Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances

Option 1: Train Pipeline (SageMaker Pipelines)

⚠ Make sure you have performed the steps described in the **Prerequisites** section before beginning this lab.

- Overview
- Prerequisites
- Build the pipeline components
  1. Import statements and declare parameters and constants
  2. Collect and prepare data
  3. Define Processing Step
  4. Define HyperParameter Tuning Step
  5. Define the evaluation script and model evaluation step
  6. Define a register model step
  7. Define a condition step to check AUC score
- Build and Trigger the pipeline run
- Conclusion

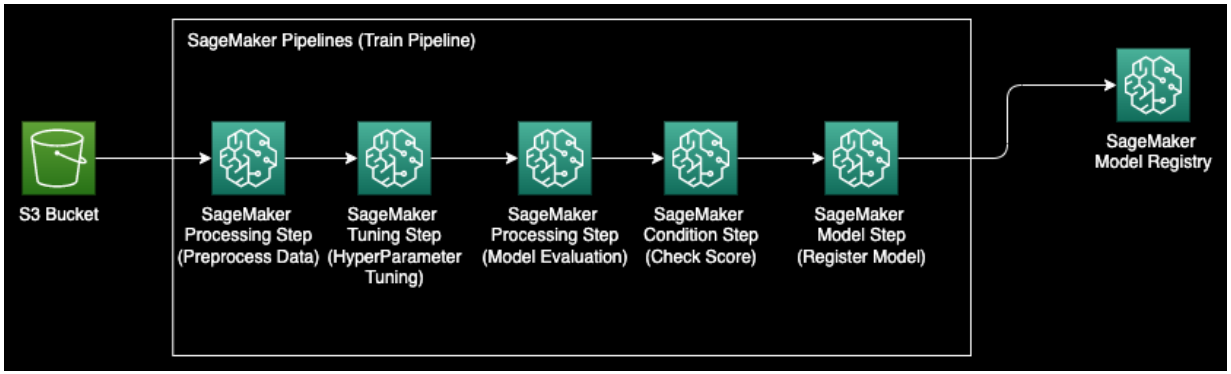
Overview

Amazon SageMaker Pipelines, is a capability of Amazon SageMaker makes it easy for data scientists and engineers to build, automate, and scale end to end machine learning pipelines. SageMaker Pipelines is a native workflow orchestration tool for building ML pipelines that take advantage of direct Amazon SageMaker integration.

This lab focuses on building a training pipeline using SageMaker Pipelines to train a model with Hyperparameter Tuning using a customer churn classification example. This pipeline includes steps to preprocess data, train model with hyperparameter tuning, model evaluation, conditional step to evaluate the score and registering model artifacts to the model registry if the score is above specified threshold.

The following diagram illustrates the high-level architecture of the ML workflow with the different steps to train the model.

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language



Train Pipeline consists of the following steps:

1. Preprocess data to build features required and split data into train, validation, and test datasets.
2. Apply hyperparameter tuning based on the ranges provided with the SageMaker XGBoost framework to give the best model, which is determined based on AUC score.
3. Evaluate the trained model using the test dataset and check if the AUC score is above a predefined threshold.
4. Check if the AUC score is greater than the threshold, if true register the model into SageMaker model registry.

Prerequisites

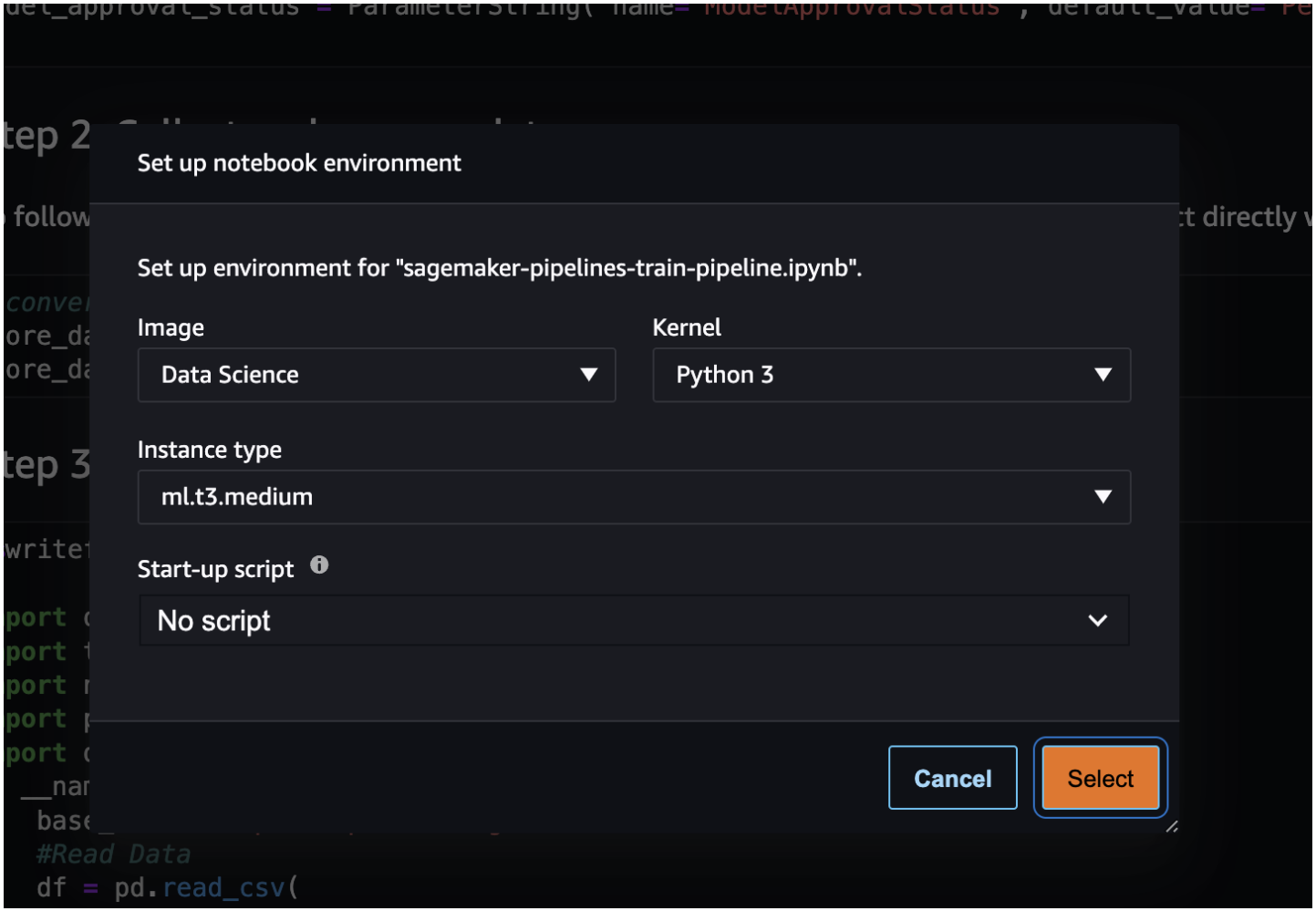
- Follow the instructions to launch Amazon SageMaker Studio
- Please ensure that you have git cloned the repository in your SageMaker Studio environment.

Build the pipeline components

Click on the “amazon-sagemaker-immersion-day” folder and then double click on the **sagemaker-pipelines-train-pipeline.ipynb** notebook within the project home directory.

If you are prompted to choose a Kernel, choose the “Python 3 (Data Science)” kernel and click “Select”.

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language



Import statements and declare parameters and constants

Run the first cell within the notebook to set up SageMaker and S3 client objects, create PipelineSession, and set up the S3 bucket location using the default bucket that comes with a SageMaker session:

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

Step 1: Import statements and declare parameters and constants

[2]:
```
import boto3
import pandas as pd
import sagemaker
from sagemaker.workflow.pipeline_context import PipelineSession

s3_client = boto3.resource('s3')
pipeline_name = f"sagemaker-immersion-train-pipeline"
sagemaker_session = sagemaker.session.Session()
region = sagemaker_session.boto_region_name
role = sagemaker.get_execution_role()
pipeline_session = PipelineSession()
default_bucket = sagemaker_session.default_bucket()
model_package_group_name = f"ChurnModelPackageGroup"
```

Pipelines supports parameterization, which allows you to specify input parameters at runtime without changing your pipeline code. You can use the modules available under the sagemaker.workflow.parameters module, such as ParameterInteger, ParameterFloat, and ParameterString, to specify pipeline parameters of various data types. Run the next cell to set up multiple input parameters:

[3]:
```
from sagemaker.workflow.parameters import (
 ParameterInteger,
 ParameterString,
 ParameterFloat)

auc_score_threshold = 0.75
base_job_prefix = "churn-example"
processing_instance_count = ParameterInteger(name="ProcessingInstanceCount", default_value=1)
processing_instance_type = ParameterString( name="ProcessingInstanceType", default_value="ml.m5.xlarge")
training_instance_type = ParameterString( name="TrainingInstanceType", default_value="ml.m5.xlarge")
input_data = "storedata_total.csv"
model_approval_status = ParameterString( name="ModelApprovalStatus", default_value="PendingManualApproval")
```

Collect and prepare data

To follow along with this lab, you need to download and save the *sample dataset* into the project directly within the SageMaker Studio environment. Then run the next cell to convert the dataset into csv format

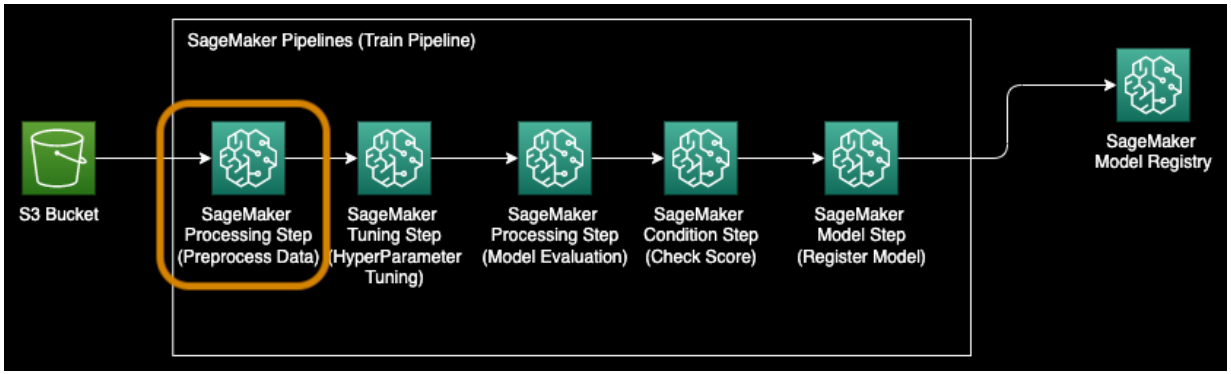
Step 2: Collect and prepare data

To follow along with this lab, you need to download and save the *sample dataset* into the project directly within the SageMaker Studio environment.

[5]:
```
# convert the store_data file into csv format
store_data = pd.read_excel("storedata_total.xlsx")
store_data.to_csv("storedata_total.csv")
```

Define Processing Step

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language



In this step, we prepare a python script to do feature engineering, one hot encoding, and curate the training, validation, and test splits to be used for model building. Run the next cell to build your processing script and save to the “scripts” folder:

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

[4]:

```
%%writefile "ML Pipelines scripts/churn_preprocess.py"

import os
import tempfile
import numpy as np
import pandas as pd
import datetime as dt
if __name__ == "__main__":
    base_dir = "/opt/ml/processing"
    #Read Data
    df = pd.read_csv(
        f"{base_dir}/input/storedata_total.csv"
    )
    # convert created column to datetime
    df["created"] = pd.to_datetime(df["created"])
    #Convert firstorder and lastorder to datetime datatype
    df["firstorder"] = pd.to_datetime(df["firstorder"],errors='coerce')
    df["lastorder"] = pd.to_datetime(df["lastorder"],errors='coerce')
    #Drop Rows with Null Values
    df = df.dropna()
    #Create column which gives the days between the last order and the first order
    df['first_last_days_diff'] = (df['lastorder'] - df['firstorder']).dt.days
    #Create column which gives the days between the customer record was created and the first order
    df['created_first_days_diff'] = (df['created'] - df['firstorder']).dt.days
    #Drop columns
    df.drop(['custid', 'created','firstorder','lastorder'], axis=1, inplace=True)
    #Apply one hot encoding on favday and city columns
    df = pd.get_dummies(df, prefix=['favday', 'city'], columns=['favday', 'city'])
    # Split into train, validation and test datasets
    y = df.pop("retained")
    X_pre = df
    y_pre = y.to_numpy().reshape(len(y), 1)
    X = np.concatenate((y_pre, X_pre), axis=1)
    np.random.shuffle(X)
    # Split in Train, Test and Validation Datasets
    train, validation, test = np.split(X, [int(.7*len(X)), int(.85*len(X))])
    train_rows = np.shape(train)[0]
    validation_rows = np.shape(validation)[0]
    test_rows = np.shape(test)[0]
    train = pd.DataFrame(train)
    test = pd.DataFrame(test)
    validation = pd.DataFrame(validation)
    # Convert the label column to integer
    train[0] = train[0].astype(int)
    test[0] = test[0].astype(int)
    validation[0] = validation[0].astype(int)
    # Save the Dataframes as csv files
    train.to_csv(f"{base_dir}/train/train.csv", header=False, index=False)
    validation.to_csv(f"{base_dir}/validation/validation.csv", header=False, index=False)
    test.to_csv(f"{base_dir}/test/test.csv", header=False, index=False)
```

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

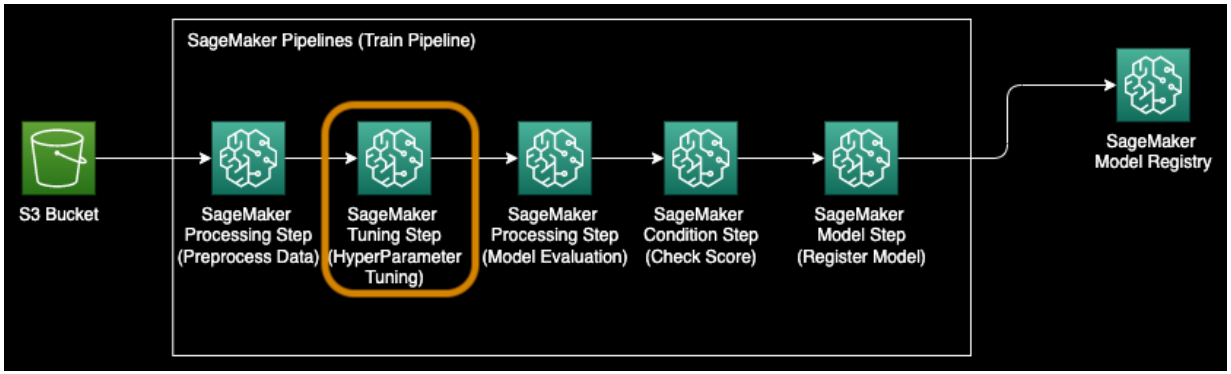
```
Overwriting ML Pipelines scripts/churn_preprocess.py
```

Run the next cell to instantiate the processor and the Pipelines step to run the processing script. Because the processing script is written in Pandas, you use a SKLearnProcessor. The Pipelines ProcessingStep function takes the following arguments: the processor, the input S3 locations for raw datasets, and the output S3 locations to save processed datasets.

[5]:
```
# Define Processing Step for Feature Engineering
from sagemaker.sklearn.processing import SKLearnProcessor
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker.workflow.steps import ProcessingStep

framework_version = "1.0-1"
sklearn_processor = SKLearnProcessor(
    framework_version=framework_version,
    instance_type="ml.m5.xlarge",
    instance_count=processing_instance_count,
    base_job_name="sklearn-churn-process",
    role=role,
    sagemaker_session=pipeline_session,
)
processor_args = sklearn_processor.run(
    inputs=[
      ProcessingInput(source=input_data, destination="/opt/ml/processing/input"),
    ],
    outputs=[
        ProcessingOutput(output_name="train", source="/opt/ml/processing/train",\
                         destination=f"s3://{default_bucket}/output/train" ),
        ProcessingOutput(output_name="validation", source="/opt/ml/processing/validation",\
                        destination=f"s3://{default_bucket}/output/validation"),
        ProcessingOutput(output_name="test", source="/opt/ml/processing/test",\
                        destination=f"s3://{default_bucket}/output/test")
    ],
    code=f"ML Pipelines scripts/churn_preprocess.py",
)
step_process = ProcessingStep(name="ChurnModelProcess", step_args=processor_args)
```

Define HyperParameter Tuning Step

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances



You can add a *model tuning step* TuningStep in the pipeline, which automatically invokes a hyperparameter tuning job (see the following code). The hyperparameter tuning finds the best version of a model by running many training jobs on the dataset using the algorithm and the ranges of hyperparameters that you specified.

Run the next cell to define the algorithm and fixed hyperparameters to be used with hyperparameter tuning.

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook
Content preferences
Language

[8]:
```
from sagemaker.estimator import Estimator
from sagemaker.inputs import TrainingInput
from sagemaker.tuner import (
    IntegerParameter,
    CategoricalParameter,
    ContinuousParameter,
    HyperparameterTuner,
)
from sagemaker.workflow.steps import TuningStep

# training step for generating model artifacts
model_path = f"s3://{default_bucket}/output"
image_uri = sagemaker.image_uris.retrieve(
    framework="xgboost",
    region=region,
    version="1.0-1",
    py_version="py3",
    instance_type=training_instance_type,
)
fixed_hyperparameters = {
"eval_metric":"auc",
"objective":"binary:logistic",
"num_round":"100",
"rate_drop":"0.3",
"tweedie_variance_power":"1.4"
}
xgb_train = Estimator(
    image_uri=image_uri,
    instance_type=training_instance_type,
    instance_count=1,
    hyperparameters=fixed_hyperparameters,
    output_path=model_path,
    base_job_name=f"churn-train",
```



```
    role=role,
)
```

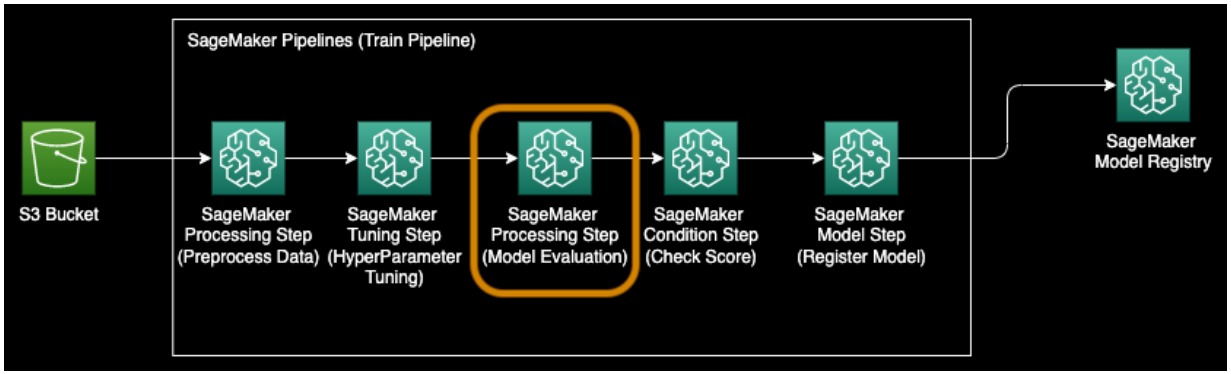
Now define the hyperparameter ranges, the metric to evaluate the best model against and the define the hyperparameter tuning step.

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

```
[ ]: hyperparameter_ranges = {
     "eta": ContinuousParameter(0, 1),
     "min_child_weight": ContinuousParameter(1, 10),
     "alpha": ContinuousParameter(0, 2),
     "max_depth": IntegerParameter(1, 10),
     }
     objective_metric_name = "validation:auc"

     tuner = HyperparameterTuner(
         xgb_train,
         objective_metric_name,
         hyperparameter_ranges,
         max_jobs=2,
         max_parallel_jobs=2,
     )

     hpo_args = tuner.fit(
         inputs={
             "train": TrainingInput(
                 s3_data=step_process.properties.ProcessingOutputConfig.Outputs["train"].S3Output.S3Uri,
                 content_type="text/csv",
             ),
             "validation": TrainingInput(
                 s3_data=step_process.properties.ProcessingOutputConfig.Outputs[
                     "validation"
                 ].S3Output.S3Uri,
                 content_type="text/csv",
             ),
         }
     )

     step_tuning = TuningStep(
         name="ChurnHyperParameterTuning",
         step_args=hpo_args,
     )
```

Define the evaluation script and model evaluation step

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language



Run the next cell to generate the script to evaluate the model once trained. This script encapsulates the logic to check if the AUC score meets the specified threshold.

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

[8]:

```
%%writefile "ML Pipelines scripts/churn_evaluate.py"

import json
import pathlib
import pickle
import tarfile
import joblib
import numpy as np
import pandas as pd
import xgboost
import datetime as dt
from sklearn.metrics import roc_curve,auc
if __name__ == "__main__":
    #Read Model Tar File
    model_path = f"/opt/ml/processing/model/model.tar.gz"
    with tarfile.open(model_path) as tar:
        tar.extractall(path=".")
    model = pickle.load(open("xgboost-model", "rb"))
    #Read Test Data using which we evaluate the model
    test_path = "/opt/ml/processing/test/test.csv"
    df = pd.read_csv(test_path, header=None)
    y_test = df.iloc[:, 0].to_numpy()
    df.drop(df.columns[0], axis=1, inplace=True)
    X_test = xgboost.DMatrix(df.values)
    #Run Predictions
    predictions = model.predict(X_test)
    #Evaluate Predictions
    fpr, tpr, thresholds = roc_curve(y_test, predictions)
    auc_score = auc(fpr, tpr)
    report_dict = {
        "classification_metrics": {
            "auc_score": {
                "value": auc_score,
            },
        },
```

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

```
    }
    #Save Evaluation Report
    output_dir = "/opt/ml/processing/evaluation"
    pathlib.Path(output_dir).mkdir(parents=True, exist_ok=True)
    evaluation_path = f"{output_dir}/evaluation.json"
    with open(evaluation_path, "w") as f:
        f.write(json.dumps(report_dict))
```

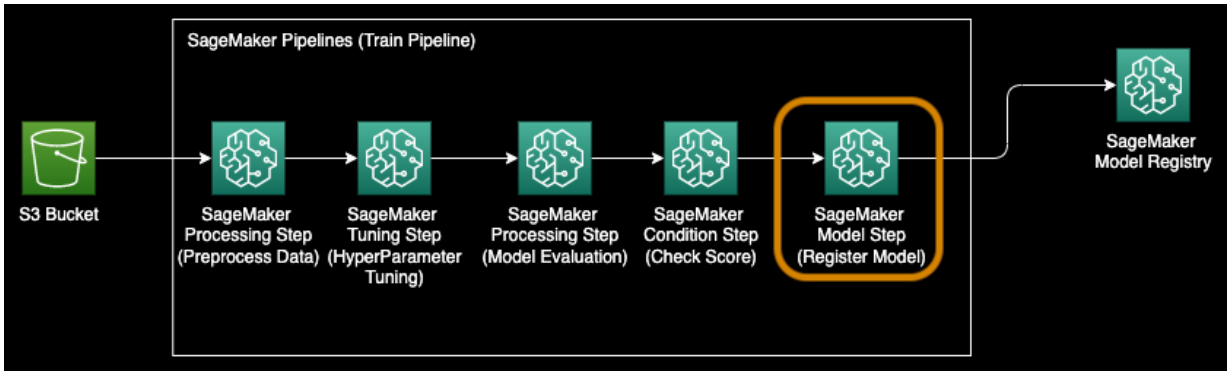
Next, run the following code block to instantiate the processor and the Pipelines step to run the evaluation script. Because the evaluation script uses the XGBoost package, you use a ScriptProcessor along with the XGBoost image. The Pipelines ProcessingStep function takes the following arguments: the processor, the input S3 locations for raw datasets, and the output S3 locations to save processed datasets.

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

[9]:
```
# define model evaluation step to evaluate the trained model
from sagemaker.processing import ScriptProcessor
script_eval = ScriptProcessor(
    image_uri=image_uri,
    command=["python3"],
    instance_type=processing_instance_type,
    instance_count=1,
    base_job_name="script-churn-eval",
    role=role,
    sagemaker_session=pipeline_session,
)
eval_args = script_eval.run(
     inputs=[
            ProcessingInput(
                source=step_tuning.get_top_model_s3_uri(top_k=0,s3_bucket=default_bucket,prefix="output"),
                destination="/opt/ml/processing/model"
            ),
            ProcessingInput(
                source=step_process.properties.ProcessingOutputConfig.Outputs[
                    "test"
                ].S3Output.S3Uri,
                destination="/opt/ml/processing/test"
            )
        ],
    outputs=[
            ProcessingOutput(output_name="evaluation", source="/opt/ml/processing/evaluation",\
                             destination=f"s3://{default_bucket}/output/evaluation"),
        ],
    code=f"ML Pipelines scripts/churn_evaluate.py",
)
from sagemaker.workflow.properties import PropertyFile

evaluation_report = PropertyFile(
    name="ChurnEvaluationReport", output_name="evaluation", path="evaluation.json"
)
step_eval = ProcessingStep(
    name="ChurnEvalModel",
    step_args=eval_args,
    property_files=[evaluation_report],
)
```

Define a register model step

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

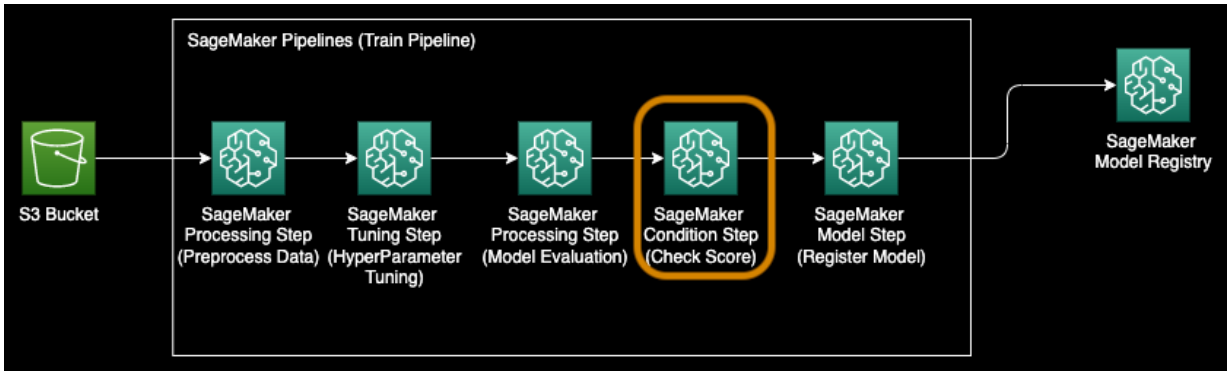


The following code registers the model within SageMaker model registry using the Pipelines model step:

[15]:
```
from sagemaker import Model
from sagemaker.workflow.model_step import ModelStep

model = Model(
    image_uri=image_uri,
    model_data=step_tuning.get_top_model_s3_uri(top_k=0,s3_bucket=default_bucket,prefix="output"),
    sagemaker_session=pipeline_session,
    role=role,
)
from sagemaker.model_metrics import MetricsSource, ModelMetrics

model_metrics = ModelMetrics(
    model_statistics=MetricsSource(
        s3_uri="{}/evaluation.json".format(
            step_eval.arguments["ProcessingOutputConfig"]["Outputs"][0]["S3Output"]["S3Uri"]
        ),
        content_type="application/json",
    )
)
register_args = model.register(
    content_types=["text/csv"],
    response_types=["text/csv"],
    inference_instances=["ml.t2.medium", "ml.m5.xlarge"],
    transform_instances=["ml.m5.xlarge"],
    model_package_group_name=model_package_group_name,
    approval_status=model_approval_status,
    model_metrics=model_metrics,
)
step_register = ModelStep(name="ChurnRegisterModel", step_args=register_args)
```

Define a condition step to check AUC score

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language



The following code defines a condition step to check the AUC score and conditionally create a model and register a model in the model registry.

```
[12]: from sagemaker.workflow.conditions import ConditionGreaterThan
      from sagemaker.workflow.condition_step import ConditionStep
      from sagemaker.workflow.functions import JsonGet
      cond_lte = ConditionGreaterThan(
          left=JsonGet(
              step_name=step_eval.name,
              property_file=evaluation_report,
              json_path="classification_metrics.auc_score.value",
          ),
          right=auc_score_threshold,
      )
      step_cond = ConditionStep(
          name="CheckAUCScoreChurnEvaluation",
          conditions=[cond_lte],
          if_steps=[step_register],
      )
```

Build and Trigger the pipeline run

After defining all of the component steps, you can assemble them into a Pipelines object. You don't need to specify the order of pipeline because Pipelines automatically infers the order sequence based on the dependencies between the steps.

Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

Run the following code in a cell in your notebook. If the pipeline already exists, the code updates the pipeline. If the pipeline doesn't exist, it creates a new one.

[16]:
```
import json
from sagemaker.workflow.pipeline import Pipeline

pipeline = Pipeline(
    name=pipeline_name,
    parameters=[
        processing_instance_count,
        processing_instance_type,
        training_instance_type,
        model_approval_status,
        input_data,
        auc_score_threshold,
    ],
    steps=[step_process, step_tuning, step_eval, step_cond],
)
definition = json.loads(pipeline.definition())
print(definition)
```

[17]:
```
# Create a new or update existing Pipeline
pipeline.upsert(role_arn=role)
# start Pipeline execution
pipeline.start()
```

Go to home within the studio domain and click on pipelines.

Script (TensorFlow)

Lab 3b. Bring your own Script (PyTorch)

Lab3c. Bring your own Container

Lab 4. Autopilot, Debugger and Model Monitor

Autopilot

Debugger

Model Monitor

Lab 5. Bias and Explainability

Bias and Explainability-Tabular Data

Lab 6. SageMaker Pipelines

Option 1: Train Pipeline (SageMaker Pipelines)

Option 2: Batch Inference Pipeline (SageMaker Pipelines)

Option 3: SageMaker Projects

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

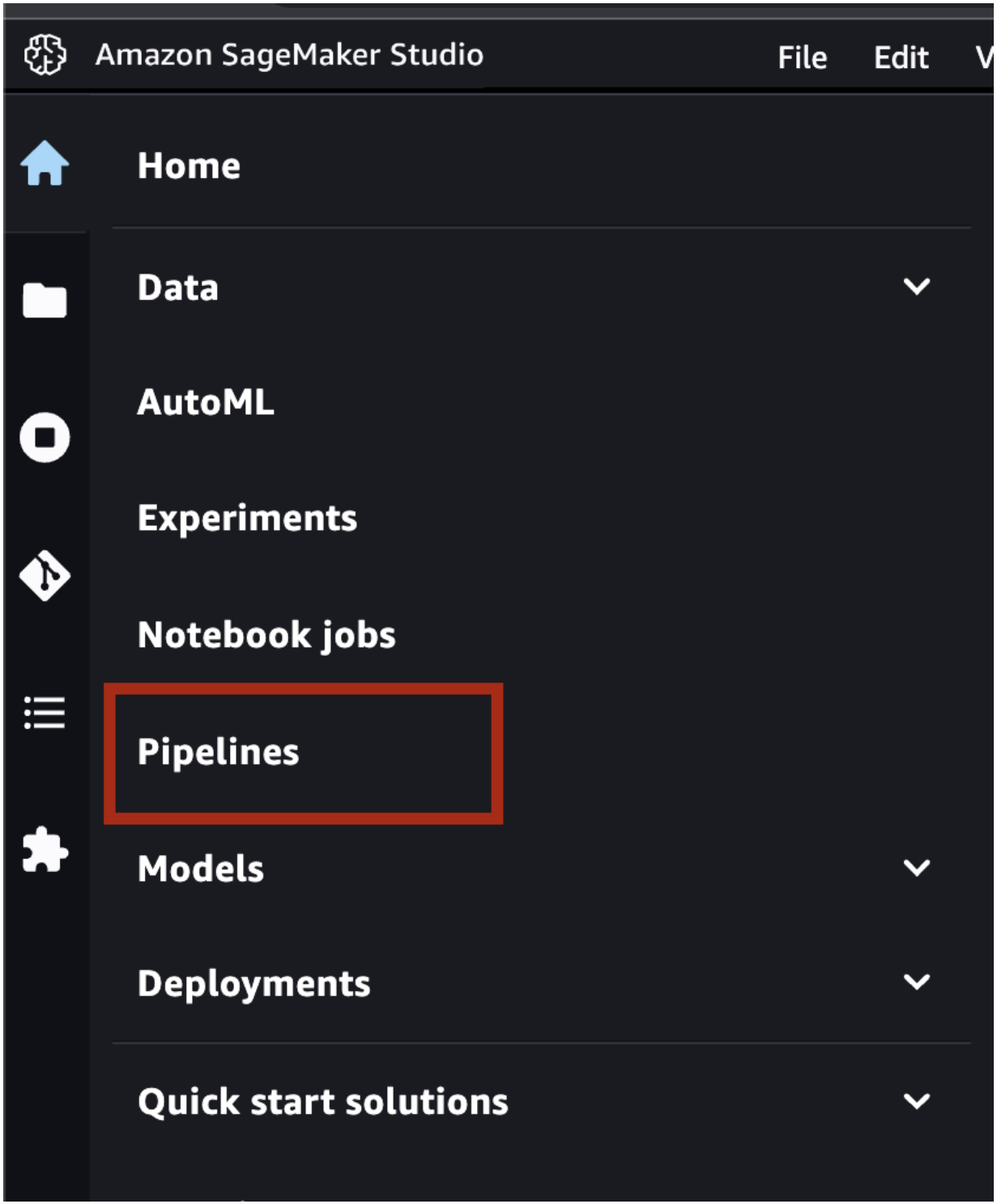
Lab 9. Amazon SageMaker JumpStart

Lab 10. ML Governance Tools for Amazon SageMaker

Lab 11. SageMaker Notebook Instances

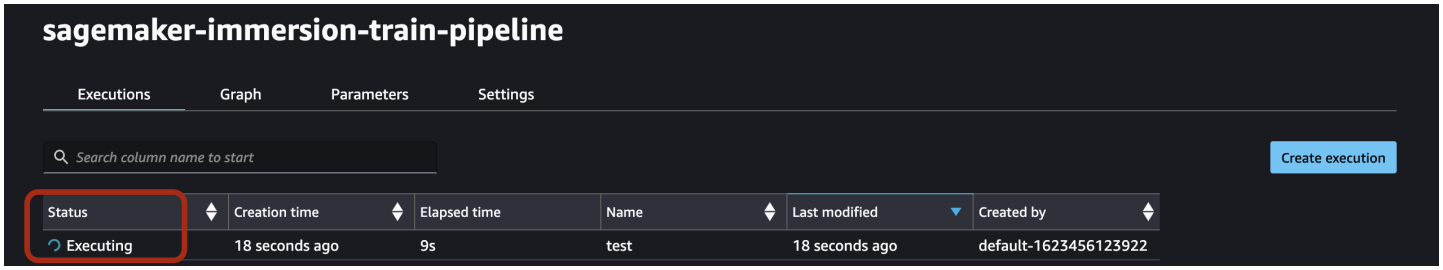
Content preferences

Language



Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

The train pipeline can be seen in execution status



If you double click on the executing pipelines, the steps of the pipeline will appear. You will be able to monitor the step that is currently running.

Script (TensorFlow)

Lab 3b. Bring your own Script (PyTorch)

Lab3c. Bring your own Container

Lab 4. Autopilot, Debugger and Model Monitor

Autopilot

Debugger

Model Monitor

Lab 5. Bias and Explainability

Bias and Explainability-Tabular Data

Lab 6. SageMaker Pipelines

Option 1: Train Pipeline (SageMaker Pipelines)

Option 2: Batch Inference Pipeline (SageMaker Pipelines)

Option 3: SageMaker Projects

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

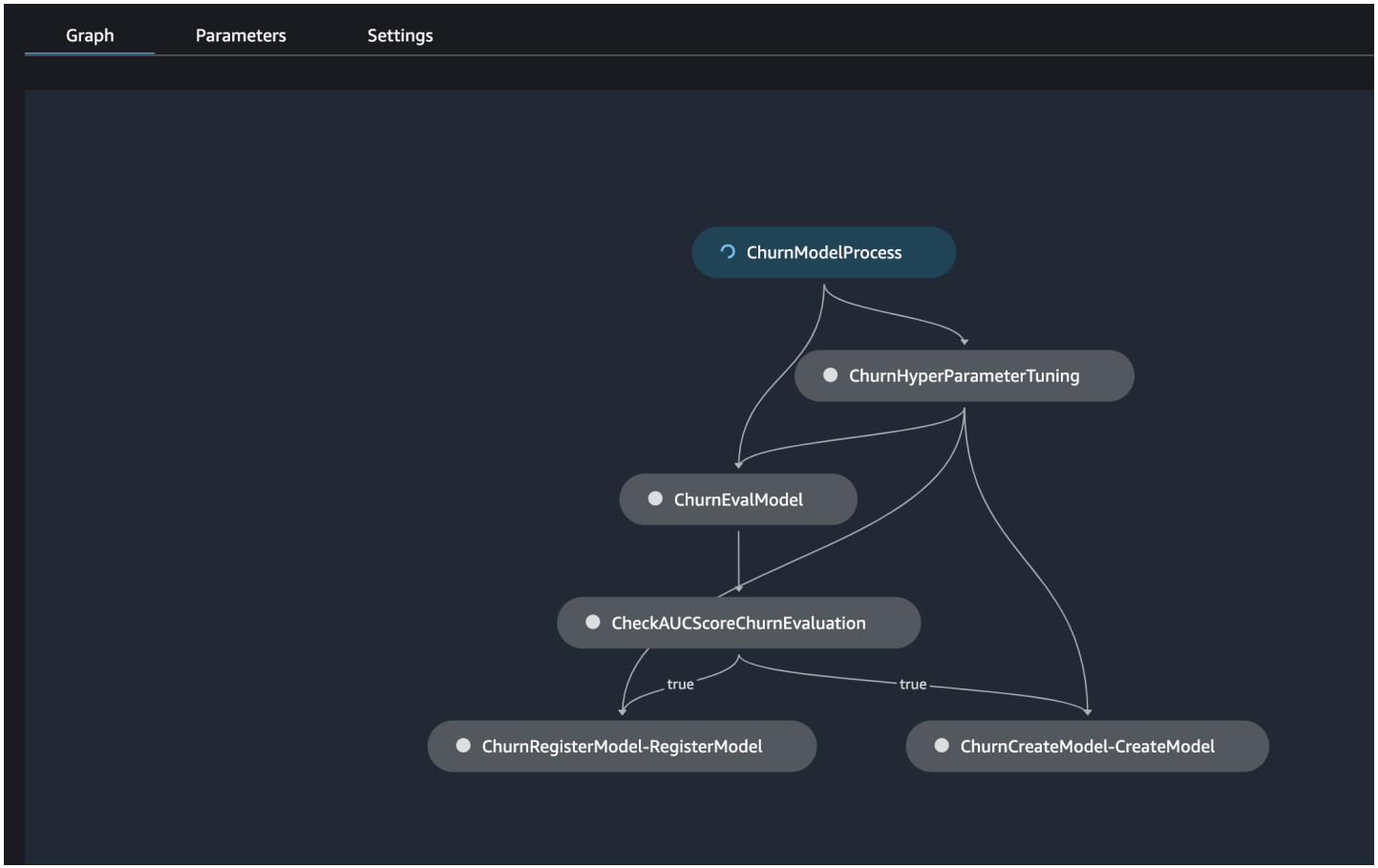
Lab 9. Amazon SageMaker JumpStart

Lab 10. ML Governance Tools for Amazon SageMaker

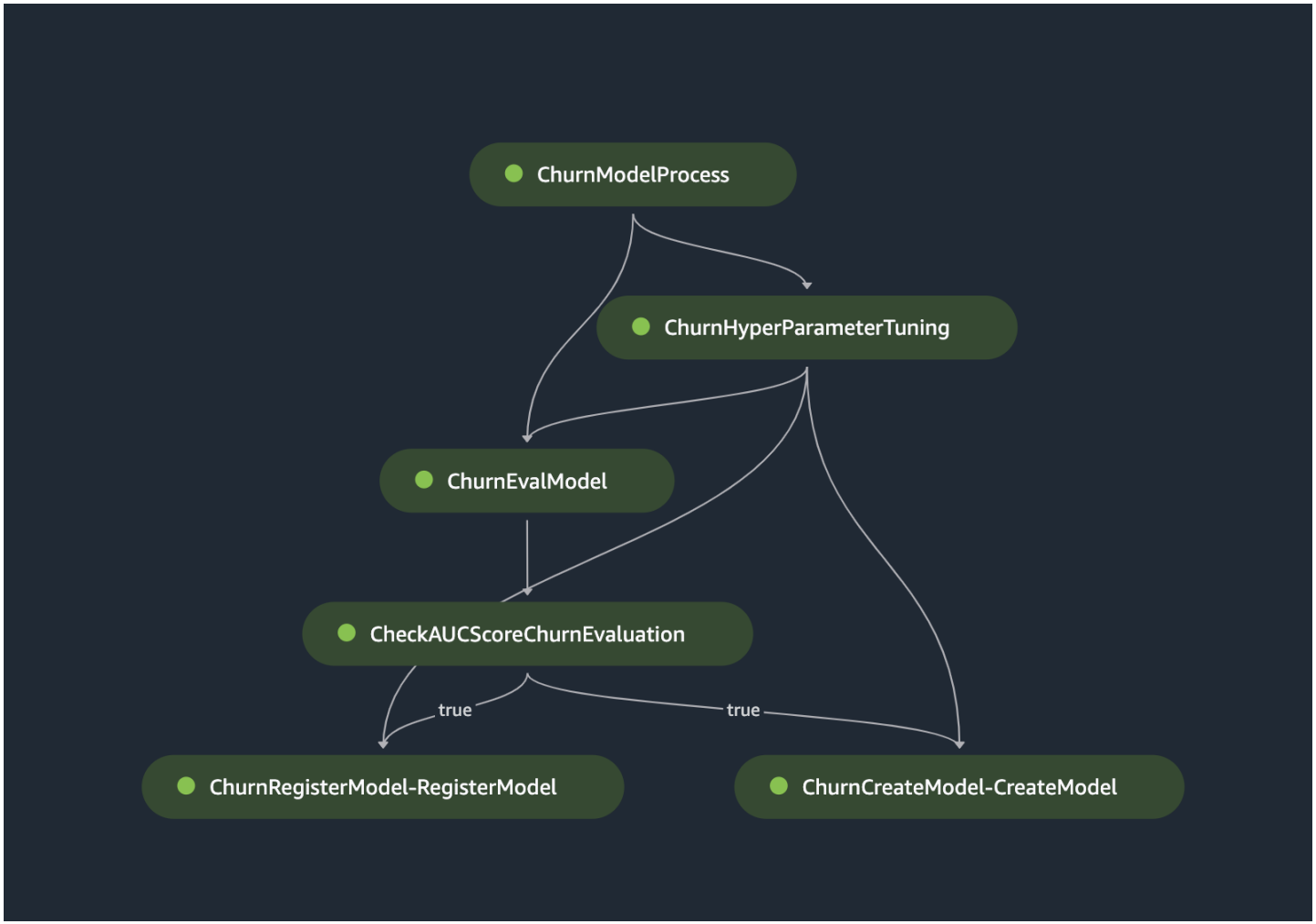
Lab 11. SageMaker Notebook Instances

Content preferences

Language



Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language



Once the pipeline execution is complete, clicking the tuning step should show the best model along with metrics

Script (TensorFlow)

Lab 3b. Bring your own Script (PyTorch)

Lab3c. Bring your own Container

Lab 4. Autopilot, Debugger and Model Monitor

Autopilot

Debugger

Model Monitor

Lab 5. Bias and Explainability

Bias and Explainability-Tabular Data

Lab 6. SageMaker Pipelines

Option 1: Train Pipeline (SageMaker Pipelines)

Option 2: Batch Inference Pipeline (SageMaker Pipelines)

Option 3: SageMaker Projects

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

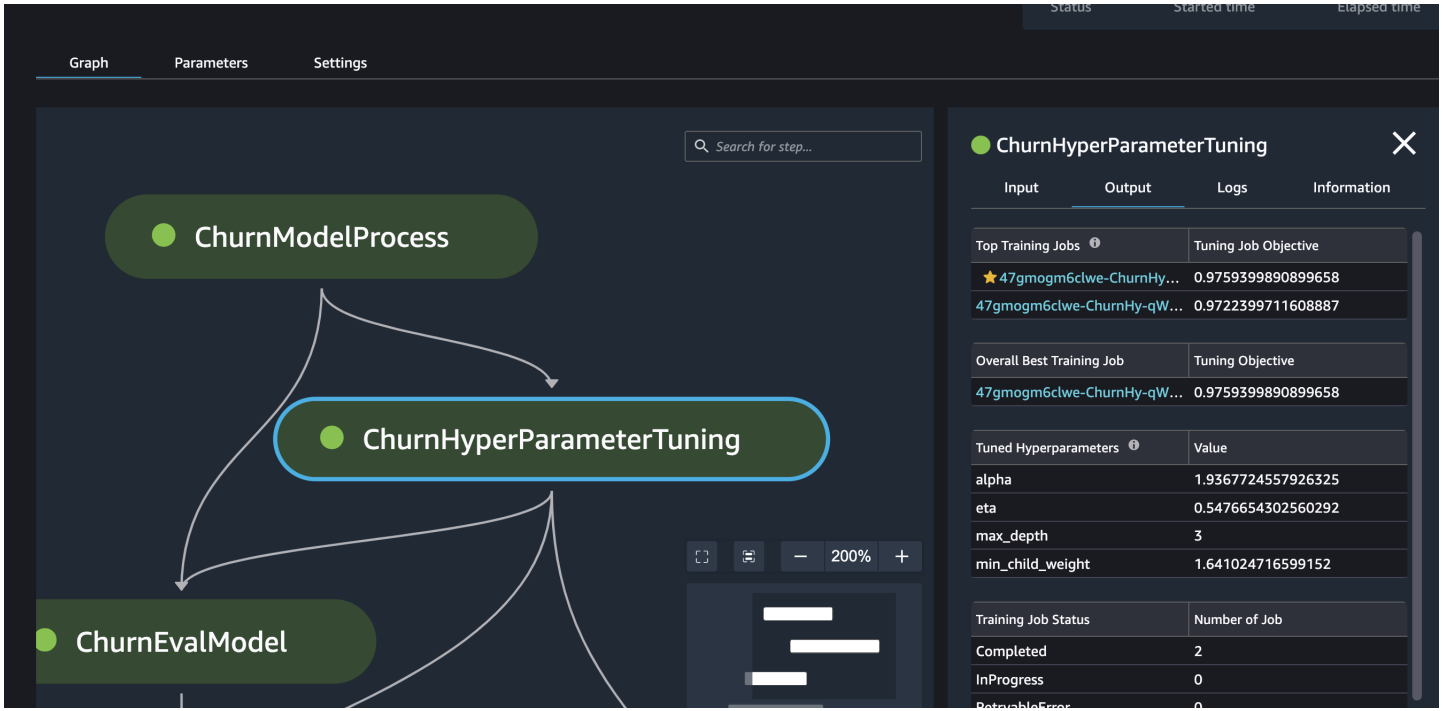
Lab 9. Amazon SageMaker JumpStart

Lab 10. ML Governance Tools for Amazon SageMaker

Lab 11. SageMaker Notebook Instances

Content preferences

Language



Given the model met with the condition with the specified threshold, the model has been registered within model registry which can be checked by going to the home screen and choosing the *Models → Model Registry.*

Script (TensorFlow)

Lab 3b. Bring your own Script (PyTorch)

Lab3c. Bring your own Container

Lab 4. Autopilot, Debugger and Model Monitor

Autopilot

Debugger

Model Monitor

Lab 5. Bias and Explainability

Bias and Explainability-Tabular Data

Lab 6. SageMaker Pipelines

Option 1: Train Pipeline (SageMaker Pipelines)

Option 2: Batch Inference Pipeline (SageMaker Pipelines)

Option 3: SageMaker Projects

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

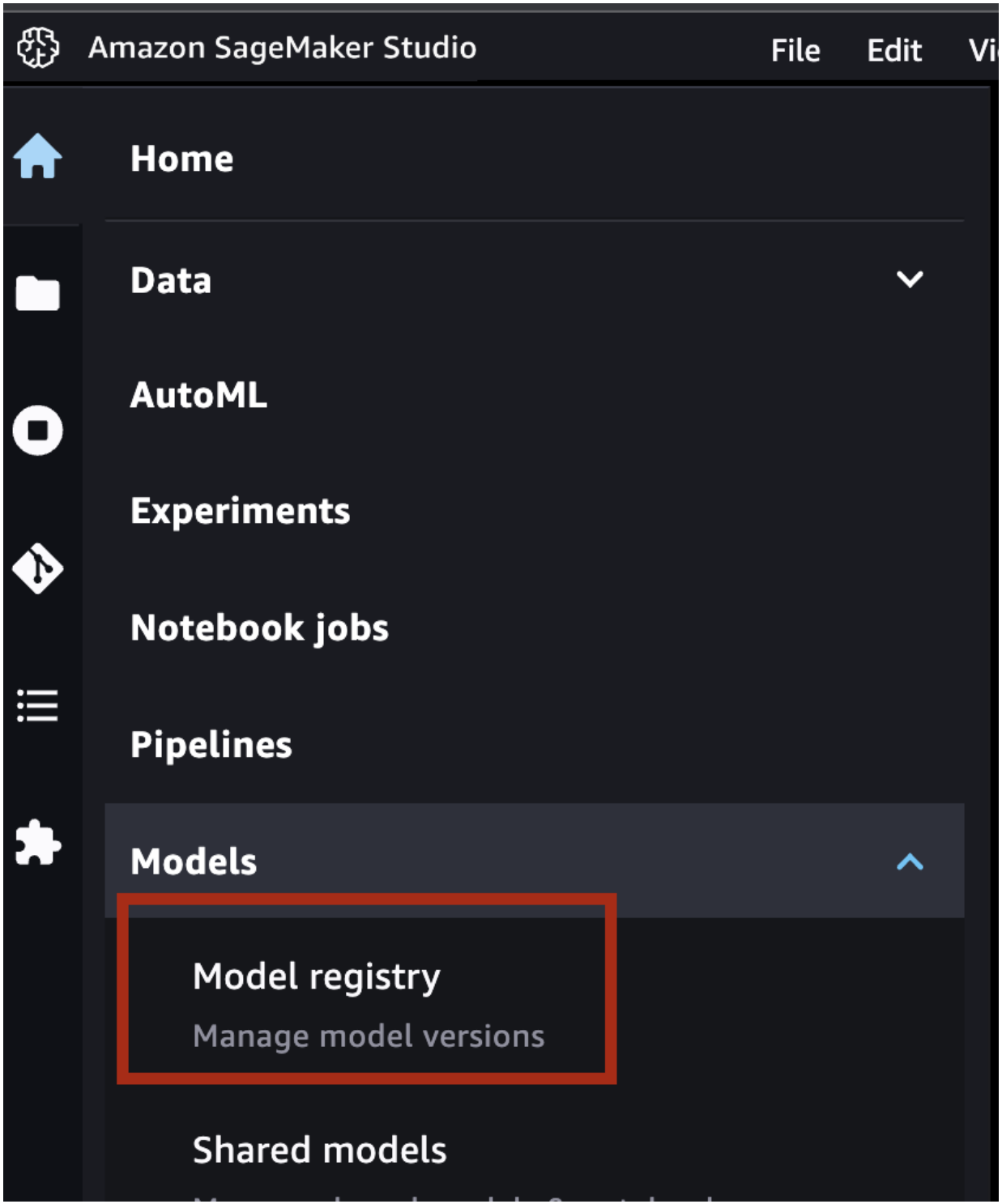
Lab 9. Amazon SageMaker JumpStart

Lab 10. ML Governance Tools for Amazon SageMaker

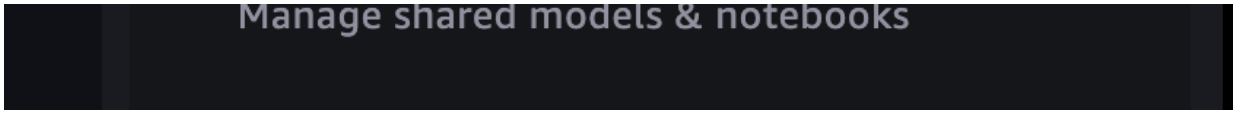
Lab 11. SageMaker Notebook Instances

Content preferences

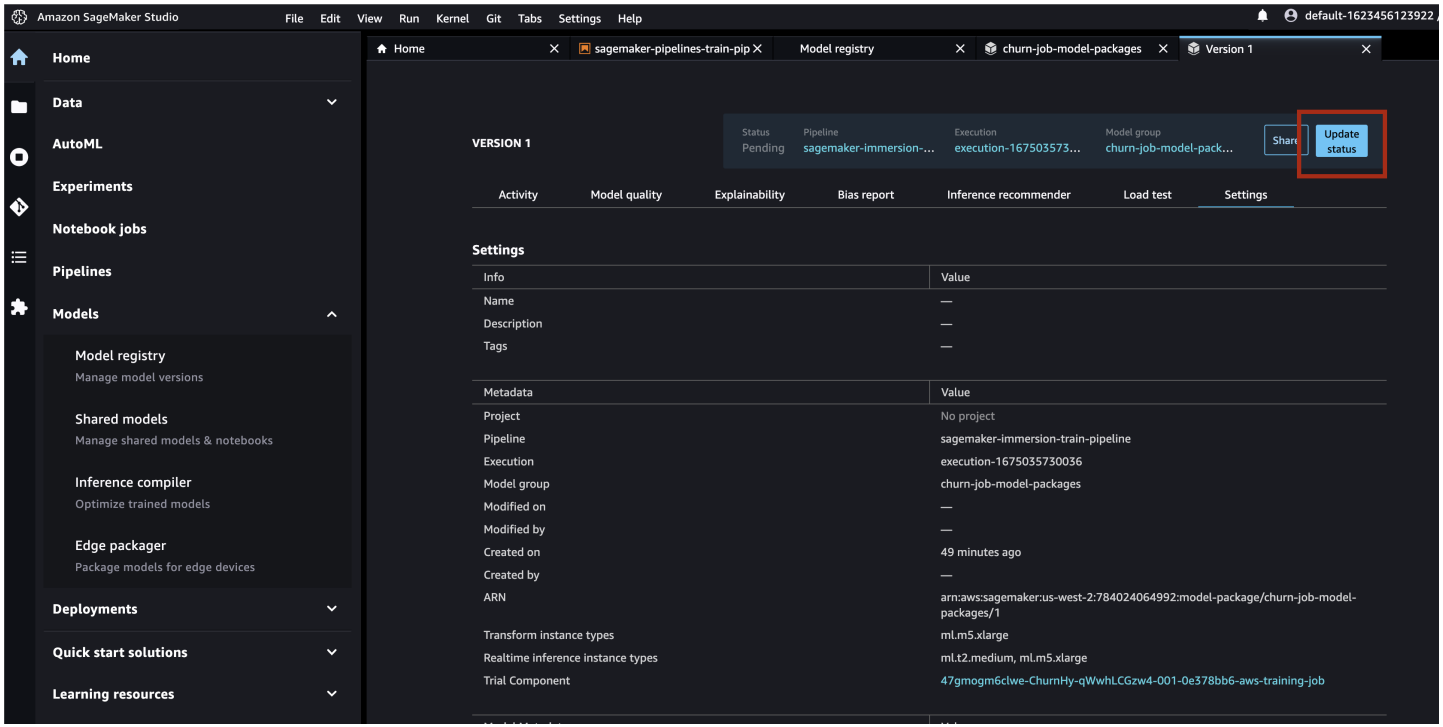
Language



Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language



You can inspect the details related to the model as shown below:



If everything looks good, you can click on the Update Status tab and manually approve the model.

Script (TensorFlow)

Lab 3b. Bring your own Script (PyTorch)

Lab3c. Bring your own Container

Lab 4. Autopilot, Debugger and Model Monitor

Autopilot

Debugger

Model Monitor

Lab 5. Bias and Explainability

Bias and Explainability-Tabular Data

Lab 6. SageMaker Pipelines

Option 1: Train Pipeline (SageMaker Pipelines)

Option 2: Batch Inference Pipeline (SageMaker Pipelines)

Option 3: SageMaker Projects

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

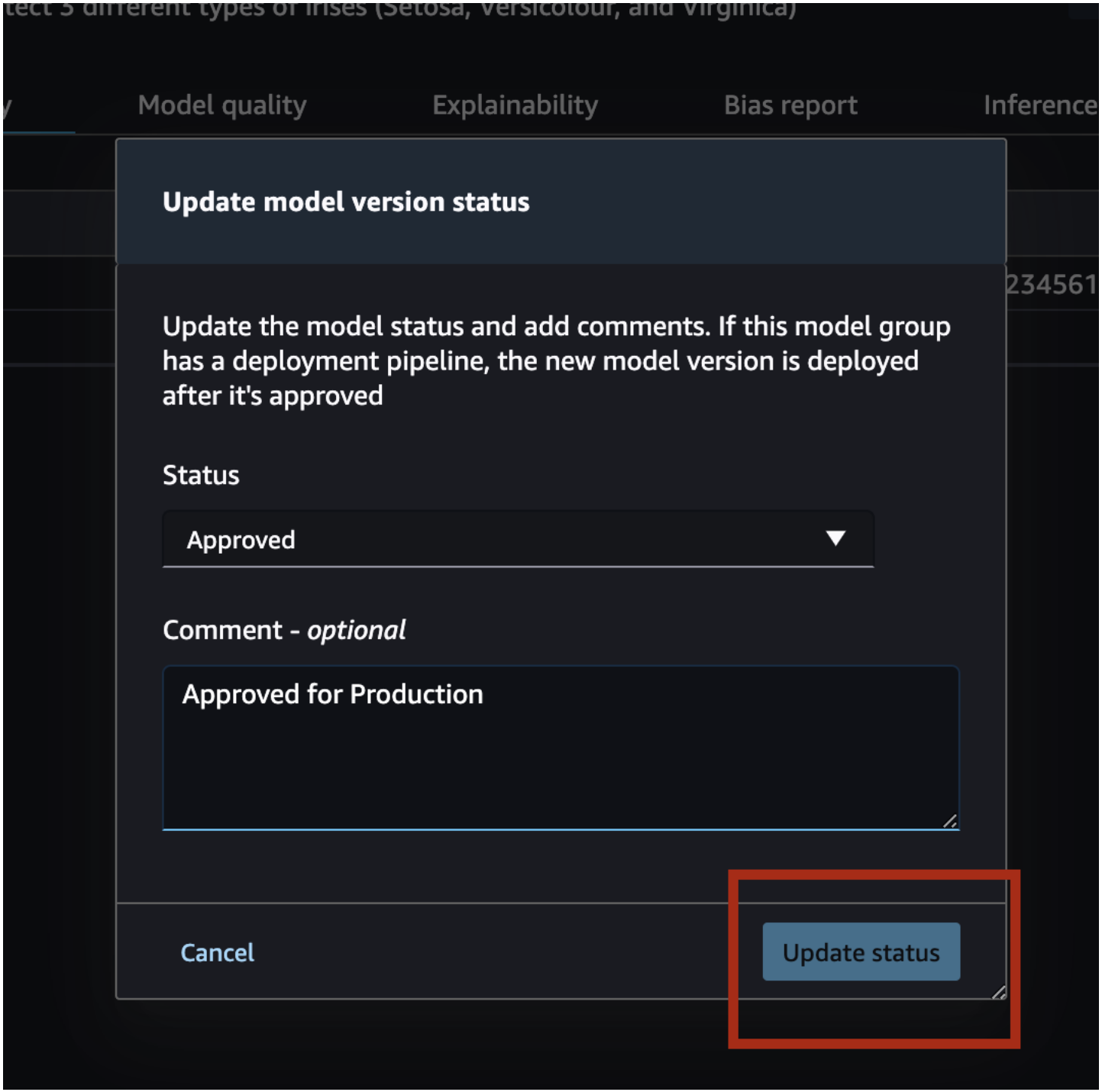
Lab 9. Amazon SageMaker JumpStart

Lab 10. ML Governance Tools for Amazon SageMaker

Lab 11. SageMaker Notebook Instances

Content preferences

Language



Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Option 1: Train Pipeline (SageMaker Pipelines)
Option 2: Batch Inference Pipeline (SageMaker Pipelines)
Option 3: SageMaker Projects
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

Conclusion

In this lab we have walked through how to build an ML workflow with the different steps to train the model train with SageMaker Pipelines along with other built-in SageMaker features and the XGBoost algorithm to develop, iterate, and deploy a model for churn prediction. The solution can be extended with additional data sources and other steps in the as needed to implement your own ML workflow. To learn more about SageMaker Pipelines, check out the website and the documentation.

Previous Next