```
!pip install -U tfx
```

```
Collecting tfx
  Downloading tfx-1.15.1-py3-none-any.whl (3.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.0/3.0 MB 36.8 MB/s eta 0:00:00
Collecting ml-pipelines-sdk==1.15.1 (from tfx)
  Downloading ml_pipelines_sdk-1.15.1-py3-none-any.whl (1.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 69.5 MB/s eta 0:00:00
Requirement already satisfied: absl-py<2.0.0,>=0.9 in /usr/local/lib/python3.10/dist-packages (from tfx) (1.4.0)
Collecting ml-metadata<1.16.0,>=1.15.0 (from tfx)
  Downloading ml_metadata-1.15.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.5/7.5 MB 96.7 MB/s eta 0:00:00
Requirement already satisfied: packaging>=22 in /usr/local/lib/python3.10/dist-packages (from tfx) (24.1)
Requirement already satisfied: portpicker<2,>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from tfx) (1.5.2)
Requirement already satisfied: protobuf<5,>=3.20.3 in /usr/local/lib/python3.10/dist-packages (from tfx) (3.20.3)
Collecting docker<5,>=4.1 (from tfx)
  Downloading docker-4.4.4-py2.py3-none-any.whl (147 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 147.0/147.0 kB 22.7 MB/s eta 0:00:00
Collecting google-apitools<1,>=0.5 (from tfx)
  Downloading google_apitools-0.5.32-py3-none-any.whl (135 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 135.7/135.7 kB 21.7 MB/s eta 0:00:00
Collecting google-api-python-client<2,>=1.8 (from tfx)
  Downloading google_api_python_client-1.12.11-py2.py3-none-any.whl (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.1/62.1 kB 8.8 MB/s eta 0:00:00
Requirement already satisfied: jinja2<4,>=2.7.3 in /usr/local/lib/python3.10/dist-packages (from tfx) (3.1.4)
Requirement already satisfied: typing-extensions<5,>=3.10.0.2 in /usr/local/lib/python3.10/dist-packages (from tfx) (4.12.2)
Collecting apache-beam[gcp]<3,>=2.47 (from tfx)
  Downloading apache_beam-2.57.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (14.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.5/14.5 MB 87.0 MB/s eta 0:00:00
Requirement already satisfied: attrs<24,>=19.3.0 in /usr/local/lib/python3.10/dist-packages (from tfx) (23.2.0)
Requirement already satisfied: click<9,>=7 in /usr/local/lib/python3.10/dist-packages (from tfx) (8.1.7)
Requirement already satisfied: google-api-core<3 in /usr/local/lib/python3.10/dist-packages (from tfx) (2.16.2)
Requirement already satisfied: google-cloud-aiplatform<2,>=1.6.2 in /usr/local/lib/python3.10/dist-packages (from tfx) (1.59.0)
Requirement already satisfied: google-cloud-bigquery<4,>=3 in /usr/local/lib/python3.10/dist-packages (from tfx) (3.21.0)
Requirement already satisfied: grpcio<2,>=1.28.1 in /usr/local/lib/python3.10/dist-packages (from tfx) (1.64.1)
Collecting keras-tuner!=1.4.0,!=1.4.1,<2,>=1.0.4 (from tfx)
  Downloading keras_tuner-1.4.7-py3-none-any.whl (129 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 129.1/129.1 kB 18.4 MB/s eta 0:00:00
Collecting kubernetes<13,>=10.0.1 (from tfx)
  Downloading kubernetes-12.0.1-py2.py3-none-any.whl (1.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 87.4 MB/s eta 0:00:00
Requirement already satisfied: numpy<2,>=1.16 in /usr/local/lib/python3.10/dist-packages (from tfx) (1.25.2)
Collecting pyarrow<11,>=10 (from tfx)
  Downloading pyarrow-10.0.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (35.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 35.9/35.9 MB 47.6 MB/s eta 0:00:00
Requirement already satisfied: scipy<1.13 in /usr/local/lib/python3.10/dist-packages (from tfx) (1.11.4)
Requirement already satisfied: pyyaml<7,>=6 in /usr/local/lib/python3.10/dist-packages (from tfx) (6.0.1)
Requirement already satisfied: tensorflow<2.16,>=2.15.0 in /usr/local/lib/python3.10/dist-packages (from tfx) (2.15.0)
Collecting tensorflow-hub<0.16,>=0.15.0 (from tfx)
  Downloading tensorflow_hub-0.15.0-py2.py3-none-any.whl (85 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 85.4/85.4 kB 13.4 MB/s eta 0:00:00
Collecting tensorflow-data-validation<1.16.0,>=1.15.1 (from tfx)
  Downloading tensorflow_data_validation-1.15.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (19.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 19.0/19.0 MB 79.1 MB/s eta 0:00:00
Collecting tensorflow-model-analysis<0.47.0,>=0.46.0 (from tfx)
  Downloading tensorflow_model_analysis-0.46.0-py3-none-any.whl (1.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 93.8 MB/s eta 0:00:00
Collecting tensorflow-serving-api<2.16,>=2.15 (from tfx)
  Downloading tensorflow_serving_api-2.15.1-py2.py3-none-any.whl (26 kB)
Collecting tensorflow-transform<1.16.0,>=1.15.0 (from tfx)
  Downloading tensorflow_transform-1.15.0-py3-none-any.whl (451 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 451.2/451.2 kB 44.8 MB/s eta 0:00:00
Collecting tfx-bsl<1.16.0,>=1.15.1 (from tfx)
  Downloading tfx_bsl-1.15.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (22.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 22.5/22.5 MB 60.4 MB/s eta 0:00:00
Collecting crcmod<2.0,>=1.7 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading crcmod-1.7.tar.gz (89 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 89.7/89.7 kB 14.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting orjson<4,>=3.9.7 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading orjson-3.10.6-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (141 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 141.1/141.1 kB 21.4 MB/s eta 0:00:00
Collecting dill<0.3.2,>=0.3.1.1 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading dill-0.3.1.1.tar.gz (151 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 152.0/152.0 kB 23.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: cloudpickle~=2.2.1 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47->tfx)
Collecting fastavro<2,>=0.23.6 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading fastavro-1.9.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.1/3.1 MB 26.9 MB/s eta 0:00:00
Collecting fasteners<1.0,>=0.3 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading fasteners-0.19-py3-none-any.whl (18 kB)
Collecting hdfs<3.0.0,>=2.1.0 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading hdfs-2.7.3.tar.gz (43 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.5/43.5 kB 6.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: httplib2<0.23.0,>=0.8 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47->tf
Requirement already satisfied: jsonschema<5.0.0,>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47-
Requirement already satisfied: jsonpickle<4.0.0,>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47-
Collecting objsize<0.8.0,>=0.6.1 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading objsize-0.7.0-py3-none-any.whl (11 kB)
Collecting pymongo<5.0.0,>=3.8.0 (from apache-beam[gcp]<3,>=2.47->tfx)
```

```
  Downloading pymongo-4.8.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 80.8 MB/s eta 0:00:00
Requirement already satisfied: proto-plus<2,>=1.7.1 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47->tfx
Requirement already satisfied: pydot<2,>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47->tfx) (1.4
Requirement already satisfied: python-dateutil<3,>=2.8.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47
Requirement already satisfied: pytz>=2018.3 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47->tfx) (2023.4
Collecting redis<6,>=5.0.0 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading redis-5.0.7-py3-none-any.whl (252 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 252.1/252.1 kB 31.8 MB/s eta 0:00:00
Requirement already satisfied: regex>=2020.6.8 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47->tfx) (202
Requirement already satisfied: requests!=2.32.*,<3.0.0,>=2.24.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,
Collecting zstandard<1,>=0.18.0 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading zstandard-0.23.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (5.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.4/5.4 MB 115.1 MB/s eta 0:00:00
Requirement already satisfied: pyarrow-hotfix<1 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47->tfx) (0
Collecting js2py<1,>=0.74 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading Js2Py-0.74-py3-none-any.whl (1.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 72.2 MB/s eta 0:00:00
Requirement already satisfied: cachetools<6,>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47->tfx
Collecting google-apitools<1,>=0.5 (from tfx)
  Downloading google-apitools-0.5.31.tar.gz (173 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 173.5/173.5 kB 24.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: google-auth<3,>=1.18.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.47->tf
Requirement already satisfied: google-auth-httplib2<0.3.0,>=0.1.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<
Requirement already satisfied: google-cloud-datastore<3,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,
Requirement already satisfied: google-cloud-pubsub<3,>=2.1.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2
Collecting google-cloud-pubsublite<2,>=1.2.0 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading google_cloud_pubsublite-1.11.0-py2.py3-none-any.whl (303 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 304.0/304.0 kB 40.3 MB/s eta 0:00:00
Collecting google-cloud-storage<3,>=2.16.0 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading google_cloud_storage-2.17.0-py2.py3-none-any.whl (126 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 126.5/126.5 kB 14.8 MB/s eta 0:00:00
Requirement already satisfied: google-cloud-bigquery-storage<3,>=2.6.3 in /usr/local/lib/python3.10/dist-packages (from apache-beam
Requirement already satisfied: google-cloud-core<3,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2.4
Requirement already satisfied: google-cloud-bigtable<3,>=2.19.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,
Collecting google-cloud-spanner<4,>=3.0.0 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading google_cloud_spanner-3.47.0-py2.py3-none-any.whl (384 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 384.6/384.6 kB 48.7 MB/s eta 0:00:00
Collecting google-cloud-dlp<4,>=3.0.0 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading google_cloud_dlp-3.18.1-py2.py3-none-any.whl (180 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 180.4/180.4 kB 27.5 MB/s eta 0:00:00
Requirement already satisfied: google-cloud-language<3,>=2.0 in /usr/local/lib/python3.10/dist-packages (from apache-beam[gcp]<3,>=2
Collecting google-cloud-videointelligence<3,>=2.0 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading google_cloud_videointelligence-2.13.4-py2.py3-none-any.whl (244 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 245.0/245.0 kB 34.8 MB/s eta 0:00:00
Collecting google-cloud-vision<4,>=2 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading google_cloud_vision-3.7.3-py2.py3-none-any.whl (466 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 466.4/466.4 kB 50.7 MB/s eta 0:00:00
Collecting google-cloud-recommendations-ai<0.11.0,>=0.1.0 (from apache-beam[gcp]<3,>=2.47->tfx)
  Downloading google_cloud_recommendations_ai-0.10.11-py2.py3-none-any.whl (183 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 183.5/183.5 kB 27.7 MB/s eta 0:00:00
Requirement already satisfied: six>=1.4.0 in /usr/local/lib/python3.10/dist-packages (from docker<5,>=4.1->tfx) (1.16.0)
Requirement already satisfied: websocket-client>=0.32.0 in /usr/local/lib/python3.10/dist-packages (from docker<5,>=4.1->tfx) (1.8.6
Requirement already satisfied: googleapis-common-protos<2.0.dev0,>=1.56.2 in /usr/local/lib/python3.10/dist-packages (from google-ap
Collecting uritemplate<4dev,>=3.0.0 (from google-api-python-client<2,>=1.8->tfx)
  Downloading uritemplate-3.0.1-py2.py3-none-any.whl (15 kB)
Requirement already satisfied: oauth2client>=1.4.12 in /usr/local/lib/python3.10/dist-packages (from google-apitools<1,>=0.5->tfx)
Requirement already satisfied: google-cloud-resource-manager<3.0.0dev,>=1.3.3 in /usr/local/lib/python3.10/dist-packages (from goog
Requirement already satisfied: shapely<3.0.0dev in /usr/local/lib/python3.10/dist-packages (from google-cloud-aiplatform<2,>=1.6.2-
Requirement already satisfied: pydantic<3 in /usr/local/lib/python3.10/dist-packages (from google-cloud-aiplatform<2,>=1.6.2->tfx)
Requirement already satisfied: docstring-parser<1 in /usr/local/lib/python3.10/dist-packages (from google-cloud-aiplatform<2,>=1.6.2
Requirement already satisfied: google-resumable-media<3.0dev,>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from google-cloud-b
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2<4,>=2.7.3->tfx) (2.1.5)
Requirement already satisfied: keras in /usr/local/lib/python3.10/dist-packages (from keras-tuner!=1.4.0,!=1.4.1,<2,>=1.0.4->tfx) (2
Collecting kt-legacy (from keras-tuner!=1.4.0,!=1.4.1,<2,>=1.0.4->tfx)
  Downloading kt_legacy-1.0.5-py3-none-any.whl (9.6 kB)
Requirement already satisfied: certifi>=14.05.14 in /usr/local/lib/python3.10/dist-packages (from kubernetes<13,>=10.0.1->tfx) (2024
Requirement already satisfied: setuptools>=21.0.0 in /usr/local/lib/python3.10/dist-packages (from kubernetes<13,>=10.0.1->tfx) (67
Requirement already satisfied: requests-oauthlib in /usr/local/lib/python3.10/dist-packages (from kubernetes<13,>=10.0.1->tfx) (1.3
Requirement already satisfied: urllib3>=1.24.2 in /usr/local/lib/python3.10/dist-packages (from kubernetes<13,>=10.0.1->tfx) (2.0.7
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from portpicker<2,>=1.3.1->tfx) (5.9.5)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15.0->tfx) (1
Requirement already satisfied: flatbuffers>=23.5.26 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15.0->tfx)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15.0->tfx)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15.0->tfx) (3.9.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15.0->tfx) (18
Requirement already satisfied: ml-dtypes~=0.2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15.0->tfx) (0.2
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15.0->tfx) (3
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15.0->tfx) (2.4
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15.0->tfx)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16
Requirement already satisfied: tensorboard<2.16,>=2.15 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>=2.15.0->tf
Requirement already satisfied: tensorflow-estimator<2.16,>=2.15.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow<2.16,>
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-data-validation<1.16.0,>=1
Collecting pandas<2,>=1.0 (from tensorflow-data-validation<1.16.0,>=1.15.1->tfx)
  Downloading pandas-1.5.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.1/12.1 MB 103.2 MB/s eta 0:00:00
Collecting pyfarmhash<0.4,>=0.2.2 (from tensorflow-data-validation<1.16.0,>=1.15.1->tfx)
  Downloading pyfarmhash-0.3.2.tar.gz (99 kB)
```

```
  Downloading pyrammash 0.9.2.tar.gz (99 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.9/99.9 kB 16.8 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: tensorflow-metadata<1.16,>=1.15.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-data-va
Requirement already satisfied: ipython<8,>=7 in /usr/local/lib/python3.10/dist-packages (from tensorflow-model-analysis<0.47.0,>=0.4
Requirement already satisfied: ipywidgets<8,>=7 in /usr/local/lib/python3.10/dist-packages (from tensorflow-model-analysis<0.47.0,>=
Requirement already satisfied: pillow>=9.4.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-model-analysis<0.47.0,>=0.4
Collecting rouge-score<2,>=0.1.2 (from tensorflow-model-analysis<0.47.0,>=0.46.0->tfx)
  Downloading rouge_score-0.1.2.tar.gz (17 kB)
  Preparing metadata (setup.py) ... done
Collecting sacrebleu<4,>=2.3 (from tensorflow-model-analysis<0.47.0,>=0.46.0->tfx)
  Downloading sacrebleu-2.4.2-py3-none-any.whl (106 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 106.7/106.7 kB 17.4 MB/s eta 0:00:00
Collecting tensorflow<2.16,>=2.15.0 (from tfx)
  Downloading tensorflow-2.15.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (475.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 475.2/475.2 MB 3.1 MB/s eta 0:00:00
Collecting ml-dtypes~=0.3.1 (from tensorflow<2.16,>=2.15.0->tfx)
  Downloading ml_dtypes-0.3.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.2/2.2 MB 95.1 MB/s eta 0:00:00
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow<2.1
Requirement already satisfied: grpcio-status<2.0.dev0,>=1.33.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core<3->t
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.18.0->apache
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.18.0->apache-beam[gc
Requirement already satisfied: grpc-google-iam-v1<1.0.0dev,>=0.12.4 in /usr/local/lib/python3.10/dist-packages (from google-cloud-bi
Collecting overrides<8.0.0,>=6.0.1 (from google-cloud-pubsublite<2,>=1.2.0->apache-beam[gcp]<3,>=2.47->tfx)
  Downloading overrides-7.7.0-py3-none-any.whl (17 kB)
Requirement already satisfied: sqlparse>=0.4.4 in /usr/local/lib/python3.10/dist-packages (from google-cloud-spanner<4,>=3.0.0->apac
Collecting grpc-interceptor>=0.15.4 (from google-cloud-spanner<4,>=3.0.0->apache-beam[gcp]<3,>=2.47->tfx)
  Downloading grpc_interceptor-0.15.4-py3-none-any.whl (20 kB)
Requirement already satisfied: google-crc32c<2.0dev,>=1.0 in /usr/local/lib/python3.10/dist-packages (from google-cloud-storage<3,>=
Collecting docopt (from hdfs<3.0.0,>=2.1.0->apache-beam[gcp]<3,>=2.47->tfx)
  Downloading docopt-0.6.2.tar.gz (25 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2 in /usr/local/lib/python3.10/dist-packages (from
Collecting jedi>=0.16 (from ipython<8,>=7->tensorflow-model-analysis<0.47.0,>=0.46.0->tfx)
  Downloading jedi-0.19.1-py2.py3-none-any.whl (1.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 81.3 MB/s eta 0:00:00
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages (from ipython<8,>=7->tensorflow-model-analysis<0
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-packages (from ipython<8,>=7->tensorflow-model-analysis
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.10/dist-packages (from ipython<8,>=7->tensorflow-model-analy
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from ipythor
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-packages (from ipython<8,>=7->tensorflow-model-analysis<0
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-packages (from ipython<8,>=7->tensorflow-model-analysis<0
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/dist-packages (from ipython<8,>=7->tensorflow-model-ar
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-packages (from ipython<8,>=7->tensorflow-model-analysis
Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.10/dist-packages (from ipywidgets<8,>=7->tensorflow-model
Requirement already satisfied: ipython-genutils~=0.2.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets<8,>=7->tensorflow
Requirement already satisfied: widgetsnbextension~=3.6.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets<8,>=7->tensorf
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from ipywidgets<8,>=7->tensorf
Requirement already satisfied: tzlocal>=1.2 in /usr/local/lib/python3.10/dist-packages (from js2py<1,>=0.74->apache-beam[gcp]<3,>=2
Collecting pyjsparser>=2.5.1 (from js2py<1,>=0.74->apache-beam[gcp]<3,>=2.47->tfx)
  Downloading pyjsparser-2.7.1.tar.gz (24 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (from jsonschema<5.0
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema<5.0.0,>=4.0.0->apache
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema<5.0.0,>=4.0.0->apache-beam
Requirement already satisfied: pyasn1>=0.1.7 in /usr/local/lib/python3.10/dist-packages (from oauth2client>=1.4.12->google-apitools<
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3->google-cloud-aipl
Requirement already satisfied: pydantic-core==2.20.1 in /usr/local/lib/python3.10/dist-packages (from pydantic<3->google-cloud-aipla
Collecting dnspython<3.0.0,>=1.16.0 (from pymongo<5.0.0,>=3.8.0->apache-beam[gcp]<3,>=2.47->tfx)
  Downloading dnspython-2.6.1-py3-none-any.whl (307 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 307.7/307.7 kB 35.5 MB/s eta 0:00:00
Requirement already satisfied: async-timeout>=4.0.3 in /usr/local/lib/python3.10/dist-packages (from redis<6,>=5.0.0->apache-beam[gc
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests!=2.32.*,<3.0.0,>=2
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests!=2.32.*,<3.0.0,>=2.24.0->apach
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (from rouge-score<2,>=0.1.2->tensorflow-model-analysi
Collecting portalocker (from sacrebleu<4,>=2.3->tensorflow-model-analysis<0.47.0,>=0.46.0->tfx)
  Downloading portalocker-2.10.1-py3-none-any.whl (18 kB)
Requirement already satisfied: tabulate>=0.8.9 in /usr/local/lib/python3.10/dist-packages (from sacrebleu<4,>=2.3->tensorflow-model
Collecting colorama (from sacrebleu<4,>=2.3->tensorflow-model-analysis<0.47.0,>=0.46.0->tfx)
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages (from sacrebleu<4,>=2.3->tensorflow-model-analysis<0
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15->tensorflow<
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15->tensorflow<
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from requests-oauthlib->kubernetes<13,>=1
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.10/dist-packages (from ipykernel>=4.5.1->ipywidgets<8,>=7->t
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.10/dist-packages (from ipykernel>=4.5.1->ipywidgets<8,>=7->ter
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from jedi>=0.16->ipython<8,>=7->tensc
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.10/dist-packages (from pexpect>4.3->ipython<8,>=7->tensorf
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.6
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.10/dist-packages (from widgetsnbextension~=3.6.0->ipywidget
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk->rouge-score<2,>=0.1.2->tensorflow-model-a
Requirement already satisfied: pyzmq<25,>=17 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6
Requirement already satisfied: jupyter-core>=4.6.1 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextens
Requirement already satisfied: nbformat in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0
Requirement already satisfied: nbconvert>=5 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3
Requirement already satisfied: nest-asyncio>=1.5 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextensic
Requirement already satisfied: Send2Trash>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextensic
```

```
Requirement already satisfied: terminado>=0.8.3 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextensior
Requirement already satisfied: prometheus-client in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextensic
Requirement already satisfied: nbclassic>=0.4.7 in /usr/local/lib/python3.10/dist-packages (from notebook>=4.4.1->widgetsnbextensior
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.10/dist-packages (from jupyter-core>=4.6.1->notebook>=4.4
Requirement already satisfied: notebook-shim>=0.2.3 in /usr/local/lib/python3.10/dist-packages (from nbclassic>=0.4.7->notebook>=4.4
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widget
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbexter
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbe
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->wi
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->w
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->wic
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widge
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbext
Requirement already satisfied: fastjsonschema>=2.15 in /usr/local/lib/python3.10/dist-packages (from nbformat->notebook>=4.4.1->widg
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/python3.10/dist-packages (from argon2-cffi->notebook>=4.4.1->w
Requirement already satisfied: jupyter-server<3,>=1.8 in /usr/local/lib/python3.10/dist-packages (from notebook-shim>=0.2.3->nbclass
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from argon2-cffi-bindings->argon2-cffi->noteb
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4->nbconvert>=5->notebook
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->nbconvert>=5->notebook>=4.4.1->
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2
Requirement already satisfied: anyio<4,>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from jupyter-server<3,>=1.8->notebook-shi
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (from anyio<4,>=3.1.0->jupyter-server<3,>=1.8
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<4,>=3.1.0->jupyter-server<3,>=1
Building wheels for collected packages: google-apitools, crcmod, dill, hdfs, pyfarmhash, rouge-score, pyjsparser, docopt
  Building wheel for google-apitools (setup.py) ... done
  Created wheel for google-apitools: filename=google_apitools-0.5.31-py3-none-any.whl size=131015 sha256=0d529315cca1a7107df4ad6b43
  Stored in directory: /root/.cache/pip/wheels/04/b7/e0/9712f8c23a5da3d9d16fb88216b897bf60e85b12f5470f26ee
  Building wheel for crcmod (setup.py) ... done
  Created wheel for crcmod: filename=crcmod-1.7-cp310-cp310-linux_x86_64.whl size=31406 sha256=9aff88a8c0cf858f9f78e0f57ba989566935
  Stored in directory: /root/.cache/pip/wheels/85/4c/07/72215c529bd59d67e3dac29711d7aba1b692f543c808ba9e86
  Building wheel for dill (setup.py) ... done
  Created wheel for dill: filename=dill-0.3.1.1-py3-none-any.whl size=78541 sha256=ef5c6c9a96a1168da24836defd7179f1d75da9e71ab22f20
  Stored in directory: /root/.cache/pip/wheels/ea/e2/86/64980d90e297e7bf2ce588c2b96e818f5399c515c4bb8a7e4f
  Building wheel for hdfs (setup.py) ... done
  Created wheel for hdfs: filename=hdfs-2.7.3-py3-none-any.whl size=34324 sha256=b40bb4eae5b1c0ba1723efaac13de4b36a33cb814a6d1663d0
  Stored in directory: /root/.cache/pip/wheels/e5/8d/b6/99c1c0a3ac5788c866b0ecd3f48b0134a5910e6ed26011800b
  Building wheel for pyfarmhash (setup.py) ... done
  Created wheel for pyfarmhash: filename=pyfarmhash-0.3.2-cp310-cp310-linux_x86_64.whl size=88655 sha256=8f281f816d9afe13001fc04858
  Stored in directory: /root/.cache/pip/wheels/e0/08/da/f66b1f3258fe3f1e767b2136c5444dbfa9fa3f7944cc5e1983
  Building wheel for rouge-score (setup.py) ... done
  Created wheel for rouge-score: filename=rouge_score-0.1.2-py3-none-any.whl size=24933 sha256=07ebb7be21e13717d1ca906c01f0a57ac16e
  Stored in directory: /root/.cache/pip/wheels/5f/dd/89/461065a73be61a532ff8599a28e9beef17985c9e9c31e541b4
  Building wheel for pyjsparser (setup.py) ... done
  Created wheel for pyjsparser: filename=pyjsparser-2.7.1-py3-none-any.whl size=25984 sha256=4465b7374e6656f75ff2107a487a995cc450957
  Stored in directory: /root/.cache/pip/wheels/5e/81/26/5956478df303e2bf5a85a5df595bb307bd25948a4bab69f7c7
  Building wheel for docopt (setup.py) ... done
  Created wheel for docopt: filename=docopt-0.6.2-py2.py3-none-any.whl size=13706 sha256=08fe6b4413e35ae8fdb87da4926377847c81b07b2e
  Stored in directory: /root/.cache/pip/wheels/fc/ab/d4/5da2067ac95b36618c629a5f93f809425700506f72c9732fac
Successfully built google-apitools crcmod dill hdfs pyfarmhash rouge-score pyjsparser docopt
Installing collected packages: pyjsparser, pyfarmhash, kt-legacy, docopt, crcmod, zstandard, uritemplate, tensorflow-hub, redis, pya
  Attempting uninstall: uritemplate
    Found existing installation: uritemplate 4.1.1
    Uninstalling uritemplate-4.1.1:
      Successfully uninstalled uritemplate-4.1.1
  Attempting uninstall: tensorflow-hub
    Found existing installation: tensorflow-hub 0.16.1
    Uninstalling tensorflow-hub-0.16.1:
      Successfully uninstalled tensorflow-hub-0.16.1
  Attempting uninstall: pyarrow
    Found existing installation: pyarrow 14.0.2
    Uninstalling pyarrow-14.0.2:
      Successfully uninstalled pyarrow-14.0.2
  Attempting uninstall: ml-dtypes
    Found existing installation: ml-dtypes 0.2.0
    Uninstalling ml-dtypes-0.2.0:
      Successfully uninstalled ml-dtypes-0.2.0
  Attempting uninstall: pandas
    Found existing installation: pandas 2.0.3
    Uninstalling pandas-2.0.3:
      Successfully uninstalled pandas-2.0.3
  Attempting uninstall: google-api-python-client
    Found existing installation: google-api-python-client 2.84.0
    Uninstalling google-api-python-client-2.84.0:
      Successfully uninstalled google-api-python-client-2.84.0
  Attempting uninstall: tensorflow
    Found existing installation: tensorflow 2.15.0
    Uninstalling tensorflow-2.15.0:
      Successfully uninstalled tensorflow-2.15.0
  Attempting uninstall: google-cloud-storage
    Found existing installation: google-cloud-storage 2.8.0
    Uninstalling google-cloud-storage-2.8.0:
      Successfully uninstalled google-cloud-storage-2.8.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
cudf-cu12 24.4.1 requires pandas<2.2.2dev0,>=2.0, but you have pandas 1.5.3 which is incompatible.
cudf-cu12 24.4.1 requires pyarrow<15.0.0a0,>=14.0.1, but you have pyarrow 10.0.1 which is incompatible.
google-colab 1.0.0 requires pandas==2.0.3, but you have pandas 1.5.3 which is incompatible.
Successfully installed apache-beam-2.57.0 colorama-0.4.6 crcmod-1.7 dill-0.3.1.1 dnspython-2.6.1 docker-4.4.4 docopt-0.6.2 fastavro
WARNING: The following packages were previously imported in this runtime:
  [google]
You must restart the runtime in order to use newly installed versions
```

You must restart the runtime in order to use newly installed versions.

RESTART SESSION

```
import os

# Path to the directory containing CSV files
directory_path = '/content/drive/MyDrive/1_movies_per_genre'

# List all files in the directory
files = os.listdir(directory_path)
print("Files in directory:", files)

# Step 3: Read All CSV Files
import pandas as pd

# Initialize an empty dictionary to hold DataFrames
dataframes = {}

# Iterate over each file in the directory
for file in files:
    # Check if the file is a CSV
    if file.endswith('.csv'):
        # Construct full file path
        file_path = os.path.join(directory_path, file)
        # Read the CSV file into a DataFrame
        df = pd.read_csv(file_path)
        # Store the DataFrame in the dictionary with the filename (without extension) as the key
        dataframes[file[:-4]] = df

        # Optionally, print the first few rows of each DataFrame
        print(f"First few rows of {file}:")
        print(df.head())
        print("\n")
```

```
1         1394  https://www.imdb.com/title/tt0482571/reviews/_...
2         2254  https://www.imdb.com/title/tt0209144/reviews/_...
3         1269  https://www.imdb.com/title/tt1130884/reviews/_...
4         1355  https://www.imdb.com/title/tt0114814/reviews/_...
```

```
import os

# Path to the directory containing CSV files
directory_path = '/content/drive/MyDrive/2_reviews_per_movie_raw'

# List all files in the directory
files = os.listdir(directory_path)
print("Files in directory:", files)

# Step 3: Read All CSV Files
import pandas as pd

# Initialize an empty dictionary to hold DataFrames
dataframes = {}

# Iterate over each file in the directory
for file in files:
    # Check if the file is a CSV
    if file.endswith('.csv'):
        # Construct full file path
        file_path = os.path.join(directory_path, file)
        # Read the CSV file into a DataFrame
        df = pd.read_csv(file_path)
        # Store the DataFrame in the dictionary with the filename (without extension) as the key
        dataframes[file[:-4]] = df

        # Optionally, print the first few rows of each DataFrame
        print(f"First few rows of {file}:")
        print(df.head())
        print("\n")
```

```
4                                   Carrey on form\n

                                                 review
0  I really do not know what people have against ...
1  Jim Carrey is back to much the same role that ...
2  Starring: Jim Carrey, Morgan Freeman, Jennifer...
3  Bruce Almighty is the story of Bruce Nolan, an...
4  Now either you like Mr Carrey's humour or you ...
```

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/2_reviews_per_movie_raw/10 Cloverfield Lane 2016.csv')
df.head()
```

	username	rating	helpful	total	date	title	review
0	Imme-van-Gorp	7	102	123	30 January 2019	Unfortunately the ending ruined an otherwise ...	This movie is full of suspense. It makes you g...
1	sonofocelot-1	5	385	500	10 May 2016	...oh dear Abrams. Again.\n	I'll leave this review fairly concise. <b...
2	mhodaee	5	110	143	4 August 2017	Fantastic, gripping, thoroughly enjoyable, un...	I give the 5/10 out of the credit I owe to the...
[illegible]	[illegible]	5	[illegible]	[illegible]	5 October [illegible]	[illegible]	First of all, I must say that I was expecting

Next steps: Generate code with df | View recommended plots

```
import dask.dataframe as dd

df1 = dd.read_csv("/content/drive/MyDrive/2_reviews_per_movie_raw//B*.csv")
df1.head()
```

	username	rating	helpful	total	date	title	review
0	lost4wurds	Null	94	116	3 August 2003	What's the Big Deal?\n	I've heard so many critics and average joes ri...
1	Mr. Gore	8	59	73	3 August 1998	Genuinely funny movie\n	David Zucker has directed one of the most enjo...
2	filmfreak-5	Null	56	72	1 June 2004	Indecently hilarious!\n	I once watched Baseketball, got hooked and sin...
3	duntrune	Null	50	65	10 February 2004	Effing brilliant!!!!\n	Just sickly and twistedly funny as hell. Parke...
4	dbarbanti	8	26	32	23 December 2003	Great!...But let me tell you why!\n	I thoroughly enjoyed this flick. I am of the ...

```
df2 = df1[["review", "rating"]]
df2.head()
```

	review	rating
0	I've heard so many critics and average joes ri...	Null
1	David Zucker has directed one of the most enjo...	8
2	I once watched Baseketball, got hooked and sin...	Null
3	Just sickly and twistedly funny as hell. Parke...	Null
4	I thoroughly enjoyed this flick. I am of the ...	8

```
df3 = df2[df2.rating != "Null"]
df3.head()
```

	review	rating
1	David Zucker has directed one of the most enjo...	8
4	I thoroughly enjoyed this flick. I am of the ...	8
5	There are so many words I want to use to descr...	10
6	BASEketball is indeed a really funny movie. Da...	7
7	Why the crap is this movie rated so low?! I've...	10

```
df3.rating = df3.rating.astype("int")
df3.head()
```

	review	rating
1	David Zucker has directed one of the most enjo...	8
4	I thoroughly enjoyed this flick. I am of the ...	8
5	There are so many words I want to use to descr...	10
6	BASEketball is indeed a really funny movie. Da...	7
7	Why the crap is this movie rated so low?! I've...	10

```
df3.rating = (df3.rating > 5).astype("int")
df3.head()
```

	review	rating
1	David Zucker has directed one of the most enjo...	1
4	I thoroughly enjoyed this flick. I am of the ...	1
5	There are so many words I want to use to descr...	1
6	BASEketball is indeed a really funny movie. Da...	1
7	Why the crap is this movie rated so low?! I've...	1

```
df3.to_csv("files.csv", index=False)
```

```
'/content/files.csv/05.part',
'/content/files.csv/06.part',
'/content/files.csv/07.part',
'/content/files.csv/08.part',
'/content/files.csv/09.part',
'/content/files.csv/10.part',
'/content/files.csv/11.part',
'/content/files.csv/12.part',
'/content/files.csv/13.part',
'/content/files.csv/14.part',
'/content/files.csv/15.part',
'/content/files.csv/16.part',
'/content/files.csv/17.part',
'/content/files.csv/18.part',
'/content/files.csv/19.part',
'/content/files.csv/20.part',
'/content/files.csv/21.part',
'/content/files.csv/22.part',
'/content/files.csv/23.part',
'/content/files.csv/24.part',
'/content/files.csv/25.part',
'/content/files.csv/26.part',
'/content/files.csv/27.part',
'/content/files.csv/28.part',
'/content/files.csv/29.part',
'/content/files.csv/30.part',
'/content/files.csv/31.part',
'/content/files.csv/32.part',
'/content/files.csv/33.part',
'/content/files.csv/34.part',
'/content/files.csv/35.part',
'/content/files.csv/36.part',
'/content/files.csv/37.part',
'/content/files.csv/38.part',
'/content/files.csv/39.part',
'/content/files.csv/40.part',
'/content/files.csv/41.part',
'/content/files.csv/42.part',
'/content/files.csv/43.part',
'/content/files.csv/44.part',
'/content/files.csv/45.part',
'/content/files.csv/46.part',
'/content/files.csv/47.part',
'/content/files.csv/48.part',
'/content/files.csv/49.part',
'/content/files.csv/50.part',
'/content/files.csv/51.part',
'/content/files.csv/52.part',
'/content/files.csv/53.part',
'/content/files.csv/54.part',
'/content/files.csv/55.part',
'/content/files.csv/56.part',
'/content/files.csv/57.part',
'/content/files.csv/58.part',
'/content/files.csv/59.part',
'/content/files.csv/60.part',
'/content/files.csv/61.part',
'/content/files.csv/62.part']
```

```
from tfx.components import CsvExampleGen
from tfx.proto import example_gen_pb2
from tfx.orchestration.experimental.interactive.interactive_context import InteractiveContext

context = InteractiveContext( pipeline_root='pipeline')
```

```
WARNING:absl:InteractiveContext metadata_connection_config not provided: using SQLite ML Metadata database at pipeline/metadata.sqli
```

```
import tensorflow as tf
import os
import pprint
pp = pprint.PrettyPrinter()


output = example_gen_pb2.Output(
    split_config = example_gen_pb2.SplitConfig(splits=[
        example_gen_pb2.SplitConfig.Split(name="train", hash_buckets=8),
        example_gen_pb2.SplitConfig.Split(name='eval', hash_buckets=2)
    ])
)

example_gen = CsvExampleGen(input_base='files.csv', output_config=output)
context.run(example_gen)
```

```
WARNING:apache_beam.runners.interactive.interactive_environment:Dependencies required for Interactive Beam PCollection visualizatior
WARNING:apache_beam.io.tfrecordio:Couldn't find python-snappy so the implementation of _TFRecordUtil._masked_crc32c is not as fast a
```

▼**ExecutionResult** at 0x7a6a1423c8b0

.execution_id	1
.component	▶**CsvExampleGen** at 0x7a6a14273010
.component.inputs	{}
.component.outputs	**['examples']** ▶**Channel** of type **'Examples'** (1 artifact) at 0x7a6a140b01f0

```
train_uri = os.path.join(example_gen.outputs['examples'].get()[0].uri, 'Split-train')

tfrecord_filenames = [os.path.join(train_uri, name) for name in os.listdir(train_uri)]

dataset = tf.data.TFRecordDataset(tfrecord_filenames, compression_type='GZIP')

for tfrecord in dataset.take(2):
    serialized_example = tfrecord.numpy()
    example = tf.train.Example()
    example.ParseFromString(serialized_example)
    pp.pprint(example)
```

```
features {
  feature {
    key: "rating"
    value {
      int64_list {
        value: 1
      }
    }
  }
  feature {
    key: "review"
    value {
      bytes_list {
        value: "Bill and Ted\'s Excellent Adventure was probably one of the most compelling films I\'ve experienced since I first wa
      }
    }
  }
}

features {
  feature {
    key: "rating"
    value {
      int64_list {
        value: 1
      }
    }
  }
  feature {
    key: "review"
    value {
      bytes_list {
        value: "\"Bill And Ted\'s Excellent Adventure\" is most definitely just that!  This is just an all around FUN movie!  The pl
      }
    }
```

```
        }
    }
```

```
from tfx.components import StatisticsGen
statistics_gen = StatisticsGen(
    examples = example_gen.outputs['examples']
)
context.run(statistics_gen)
```

▼**ExecutionResult** at 0x7a6ab450ae60

.execution_id	2
.component	▶**StatisticsGen** at 0x7a6a14270c10
.component.inputs	**['examples']** ▶**Channel** of type **'Examples'** (1 artifact) at 0x7a6a140b01f0
.component.outputs	**['statistics']** ▶**Channel** of type **'ExampleStatistics'** (1 artifact) at 0x7a6a14273e20

```
context.show(statistics_gen.outputs['statistics'])
```

Artifact at pipeline/StatisticsGen/statistics/2

'train' split:

Sort by Feature order — Reverse order — Feature search (regex enabled)

Features: int(1) string(1)

Numeric Features (1)

	count	missing	mean	std dev	zeros	min	median	max
rating	37.2k	0%	0.76	0.43	23.8%	0	1	1

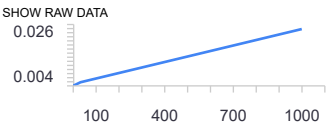
Chart to show: Standard — log expand

Categorical Features (1)

	count	missing	unique	top	freq top	avg str len
review	37.2k	0%	37.2k	Wow! I f…	2	1,289.98

Chart to show: Standard — log expand

SHOW RAW DATA



'eval' split:

Sort by Feature order — Reverse order — Feature search (regex enabled)

Features: int(1) string(1)

Numeric Features (1)

	count	missing	mean	std dev	zeros	min	median	max
rating	9,310	0%	0.76	0.42	23.56%	0	1	1

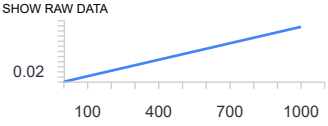
Chart to show: Standard — log expand

Categorical Features (1)

	count	missing	unique	top	freq top	avg str len
review	9,310	0%	9,303	While at …	2	1,288.25

Chart to show: Standard — log expand

SHOW RAW DATA



```
from tfx.components import SchemaGen
schema_gen =  SchemaGen(
    statistics=statistics_gen.outputs['statistics']
)
context.run(schema_gen)
context.show(schema_gen.outputs['schema'])
```

Artifact at pipeline/SchemaGen/schema/3

Feature name	Type	Presence	Valency	Domain
'rating'	INT	required		-
'review'	BYTES	required		-

```
from tfx.components import ExampleValidator
example_validator = ExampleValidator(
    statistics=statistics_gen.outputs['statistics'],
    schema=schema_gen.outputs['schema']
)

context.run(example_validator)
context.show(example_validator.outputs['anomalies'])
```

Artifact at pipeline/ExampleValidator/anomalies/4

'train' split:

No anomalies found.

'eval' split:

No anomalies found.

```
_transform_module_file = '_transform.py'
```

```
%%writefile {_transform_module_file}

import tensorflow as tf
import tensorflow_transform as tft

stopwords = ["i", "me", "my", "myself", "we", "our", "ours", "ourselves", "you", "your",
             "yours", "yourself", "yourselves", "he", "him", "his", "himself", "she",
             "her", "hers", "herself", "it", "its", "itself", "they", "them", "their",
             "theirs", "themselves", "what", "which", "who", "whom", "this", "that", "these",
             "those", "am", "is", "are", "was", "were", "be", "been", "being", "have", "has", "had",
             "having", "do", "does", "did", "doing", "a", "an", "the", "and", "but", "if", "or", "because",
             "as", "until", "while", "of", "at", "by", "for", "with", "about", "against", "between", "into",
             "through", "during", "before", "after", "above", "below", "to", "from", "up", "down", "in", "out", "on",
             "off", "over", "under", "again", "further", "then", "once", "here", "there", "when", "where", "why", "how",
             "all", "any", "both", "each", "few", "more", "most", "other", "some", "such", "no", "nor", "not", "only",
             "own", "same", "so", "than", "too", "very", "s", "t", "can", "will", "just", "don", "should", "now"]

_LABEL_KEY = 'rating'

# Renaming transformed features
def _transformed_name(key):
    return key + '_xf'

# Define the transformations
def preprocessing_fn(inputs):

    outputs = {}

    outputs[_transformed_name('review')] = tf.strings.lower(inputs['review'])
    outputs[_transformed_name('review')] = tf.strings.regex_replace(outputs[_transformed_name('review')], r"(?:<br />)", " ")
    outputs[_transformed_name('review')] = tf.strings.regex_replace(outputs[_transformed_name('review')], "n\'t", " not ")
    outputs[_transformed_name('review')] = tf.strings.regex_replace(outputs[_transformed_name('review')], r"(?:\'ll |\'re |\'d |\'ve)",
    outputs[_transformed_name('review')] = tf.strings.regex_replace(outputs[_transformed_name('review')], r"\W+", " ")
    outputs[_transformed_name('review')] = tf.strings.regex_replace(outputs[_transformed_name('review')], r"\d+", " ")
    outputs[_transformed_name('review')] = tf.strings.regex_replace(outputs[_transformed_name('review')], r"\b[a-zA-Z]\b", " ")
    outputs[_transformed_name('review')] = tf.strings.regex_replace(outputs[_transformed_name('review')], r'\b(' + r'|'.join(stopwords)

    outputs[_transformed_name(_LABEL_KEY)] = tf.cast(inputs[_LABEL_KEY], tf.int64)

    return outputs
```

```
Writing _transform.py
```

```
# Run the transform component
from tfx.components import Transform
transform  = Transform(
    examples=example_gen.outputs['examples'],
    schema= schema_gen.outputs['schema'],
    module_file=_transform_module_file
)

context.run(transform)
```

▼**ExecutionResult** at 0x7a6b5c4f7130

.execution_id	5
.component	▶**Transform** at 0x7a6a12c7f700
.component.inputs	**['examples']** ▶**Channel** of type **'Examples'** (1 artifact) at 0x7a6a140b01f0 **['schema']** ▶**Channel** of type **'Schema'** (1 artifact) at 0x7a6a12be6410
.component.outputs	**['transform_graph']** ▶**Channel** of type **'TransformGraph'** (1 artifact) at 0x7a6a12c7dcf0 **['transformed_examples']** ▶**Channel** of type **'Examples'** (1 artifact) at 0x7a6a12c7dd20 **['updated_analyzer_cache']** ▶**Channel** of type **'TransformCache'** (1 artifact) at 0x7a6a12c7c0a0 **['pre_transform_schema']** ▶**Channel** of type **'Schema'** (1 artifact) at 0x7a6a12c7c1c0 **['pre_transform_stats']** ▶**Channel** of type **'ExampleStatistics'** (1 artifact) at 0x7a6a12c7c460 **['post_transform_schema']** ▶**Channel** of type **'Schema'** (1 artifact) at 0x7a6a12c7cc40 **['post_transform_stats']** ▶**Channel** of type **'ExampleStatistics'** (1 artifact) at 0x7a6a12c7c400 **['post_transform_anomalies']** ▶**Channel** of type **'ExampleAnomalies'** (1 artifact) at 0x7a6a12c7d2a0

```
train_uri = os.path.join(transform.outputs['transformed_examples'].get()[0].uri, 'Split-train')

tfrecord_filenames = [os.path.join(train_uri, name) for name in os.listdir(train_uri)]

dataset = tf.data.TFRecordDataset(tfrecord_filenames, compression_type='GZIP')

for tfrecord in dataset.take(2):
    serialized_example = tfrecord.numpy()
    example = tf.train.Example()
    example.ParseFromString(serialized_example)
    pp.pprint(example)
```

```
features {
  feature {
    key: "rating_xf"
    value {
      int64_list {
        value: 1
      }
    }
  }
  feature {
    key: "review_xf"
    value {
      bytes_list {
        value: "bill  ted   excellent adventure  probably one    compelling films   experienced since   first watched     idea   inc
      }
    }
  }
}

features {
  feature {
    key: "rating_xf"
    value {
      int64_list {
        value: 1
      }
    }
  }
  feature {
    key: "review_xf"
    value {
      bytes_list {
        value: " bill  ted   excellent adventure   definitely        around fun movie  plot  far  anything serious  deep     good mo
      }
    }
  }
}
```

```
# Declare a trainer module file
_trainer_module_file = '_trainer.py'
```

```
%%writefile {_trainer_module_file}

import tensorflow as tf
import tensorflow_transform as tft
from tensorflow.keras import layers
import tensorflow_hub as hub
from tfx.components.trainer.fn_args_utils import FnArgs

_LABEL_KEY = 'rating'
_FEATURE = 'review'

def _transformed_name(key):
    return key + '_xf'

def _gzip_reader_fn(filenames):

    '''Loads compressed data'''
    return tf.data.TFRecordDataset(filenames, compression_type='GZIP')


def _input_fn(file_pattern,
              tf_transform_output,
              num_epochs,
              batch_size=64)->tf.data.Dataset:

    # Get post_transform feature spec
    transform_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=transform_feature_spec,
        reader=_gzip_reader_fn,
        num_epochs=num_epochs,
        label_key = _transformed_name(_LABEL_KEY))
    return dataset



embed = hub.KerasLayer("https://tfhub.dev/google/universal-sentence-encoder/4")
def model_builder():

    rate = 0.2

    inputs = tf.keras.Input(shape=(1,), name=_transformed_name('review'), dtype=tf.string)
    reshaped_narrative = tf.reshape(inputs, [-1])
    x = embed(reshaped_narrative)
    x = tf.keras.layers.Reshape((1,512), input_shape=(1,512))(x)
    x = layers.Dense(64, activation='elu', kernel_initializer='glorot_uniform')(x)

    attn_output = layers.MultiHeadAttention(num_heads=2, key_dim=64)(x, x, x)
    attn_output = layers.Dropout(rate)(attn_output)

    out1 = layers.LayerNormalization(epsilon=1e-7)(x + attn_output)
    ffn_output = layers.Dense(64, activation="elu", kernel_initializer="glorot_uniform")(out1)
    ffn_output = layers.Dense(64, kernel_initializer='glorot_uniform')(ffn_output)
    ffn_output = layers.Dropout(rate)(ffn_output)

    x = layers.LayerNormalization(epsilon=1e-7)(out1 + ffn_output)
    x = layers.GlobalAveragePooling1D()(x)
    x = layers.Dropout(rate)(x)
    x = layers.Dense(32, activation="elu", kernel_initializer="glorot_uniform")(x)
    x = layers.Dropout(rate)(x)
    outputs = layers.Dense(1, activation='sigmoid')(x)


    model = tf.keras.Model(inputs=inputs, outputs = outputs)

    model.compile(
        loss = 'binary_crossentropy',
        optimizer=tf.keras.optimizers.Adam(0.01),
        metrics=[tf.keras.metrics.BinaryAccuracy()]

    )

    # print(model)
    model.summary()
    return model
```

```
def _get_serve_tf_examples_fn(model, tf_transform_output):

    model.tft_layer = tf_transform_output.transform_features_layer()

    @tf.function
    def serve_tf_examples_fn(serialized_tf_examples):

        feature_spec = tf_transform_output.raw_feature_spec()

        feature_spec.pop("rating")

        parsed_features = tf.io.parse_example(serialized_tf_examples, feature_spec)

        transformed_features = model.tft_layer(parsed_features)

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

    return serve_tf_examples_fn

def run_fn(fn_args: FnArgs) -> None:

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

    es = tf.keras.callbacks.EarlyStopping(monitor='val_binary_accuracy', mode='max', verbose=1, patience=10)
    mc = tf.keras.callbacks.ModelCheckpoint(fn_args.serving_model_dir, monitor='val_binary_accuracy', mode='max', verbose=1, save_best_(


    # Load the transform output
    tf_transform_output = tft.TFTransformOutput(fn_args.transform_graph_path)

    # Create batches of data
    train_set = _input_fn(fn_args.train_files, tf_transform_output, 10)
    val_set = _input_fn(fn_args.eval_files, tf_transform_output, 10)


    # Build the model
    model = model_builder()


    # Train the model
    model.fit(x = train_set,
             validation_data = val_set,
             callbacks = [tensorboard_callback, es, mc],
               steps_per_epoch = 1000,
             validation_steps= 1000,
             epochs=1)
    signatures = {
        'serving_default':
        _get_serve_tf_examples_fn(model,
                                  tf_transform_output).get_concrete_function(
                                     tf.TensorSpec(
                                     shape=[None],
                                     dtype=tf.string,
                                     name='examples'))
    }
    model.save(fn_args.serving_model_dir, save_format='tf', signatures=signatures)
```

```
Writing _trainer.py
```

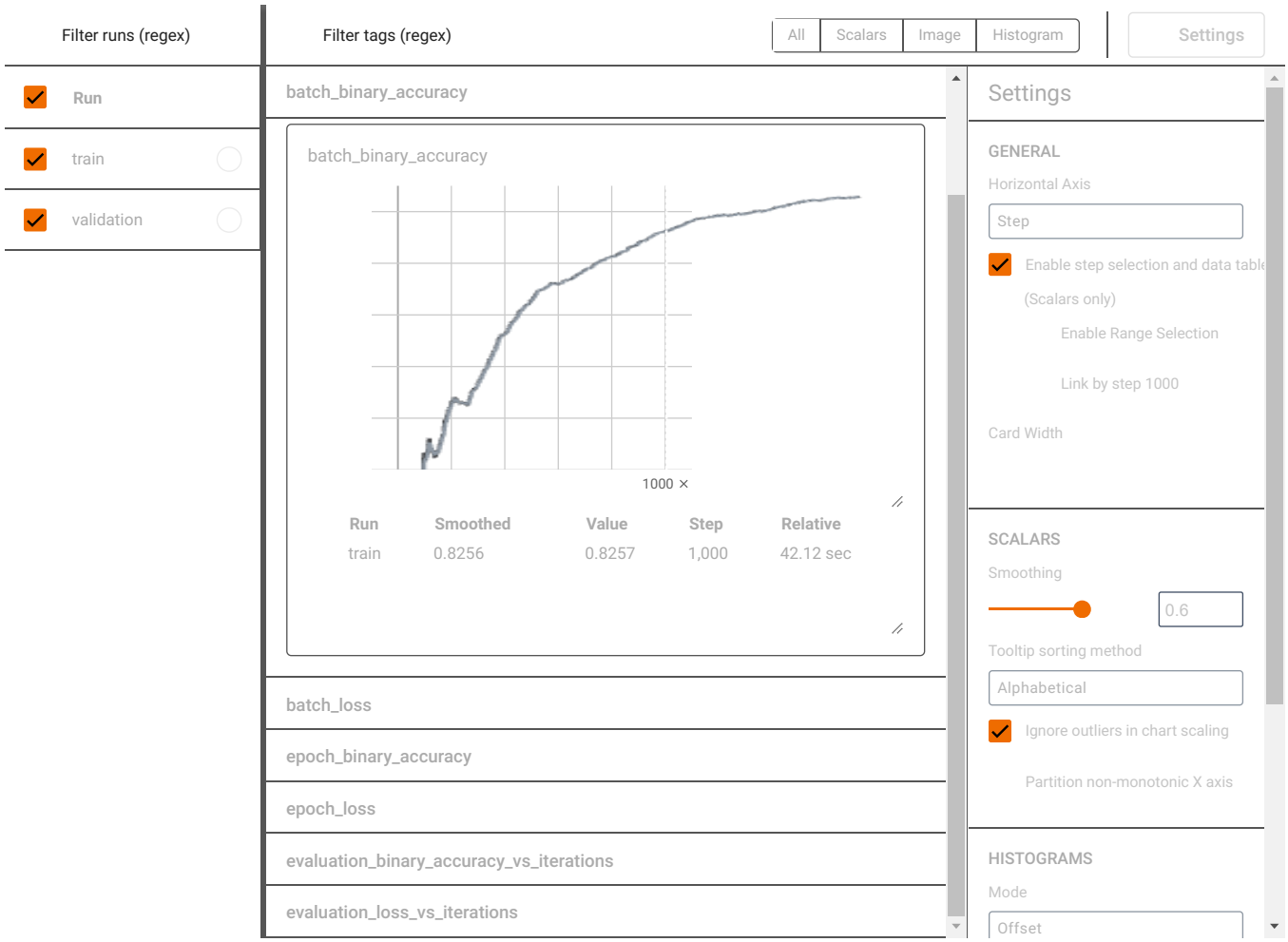
```
from tfx.components import Trainer
from tfx.proto import trainer_pb2

trainer  = Trainer(
    module_file=_trainer_module_file,
    examples = transform.outputs['transformed_examples'],
    transform_graph=transform.outputs['transform_graph'],
    schema=schema_gen.outputs['schema'],
    train_args=trainer_pb2.TrainArgs(splits=['train']),
    eval_args=trainer_pb2.EvalArgs(splits=['eval'])
)

context.run(trainer)
```

```
WARNING:absl:Examples artifact does not have payload_format custom property. Falling back to FORMAT_TF_EXAMPLE
WARNING:absl:Examples artifact does not have payload_format custom property. Falling back to FORMAT_TF_EXAMPLE
WARNING:absl:Examples artifact does not have payload_format custom property. Falling back to FORMAT_TF_EXAMPLE
WARNING:tensorflow:From /usr/local/lib/python3.10/dist-packages/tensorflow/python/data/experimental/ops/readers.py:1086: parse_examp
Instructions for updating:
Use `tf.data.Dataset.map(tf.io.parse_example(...))` instead.
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                Output Shape                 Param #   Connected to
==================================================================================================
 review_xf (InputLayer)      [(None, 1)]                  0         []

 tf.reshape (TFOpLambda)     (None,)                      0         ['review_xf[0][0]']

 keras_layer (KerasLayer)    (None, 512)                  2567978   ['tf.reshape[0][0]']
                                                          24

 reshape (Reshape)           (None, 1, 512)               0         ['keras_layer[0][0]']

 dense (Dense)               (None, 1, 64)                32832     ['reshape[0][0]']

 multi_head_attention (Mult  (None, 1, 64)                33216     ['dense[0][0]',
 iHeadAttention)                                                     'dense[0][0]',
                                                                     'dense[0][0]']

 dropout (Dropout)           (None, 1, 64)                0         ['multi_head_attention[0][0]']

 tf.__operators__.add (TFOp  (None, 1, 64)                0         ['dense[0][0]',
 Lambda)                                                             'dropout[0][0]']

 layer_normalization (Layer  (None, 1, 64)                128       ['tf.__operators__.add[0][0]']
 Normalization)

 dense_1 (Dense)             (None, 1, 64)                4160      ['layer_normalization[0][0]']

 dense_2 (Dense)             (None, 1, 64)                4160      ['dense_1[0][0]']

 dropout_1 (Dropout)         (None, 1, 64)                0         ['dense_2[0][0]']

 tf.__operators__.add_1 (TF  (None, 1, 64)                0         ['layer_normalization[0][0]',
 OpLambda)                                                           'dropout_1[0][0]']

 layer_normalization_1 (Lay  (None, 1, 64)                128       ['tf.__operators__.add_1[0][0]
 erNormalization)                                                    ']

 global_average_pooling1d (  (None, 64)                   0         ['layer_normalization_1[0][0]'
 GlobalAveragePooling1D)                                             ]

 dropout_2 (Dropout)         (None, 64)                   0         ['global_average_pooling1d[0][
                                                                     0]']

 dense_3 (Dense)             (None, 32)                   2080      ['dropout_2[0][0]']

 dropout_3 (Dropout)         (None, 32)                   0         ['dense_3[0][0]']

 dense_4 (Dense)             (None, 1)                    33        ['dropout_3[0][0]']

==================================================================================================
Total params: 256874561 (979.90 MB)
Trainable params: 76737 (299.75 KB)
Non-trainable params: 256797824 (979.61 MB)
__________________________________________________________________________________________________
 999/1000 [============================>.] - ETA: 0s - loss: 0.4021 - binary_accuracy: 0.8256
Epoch 1: val_binary_accuracy improved from -inf to 0.84198, saving model to pipeline/Trainer/model/6/Format-Serving
1000/1000 [==============================] - 96s 86ms/step - loss: 0.4020 - binary_accuracy: 0.8257 - val_loss: 0.3550 - val_binary_
```

▼**ExecutionResult** at 0x7a6ab49a7dc0

.execution_id	6
.component	▶**Trainer** at 0x7a6a13418790
.component.inputs	**['examples']** ▶**Channel** of type **'Examples'** (1 artifact) at 0x7a6a12c7dd20 **['transform_graph']** ▶**Channel** of type **'TransformGraph'** (1 artifact) at 0x7a6a12c7dcf0 **['schema']** ▶**Channel** of type **'Schema'** (1 artifact) at 0x7a6a12be6410
.component.outputs	**['model']** ▶**Channel** of type **'Model'** (1 artifact) at 0x7a6a1341b400 **['model_run']** ▶**Channel** of type **'ModelRun'** (1 artifact) at 0x7a6a134189d0

```
model_run_artifact_dir = trainer.outputs['model_run'].get()[0].uri

%load_ext tensorboard
%tensorboard --logdir {model_run_artifact_dir}
```




```
# Declare a tuner module file
_tuner_module_file = '_tuner.py'
```

```
%%writefile {_tuner_module_file}
import os
import tensorflow as tf
import tensorflow_transform as tft
import keras_tuner as kt
from tensorflow.keras import layers
from tfx.components.trainer.fn_args_utils import FnArgs
from keras_tuner.engine import base_tuner
from typing import NamedTuple, Dict, Text, Any

_LABEL_KEY = 'rating'
_FEATURE_KEY = 'review'

def transformed_name(key):
    """Renaming transformed features"""
    return key + "_xf"

def gzip_reader_fn(filenames):
    """Loads compressed data"""
    return tf.data.TFRecordDataset(filenames, compression_type='GZIP')

def input_fn(file_pattern, tf_transform_output, num_epochs, batch_size=64) -> tf.data.Dataset:
    """Get post_tranform feature & create batches of data"""

    # Get post_transform feature spec
    transform_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     = transform_feature_spec,
        reader       = gzip_reader_fn,
        num_epochs   = num_epochs,
        label_key    = transformed_name(_LABEL_KEY)
    )

    return dataset

# Vocabulary size and number of words in a sequence.
VOCAB_SIZE      = 10000
SEQUENCE_LENGTH = 100

vectorize_layer = layers.TextVectorization(
    standardize            = 'lower_and_strip_punctuation',
    max_tokens             = VOCAB_SIZE,
    output_mode            = 'int',
    output_sequence_length = SEQUENCE_LENGTH
)

def model_builder(hp):
    """Build keras tuner model"""
    embedding_dim = hp.Int('embedding_dim', min_value=16, max_value=128, step=16)
    lstm_units    = hp.Int('lstm_units', min_value=16, max_value=128, step=16)
    num_layers    = hp.Choice('num_layers', values=[1, 2, 3])
    dense_units   = hp.Int('dense_units', min_value=16, max_value=128, step=16)
    dropout_rate = hp.Float('dropout_rate', min_value=0.1, max_value=0.5, step=0.1)
    learning_rate = hp.Choice('learning_rate', values=[1e-2, 1e-3, 1e-4])

    inputs = tf.keras.Input(shape=(1,), name=transformed_name(_FEATURE_KEY), dtype=tf.string)

    reshaped_narrative = tf.reshape(inputs, [-1])
    x = vectorize_layer(reshaped_narrative)
    x = layers.Embedding(VOCAB_SIZE, embedding_dim, name='embedding')(x)
    x = layers.Bidirectional(layers.LSTM(lstm_units))(x)
    for _ in range(num_layers):
        x = layers.Dense(dense_units, activation='relu')(x)
    x = layers.Dropout(dropout_rate)(x)
    outputs = layers.Dense(1, activation='sigmoid')(x)

    model = tf.keras.Model(inputs = inputs, outputs = outputs)
    model.compile(
        loss      = tf.keras.losses.BinaryCrossentropy(from_logits=True),
        optimizer = tf.keras.optimizers.Adam(learning_rate),
        metrics   = [tf.keras.metrics.BinaryAccuracy()]
    )

    model.summary()
    return model
```

```
TunerFnResult = NamedTuple('TunerFnResult', [
    ('tuner', base_tuner.BaseTuner),
    ('fit_kwargs', Dict[Text, Any]),
])

early_stop_callback = tf.keras.callbacks.EarlyStopping(
    monitor  = 'val_binary_accuracy',
    mode     = 'max',
    verbose  = 1,
    patience = 10
)

def tuner_fn(fn_args: FnArgs) -> None:
    # Load the transform output
    tf_transform_output = tft.TFTransformOutput(fn_args.transform_graph_path)

    # Create batches of data
    train_set = input_fn(fn_args.train_files[0], tf_transform_output, 10)
    val_set   = input_fn(fn_args.eval_files[0],  tf_transform_output, 10)

    vectorize_layer.adapt(
        [j[0].numpy()[0] for j in [
            i[0][transformed_name(_FEATURE_KEY)]
                for i in list(train_set)
        ]]
    )

    # Build the model tuner
    model_tuner = kt.RandomSearch(
        hypermodel   = lambda hp: model_builder(hp),
        objective    = kt.Objective('val_binary_accuracy', direction='max'),
        max_trials   = 3,
        executions_per_trial = 1,
        directory    = fn_args.working_dir,
        project_name = 'imdb_kt'
    )

    return TunerFnResult(
        tuner      = model_tuner,
        fit_kwargs = {
            'callbacks'        : [early_stop_callback],
            'x'                : train_set,
            'validation_data'  : val_set,
            'steps_per_epoch'  : fn_args.train_steps,
            'validation_steps' : fn_args.eval_steps
        }
    )
```

```
Writing _tuner.py
```

```
from tfx.components import Tuner
tuner = Tuner(
    module_file     = os.path.abspath(_tuner_module_file),
    examples        = transform.outputs['transformed_examples'],
    transform_graph = transform.outputs['transform_graph'],
    schema          = schema_gen.outputs['schema'],
    train_args      = trainer_pb2.TrainArgs(splits=['train']),
    eval_args       = trainer_pb2.EvalArgs(splits=['eval'])
)

context.run(tuner)
```

```
Trial 3 Complete [00h 02m 25s]
val_binary_accuracy: 0.8371643424034119

Best val_binary_accuracy So Far: 0.8380236029624939
Total elapsed time: 00h 07m 55s
Results summary
Results in pipeline/.temp/7/imdb_kt
Showing 10 best trials
Objective(name="val_binary_accuracy", direction="max")

Trial 0 summary
Hyperparameters:
embedding_dim: 128
lstm_units: 48
num_layers: 3
dense_units: 64
dropout_rate: 0.5
learning_rate: 0.01
Score: 0.8380236029624939

Trial 2 summary
Hyperparameters:
embedding_dim: 128
lstm_units: 64
num_layers: 3
dense_units: 112
dropout_rate: 0.2
learning_rate: 0.001
Score: 0.8371643424034119

Trial 1 summary
Hyperparameters:
embedding_dim: 64
lstm_units: 32
num_layers: 2
dense_units: 16
dropout_rate: 0.2
learning_rate: 0.001
Score: 0.8274973034858704
```

▼**ExecutionResult** at 0x7a6b232c9390

.execution_id	7
.component	▶**Tuner** at 0x7a69d03963e0
.component.inputs	**['examples']** ▶**Channel** of type **'Examples'** (1 artifact) at 0x7a6a12c7dd20
	['schema'] ▶**Channel** of type **'Schema'** (1 artifact) at 0x7a6a12be6410
	['transform_graph'] ▶**Channel** of type **'TransformGraph'** (1 artifact) at 0x7a6a12c7dcf0
.component.outputs	**['best_hyperparameters']** ▶**Channel** of type **'HyperParameters'** (1 artifact) at 0x7a6a10695150
	['tuner_results'] ▶**Channel** of type **'TunerResults'** (1 artifact) at 0x7a6a10694580

```
from tfx.dsl.components.common.resolver import Resolver
from tfx.dsl.input_resolution.strategies.latest_blessed_model_strategy import LatestBlessedModelStrategy
from tfx.types import Channel
from tfx.types.standard_artifacts import Model, ModelBlessing

model_resolver = Resolver(
    strategy_class= LatestBlessedModelStrategy,
    model = Channel(type=Model),
    model_blessing = Channel(type=ModelBlessing)
).with_id('Latest_blessed_model_resolver')

context.run(model_resolver)
```

▼**ExecutionResult** at 0x7a6a10697c70

.execution_id	8
.component	<tfx.dsl.components.common.resolver.Resolver object at 0x7a69d06c9ba0>
.component.inputs	**['model']** ResolvedChannel(artifact_type=Model, LatestBlessedModelStrategy(Dict(model=Input(), model_blessing=Input()))["model"])
	['model_blessing'] ResolvedChannel(artifact_type=ModelBlessing, LatestBlessedModelStrategy(Dict(model=Input(), model_blessing=Input()))["model_blessing"])
.component.outputs	**['model']** ▶**Channel** of type **'Model'** (0 artifacts) at 0x7a69d06c8640

```
import tensorflow_model_analysis as tfma

eval_config = tfma.EvalConfig(
    model_specs=[tfma.ModelSpec(label_key='rating')],
    slicing_specs=[tfma.SlicingSpec()],
    metrics_specs=[
        tfma.MetricsSpec(metrics=[
```

```
            tfma.MetricConfig(class_name='ExampleCount'),
            tfma.MetricConfig(class_name='AUC'),
            tfma.MetricConfig(class_name='FalsePositives'),
            tfma.MetricConfig(class_name='TruePositives'),
            tfma.MetricConfig(class_name='FalseNegatives'),
            tfma.MetricConfig(class_name='TrueNegatives'),
            tfma.MetricConfig(class_name='BinaryAccuracy',
                threshold=tfma.MetricThreshold(
                    value_threshold=tfma.GenericValueThreshold(
                        lower_bound={'value':0.5}),
                    change_threshold=tfma.GenericChangeThreshold(
                        direction=tfma.MetricDirection.HIGHER_IS_BETTER,
                        absolute={'value':0.0001})
                    )
            )
        ])
    ]

)

from tfx.components import 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)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.10/dist-packages/tensorflow_model_analysis/writers/metrics_plots_and_validations_writ
Instructions for updating:
Use eager execution and:
`tf.data.TFRecordDataset(path)`
```

▼**ExecutionResult** at 0x7a6a10172050

.execution_id	9
.component	▶**Evaluator** at 0x7a6a135ba110
.component.inputs	**['examples']** ▶**Channel** of type **'Examples'** (1 artifact) at 0x7a6a140b01f0 **['model']** ▶**Channel** of type **'Model'** (1 artifact) at 0x7a6a1341b400 **['baseline_model']** ▶**Channel** of type **'Model'** (0 artifacts) at 0x7a69d06c8640
.component.outputs	**['evaluation']** ▶**Channel** of type **'ModelEvaluation'** (1 artifact) at 0x7a69d0395540 **['blessing']** ▶**Channel** of type **'ModelBlessing'** (1 artifact) at 0x7a69d0395270

```
# Visualize the evaluation results
eval_result = evaluator.outputs['evaluation'].get()[0].uri
tfma_result = tfma.load_eval_result(eval_result)
tfma.view.render_slicing_metrics(tfma_result)
```

Examples (Weighted) Threshold
0

Visualization
Slices Overview

Show
example_count

Sort by
Slice

feature	auc	binary_accuracy	example_count	false_negatives	false_positives	loss
Overall	0.87975	0.84211	9310	264.00000	1206.00000	0.3548

```
tfma.addons.fairness.view.widget_view.render_fairness_indicator(tfma_result)
```

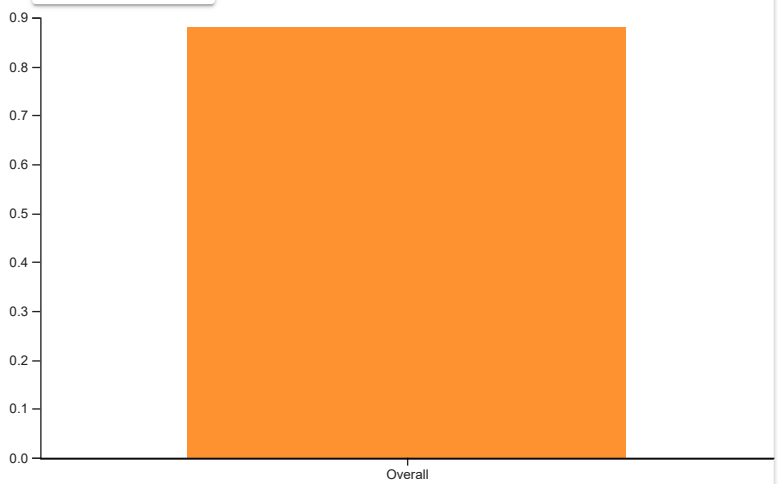
Select metrics to display:

- [ ] Select all
- [ ] **auc**
- [ ] example_count
- [ ] binary_accuracy
- [ ] false_negatives
- [ ] **false_positives**
- [ ] loss
- [ ] true_negatives
- [ ] true_positives

Baseline
Overall

auc

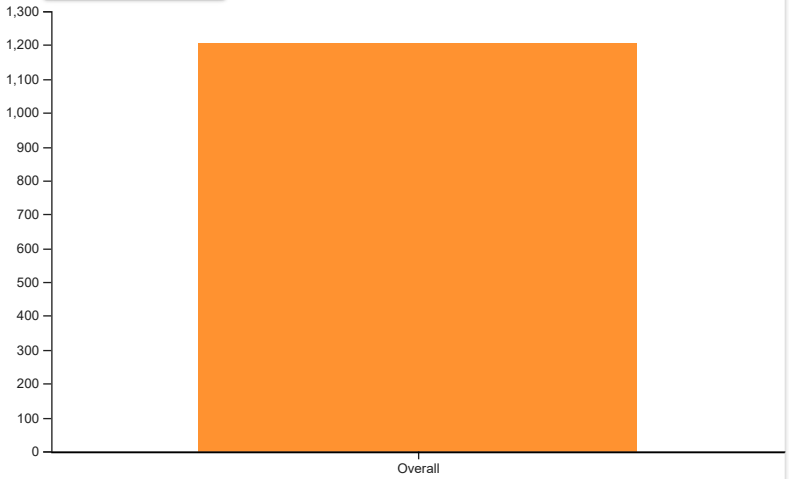
SELECT SLICES



feature	auc	auc against Overall	example count
Overall	**0.88**	**0%**	**9310**

false_positives

SELECT SLICES



feature	false_positives	false_positives against Overall	example count
Overall	**1206**	**0%**	**9310**

```
# Print validation results
eval_result = evaluator.outputs['evaluation'].get()[0].uri
print(tfma.load_validation_result(eval_result))
```

```
validation_ok: true
validation_details {
  slicing_details {
    slicing_spec {
    }
    num_matching_slices: 1
  }
}
```

```
from tfx.components import Pusher
from tfx.proto import pusher_pb2

pusher = Pusher(
model=trainer.outputs['model'],
model_blessing=evaluator.outputs['blessing'],
push_destination=pusher_pb2.PushDestination(
    filesystem=pusher_pb2.PushDestination.Filesystem(
        base_directory='serving_model_dir'))

)

context.run(pusher)
```

▼**ExecutionResult** at 0x7a69d04971f0

.execution_id	10
.component	▶**Pusher** at 0x7a6a135bbfd0
.component.inputs	**['model']** ▶**Channel** of type **'Model'** (1 artifact) at 0x7a6a1341b400 **['model_blessing']** ▶**Channel** of type **'ModelBlessing'** (1 artifact) at 0x7a69d0395270
.component.outputs	**['pushed_model']** ▶**Channel** of type **'PushedModel'** (1 artifact) at 0x7a69d04970a0

```
import os
from tfx.components.base import base_component, base_executor, executor_spec
from tfx.types import standard_artifacts, component_spec, channel_utils
import tensorflow_data_validation as tfdv
from scipy.stats import ks_2samp

class DriftDetectionSpec(component_spec.ComponentSpec):
    PARAMETERS = {}
    INPUTS = {
        'anomalies': component_spec.ChannelParameter(type=standard_artifacts.ExampleAnomalies),
    }
    OUTPUTS = {
        'drift_detected': component_spec.ChannelParameter(type=standard_artifacts.String),
    }

# Assuming anomalies_dir is the directory containing the anomalies
anomalies_dir = '/content/pipeline/ExampleValidator/anomalies/4'

# Define paths to check for SchemaDiff.pb
anomalies_files = [
    os.path.join(anomalies_dir, 'Split-train', 'SchemaDiff.pb'),
    os.path.join(anomalies_dir, 'Split-eval', 'SchemaDiff.pb')
]

# Example usage in the DriftDetectionExecutor class
class DriftDetectionExecutor(base_executor.BaseExecutor):
    def Do(self, input_dict, output_dict, exec_properties):
        anomalies_channel = input_dict.get('anomalies', None)

        # Print input_dict for debugging
        print("input_dict:", input_dict)

        # Check if anomalies_channel is valid and contains artifacts
        if anomalies_channel and len(anomalies_channel) > 0:
            anomalies_uri = anomalies_channel[0].uri
            print(f"Anomalies found at: {anomalies_uri}")
        else:
            print("No anomalies found. Proceeding with drift detection on available data.")
            anomalies_uri = None

        # List contents of the anomalies directory
        print("Contents of anomalies directory:")
        for root, dirs, files in os.walk(anomalies_dir):
            for name in files:
                print(os.path.join(root, name))

        # Check if specific files exist and print their paths
        for anomalies_file in anomalies_files:
            if os.path.exists(anomalies_file):
                print(f"File exists: {anomalies_file}")
            else:
                print(f"File does not exist: {anomalies_file}")

        # Iterate over each anomalies file path and perform drift detection
        for anomalies_file in anomalies_files:
            if os.path.exists(anomalies_file):
                reference_stats_path = anomalies_file
```

```
                break
        else:
            # If no valid anomalies file found, handle accordingly
            print("No valid SchemaDiff.pb file found in the specified paths.")
            return

        # Load new data statistics if anomalies are found
        if anomalies_uri:
            new_data_stats_path = os.path.join(anomalies_uri, 'Split-eval', 'stats_tfrecord')
            if os.path.exists(new_data_stats_path):
                new_stats = tfdv.load_statistics(new_data_stats_path)
            else:
                print(f"New data statistics path does not exist: {new_data_stats_path}")
                new_stats = None
        else:
            new_stats = None

        # Log reference and new stats for debugging
        print(f"Reference stats path: {reference_stats_path}")
        if new_stats:
            print(f"New stats loaded from: {new_data_stats_path}")
        else:
            print("No new stats loaded.")

        drift_detected = self.detect_statistical_drift(reference_stats_path, new_stats)

        # Save the result
        drift_detected_output_dir = output_dict['drift_detected'][0].uri
        os.makedirs(drift_detected_output_dir, exist_ok=True)
        drift_detected_output_file = os.path.join(drift_detected_output_dir, 'drift_detected.txt')

        with open(drift_detected_output_file, 'w') as f:
            f.write(drift_detected)

        print("Drift detected" if drift_detected == 'true' else "No drift detected")

    def detect_statistical_drift(self, reference_stats_path, new_stats):
        if not new_stats:
            print("No new statistics to compare. Skipping drift detection.")
            return 'false'

        reference_stats = tfdv.load_statistics(reference_stats_path)
        drift_detected = False

        for feature in new_stats.datasets[0].features:
            if feature.type == tfdv.FeatureType.FLOAT:
                ref_values = [x.num for x in reference_stats.datasets[0].features[feature.name].num_stats.histograms[0].buckets]
                new_values = [x.num for x in new_stats.datasets[0].features[feature.name].num_stats.histograms[0].buckets]

                statistic, p_value = ks_2samp(ref_values, new_values)

                # Log feature values and p-value for debugging
                print(f"Feature: {feature.name}, Ref values: {ref_values}, New values: {new_values}, p-value: {p_value}")

                # Set a threshold for drift detection
                if p_value < 0.05:  # Example threshold, adjust as needed
                    print(f"Drift detected in feature: {feature.name}")
                    drift_detected = True

        return 'true' if drift_detected else 'false'

class DriftDetection(base_component.BaseComponent):
    SPEC_CLASS = DriftDetectionSpec
    EXECUTOR_SPEC = executor_spec.ExecutorClassSpec(DriftDetectionExecutor)

    def __init__(self, anomalies):
        spec = DriftDetectionSpec(
            anomalies=anomalies,
            drift_detected=channel_utils.as_channel([standard_artifacts.String()])
        )
        super(DriftDetection, self).__init__(spec=spec)

# Assuming example_validator is already defined and outputs 'anomalies'
example_validator_output = example_validator.outputs['anomalies']
drift_detection = DriftDetection(anomalies=example_validator_output)
context.run(drift_detection)
```

```
input_dict: {'anomalies': [Artifact(artifact: id: 4
type_id: 20
uri: "pipeline/ExampleValidator/anomalies/4"
properties {
  key: "split_names"
  value {
    string_value: "[\"train\", \"eval\"]"
  }
}
custom_properties {
  key: "blessed"
  value {
    struct_value {
      fields {
        key: "eval"
        value {
          number_value: 1.0
        }
      }
      fields {
        key: "train"
        value {
          number_value: 1.0
        }
      }
    }
  }
}
custom_properties {
  key: "name"
  value {
    string_value: "anomalies:2024-07-19T15:50:40.814623"
  }
}
custom_properties {
  key: "producer_component"
  value {
    string_value: "ExampleValidator"
  }
}
custom_properties {
  key: "tfx_version"
  value {
    string_value: "1.15.1"
  }
}
state: LIVE
name: "anomalies:2024-07-19T15:50:40.814623"
, artifact_type: id: 20
name: "ExampleAnomalies"
properties {
  key: "span"
  value: INT
}
properties {
  key: "split_names"
  value: STRING
}
)]}
Anomalies found at: pipeline/ExampleValidator/anomalies/4
Contents of anomalies directory:
/content/pipeline/ExampleValidator/anomalies/4/Split-eval/SchemaDiff.pb
/content/pipeline/ExampleValidator/anomalies/4/Split-train/SchemaDiff.pb
File exists: /content/pipeline/ExampleValidator/anomalies/4/Split-train/SchemaDiff.pb
File exists: /content/pipeline/ExampleValidator/anomalies/4/Split-eval/SchemaDiff.pb
New data statistics path does not exist: pipeline/ExampleValidator/anomalies/4/Split-eval/stats_tfrecord
Reference stats path: /content/pipeline/ExampleValidator/anomalies/4/Split-train/SchemaDiff.pb
No new stats loaded.
No new statistics to compare. Skipping drift detection.
No drift detected
```

▼**ExecutionResult** at 0x7a6955065510

.execution_id	11
.component	▶**DriftDetection** at 0x7a691ab9c370
.component.inputs	**['anomalies']** ▶**Channel** of type **'ExampleAnomalies'** (1 artifact) at 0x7a6a12be7460
.component.outputs	**['drift_detected']** ▶**Channel** of type **'String'** (1 artifact) at 0x7a6955065750

```
import os
import glob
import pandas as pd
from tfx.components import CsvExampleGen, StatisticsGen, SchemaGen, ExampleValidator
from tfx.orchestration.experimental.interactive.interactive_context import InteractiveContext
from tfx.proto import example_gen_pb2

# Define datasets and paths dynamically or through configuration
```

```
datasets = [
    '/content/drive/MyDrive/Horror.csv',
    '/content/drive/MyDrive/Animation.csv'
]
current_dataset_index = 0

# Assuming anomalies_dir is the directory containing the anomalies
anomalies_dir = '/content/pipeline/ExampleValidator/anomalies'

context = InteractiveContext()

# Function to ensure consistent headers in CSV files
def ensure_consistent_headers(csv_files):
    if not csv_files:
        return  # Handle the case where no CSV files are found

    # Use pandas to read and standardize headers
    reference_df = pd.read_csv(csv_files[0])
    reference_header = reference_df.columns

    for file in csv_files[1:]:
        df = pd.read_csv(file)
        if not df.columns.equals(reference_header):
            print(f"Adjusting headers for {file}...")
            df.columns = reference_header
            df.to_csv(file, index=False)

# Define initial pipeline components for dataset_1
dataset_path = datasets[current_dataset_index]
csv_files = glob.glob(os.path.join(dataset_path, '*.csv'))
ensure_consistent_headers(csv_files)  # Ensure headers are consistent before ExampleGen

# Corrected example_gen initialization
example_gen = CsvExampleGen(
    input_base=os.path.dirname(dataset_path),  # Use the directory containing the CSV
    input_config=example_gen_pb2.Input(splits=[
        example_gen_pb2.Input.Split(name='train', pattern=os.path.basename(dataset_path))  # Use the filename
    ]),
    output_config=example_gen_pb2.Output(
        split_config=example_gen_pb2.SplitConfig(splits=[
            example_gen_pb2.SplitConfig.Split(name='train', hash_buckets=2),
        ])
    )
)

statistics_gen = StatisticsGen(examples=example_gen.outputs['examples'])
schema_gen = SchemaGen(statistics=statistics_gen.outputs['statistics'])
example_validator = ExampleValidator(
    statistics=statistics_gen.outputs['statistics'],
    schema=schema_gen.outputs['schema']
)

# DriftDetection component placeholder
# Define and implement DriftDetection component as discussed earlier

# Pipeline execution with iterative logic for drift detection and retraining
while current_dataset_index < len(datasets):
    # Run the components for the current dataset
    context.run(example_gen)
    context.run(statistics_gen)
    context.run(schema_gen)
    context.run(example_validator)  # Make sure ExampleValidator is run

    # Print ExampleValidator output artifact URI
    print("ExampleValidator anomalies URI:", example_validator.outputs['anomalies'].get()[0].uri)

    # Perform drift detection
    context.run(drift_detection)
    drift_detected = context.show(drift_detection.outputs['drift_detected'])

    if drift_detected == 'true':
        print(f"Drift detected in {dataset_path}. Retraining on next dataset.")
    else:
        print("No drift detected. Proceeding to the next dataset.")

    # Move to the next dataset
    current_dataset_index += 1
    if current_dataset_index < len(datasets):
        dataset_path = datasets[current_dataset_index]
        csv_files = glob.glob(os.path.join(dataset_path, '*.csv'))
        ensure_consistent_headers(csv_files)  # Ensure headers are consistent before updating ExampleGen
        example_gen.input_base = dataset_path  # Update input_base for ExampleGen
```

```
# Finalize the pipeline execution
# context.close()
```

```
WARNING:absl:InteractiveContext pipeline_root argument not provided: using temporary directory /tmp/tfx-interactive-2024-07-19T16_12
WARNING:absl:InteractiveContext metadata_connection_config not provided: using SQLite ML Metadata database at /tmp/tfx-interactive-2
ExampleValidator anomalies URI: /tmp/tfx-interactive-2024-07-19T16_12_49.565243-5t1q369w/ExampleValidator/anomalies/4
input_dict: {'anomalies': [Artifact(artifact: id: 4
type_id: 20
uri: "pipeline/ExampleValidator/anomalies/4"
properties {
  key: "split_names"
  value {
    string_value: "[\"train\", \"eval\"]"
  }
}
custom_properties {
  key: "blessed"
  value {
    struct_value {
      fields {
        key: "eval"
        value {
          number_value: 1.0
        }
      }
      fields {
        key: "train"
        value {
          number_value: 1.0
        }
      }
    }
  }
}
custom_properties {
  key: "name"
  value {
    string_value: "anomalies:2024-07-19T15:50:40.814623"
  }
}
custom_properties {
  key: "producer_component"
  value {
    string_value: "ExampleValidator"
  }
}
custom_properties {
  key: "tfx_version"
  value {
    string_value: "1.15.1"
  }
}
state: LIVE
name: "anomalies:2024-07-19T15:50:40.814623"
, artifact_type: id: 20
name: "ExampleAnomalies"
properties {
  key: "span"
  value: INT
}
properties {
  key: "split_names"
  value: STRING
}
)]}
Anomalies found at: pipeline/ExampleValidator/anomalies/4
Contents of anomalies directory:
/content/pipeline/ExampleValidator/anomalies/4/Split-eval/SchemaDiff.pb
/content/pipeline/ExampleValidator/anomalies/4/Split-train/SchemaDiff.pb
File exists: /content/pipeline/ExampleValidator/anomalies/4/Split-train/SchemaDiff.pb
File exists: /content/pipeline/ExampleValidator/anomalies/4/Split-eval/SchemaDiff.pb
New data statistics path does not exist: pipeline/ExampleValidator/anomalies/4/Split-eval/stats_tfrecord
Reference stats path: /content/pipeline/ExampleValidator/anomalies/4/Split-train/SchemaDiff.pb
No new stats loaded.
No new statistics to compare. Skipping drift detection.
No drift detected
```

Artifact at /tmp/tfx-interactive-2024-07-19T16_12_49.565243-5t1q369w/DriftDetection/drift_detected/5/value

```
No drift detected. Proceeding to the next dataset.
ExampleValidator anomalies URI: pipeline/ExampleValidator/anomalies/4
```

Artifact at /tmp/tfx-interactive-2024-07-19T16_12_49.565243-5t1q369w/DriftDetection/drift_detected/5/value

```
No drift detected. Proceeding to the next dataset.
```

```
!pip install tensorflow-model-analysis

import tensorflow_model_analysis as tfma
import os
import json
import tensorflow as tf

# Define the log directory for TensorBoard
log_dir = 'logs/metrics'
file_writer = tf.summary.create_file_writer(log_dir)

# Load and render evaluation metrics
eval_result = evaluator.outputs['evaluation'].get()[0].uri
tfma_result = tfma.load_eval_result(eval_result)
tfma.view.render_slicing_metrics(tfma_result)
tfma.addons.fairness.view.widget_view.render_fairness_indicator(tfma_result)

# Print validation results
print("Validation Results:")
validation_result = tfma.load_validation_result(eval_result)
print(validation_result)

# Calculate and print precision, recall, and F1-score
try:
    slicing_metrics = tfma_result.slicing_metrics[0][1][''][''']
    true_positives = slicing_metrics['true_positives']['doubleValue']
    false_positives = slicing_metrics['false_positives']['doubleValue']
    false_negatives = slicing_metrics['false_negatives']['doubleValue']

    precision = true_positives / (true_positives + false_positives) if (true_positives + false_positives) > 0 else 0.0
    recall = true_positives / (true_positives + false_negatives) if (true_positives + false_negatives) > 0 else 0.0
    f1_score = 2 * (precision * recall) / (precision + recall) if (precision + recall) > 0 else 0.0

    print(f"Precision: {precision}")
    print(f"Recall: {recall}")
    print(f"F1-Score: {f1_score}")

    # Log metrics to TensorBoard
    with file_writer.as_default():
        tf.summary.scalar('Precision', precision, step=0)
        tf.summary.scalar('Recall', recall, step=0)
        tf.summary.scalar('F1-Score', f1_score, step=0)

except (KeyError, IndexError, TypeError) as e:
    print(f"Error accessing metrics: {e}")

# Start TensorBoard within Colab
%reload_ext tensorboard
%tensorboard --logdir logs/metrics
```