs['model'].get()[0].uri
```

Out[41]: './pipeline/Trainer/model/22'

Finally, the `evaluation` output of the `Evaluator` component will now be able to produce the `diff` results you saw in the previous lab. This will signify if the metrics from the candidate model has indeed improved compared to the baseline. Unlike when using TFMA as a standalone library, visualizing this will just show the results for the candidate (i.e. only one row instead of two in the tabular output in the graph below).

Note: You can ignore the warning about failing to find plots.

```
In [42]:    context.show(evaluator.outputs['evaluation'])
```

Artifact at ./pipeline/Evaluator/evaluation/23

```
WARNING:absl:Fail to find plots for model name: None . Available model name
s are [candidate, baseline]
WARNING:absl:Fail to find plots for model name: None . Available model name
s are [candidate, baseline]
WARNING:absl:Fail to find plots for model name: None . Available model name
```

```
s are [candidate, baseline]
WARNING:absl:Fail to find plots for model name: None . Available model name
s are [candidate, baseline]
WARNING:absl:Fail to find plots for model name: None . Available model name
s are [candidate, baseline]
WARNING:absl:Fail to find plots for model name: None . Available model name
s are [candidate, baseline]
WARNING:absl:Fail to find plots for model name: None . Available model name
s are [candidate, baseline]
WARNING:absl:Fail to find plots for model name: None . Available model name
s are [candidate, baseline]
WARNING:absl:Fail to find plots for model name: None . Available model name
s are [candidate, baseline]
```

Congratulations! You can now successfully evaluate your models in a TFX pipeline! This is a critical part of production ML because you want to make sure that subsequent deployments are indeed improving your results. Moreover, you can extract the evaluation results from your pipeline directory for further investigation with TFMA. In the next sections, you will continue to study techniques related to model evaluation and ensuring fairness.