Note: Tested in SageMaker Studio using the Python 3 (Data Science) kernel.

In [1]:
```
%pip install xgboost==1.3.1
```

```
/opt/conda/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16: CryptographyDeprecationWarn
ing: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
/opt/conda/lib/python3.7/site-packages/secretstorage/util.py:25: CryptographyDeprecationWarning:
int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
Requirement already satisfied: xgboost==1.3.1 in /opt/conda/lib/python3.7/site-packages (1.3.1)
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages (from xgboost==1.
3.1) (1.4.1)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages (from xgboost==1.
3.1) (1.20.3)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behavio
ur with the system package manager. It is recommended to use a virtual environment instead: http
s://pip.pypa.io/warnings/venv
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import boto3
import sagemaker
import os
from time import strftime
import sagemaker
from sagemaker import get_execution_role, session
from sagemaker.xgboost.estimator import XGBoost
from sagemaker.tuner import ContinuousParameter, HyperparameterTuner

sess = sagemaker.Session()
default_bucket = sess.default_bucket()
role = get_execution_role()
```

In [3]:
```
# get data
sm_session = sagemaker.session.Session()
sm_session.download_data(".", bucket="sagemaker-sample-files", key_prefix="datasets/tabular/xgb-churn/test-dataset.csv")

# upload to default bucket
s3_input_uri = sm_session.upload_data(".", bucket=default_bucket, key_prefix="datasets/tabular/xgb-churn/test-dataset.csv")
```

In [4]:
```
content_type = "text/csv"

s3_input_train = sagemaker.inputs.TrainingInput(
    s3_input_uri,
    content_type=content_type
)

s3_input_validation = sagemaker.inputs.TrainingInput(
    s3_input_uri,
    content_type=content_type
)
```

In [5]:
```
%%writefile xgb_train.py

import argparse
import os
import numpy as np
import pandas as pd
import pickle as pkl
import xgboost as xgb

def model_fn(model_dir):
    """Deserialize and return fitted model.

    Note that this should have the same name as the serialized model in the _xgb_train method
    """
    model_file = 'xgboost-model.pkl'
    with open(os.path.join(model_dir, model_file), 'rb') as f:
        booster = pkl.load(f)
    return booster

def _parse_args():
    """Parse job parameters."""

    parser = argparse.ArgumentParser()

    # Hyperparameters are described here.
    parser.add_argument('--objective', type=str, default="binary:logistic")
    parser.add_argument('--booster', type=str, default="gbtree")
    parser.add_argument('--eval_metric', type=str, default="error")
    parser.add_argument('--n_estimators', type=int, default=15)
    parser.add_argument('--max_depth', type=int, default=6)
    parser.add_argument('--min_child_weight', type=float, default=1)
    parser.add_argument('--subsample', type=float, default=1)
    parser.add_argument('--gamma', type=float, default=0)
    parser.add_argument('--alpha', type=float, default=0)
    parser.add_argument("--model-dir", type=str, default=os.environ.get("SM_MODEL_DIR"))
    parser.add_argument("--train", type=str, default=os.environ.get("SM_CHANNEL_TRAIN"))
    parser.add_argument("--validation", type=str, default=os.environ.get("SM_CHANNEL_VALIDATIO
N"))
    parser.add_argument("--train-file", type=str, default="test-dataset.csv")
    parser.add_argument("--validation-file", type=str, default="test-dataset.csv")
```

```
    return parser.parse_known_args()

if __name__ == '__main__':

    print("extracting arguments")
    args, _ = _parse_args()
    print(args)

    hyper_params_dict = {
        'objective': args.objective,
        'booster': args.booster,
        'eval_metric': args.eval_metric,
        'n_estimators': args.n_estimators,
        'max_depth': args.max_depth,
        'min_child_weight': args.min_child_weight,
        'subsample': args.subsample,
        'gamma': args.gamma,
        'alpha': args.alpha
    }

    print("Preparing data")
    train_df = pd.read_csv(os.path.join(args.train, args.train_file))
    validation_df = pd.read_csv(os.path.join(args.validation, args.validation_file))

    train_labels = np.array(train_df.iloc[:,0])
    validation_labels = np.array(validation_df.iloc[:,0])

    train_np = np.array(train_df.iloc[:,1:])
    validation_np = np.array(validation_df.iloc[:,1:])

    classifier = xgb.XGBClassifier(
        use_label_encoder=False,
        **hyper_params_dict
    )

    print("Fitting model")
    classifier.fit(
        train_np,
        train_labels,
        eval_set=[(train_np, train_labels), (validation_np, validation_labels)],
        verbose=True
    )
```

```
    print("Evaluating model")
    results = classifier.evals_result()
    for epoch, value in enumerate(results["validation_0"]["error"]):
        print(f"[{epoch}]#011train-error:{value}")  # Required for SageMaker to pick up this met
ric in the logs during HPO (See section 6)

    for epoch, value in enumerate(results["validation_1"]["error"]):
        print(f"[{epoch}]#011validation-error:{value}")  # Required for SageMaker to pick up thi
s metric in the logs during HPO (See section 6)

    print("Saving model")
    path = os.path.join(args.model_dir, "xgboost-model.pkl")
    pkl.dump(classifier, open(path, 'wb'))
    print("Model saved to " + path)
```

```
Overwriting xgb_train.py
```

In [6]:

```
# Define an XGBoost estimator, using the training script xgb_train.py as the entry_point.
# Note that we're specifying a framework version, py_version, and instance_type but NOT an image
_uri

hyper_params_dict = {
    'objective':            "binary:logistic",
    'booster':              "gbtree",
    'eval_metric':          "error"
}

xgb_estimator = XGBoost(entry_point = "xgb_train.py",
                        output_path = f"s3://{default_bucket}/",
                        framework_version="1.3-1",
                        py_version="py3",
                        hyperparameters=hyper_params_dict,
                        role=role,
                        instance_count=1,
                        instance_type='ml.m5.2xlarge',
                        enable_sagemaker_metrics=True,
                       )
```

In [7]:
```
#What Training image was used for this model?
xgb_estimator.training_image_uri()
```

Out[7]:
```
'257758044811.dkr.ecr.us-east-2.amazonaws.com/sagemaker-xgboost:1.3-1'
```

In [8]:
```
# This is the same value returned from sagemaker.image_uris.retrieve. Note that the tag is just
"1.3.1".
sagemaker.image_uris.retrieve(framework="xgboost", region="us-east-2", version="1.3-1", py_versi
on="py3", instance_type="ml.m4.2xlarge")
```

Out[8]:
```
'257758044811.dkr.ecr.us-east-2.amazonaws.com/sagemaker-xgboost:1.3-1'
```

In [9]:
```
# Now lets run a HPO job
hpo_tuner = HyperparameterTuner(
    xgb_estimator,
    objective_metric_name = "validation:error",
    objective_type = "Minimize",
    hyperparameter_ranges = {"alpha": ContinuousParameter(0, 1000)},
    max_jobs=5,
    max_parallel_jobs=5,
)

hpo_tuner.fit(
    {'train': s3_input_train, 'validation': s3_input_validation},
#     include_cls_metadata=True,
)
```

```
.....................................................!
```

In [10]:
```
# Let's look at the results
tuner = hpo_tuner.analytics()
tuning_results = tuner.dataframe().sort_values(by="FinalObjectiveValue")
tuning_results
```

Out[10]:

	alpha	TrainingJobName	TrainingJobStatus	FinalObjectiveValue	TrainingStartTime	TrainingEndTime	TrainingElapsed
0	359.843406	sagemaker-xgboost-220127-1730-005-1e00cd3b	Completed	0.144144	2022-01-27 17:33:34+00:00	2022-01-27 17:34:30+00:00	
1	152.070202	sagemaker-xgboost-220127-1730-004-18af9201	Completed	0.144144	2022-01-27 17:32:56+00:00	2022-01-27 17:33:52+00:00	
2	335.732962	sagemaker-xgboost-220127-1730-003-4abb84b2	Completed	0.144144	2022-01-27 17:32:51+00:00	2022-01-27 17:33:48+00:00	
3	322.061990	sagemaker-xgboost-220127-1730-002-1a49496e	Completed	0.144144	2022-01-27 17:32:42+00:00	2022-01-27 17:33:42+00:00	
4	405.497053	sagemaker-xgboost-220127-1730-001-aecaec31	Completed	0.144144	2022-01-27 17:32:51+00:00	2022-01-27 17:33:46+00:00	

In [11]:
```
# Let's take a look at the image used for the HPO training jobs - same as before, tagged with "1.3-1"
tuner_description = hpo_tuner.describe()
tuner_description["TrainingJobDefinition"]["AlgorithmSpecification"]["TrainingImage"]
```

Out[11]:
```
'257758044811.dkr.ecr.us-east-2.amazonaws.com/sagemaker-xgboost:1.3-1'
```

In [12]:
```
# But wait, when we try to deploy the model, we get an error!
hpo_tuner.deploy(
    initial_instance_count=1,
    instance_type="ml.m5.2xlarge",
)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-12-55756f1c9135> in <module>
      2 hpo_tuner.deploy(
      3     initial_instance_count=1,
----> 4     instance_type="ml.m5.2xlarge",
      5 )

/opt/conda/lib/python3.7/site-packages/sagemaker/tuner.py in deploy(self, initial_instance_coun
t, instance_type, serializer, deserializer, accelerator_type, endpoint_name, wait, model_name, k
ms_key, data_capture_config, **kwargs)
    757         """
    758         best_training_job = self._get_best_training_job()
--> 759         best_estimator = self.best_estimator(best_training_job)
    760
    761         return best_estimator.deploy(

/opt/conda/lib/python3.7/site-packages/sagemaker/tuner.py in best_estimator(self, best_training_
job)
    824         return best_estimator.attach(
    825             training_job_name=best_training_job["TrainingJobName"],
--> 826             sagemaker_session=self.sagemaker_session,
    827         )
    828

/opt/conda/lib/python3.7/site-packages/sagemaker/xgboost/estimator.py in attach(cls, training_jo
b_name, sagemaker_session, model_channel_name)
    226             TrainingJobName=training_job_name
    227         )
--> 228         init_params = cls._prepare_init_params_from_job_description(job_details, model_c
hannel_name)
    229         tags = sagemaker_session.sagemaker_client.list_tags(
    230             ResourceArn=job_details["TrainingJobArn"]

/opt/conda/lib/python3.7/site-packages/sagemaker/xgboost/estimator.py in _prepare_init_params_fr
om_job_description(cls, job_details, model_channel_name)
    269                 )
    270             )
--> 271         init_params["framework_version"] = framework_version_from_tag(tag)
    272
    273         if not framework:
```

```
/opt/conda/lib/python3.7/site-packages/sagemaker/fw_utils.py in framework_version_from_tag(image
_tag)
    332     """
    333     tag_pattern = re.compile(r"^(.*)-(cpu|gpu)-(py2|py3\d*)$")
--> 334     tag_match = tag_pattern.match(image_tag)
    335     return None if tag_match is None else tag_match.group(1)
    336

TypeError: expected string or bytes-like object
```

In [13]:
```
# This should identify one of the training jobs - but it throws the same error instead!
best_estimator = hpo_tuner.best_estimator()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-13-c9c78c2eaba6> in <module>
      1 # This should identify one of the training jobs - but it throws the same error instead!
----> 2 best_estimator = hpo_tuner.best_estimator()

/opt/conda/lib/python3.7/site-packages/sagemaker/tuner.py in best_estimator(self, best_training_
job)
    824         return best_estimator.attach(
    825             training_job_name=best_training_job["TrainingJobName"],
--> 826             sagemaker_session=self.sagemaker_session,
    827         )
    828

/opt/conda/lib/python3.7/site-packages/sagemaker/xgboost/estimator.py in attach(cls, training_jo
b_name, sagemaker_session, model_channel_name)
    226             TrainingJobName=training_job_name
    227         )
--> 228         init_params = cls._prepare_init_params_from_job_description(job_details, model_c
hannel_name)
    229         tags = sagemaker_session.sagemaker_client.list_tags(
    230             ResourceArn=job_details["TrainingJobArn"]

/opt/conda/lib/python3.7/site-packages/sagemaker/xgboost/estimator.py in _prepare_init_params_fr
om_job_description(cls, job_details, model_channel_name)
    269                     )
    270             )
--> 271         init_params["framework_version"] = framework_version_from_tag(tag)
    272
    273         if not framework:

/opt/conda/lib/python3.7/site-packages/sagemaker/fw_utils.py in framework_version_from_tag(image
_tag)
    332     """
    333     tag_pattern = re.compile(r"^(.*)-(cpu|gpu)-(py2|py3\d*)$")
--> 334     tag_match = tag_pattern.match(image_tag)
    335     return None if tag_match is None else tag_match.group(1)
    336

TypeError: expected string or bytes-like object
```

In [14]:
```
# What about the best training job? This works!
hpo_tuner.best_training_job()
```

Out[14]:
```
'sagemaker-xgboost-220127-1730-001-aecaec31'
```

So, what's going on? If you track down the `framework_version_from_tag` function in the SageMaker code (https://github.com/aws/sagemaker-python-sdk/blob/d9e2567e86ece46100f9bea6d21cd1ed73105521/src/sagemaker/fw_utils.py#L360 (https://github.com/aws/sagemaker-python-sdk/blob/d9e2567e86ece46100f9bea6d21cd1ed73105521/src/sagemaker/fw_utils.py#L360)) you find a docstring that says:

```
Args:
        image_tag (str): Image tag, which should take the form
            '<framework_version>-<device>-<py_version>'
```

That is NOT the tag our image is using (1.3-1). Instead, it seems like it should be tagged like this: `xgboost-cpu-1.3-1`

Interestingly enough, that tag DOES appear in the docs for the SageMaker XGBoost container image (https://github.com/aws/sagemaker-xgboost-container (https://github.com/aws/sagemaker-xgboost-container)). The README states:

```
Tagging scheme is based on <SageMaker-XGBoost-version>-cpu-py3 (e.g. 1.3-1-cpu-py3), where
    <SageMaker-XGBoost-version> is comprised of <XGBoost-version>-<SageMaker-version>.
```

What happens if we specify that container tag directly in the estimator creation? (I'm using US-EAST-2, so the ECR registry is 257758044811)

In [15]:
```
# Define an XGBoost estimator, using the training script xgb_train.py as the entry_point.
# Note that we're specifying an image_uri now, in addition to the other parameters

hyper_params_dict = {
    'objective':            "binary:logistic",
    'booster':              "gbtree",
    'eval_metric':          "error"
}

xgb_estimator = XGBoost(entry_point = "xgb_train.py",
                    output_path = f"s3://{default_bucket}/",
                    framework_version="1.3-1",
                    py_version="py3",
                    hyperparameters=hyper_params_dict,
                    role=role,
                    instance_count=1,
                    instance_type='ml.m5.2xlarge',
                    image_uri='257758044811.dkr.ecr.us-east-2.amazonaws.com/sagemaker-xgboost:1.
3-1-cpu-py3',
                    enable_sagemaker_metrics=True,
                   )
```

In [16]:
```
# Now lets run a HPO job
new_hpo_tuner = HyperparameterTuner(
    xgb_estimator,
    objective_metric_name = "validation:error",
    objective_type = "Minimize",
    hyperparameter_ranges = {"alpha": ContinuousParameter(0, 1000)},
    max_jobs=5,
    max_parallel_jobs=5,
)

new_hpo_tuner.fit(
    {'train': s3_input_train, 'validation': s3_input_validation},
)
```

```
.............................................................!
```

In [17]:
```
# Let's take a look at the image used for the HPO training jobs - it's different than the one we used before
tuner_description = new_hpo_tuner.describe()
tuner_description["TrainingJobDefinition"]["AlgorithmSpecification"]["TrainingImage"]
```

Out[17]:
```
'257758044811.dkr.ecr.us-east-2.amazonaws.com/sagemaker-xgboost:1.3-1-cpu-py3'
```

In [18]:
```
# Can we get the best estimator now? YES!
best_estimator = new_hpo_tuner.best_estimator()
best_estimator
```

```
2022-01-27 18:14:31 Starting - Preparing the instances for training
2022-01-27 18:14:31 Downloading - Downloading input data
2022-01-27 18:14:31 Training - Training image download completed. Training in progress.
2022-01-27 18:14:31 Uploading - Uploading generated training model
2022-01-27 18:14:31 Completed - Training job completed[2022-01-27 18:13:40.181 ip-10-0-67-179.us
-east-2.compute.internal:1 INFO utils.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: None
[2022-01-27:18:13:40:INFO] Imported framework sagemaker_xgboost_container.training
[2022-01-27:18:13:40:INFO] Failed to parse hyperparameter _tuning_objective_metric value validat
ion:error to Json.
Returning the value itself
[2022-01-27:18:13:40:INFO] No GPUs detected (normal if no gpus installed)
[2022-01-27:18:13:40:INFO] Invoking user training script.
[2022-01-27:18:13:40:INFO] Module xgb_train does not provide a setup.py.
Generating setup.py
[2022-01-27:18:13:40:INFO] Generating setup.cfg
[2022-01-27:18:13:40:INFO] Generating MANIFEST.in
[2022-01-27:18:13:40:INFO] Installing module with the following command:
/miniconda3/bin/python3 -m pip install .
Processing /opt/ml/code
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Building wheels for collected packages: xgb-train
  Building wheel for xgb-train (setup.py): started
  Building wheel for xgb-train (setup.py): finished with status 'done'
  Created wheel for xgb-train: filename=xgb_train-1.0.0-py2.py3-none-any.whl size=5078 sha256=f6
3f81bfcf39e50b7fe7f869f7b488ce3c8589f08c01fa64a30fad2ae45a72b3
  Stored in directory: /home/model-server/tmp/pip-ephem-wheel-cache-dron06k5/wheels/3e/0f/51/2f1
df833dd0412c1bc2f5ee56baac195b5be563353d111dca6
Successfully built xgb-train
Installing collected packages: xgb-train
Successfully installed xgb-train-1.0.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behavio
ur with the system package manager. It is recommended to use a virtual environment instead: http
s://pip.pypa.io/warnings/venv
[2022-01-27:18:13:42:INFO] Failed to parse hyperparameter _tuning_objective_metric value validat
ion:error to Json.
Returning the value itself
[2022-01-27:18:13:42:INFO] No GPUs detected (normal if no gpus installed)
[2022-01-27:18:13:42:INFO] Invoking user script
Training Env:
{
```

```
    "additional_framework_parameters": {
        "sagemaker_estimator_class_name": "XGBoost",
        "sagemaker_estimator_module": "sagemaker.xgboost.estimator"
    },
    "channel_input_dirs": {
        "validation": "/opt/ml/input/data/validation",
        "train": "/opt/ml/input/data/train"
    },
    "current_host": "algo-1",
    "framework_module": "sagemaker_xgboost_container.training:main",
    "hosts": [
        "algo-1"
    ],
    "hyperparameters": {
        "eval_metric": "error",
        "alpha": 952.2896578085882,
        "booster": "gbtree",
        "objective": "binary:logistic"
    },
    "input_config_dir": "/opt/ml/input/config",
    "input_data_config": {
        "validation": {
            "ContentType": "text/csv",
            "TrainingInputMode": "File",
            "S3DistributionType": "FullyReplicated",
            "RecordWrapperType": "None"
        },
        "train": {
            "ContentType": "text/csv",
            "TrainingInputMode": "File",
            "S3DistributionType": "FullyReplicated",
            "RecordWrapperType": "None"
        }
    },
    "input_dir": "/opt/ml/input",
    "is_master": true,
    "job_name": "sagemaker-xgboost-220127-1810-001-120a4eef",
    "log_level": 20,
    "master_hostname": "algo-1",
    "model_dir": "/opt/ml/model",
    "module_dir": "s3://sagemaker-us-east-2-167428594774/sagemaker-xgboost-2022-01-27-18-10-22-5
85/source/sourcedir.tar.gz",
```

```
    "module_name": "xgb_train",
    "network_interface_name": "eth0",
    "num_cpus": 8,
    "num_gpus": 0,
    "output_data_dir": "/opt/ml/output/data",
    "output_dir": "/opt/ml/output",
    "output_intermediate_dir": "/opt/ml/output/intermediate",
    "resource_config": {
        "current_host": "algo-1",
        "hosts": [
            "algo-1"
        ],
        "network_interface_name": "eth0"
    },
    "user_entry_point": "xgb_train.py"
}
Environment variables:
SM_HOSTS=["algo-1"]
SM_NETWORK_INTERFACE_NAME=eth0
SM_HPS={"alpha":952.2896578085882,"booster":"gbtree","eval_metric":"error","objective":"binary:l
ogistic"}
SM_USER_ENTRY_POINT=xgb_train.py
SM_FRAMEWORK_PARAMS={"sagemaker_estimator_class_name":"XGBoost","sagemaker_estimator_module":"sa
gemaker.xgboost.estimator"}
SM_RESOURCE_CONFIG={"current_host":"algo-1","hosts":["algo-1"],"network_interface_name":"eth0"}
SM_INPUT_DATA_CONFIG={"train":{"ContentType":"text/csv","RecordWrapperType":"None","S3Distributi
onType":"FullyReplicated","TrainingInputMode":"File"},"validation":{"ContentType":"text/csv","Re
cordWrapperType":"None","S3DistributionType":"FullyReplicated","TrainingInputMode":"File"}}
SM_OUTPUT_DATA_DIR=/opt/ml/output/data
SM_CHANNELS=["train","validation"]
SM_CURRENT_HOST=algo-1
SM_MODULE_NAME=xgb_train
SM_LOG_LEVEL=20
SM_FRAMEWORK_MODULE=sagemaker_xgboost_container.training:main
SM_INPUT_DIR=/opt/ml/input
SM_INPUT_CONFIG_DIR=/opt/ml/input/config
SM_OUTPUT_DIR=/opt/ml/output
SM_NUM_CPUS=8
SM_NUM_GPUS=0
SM_MODEL_DIR=/opt/ml/model
SM_MODULE_DIR=s3://sagemaker-us-east-2-167428594774/sagemaker-xgboost-2022-01-27-18-10-22-585/so
urce/sourcedir.tar.gz
```

```
SM_TRAINING_ENV={"additional_framework_parameters":{"sagemaker_estimator_class_name":"XGBoost","
sagemaker_estimator_module":"sagemaker.xgboost.estimator"},"channel_input_dirs":{"train":"/opt/m
l/input/data/train","validation":"/opt/ml/input/data/validation"},"current_host":"algo-1","frame
work_module":"sagemaker_xgboost_container.training:main","hosts":["algo-1"],"hyperparameters":{"
alpha":952.2896578085882,"booster":"gbtree","eval_metric":"error","objective":"binary:logisti
c"},"input_config_dir":"/opt/ml/input/config","input_data_config":{"train":{"ContentType":"text/
csv","RecordWrapperType":"None","S3DistributionType":"FullyReplicated","TrainingInputMode":"Fil
e"},"validation":{"ContentType":"text/csv","RecordWrapperType":"None","S3DistributionType":"Full
yReplicated","TrainingInputMode":"File"}},"input_dir":"/opt/ml/input","is_master":true,"job_nam
e":"sagemaker-xgboost-220127-1810-001-120a4eef","log_level":20,"master_hostname":"algo-1","model
_dir":"/opt/ml/model","module_dir":"s3://sagemaker-us-east-2-167428594774/sagemaker-xgboost-2022
-01-27-18-10-22-585/source/sourcedir.tar.gz","module_name":"xgb_train","network_interface_nam
e":"eth0","num_cpus":8,"num_gpus":0,"output_data_dir":"/opt/ml/output/data","output_dir":"/opt/m
l/output","output_intermediate_dir":"/opt/ml/output/intermediate","resource_config":{"current_ho
st":"algo-1","hosts":["algo-1"],"network_interface_name":"eth0"},"user_entry_point":"xgb_train.p
y"}
SM_USER_ARGS=["--alpha","952.2896578085882","--booster","gbtree","--eval_metric","error","--obje
ctive","binary:logistic"]
SM_OUTPUT_INTERMEDIATE_DIR=/opt/ml/output/intermediate
SM_CHANNEL_VALIDATION=/opt/ml/input/data/validation
SM_CHANNEL_TRAIN=/opt/ml/input/data/train
SM_HP_EVAL_METRIC=error
SM_HP_ALPHA=952.2896578085882
SM_HP_BOOSTER=gbtree
SM_HP_OBJECTIVE=binary:logistic
PYTHONPATH=/miniconda3/bin:/:/miniconda3/lib/python/site-packages/xgboost/dmlc-core/tracker:/min
iconda3/lib/python37.zip:/miniconda3/lib/python3.7:/miniconda3/lib/python3.7/lib-dynload:/minico
nda3/lib/python3.7/site-packages
Invoking script with the following command:
/miniconda3/bin/python3 -m xgb_train --alpha 952.2896578085882 --booster gbtree --eval_metric er
ror --objective binary:logistic
extracting arguments
Namespace(alpha=952.2896578085882, booster='gbtree', eval_metric='error', gamma=0, max_depth=6,
min_child_weight=1, model_dir='/opt/ml/model', n_estimators=15, objective='binary:logistic', sub
sample=1, train='/opt/ml/input/data/train', train_file='test-dataset.csv', validation='/opt/ml/i
nput/data/validation', validation_file='test-dataset.csv')
Preparing data
Fitting model
[0]#011validation_0-error:0.14414#011validation_1-error:0.14414
[1]#011validation_0-error:0.14414#011validation_1-error:0.14414
[2]#011validation_0-error:0.14414#011validation_1-error:0.14414
[3]#011validation_0-error:0.14414#011validation_1-error:0.14414
```

```
[4]#011validation_0-error:0.14414#011validation_1-error:0.14414
[5]#011validation_0-error:0.14414#011validation_1-error:0.14414
[6]#011validation_0-error:0.14414#011validation_1-error:0.14414
[7]#011validation_0-error:0.14414#011validation_1-error:0.14414
[8]#011validation_0-error:0.14414#011validation_1-error:0.14414
[9]#011validation_0-error:0.14414#011validation_1-error:0.14414
[10]#011validation_0-error:0.14414#011validation_1-error:0.14414
[11]#011validation_0-error:0.14414#011validation_1-error:0.14414
[12]#011validation_0-error:0.14414#011validation_1-error:0.14414
[13]#011validation_0-error:0.14414#011validation_1-error:0.14414
[14]#011validation_0-error:0.14414#011validation_1-error:0.14414
Evaluating model
[0]#011train-error:0.144144
[1]#011train-error:0.144144
[2]#011train-error:0.144144
[3]#011train-error:0.144144
[4]#011train-error:0.144144
[5]#011train-error:0.144144
[6]#011train-error:0.144144
[7]#011train-error:0.144144
[8]#011train-error:0.144144
[9]#011train-error:0.144144
[10]#011train-error:0.144144
[11]#011train-error:0.144144
[12]#011train-error:0.144144
[13]#011train-error:0.144144
[14]#011train-error:0.144144
[0]#011validation-error:0.144144
[1]#011validation-error:0.144144
[2]#011validation-error:0.144144
[3]#011validation-error:0.144144
[4]#011validation-error:0.144144
[5]#011validation-error:0.144144
[6]#011validation-error:0.144144
[7]#011validation-error:0.144144
[8]#011validation-error:0.144144
[9]#011validation-error:0.144144
[10]#011validation-error:0.144144
[11]#011validation-error:0.144144
[12]#011validation-error:0.144144
[13]#011validation-error:0.144144
[14]#011validation-error:0.144144
```

```
Saving model
Model saved to /opt/ml/model/xgboost-model.pkl
```

Out[18]: <sagemaker.xgboost.estimator.XGBoost at 0x7f2ef94b8990>

In [19]:
```
# Can we deploy? YES!
new_hpo_tuner.deploy(
    initial_instance_count=1,
    instance_type="ml.m5.2xlarge",
)
```

```
2022-01-27 18:14:31 Starting - Preparing the instances for training
2022-01-27 18:14:31 Downloading - Downloading input data
2022-01-27 18:14:31 Training - Training image download completed. Training in progress.
2022-01-27 18:14:31 Uploading - Uploading generated training model
2022-01-27 18:14:31 Completed - Training job completed[2022-01-27 18:13:40.181 ip-10-0-67-179.us
-east-2.compute.internal:1 INFO utils.py:27] RULE_JOB_STOP_SIGNAL_FILENAME: None
[2022-01-27:18:13:40:INFO] Imported framework sagemaker_xgboost_container.training
[2022-01-27:18:13:40:INFO] Failed to parse hyperparameter _tuning_objective_metric value validat
ion:error to Json.
Returning the value itself
[2022-01-27:18:13:40:INFO] No GPUs detected (normal if no gpus installed)
[2022-01-27:18:13:40:INFO] Invoking user training script.
[2022-01-27:18:13:40:INFO] Module xgb_train does not provide a setup.py.
Generating setup.py
[2022-01-27:18:13:40:INFO] Generating setup.cfg
[2022-01-27:18:13:40:INFO] Generating MANIFEST.in
[2022-01-27:18:13:40:INFO] Installing module with the following command:
/miniconda3/bin/python3 -m pip install .
Processing /opt/ml/code
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Building wheels for collected packages: xgb-train
  Building wheel for xgb-train (setup.py): started
  Building wheel for xgb-train (setup.py): finished with status 'done'
  Created wheel for xgb-train: filename=xgb_train-1.0.0-py2.py3-none-any.whl size=5078 sha256=f6
3f81bfcf39e50b7fe7f869f7b488ce3c8589f08c01fa64a30fad2ae45a72b3
  Stored in directory: /home/model-server/tmp/pip-ephem-wheel-cache-dron06k5/wheels/3e/0f/51/2f1
df833dd0412c1bc2f5ee56baac195b5be563353d111dca6
Successfully built xgb-train
Installing collected packages: xgb-train
Successfully installed xgb-train-1.0.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behavio
ur with the system package manager. It is recommended to use a virtual environment instead: http
s://pip.pypa.io/warnings/venv
[2022-01-27:18:13:42:INFO] Failed to parse hyperparameter _tuning_objective_metric value validat
ion:error to Json.
Returning the value itself
[2022-01-27:18:13:42:INFO] No GPUs detected (normal if no gpus installed)
[2022-01-27:18:13:42:INFO] Invoking user script
Training Env:
{
```

```
    "additional_framework_parameters": {
        "sagemaker_estimator_class_name": "XGBoost",
        "sagemaker_estimator_module": "sagemaker.xgboost.estimator"
    },
    "channel_input_dirs": {
        "validation": "/opt/ml/input/data/validation",
        "train": "/opt/ml/input/data/train"
    },
    "current_host": "algo-1",
    "framework_module": "sagemaker_xgboost_container.training:main",
    "hosts": [
        "algo-1"
    ],
    "hyperparameters": {
        "eval_metric": "error",
        "alpha": 952.2896578085882,
        "booster": "gbtree",
        "objective": "binary:logistic"
    },
    "input_config_dir": "/opt/ml/input/config",
    "input_data_config": {
        "validation": {
            "ContentType": "text/csv",
            "TrainingInputMode": "File",
            "S3DistributionType": "FullyReplicated",
            "RecordWrapperType": "None"
        },
        "train": {
            "ContentType": "text/csv",
            "TrainingInputMode": "File",
            "S3DistributionType": "FullyReplicated",
            "RecordWrapperType": "None"
        }
    },
    "input_dir": "/opt/ml/input",
    "is_master": true,
    "job_name": "sagemaker-xgboost-220127-1810-001-120a4eef",
    "log_level": 20,
    "master_hostname": "algo-1",
    "model_dir": "/opt/ml/model",
    "module_dir": "s3://sagemaker-us-east-2-167428594774/sagemaker-xgboost-2022-01-27-18-10-22-5
85/source/sourcedir.tar.gz",
```

```
    "module_name": "xgb_train",
    "network_interface_name": "eth0",
    "num_cpus": 8,
    "num_gpus": 0,
    "output_data_dir": "/opt/ml/output/data",
    "output_dir": "/opt/ml/output",
    "output_intermediate_dir": "/opt/ml/output/intermediate",
    "resource_config": {
        "current_host": "algo-1",
        "hosts": [
            "algo-1"
        ],
        "network_interface_name": "eth0"
    },
    "user_entry_point": "xgb_train.py"
}
Environment variables:
SM_HOSTS=["algo-1"]
SM_NETWORK_INTERFACE_NAME=eth0
SM_HPS={"alpha":952.2896578085882,"booster":"gbtree","eval_metric":"error","objective":"binary:l
ogistic"}
SM_USER_ENTRY_POINT=xgb_train.py
SM_FRAMEWORK_PARAMS={"sagemaker_estimator_class_name":"XGBoost","sagemaker_estimator_module":"sa
gemaker.xgboost.estimator"}
SM_RESOURCE_CONFIG={"current_host":"algo-1","hosts":["algo-1"],"network_interface_name":"eth0"}
SM_INPUT_DATA_CONFIG={"train":{"ContentType":"text/csv","RecordWrapperType":"None","S3Distributi
onType":"FullyReplicated","TrainingInputMode":"File"},"validation":{"ContentType":"text/csv","Re
cordWrapperType":"None","S3DistributionType":"FullyReplicated","TrainingInputMode":"File"}}
SM_OUTPUT_DATA_DIR=/opt/ml/output/data
SM_CHANNELS=["train","validation"]
SM_CURRENT_HOST=algo-1
SM_MODULE_NAME=xgb_train
SM_LOG_LEVEL=20
SM_FRAMEWORK_MODULE=sagemaker_xgboost_container.training:main
SM_INPUT_DIR=/opt/ml/input
SM_INPUT_CONFIG_DIR=/opt/ml/input/config
SM_OUTPUT_DIR=/opt/ml/output
SM_NUM_CPUS=8
SM_NUM_GPUS=0
SM_MODEL_DIR=/opt/ml/model
SM_MODULE_DIR=s3://sagemaker-us-east-2-167428594774/sagemaker-xgboost-2022-01-27-18-10-22-585/so
urce/sourcedir.tar.gz
```

```
SM_TRAINING_ENV={"additional_framework_parameters":{"sagemaker_estimator_class_name":"XGBoost","
sagemaker_estimator_module":"sagemaker.xgboost.estimator"},"channel_input_dirs":{"train":"/opt/m
l/input/data/train","validation":"/opt/ml/input/data/validation"},"current_host":"algo-1","frame
work_module":"sagemaker_xgboost_container.training:main","hosts":["algo-1"],"hyperparameters":{"
alpha":952.2896578085882,"booster":"gbtree","eval_metric":"error","objective":"binary:logisti
c"},"input_config_dir":"/opt/ml/input/config","input_data_config":{"train":{"ContentType":"text/
csv","RecordWrapperType":"None","S3DistributionType":"FullyReplicated","TrainingInputMode":"Fil
e"},"validation":{"ContentType":"text/csv","RecordWrapperType":"None","S3DistributionType":"Full
yReplicated","TrainingInputMode":"File"}},"input_dir":"/opt/ml/input","is_master":true,"job_nam
e":"sagemaker-xgboost-220127-1810-001-120a4eef","log_level":20,"master_hostname":"algo-1","model
_dir":"/opt/ml/model","module_dir":"s3://sagemaker-us-east-2-167428594774/sagemaker-xgboost-2022
-01-27-18-10-22-585/source/sourcedir.tar.gz","module_name":"xgb_train","network_interface_nam
e":"eth0","num_cpus":8,"num_gpus":0,"output_data_dir":"/opt/ml/output/data","output_dir":"/opt/m
l/output","output_intermediate_dir":"/opt/ml/output/intermediate","resource_config":{"current_ho
st":"algo-1","hosts":["algo-1"],"network_interface_name":"eth0"},"user_entry_point":"xgb_train.p
y"}
SM_USER_ARGS=["--alpha","952.2896578085882","--booster","gbtree","--eval_metric","error","--obje
ctive","binary:logistic"]
SM_OUTPUT_INTERMEDIATE_DIR=/opt/ml/output/intermediate
SM_CHANNEL_VALIDATION=/opt/ml/input/data/validation
SM_CHANNEL_TRAIN=/opt/ml/input/data/train
SM_HP_EVAL_METRIC=error
SM_HP_ALPHA=952.2896578085882
SM_HP_BOOSTER=gbtree
SM_HP_OBJECTIVE=binary:logistic
PYTHONPATH=/miniconda3/bin:/:/miniconda3/lib/python/site-packages/xgboost/dmlc-core/tracker:/min
iconda3/lib/python37.zip:/miniconda3/lib/python3.7:/miniconda3/lib/python3.7/lib-dynload:/minico
nda3/lib/python3.7/site-packages
Invoking script with the following command:
/miniconda3/bin/python3 -m xgb_train --alpha 952.2896578085882 --booster gbtree --eval_metric er
ror --objective binary:logistic
extracting arguments
Namespace(alpha=952.2896578085882, booster='gbtree', eval_metric='error', gamma=0, max_depth=6,
min_child_weight=1, model_dir='/opt/ml/model', n_estimators=15, objective='binary:logistic', sub
sample=1, train='/opt/ml/input/data/train', train_file='test-dataset.csv', validation='/opt/ml/i
nput/data/validation', validation_file='test-dataset.csv')
Preparing data
Fitting model
[0]#011validation_0-error:0.14414#011validation_1-error:0.14414
[1]#011validation_0-error:0.14414#011validation_1-error:0.14414
[2]#011validation_0-error:0.14414#011validation_1-error:0.14414
[3]#011validation_0-error:0.14414#011validation_1-error:0.14414
```

```
[4]#011validation_0-error:0.14414#011validation_1-error:0.14414
[5]#011validation_0-error:0.14414#011validation_1-error:0.14414
[6]#011validation_0-error:0.14414#011validation_1-error:0.14414
[7]#011validation_0-error:0.14414#011validation_1-error:0.14414
[8]#011validation_0-error:0.14414#011validation_1-error:0.14414
[9]#011validation_0-error:0.14414#011validation_1-error:0.14414
[10]#011validation_0-error:0.14414#011validation_1-error:0.14414
[11]#011validation_0-error:0.14414#011validation_1-error:0.14414
[12]#011validation_0-error:0.14414#011validation_1-error:0.14414
[13]#011validation_0-error:0.14414#011validation_1-error:0.14414
[14]#011validation_0-error:0.14414#011validation_1-error:0.14414
Evaluating model
[0]#011train-error:0.144144
[1]#011train-error:0.144144
[2]#011train-error:0.144144
[3]#011train-error:0.144144
[4]#011train-error:0.144144
[5]#011train-error:0.144144
[6]#011train-error:0.144144
[7]#011train-error:0.144144
[8]#011train-error:0.144144
[9]#011train-error:0.144144
[10]#011train-error:0.144144
[11]#011train-error:0.144144
[12]#011train-error:0.144144
[13]#011train-error:0.144144
[14]#011train-error:0.144144
[0]#011validation-error:0.144144
[1]#011validation-error:0.144144
[2]#011validation-error:0.144144
[3]#011validation-error:0.144144
[4]#011validation-error:0.144144
[5]#011validation-error:0.144144
[6]#011validation-error:0.144144
[7]#011validation-error:0.144144
[8]#011validation-error:0.144144
[9]#011validation-error:0.144144
[10]#011validation-error:0.144144
[11]#011validation-error:0.144144
[12]#011validation-error:0.144144
[13]#011validation-error:0.144144
[14]#011validation-error:0.144144
```

```
Saving model
Model saved to /opt/ml/model/xgboost-model.pkl
Training seconds: 98
```

Out[19]:
```
<sagemaker.xgboost.model.XGBoostPredictor at 0x7f2ef8ef84d0>
```

So, specifying thr `image_uri` when creating the estimator is a good work-around, but you shouldn't have to do this. One of the whole reasons we have higher-level classes is so that users don't need to mess around with containers. Why doesn't this work with just the framework information?

In other words, why is the call to `sagemaker.image_uris.retrieve` returning the wrong image tag? (`1.3-1` instead of `1.3-1-cpu-py3`)

In [20]:
```
sagemaker.image_uris.retrieve(
                framework="xgboost",
                region = "us-east-2",
                version="1.3-1",
                py_version="py3",
                instance_type="m1.,5.2xlarge",
                image_scope="training",
            )
```

Out[20]:
```
'257758044811.dkr.ecr.us-east-2.amazonaws.com/sagemaker-xgboost:1.3-1'
```

Looking at the function definition (https://github.com/aws/sagemaker-python-sdk/blob/d9e2567e86ece46100f9bea6d21cd1ed73105521/src/sagemaker/image_uris.py#L30 (https://github.com/aws/sagemaker-python-sdk/blob/d9e2567e86ece46100f9bea6d21cd1ed73105521/src/sagemaker/image_uris.py#L30)) shows a few things:

In [22]:
```
# (Some code redacted for clarity)

from sagemaker import utils

framework="xgboost"
region = "us-east-2"
version="1.3-1"
py_version="py3"
instance_type="m1.,5.2xlarge"
image_scope="training"
accelerator_type=None
container_version=None
distribution=None
training_compiler_config=None
tag_prefix=None
ECR_URI_TEMPLATE = "{registry}.dkr.{hostname}/{repository}"

if training_compiler_config is None:
    config = sagemaker.image_uris._config_for_framework_and_scope(framework, image_scope, accele
rator_type)

original_version = version
version = sagemaker.image_uris._validate_version_and_set_if_needed(version, config, framework)
version_config = config["versions"][sagemaker.image_uris._version_for_config(version, config)]
version_config = version_config.get(py_version) or version_config
processor = sagemaker.image_uris._processor(
        instance_type, config.get("processors") or version_config.get("processors")
    )
py_version = sagemaker.image_uris._validate_py_version_and_set_if_needed(py_version, version_con
fig, framework)
version_config = version_config.get(py_version) or version_config

tag = sagemaker.image_uris._format_tag(
    tag_prefix,
    processor,
    py_version,
    container_version,
)
print(f"Version is {version}")
print(f"Processor is {processor}")
print(f"Py Version is {py_version}")
```

```
print(f"Version Config is {version_config}")
print(f"Tag is {tag}")
```

```
Version is 1.3-1
Processor is None
Py Version is None
Version Config is {'registries': {'af-south-1': '510948584623', 'ap-east-1': '651117190479', 'ap
-northeast-1': '354813040037', 'ap-northeast-2': '366743142698', 'ap-northeast-3': '867004704886
', 'ap-south-1': '720646828776', 'ap-southeast-1': '121021644041', 'ap-southeast-2': '7833576542
85', 'ca-central-1': '341280168497', 'cn-north-1': '450853457545', 'cn-northwest-1': '4510491205
00', 'eu-central-1': '492215442770', 'eu-north-1': '662702820516', 'eu-west-1': '141502667606',
'eu-west-2': '764974769150', 'eu-west-3': '659782779980', 'eu-south-1': '978288397137', 'me-sout
h-1': '801668240914', 'sa-east-1': '737474898029', 'us-east-1': '683313688378', 'us-east-2': '25
7758044811', 'us-gov-west-1': '414596584902', 'us-iso-east-1': '833128469047', 'us-west-1': '746
614075791', 'us-west-2': '246618743249'}, 'repository': 'sagemaker-xgboost'}
Tag is
```

Because `processor` and `py_version` are set to `None`, the tag doesn't include them. Why are they set to none? Because both validation functions refer to a xgboost config object (Defined in https://github.com/aws/sagemaker-python-sdk/blob/dev/src/sagemaker/image_uri_config/xgboost.json (https://github.com/aws/sagemaker-python-sdk/blob/dev/src/sagemaker/image_uri_config/xgboost.json)). In this file, all XGBoost versions 1.0-1 and older have a `processors` and `py_versions` key defined. However, newer ones (including 1.3-1) do not! This means that the validation functions can't find them, which means that the `image_uris.retrieve` function returns the wrong image tag, which means that the `fw_utils.framework_version_from_tag` function can't parse it, which means that the HPO deploy commands don't know what image to use for inference, which means you can't deploy the best model from a HPO framework job without manually specifying the correct image URI!.

So, what's the solution? Update https://github.com/aws/sagemaker-python-sdk/blob/dev/src/sagemaker/image_uri_config/xgboost.json (https://github.com/aws/sagemaker-python-sdk/blob/dev/src/sagemaker/image_uri_config/xgboost.json) so that all version objects include the following items:

```
"processors": ["cpu"],
"py_versions": ["py3"]
```

Will that break something else? Don't know.... but it should fix this issue.

In [ ]: