ArupChakraborty_Assignment4.1

June 1, 2025

1 Amazon SageMaker Batch Transform: Associate prediction results with their corresponding input records

[4]:
```
!pip3 install -U sagemaker
```

```
Requirement already satisfied: sagemaker in /opt/conda/lib/python3.12/site-
packages (2.245.0)
Requirement already satisfied: attrs<24,>=23.1.0 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (23.2.0)
Requirement already satisfied: boto3<2.0,>=1.35.75 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (1.37.1)
Requirement already satisfied: cloudpickle>=2.2.1 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (3.1.1)
Requirement already satisfied: docker in /opt/conda/lib/python3.12/site-packages
(from sagemaker) (7.1.0)
Requirement already satisfied: fastapi in /opt/conda/lib/python3.12/site-
packages (from sagemaker) (0.115.12)
Requirement already satisfied: google-pasta in /opt/conda/lib/python3.12/site-
packages (from sagemaker) (0.2.0)
Requirement already satisfied: graphene<4,>=3 in /opt/conda/lib/python3.12/site-
packages (from sagemaker) (3.4.3)
Requirement already satisfied: importlib-metadata<7.0,>=1.4.0 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (6.10.0)
Requirement already satisfied: jsonschema in /opt/conda/lib/python3.12/site-
packages (from sagemaker) (4.23.0)
Requirement already satisfied: numpy==1.26.4 in /opt/conda/lib/python3.12/site-
packages (from sagemaker) (1.26.4)
Requirement already satisfied: omegaconf<3,>=2.2 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (2.3.0)
Requirement already satisfied: packaging<25,>=23.0 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (24.2)
Requirement already satisfied: pandas in /opt/conda/lib/python3.12/site-packages
(from sagemaker) (2.2.3)
Requirement already satisfied: pathos in /opt/conda/lib/python3.12/site-packages
(from sagemaker) (0.3.4)
Requirement already satisfied: platformdirs in /opt/conda/lib/python3.12/site-
packages (from sagemaker) (4.3.7)
```

```
Requirement already satisfied: protobuf<6.0,>=3.12 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (5.28.3)
Requirement already satisfied: psutil in /opt/conda/lib/python3.12/site-packages
(from sagemaker) (5.9.8)
Requirement already satisfied: pyyaml>=6.0.1 in /opt/conda/lib/python3.12/site-
packages (from sagemaker) (6.0.2)
Requirement already satisfied: requests in /opt/conda/lib/python3.12/site-
packages (from sagemaker) (2.32.3)
Requirement already satisfied: sagemaker-core<2.0.0,>=1.0.17 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (1.0.31)
Requirement already satisfied: schema in /opt/conda/lib/python3.12/site-packages
(from sagemaker) (0.7.7)
Requirement already satisfied: smdebug-rulesconfig==1.0.1 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (1.0.1)
Requirement already satisfied: tblib<4,>=1.7.0 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (3.1.0)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.12/site-packages
(from sagemaker) (4.67.1)
Requirement already satisfied: urllib3<3.0.0,>=1.26.8 in
/opt/conda/lib/python3.12/site-packages (from sagemaker) (2.4.0)
Requirement already satisfied: uvicorn in /opt/conda/lib/python3.12/site-
packages (from sagemaker) (0.34.2)
Requirement already satisfied: botocore<1.38.0,>=1.37.1 in
/opt/conda/lib/python3.12/site-packages (from boto3<2.0,>=1.35.75->sagemaker)
(1.37.1)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in
/opt/conda/lib/python3.12/site-packages (from boto3<2.0,>=1.35.75->sagemaker)
(1.0.1)
Requirement already satisfied: s3transfer<0.12.0,>=0.11.0 in
/opt/conda/lib/python3.12/site-packages (from boto3<2.0,>=1.35.75->sagemaker)
(0.11.3)
Requirement already satisfied: graphql-core<3.3,>=3.1 in
/opt/conda/lib/python3.12/site-packages (from graphene<4,>=3->sagemaker) (3.2.6)
Requirement already satisfied: graphql-relay<3.3,>=3.1 in
/opt/conda/lib/python3.12/site-packages (from graphene<4,>=3->sagemaker) (3.2.0)
Requirement already satisfied: python-dateutil<3,>=2.7.0 in
/opt/conda/lib/python3.12/site-packages (from graphene<4,>=3->sagemaker)
(2.9.0.post0)
Requirement already satisfied: typing-extensions<5,>=4.7.1 in
/opt/conda/lib/python3.12/site-packages (from graphene<4,>=3->sagemaker)
(4.13.2)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.12/site-
packages (from importlib-metadata<7.0,>=1.4.0->sagemaker) (3.21.0)
Requirement already satisfied: antlr4-python3-runtime==4.9.* in
/opt/conda/lib/python3.12/site-packages (from omegaconf<3,>=2.2->sagemaker)
(4.9.3)
Requirement already satisfied: pydantic<3.0.0,>=2.0.0 in
/opt/conda/lib/python3.12/site-packages (from sagemaker-
```

```
core<2.0.0,>=1.0.17->sagemaker) (2.11.3)
Requirement already satisfied: rich<14.0.0,>=13.0.0 in
/opt/conda/lib/python3.12/site-packages (from sagemaker-
core<2.0.0,>=1.0.17->sagemaker) (13.9.4)
Requirement already satisfied: mock<5.0,>4.0 in /opt/conda/lib/python3.12/site-
packages (from sagemaker-core<2.0.0,>=1.0.17->sagemaker) (4.0.3)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/opt/conda/lib/python3.12/site-packages (from jsonschema->sagemaker) (2025.4.1)
Requirement already satisfied: referencing>=0.28.4 in
/opt/conda/lib/python3.12/site-packages (from jsonschema->sagemaker) (0.36.2)
Requirement already satisfied: rpds-py>=0.7.1 in /opt/conda/lib/python3.12/site-
packages (from jsonschema->sagemaker) (0.24.0)
Requirement already satisfied: charset_normalizer<4,>=2 in
/opt/conda/lib/python3.12/site-packages (from requests->sagemaker) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.12/site-
packages (from requests->sagemaker) (3.10)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.12/site-packages (from requests->sagemaker) (2025.1.31)
Requirement already satisfied: starlette<0.47.0,>=0.40.0 in
/opt/conda/lib/python3.12/site-packages (from fastapi->sagemaker) (0.46.2)
Requirement already satisfied: six in /opt/conda/lib/python3.12/site-packages
(from google-pasta->sagemaker) (1.17.0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.12/site-
packages (from pandas->sagemaker) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.12/site-
packages (from pandas->sagemaker) (2025.2)
Requirement already satisfied: ppft>=1.7.7 in /opt/conda/lib/python3.12/site-
packages (from pathos->sagemaker) (1.7.7)
Requirement already satisfied: dill>=0.4.0 in /opt/conda/lib/python3.12/site-
packages (from pathos->sagemaker) (0.4.0)
Requirement already satisfied: pox>=0.3.6 in /opt/conda/lib/python3.12/site-
packages (from pathos->sagemaker) (0.3.6)
Requirement already satisfied: multiprocess>=0.70.18 in
/opt/conda/lib/python3.12/site-packages (from pathos->sagemaker) (0.70.18)
Requirement already satisfied: click>=7.0 in /opt/conda/lib/python3.12/site-
packages (from uvicorn->sagemaker) (8.1.8)
Requirement already satisfied: h11>=0.8 in /opt/conda/lib/python3.12/site-
packages (from uvicorn->sagemaker) (0.16.0)
Requirement already satisfied: annotated-types>=0.6.0 in
/opt/conda/lib/python3.12/site-packages (from pydantic<3.0.0,>=2.0.0->sagemaker-
core<2.0.0,>=1.0.17->sagemaker) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.1 in
/opt/conda/lib/python3.12/site-packages (from pydantic<3.0.0,>=2.0.0->sagemaker-
core<2.0.0,>=1.0.17->sagemaker) (2.33.1)
Requirement already satisfied: typing-inspection>=0.4.0 in
/opt/conda/lib/python3.12/site-packages (from pydantic<3.0.0,>=2.0.0->sagemaker-
core<2.0.0,>=1.0.17->sagemaker) (0.4.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in
```

```
/opt/conda/lib/python3.12/site-packages (from rich<14.0.0,>=13.0.0->sagemaker-
core<2.0.0,>=1.0.17->sagemaker) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/opt/conda/lib/python3.12/site-packages (from rich<14.0.0,>=13.0.0->sagemaker-
core<2.0.0,>=1.0.17->sagemaker) (2.19.1)
Requirement already satisfied: anyio<5,>=3.6.2 in
/opt/conda/lib/python3.12/site-packages (from
starlette<0.47.0,>=0.40.0->fastapi->sagemaker) (4.9.0)
Requirement already satisfied: sniffio>=1.1 in /opt/conda/lib/python3.12/site-
packages (from anyio<5,>=3.6.2->starlette<0.47.0,>=0.40.0->fastapi->sagemaker)
(1.3.1)
Requirement already satisfied: mdurl~=0.1 in /opt/conda/lib/python3.12/site-
packages (from markdown-it-py>=2.2.0->rich<14.0.0,>=13.0.0->sagemaker-
core<2.0.0,>=1.0.17->sagemaker) (0.1.2)
```

[5]:
```
import os
import boto3
import sagemaker

role = sagemaker.get_execution_role()
sess = sagemaker.Session()
region = sess.boto_region_name

bucket = sess.default_bucket()
prefix = "DEMO-breast-cancer-prediction-xgboost-highlevel"
```

```
sagemaker.config INFO - Not applying SDK defaults from location:
/etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location:
/home/sagemaker-user/.config/sagemaker/config.yaml
```

1.1 Data sources

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Breast Cancer Wisconsin (Diagnostic) Data Set [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wi

Also see: Breast Cancer Wisconsin (Diagnostic) Data Set [https://www.kaggle.com/uciml/breast-cancer-wisconsin-data].

1.2 Data preparation

Let's download the data and save it in the local folder with the name data.csv and take a look at it.

[6]:
```
import pandas as pd
import numpy as np

s3 = boto3.client("s3")

filename = "wdbc.csv"
s3.download_file(
    f"sagemaker-example-files-prod-{region}", "datasets/tabular/breast_cancer/
 ↪wdbc.csv", filename
)
data = pd.read_csv(filename, header=None)

# specify columns extracted from wbdc.names
data.columns = [
    "id",
    "diagnosis",
    "radius_mean",
    "texture_mean",
    "perimeter_mean",
    "area_mean",
    "smoothness_mean",
    "compactness_mean",
    "concavity_mean",
    "concave points_mean",
    "symmetry_mean",
    "fractal_dimension_mean",
    "radius_se",
    "texture_se",
    "perimeter_se",
    "area_se",
    "smoothness_se",
    "compactness_se",
    "concavity_se",
    "concave points_se",
    "symmetry_se",
    "fractal_dimension_se",
    "radius_worst",
    "texture_worst",
    "perimeter_worst",
    "area_worst",
    "smoothness_worst",
    "compactness_worst",
    "concavity_worst",
    "concave points_worst",
    "symmetry_worst",
    "fractal_dimension_worst",
]
```

```
# save the data
data.to_csv("data.csv", sep=",", index=False)

data.sample(8)
```

[6]:

```
           id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
24     852552         M        16.65         21.38          110.00      904.6
408  90524101         M        17.99         20.66          117.80      991.7
372   9012795         M        21.37         15.10          141.30     1386.0
278   8911800         B        13.59         17.84           86.24      572.3
366   9011494         M        20.20         26.83          133.70     1234.0
189    874839         B        12.30         15.90           78.83      463.7
258    887181         M        15.66         23.20          110.20      773.5
167   8712729         M        16.78         18.80          109.30      886.3

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
24           0.11210           0.14570         0.15250              0.09170
408          0.10360           0.13040         0.12010              0.08824
372          0.10010           0.15150         0.19320              0.12550
278          0.07948           0.04052         0.01997              0.01238
366          0.09905           0.16690         0.16410              0.12650
189          0.08080           0.07253         0.03844              0.01654
258          0.11090           0.31140         0.31760              0.13770
167          0.08865           0.09182         0.08422              0.06576

     …  radius_worst  texture_worst  perimeter_worst  area_worst  \
24   …         26.46          31.56           177.00      2215.0
408  …         21.08          25.41           138.10      1349.0
372  …         22.69          21.84           152.10      1535.0
278  …         15.50          26.10            98.91       739.1
366  …         24.19          33.81           160.00      1671.0
189  …         13.35          19.59            86.65       546.7
258  …         19.85          31.64           143.70      1226.0
167  …         20.05          26.30           130.70      1260.0

     smoothness_worst  compactness_worst  concavity_worst  \
24             0.1805            0.35780           0.4695
408            0.1482            0.37350           0.3301
372            0.1192            0.28400           0.4024
278            0.1050            0.07622           0.1060
366            0.1278            0.34160           0.3703
189            0.1096            0.16500           0.1423
258            0.1504            0.51720           0.6181
167            0.1168            0.21190           0.2318

     concave points_worst  symmetry_worst  fractal_dimension_worst
```

```
24                0.20950          0.3613                0.09564
408               0.19740          0.3060                0.08503
372               0.19660          0.2730                0.08666
278               0.05185          0.2335                0.06263
366               0.21520          0.3271                0.07632
189               0.04815          0.2482                0.06306
258               0.24620          0.3277                0.10190
167               0.14740          0.2810                0.07228

[8 rows x 32 columns]
```

Key observations:

- The data has 569 observations and 32 columns.
- The first field is the 'id' attribute that we will want to drop before batch inference and add to the final inference output next to the probability of malignancy.
- Second field, 'diagnosis', is an indicator of the actual diagnosis ('M' = Malignant; 'B' = Benign).
- There are 30 other numeric features that we will use for training and inferencing.

Let's replace the M/B diagnosis with a 1/0 boolean value.

[7]:
```
data["diagnosis"] = data["diagnosis"].apply(lambda x: ((x == "M")) + 0)
data.sample(8)
```

[7]:
```
           id  diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
455  9112085          0       13.380         30.72           86.34      557.2
103   862980          0        9.876         19.40           63.95      298.3
511   915664          0       14.810         14.70           94.66      680.7
27    852781          1       18.610         20.25          122.10     1094.0
105   863030          1       13.110         15.56           87.21      530.2
188   874662          0       11.810         17.39           75.27      428.9
163  8712064          0       12.340         22.22           79.85      464.5
78   8610862          1       20.180         23.97          143.70     1245.0

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
455          0.09245           0.07426         0.02819              0.03264
103          0.10050           0.09697         0.06154              0.03029
511          0.08472           0.05016         0.03416              0.02541
27           0.09440           0.10660         0.14900              0.07731
105          0.13980           0.17650         0.20710              0.09601
188          0.10070           0.05562         0.02353              0.01553
163          0.10120           0.10150         0.05370              0.02822
78           0.12860           0.34540         0.37540              0.16040

     …  radius_worst  texture_worst  perimeter_worst  area_worst  \
455  …         15.05          41.61            96.69       705.6
103  …         10.76          26.83            72.22       361.2
```

```
511  …         15.61          17.58           101.70       760.2
27   …         21.31          27.26           139.90      1403.0
105  …         16.31          22.40           106.40       827.2
188  …         12.57          26.48            79.57       489.5
163  …         13.58          28.68            87.36       553.0
78   …         23.37          31.72           170.30      1623.0

     smoothness_worst  compactness_worst  concavity_worst  \
455            0.1172             0.1421          0.07003
103            0.1559             0.2302          0.26440
511            0.1139             0.1011          0.11010
27             0.1338             0.2117          0.34460
105            0.1862             0.4099          0.63760
188            0.1356             0.1000          0.08803
163            0.1452             0.2338          0.16880
78             0.1639             0.6164          0.76810

     concave points_worst  symmetry_worst  fractal_dimension_worst
455               0.07763          0.2196                  0.07675
103               0.09749          0.2622                  0.08490
511               0.07955          0.2334                  0.06142
27                0.14900          0.2341                  0.07421
105               0.19860          0.3147                  0.14050
188               0.04306          0.3200                  0.06576
163               0.08194          0.2268                  0.09082
78                0.25080          0.5440                  0.09964

[8 rows x 32 columns]
```

Let's split the data as follows: 80% for training, 10% for validation and let's set 10% aside for our batch inference job. In addition, let's drop the 'id' field on the training set and validation set as 'id' is not a training feature. For our batch set however, we keep the 'id' feature. We'll want to filter it out prior to running our inferences so that the input data features match the ones of training set and then ultimately, we'll want to join it with inference result. We are however dropping the diagnosis attribute for the batch set since this is what we'll try to predict.

[8]:
```
# data split in three sets, training, validation and batch inference
rand_split = np.random.rand(len(data))
train_list = rand_split < 0.8
val_list = (rand_split >= 0.8) & (rand_split < 0.9)
batch_list = rand_split >= 0.9

data_train = data[train_list].drop(["id"], axis=1)
data_val = data[val_list].drop(["id"], axis=1)
data_batch = data[batch_list].drop(["diagnosis"], axis=1)
data_batch_noID = data_batch.drop(["id"], axis=1)
```

Let's upload those data sets in S3

[9]:
```
train_file = "train_data.csv"
data_train.to_csv(train_file, index=False, header=False)
sess.upload_data(train_file, key_prefix="{}/train".format(prefix))

validation_file = "validation_data.csv"
data_val.to_csv(validation_file, index=False, header=False)
sess.upload_data(validation_file, key_prefix="{}/validation".format(prefix))

batch_file = "batch_data.csv"
data_batch.to_csv(batch_file, index=False, header=False)
sess.upload_data(batch_file, key_prefix="{}/batch".format(prefix))

batch_file_noID = "batch_data_noID.csv"
data_batch_noID.to_csv(batch_file_noID, index=False, header=False)
sess.upload_data(batch_file_noID, key_prefix="{}/batch".format(prefix))
```

[9]:
```
's3://sagemaker-us-east-1-672518276407/DEMO-breast-cancer-prediction-xgboost-
highlevel/batch/batch_data_noID.csv'
```

1.3 Training job and model creation

The below cell uses the SageMaker Python SDK to kick off the training job using both our training set and validation set. Not that the objective is set to 'binary:logistic' which trains a model to output a probability between 0 and 1 (here the probability of a tumor being malignant).

[10]:
```
%%time
from time import gmtime, strftime

job_name = "xgb-" + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
output_location = "s3://{}/{}/output/{}".format(bucket, prefix, job_name)
image = sagemaker.image_uris.retrieve(
    framework="xgboost", region=boto3.Session().region_name, version="1.7-1"
)

sm_estimator = sagemaker.estimator.Estimator(
    image,
    role,
    instance_count=1,
    instance_type="ml.m5.xlarge",
    volume_size=50,
    input_mode="File",
    output_path=output_location,
    sagemaker_session=sess,
)

sm_estimator.set_hyperparameters(
```

```
    objective="binary:logistic",
    max_depth=5,
    eta=0.2,
    gamma=4,
    min_child_weight=6,
    subsample=0.8,
    verbosity=0,
    num_round=100,
)

train_data = sagemaker.inputs.TrainingInput(
    "s3://{}/{}/train".format(bucket, prefix),
    distribution="FullyReplicated",
    content_type="text/csv",
    s3_data_type="S3Prefix",
)
validation_data = sagemaker.inputs.TrainingInput(
    "s3://{}/{}/validation".format(bucket, prefix),
    distribution="FullyReplicated",
    content_type="text/csv",
    s3_data_type="S3Prefix",
)
data_channels = {"train": train_data, "validation": validation_data}

# Start training by calling the fit method in the estimator
sm_estimator.fit(inputs=data_channels, job_name=job_name, logs=True)
```

```
INFO:sagemaker:Creating training-job with name: xgb-2025-06-01-06-11-34

2025-06-01 06:11:37 Starting - Starting the training job…
2025-06-01 06:11:51 Starting - Preparing the instances for training…
2025-06-01 06:12:33 Downloading - Downloading the training image…
2025-06-01 06:13:40 Training - Training image download completed. Training in
progress.
2025-06-01 06:13:40 Uploading - Uploading generated training
model.[2025-06-01 06:13:35.865 ip-10-0-230-145.ec2.internal:7 INFO

utils.py:28] RULE_JOB_STOP_SIGNAL_FILENAME: None
[2025-06-01 06:13:35.888 ip-10-0-230-145.ec2.internal:7 INFO

profiler_config_parser.py:111] User has disabled profiler.
[2025-06-01:06:13:36:INFO] Imported framework

sagemaker_xgboost_container.training
[2025-06-01:06:13:36:INFO] Failed to parse hyperparameter objective value

binary:logistic to Json.
Returning the value itself
```

```
[2025-06-01:06:13:36:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:13:36:INFO] Running XGBoost Sagemaker in algorithm mode
[2025-06-01:06:13:36:INFO] Determined 0 GPU(s) available on the
instance.
[2025-06-01:06:13:36:INFO] Determined delimiter of CSV input is ','
[2025-06-01:06:13:36:INFO] Determined delimiter of CSV input is ','
[2025-06-01:06:13:36:INFO] File path /opt/ml/input/data/train of input
files
[2025-06-01:06:13:36:INFO] Making smlinks from folder
/opt/ml/input/data/train to folder /tmp/sagemaker_xgboost_input_data
[2025-06-01:06:13:36:INFO] creating symlink between Path
/opt/ml/input/data/train/train_data.csv and destination
/tmp/sagemaker_xgboost_input_data/train_data.csv2161540189430046313
[2025-06-01:06:13:36:INFO] files path:
/tmp/sagemaker_xgboost_input_data
[2025-06-01:06:13:36:INFO] Determined delimiter of CSV input is ','
[2025-06-01:06:13:36:INFO] File path /opt/ml/input/data/validation of input
files
[2025-06-01:06:13:36:INFO] Making smlinks from folder
/opt/ml/input/data/validation to folder /tmp/sagemaker_xgboost_input_data
[2025-06-01:06:13:36:INFO] creating symlink between Path
/opt/ml/input/data/validation/validation_data.csv and destination
/tmp/sagemaker_xgboost_input_data/validation_data.csv-5955050547540261129
[2025-06-01:06:13:36:INFO] files path:
/tmp/sagemaker_xgboost_input_data
[2025-06-01:06:13:36:INFO] Determined delimiter of CSV input is ','
[2025-06-01:06:13:36:INFO] Single node training.
[2025-06-01:06:13:36:INFO] Train matrix has 457 rows and 30 columns
[2025-06-01:06:13:36:INFO] Validation matrix has 60 rows
[2025-06-01 06:13:36.312 ip-10-0-230-145.ec2.internal:7 INFO
json_config.py:92] Creating hook from json_config at
/opt/ml/input/config/debughookconfig.json.
[2025-06-01 06:13:36.312 ip-10-0-230-145.ec2.internal:7 INFO hook.py:206]
tensorboard_dir has not been set for the hook. SMDebug will not be exporting
tensorboard summaries.
[2025-06-01 06:13:36.313 ip-10-0-230-145.ec2.internal:7 INFO hook.py:259]
Saving to /opt/ml/output/tensors
```

```
[2025-06-01 06:13:36.313 ip-10-0-230-145.ec2.internal:7 INFO
state_store.py:77] The checkpoint config file
/opt/ml/input/config/checkpointconfig.json does not exist.
[2025-06-01:06:13:36:INFO] Debug hook created from config
[2025-06-01 06:13:36.317 ip-10-0-230-145.ec2.internal:7 INFO hook.py:427]
Monitoring the collections: metrics
[2025-06-01 06:13:36.322 ip-10-0-230-145.ec2.internal:7 INFO hook.py:491]
Hook is writing from the hook with pid: 7
[0]#011train-logloss:0.54611#011validation-logloss:0.55917
[1]#011train-logloss:0.44766#011validation-logloss:0.47712
[2]#011train-logloss:0.38059#011validation-logloss:0.41633
[3]#011train-logloss:0.32446#011validation-logloss:0.37017
[4]#011train-logloss:0.27831#011validation-logloss:0.34023
[5]#011train-logloss:0.24352#011validation-logloss:0.31670
[6]#011train-logloss:0.21547#011validation-logloss:0.29658
[7]#011train-logloss:0.19209#011validation-logloss:0.27445
[8]#011train-logloss:0.17297#011validation-logloss:0.25836
[9]#011train-logloss:0.15708#011validation-logloss:0.24525
[10]#011train-logloss:0.14513#011validation-logloss:0.24271
[11]#011train-logloss:0.13414#011validation-logloss:0.23880
[12]#011train-logloss:0.12379#011validation-logloss:0.23306
[13]#011train-logloss:0.11496#011validation-logloss:0.23144
[14]#011train-logloss:0.10642#011validation-logloss:0.22973
[15]#011train-logloss:0.10015#011validation-logloss:0.23000
[16]#011train-logloss:0.09344#011validation-logloss:0.22470
[17]#011train-logloss:0.09038#011validation-logloss:0.22715
[18]#011train-logloss:0.08738#011validation-logloss:0.22621
[19]#011train-logloss:0.08278#011validation-logloss:0.22324
[20]#011train-logloss:0.07996#011validation-logloss:0.22027
[21]#011train-logloss:0.07761#011validation-logloss:0.21928
[22]#011train-logloss:0.07570#011validation-logloss:0.22093
[23]#011train-logloss:0.07399#011validation-logloss:0.21968
[24]#011train-logloss:0.07238#011validation-logloss:0.21771
[25]#011train-logloss:0.07240#011validation-logloss:0.21797
[26]#011train-logloss:0.07239#011validation-logloss:0.21774
[27]#011train-logloss:0.07238#011validation-logloss:0.21762
[28]#011train-logloss:0.07239#011validation-logloss:0.21775
[29]#011train-logloss:0.07241#011validation-logloss:0.21807
[30]#011train-logloss:0.07240#011validation-logloss:0.21794
[31]#011train-logloss:0.07239#011validation-logloss:0.21784
[32]#011train-logloss:0.07101#011validation-logloss:0.21416
[33]#011train-logloss:0.07102#011validation-logloss:0.21426
[34]#011train-logloss:0.07100#011validation-logloss:0.21407
[35]#011train-logloss:0.07101#011validation-logloss:0.21412
[36]#011train-logloss:0.07101#011validation-logloss:0.21411
```

[37]#011train-logloss:0.07099#011validation-logloss:0.21391
[38]#011train-logloss:0.07099#011validation-logloss:0.21388
[39]#011train-logloss:0.07099#011validation-logloss:0.21364
[40]#011train-logloss:0.07100#011validation-logloss:0.21406
[41]#011train-logloss:0.07100#011validation-logloss:0.21405
[42]#011train-logloss:0.07100#011validation-logloss:0.21407
[43]#011train-logloss:0.07102#011validation-logloss:0.21424
[44]#011train-logloss:0.07099#011validation-logloss:0.21392
[45]#011train-logloss:0.07099#011validation-logloss:0.21379
[46]#011train-logloss:0.07099#011validation-logloss:0.21383
[47]#011train-logloss:0.07099#011validation-logloss:0.21382
[48]#011train-logloss:0.07099#011validation-logloss:0.21372
[49]#011train-logloss:0.07099#011validation-logloss:0.21383
[50]#011train-logloss:0.07099#011validation-logloss:0.21378
[51]#011train-logloss:0.07099#011validation-logloss:0.21376
[52]#011train-logloss:0.07100#011validation-logloss:0.21351
[53]#011train-logloss:0.06935#011validation-logloss:0.21342
[54]#011train-logloss:0.06935#011validation-logloss:0.21367
[55]#011train-logloss:0.06935#011validation-logloss:0.21351
[56]#011train-logloss:0.06935#011validation-logloss:0.21348
[57]#011train-logloss:0.06935#011validation-logloss:0.21352
[58]#011train-logloss:0.06936#011validation-logloss:0.21370
[59]#011train-logloss:0.06935#011validation-logloss:0.21359
[60]#011train-logloss:0.06936#011validation-logloss:0.21327
[61]#011train-logloss:0.06935#011validation-logloss:0.21343
[62]#011train-logloss:0.06935#011validation-logloss:0.21337
[63]#011train-logloss:0.06935#011validation-logloss:0.21363
[64]#011train-logloss:0.06935#011validation-logloss:0.21361
[65]#011train-logloss:0.06935#011validation-logloss:0.21356
[66]#011train-logloss:0.06935#011validation-logloss:0.21353
[67]#011train-logloss:0.06935#011validation-logloss:0.21359
[68]#011train-logloss:0.06807#011validation-logloss:0.21138
[69]#011train-logloss:0.06807#011validation-logloss:0.21142
[70]#011train-logloss:0.06807#011validation-logloss:0.21130
[71]#011train-logloss:0.06808#011validation-logloss:0.21107
[72]#011train-logloss:0.06808#011validation-logloss:0.21099
[73]#011train-logloss:0.06808#011validation-logloss:0.21106
[74]#011train-logloss:0.06808#011validation-logloss:0.21154
[75]#011train-logloss:0.06808#011validation-logloss:0.21158
[76]#011train-logloss:0.06810#011validation-logloss:0.21172
[77]#011train-logloss:0.06808#011validation-logloss:0.21161
[78]#011train-logloss:0.06809#011validation-logloss:0.21168
[79]#011train-logloss:0.06807#011validation-logloss:0.21143
[80]#011train-logloss:0.06811#011validation-logloss:0.21182
[81]#011train-logloss:0.06811#011validation-logloss:0.21186
[82]#011train-logloss:0.06811#011validation-logloss:0.21182
[83]#011train-logloss:0.06811#011validation-logloss:0.21181
[84]#011train-logloss:0.06811#011validation-logloss:0.21185

```
[85]#011train-logloss:0.06815#011validation-logloss:0.21211
[86]#011train-logloss:0.06813#011validation-logloss:0.21199
[87]#011train-logloss:0.06810#011validation-logloss:0.21171
[88]#011train-logloss:0.06807#011validation-logloss:0.21148
[89]#011train-logloss:0.06807#011validation-logloss:0.21120
[90]#011train-logloss:0.06807#011validation-logloss:0.21127
[91]#011train-logloss:0.06807#011validation-logloss:0.21147
[92]#011train-logloss:0.06807#011validation-logloss:0.21142
[93]#011train-logloss:0.06807#011validation-logloss:0.21125
[94]#011train-logloss:0.06807#011validation-logloss:0.21125
[95]#011train-logloss:0.06807#011validation-logloss:0.21109
[96]#011train-logloss:0.06807#011validation-logloss:0.21108
[97]#011train-logloss:0.06807#011validation-logloss:0.21121
[98]#011train-logloss:0.06807#011validation-logloss:0.21119
[99]#011train-logloss:0.06807#011validation-logloss:0.21124

2025-06-01 06:13:53 Completed - Training job completed
Training seconds: 99
Billable seconds: 99
CPU times: user 393 ms, sys: 42.7 ms, total: 436 ms
Wall time: 2min 46s
```

1.4 Batch Transform

In SageMaker Batch Transform, we introduced 3 new attributes - **input_filter**, **join_source** and **output_filter**. In the below cell, we use the SageMaker Python SDK to kick-off several Batch Transform jobs using different configurations of these 3 new attributes. Please refer to this page to learn more about how to use them.

1. Create a transform job with the default configurations Let's first skip these 3 new attributes and inspect the inference results. We'll use it as a baseline to compare to the results with data processing.

[11]:
```
%%time

sm_transformer = sm_estimator.transformer(1, "ml.m5.xlarge")

# start a transform job
input_location = "s3://{}/{}/batch/{}".format(
    bucket, prefix, batch_file_noID
)  # use input data without ID column
sm_transformer.transform(input_location, content_type="text/csv",␣
 ↪split_type="Line")
sm_transformer.wait()
```

```
INFO:sagemaker:Creating model with name: sagemaker-
xgboost-2025-06-01-06-14-21-162
```

```
INFO:sagemaker:Creating transform job with name: sagemaker-
xgboost-2025-06-01-06-14-21-928
…[2025-06-01:06:19:37:INFO] No GPUs detected
(normal if no gpus installed)
[2025-06-01:06:19:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:37:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
```

```
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2025-06-01 06:19:37 +0000] [18] [INFO] Starting gunicorn 19.10.0
[2025-06-01 06:19:37 +0000] [18] [INFO] Listening at:
unix:/tmp/gunicorn.sock (18)
[2025-06-01 06:19:37 +0000] [18] [INFO] Using worker: gevent
/miniconda3/lib/python3.9/os.py:1023: RuntimeWarning: line buffering
(buffering=1) isn't supported in binary mode, the default buffer size will be
used
  return io.open(fd, *args, **kwargs)
[2025-06-01 06:19:37 +0000] [23] [INFO] Booting worker with pid: 23
[2025-06-01 06:19:37 +0000] [24] [INFO] Booting worker with pid: 24
[2025-06-01 06:19:38 +0000] [25] [INFO] Booting worker with pid: 25
[2025-06-01 06:19:38 +0000] [26] [INFO] Booting worker with pid: 26
```

```
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
```

```
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:43:INFO] Determined delimiter of CSV input is ','
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
[2025-06-01:06:19:43:INFO] Determined delimiter of CSV input is ','
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
2025-06-01T06:19:43.539:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD

[2025-06-01:06:19:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:37:INFO] nginx config:
worker_processes auto;
[2025-06-01:06:19:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:37:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
```

```
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2025-06-01 06:19:37 +0000] [18] [INFO] Starting gunicorn 19.10.0
[2025-06-01 06:19:37 +0000] [18] [INFO] Listening at:
unix:/tmp/gunicorn.sock (18)
[2025-06-01 06:19:37 +0000] [18] [INFO] Using worker: gevent
/miniconda3/lib/python3.9/os.py:1023: RuntimeWarning: line buffering
(buffering=1) isn't supported in binary mode, the default buffer size will be
used
  return io.open(fd, *args, **kwargs)
[2025-06-01 06:19:37 +0000] [23] [INFO] Booting worker with pid: 23
[2025-06-01 06:19:37 +0000] [24] [INFO] Booting worker with pid: 24
daemon off;
pid /tmp/nginx.pid;
```

```
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2025-06-01 06:19:37 +0000] [18] [INFO] Starting gunicorn 19.10.0
[2025-06-01 06:19:37 +0000] [18] [INFO] Listening at:
unix:/tmp/gunicorn.sock (18)
[2025-06-01 06:19:37 +0000] [18] [INFO] Using worker: gevent
```

```
/miniconda3/lib/python3.9/os.py:1023: RuntimeWarning: line buffering
(buffering=1) isn't supported in binary mode, the default buffer size will be
used
  return io.open(fd, *args, **kwargs)
[2025-06-01 06:19:37 +0000] [23] [INFO] Booting worker with pid: 23
[2025-06-01 06:19:37 +0000] [24] [INFO] Booting worker with pid: 24
[2025-06-01 06:19:38 +0000] [25] [INFO] Booting worker with pid: 25
[2025-06-01 06:19:38 +0000] [26] [INFO] Booting worker with pid: 26
[2025-06-01 06:19:38 +0000] [25] [INFO] Booting worker with pid: 25
[2025-06-01 06:19:38 +0000] [26] [INFO] Booting worker with pid: 26
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
```

```
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:40:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:40:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:19:40:INFO] Model objective : binary:logistic
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2025-06-01:06:19:43:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:19:43:INFO] Determined delimiter of CSV input is ','
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
[2025-06-01:06:19:43:INFO] Determined delimiter of CSV input is ','
```

```
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:19:43 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
2025-06-01T06:19:43.539:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
CPU times: user 670 ms, sys: 72 ms, total: 742 ms
Wall time: 6min 4s
```

Let's inspect the output of the Batch Transform job in S3. It should show the list probabilities of tumors being malignant.

[12]:
```
import re


def get_csv_output_from_s3(s3uri, batch_file):
    file_name = "{}.out".format(batch_file)
    match = re.match("s3://([^/]+)/(.*)", "{}/{}".format(s3uri, file_name))
    output_bucket, output_prefix = match.group(1), match.group(2)
    s3.download_file(output_bucket, output_prefix, file_name)
    return pd.read_csv(file_name, sep=",", header=None)
```

[13]:
```
output_df = get_csv_output_from_s3(sm_transformer.output_path, batch_file_noID)
output_df.head(8)
```

[13]:

	0
0	0.902768
1	0.933148
2	0.903114
3	0.987099
4	0.989131
5	0.944575
6	0.993661
7	0.993661

2. Join the input and the prediction results Now, let's associate the prediction results with their corresponding input records. We can also use the **input_filter** to exclude the ID column easily and there's no need to have a separate file in S3.

- Set **input_filter** to "$[1:]": indicates that we are excluding column 0 (the 'ID') before processing the inferences and keeping everything from column 1 to the last column (all the features or predictors)
- Set **join_source** to "Input": indicates our desire to join the input data with the inference results

- Leave **output_filter** to default ('$'), indicating that the joined input and inference results be will saved as output.

[14]:
```
# content_type / accept and split_type / assemble_with are required to use IO joining feature
sm_transformer.assemble_with = "Line"
sm_transformer.accept = "text/csv"

# start a transform job
input_location = "s3://{}/{}/batch/{}".format(
    bucket, prefix, batch_file
)  # use input data with ID column cause InputFilter will filter it out
sm_transformer.transform(
    input_location,
    split_type="Line",
    content_type="text/csv",
    input_filter="$[1:]",
    join_source="Input",
)
sm_transformer.wait()
```

```
INFO:sagemaker:Creating transform job with name: sagemaker-
xgboost-2025-06-01-06-20-25-618
```

```
…[2025-06-01:06:26:13:INFO] No GPUs

detected (normal if no gpus installed)
[2025-06-01:06:26:14:INFO] No GPUs detected (normal if no gpus

installed)
[2025-06-01:06:26:14:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {

  worker_connections 2048;
}
```

```
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2025-06-01 06:26:14 +0000] [18] [INFO] Starting gunicorn 19.10.0
[2025-06-01 06:26:14 +0000] [18] [INFO] Listening at:
unix:/tmp/gunicorn.sock (18)
[2025-06-01 06:26:14 +0000] [18] [INFO] Using worker: gevent
/miniconda3/lib/python3.9/os.py:1023: RuntimeWarning: line buffering
(buffering=1) isn't supported in binary mode, the default buffer size will be
used
  return io.open(fd, *args, **kwargs)
[2025-06-01 06:26:14 +0000] [23] [INFO] Booting worker with pid: 23
[2025-06-01 06:26:14 +0000] [24] [INFO] Booting worker with pid: 24
[2025-06-01 06:26:14 +0000] [25] [INFO] Booting worker with pid: 25
[2025-06-01 06:26:14 +0000] [26] [INFO] Booting worker with pid: 26
```

```
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:19:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:26:19 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2025-06-01:06:26:19:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:26:19 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2025-06-01:06:26:19:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:19:INFO] Determined delimiter of CSV input is ','
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:26:19 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
```

```
2025-06-01T06:26:19.190:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
[2025-06-01:06:26:13:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:14:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:14:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
```

```
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2025-06-01 06:26:14 +0000] [18] [INFO] Starting gunicorn 19.10.0
[2025-06-01 06:26:14 +0000] [18] [INFO] Listening at:
unix:/tmp/gunicorn.sock (18)
[2025-06-01 06:26:14 +0000] [18] [INFO] Using worker: gevent
[2025-06-01:06:26:13:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:14:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:14:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
```

```
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2025-06-01 06:26:14 +0000] [18] [INFO] Starting gunicorn 19.10.0
[2025-06-01 06:26:14 +0000] [18] [INFO] Listening at:
unix:/tmp/gunicorn.sock (18)
[2025-06-01 06:26:14 +0000] [18] [INFO] Using worker: gevent
```

```
/miniconda3/lib/python3.9/os.py:1023: RuntimeWarning: line buffering
(buffering=1) isn't supported in binary mode, the default buffer size will be
used
  return io.open(fd, *args, **kwargs)
[2025-06-01 06:26:14 +0000] [23] [INFO] Booting worker with pid: 23
[2025-06-01 06:26:14 +0000] [24] [INFO] Booting worker with pid: 24
[2025-06-01 06:26:14 +0000] [25] [INFO] Booting worker with pid: 25
[2025-06-01 06:26:14 +0000] [26] [INFO] Booting worker with pid: 26
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
/miniconda3/lib/python3.9/os.py:1023: RuntimeWarning: line buffering
(buffering=1) isn't supported in binary mode, the default buffer size will be
used
  return io.open(fd, *args, **kwargs)
[2025-06-01 06:26:14 +0000] [23] [INFO] Booting worker with pid: 23
[2025-06-01 06:26:14 +0000] [24] [INFO] Booting worker with pid: 24
[2025-06-01 06:26:14 +0000] [25] [INFO] Booting worker with pid: 25
[2025-06-01 06:26:14 +0000] [26] [INFO] Booting worker with pid: 26
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
```

```
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:16:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:16:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:26:16:INFO] Model objective : binary:logistic
[2025-06-01:06:26:19:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:26:19 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2025-06-01:06:26:19:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:26:19 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2025-06-01:06:26:19:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:19:INFO] Determined delimiter of CSV input is ','
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:26:19 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
[2025-06-01:06:26:19:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:26:19 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
```

```
[2025-06-01:06:26:19:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:26:19 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2025-06-01:06:26:19:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:26:19:INFO] Determined delimiter of CSV input is ','
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:26:19 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
2025-06-01T06:26:19.190:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
```

Let's inspect the output of the Batch Transform job in S3. It should show the list of tumors identified by their original feature columns and their corresponding probabilities of being malignant.

[15]:
```
output_df = get_csv_output_from_s3(sm_transformer.output_path, batch_file)
output_df.head(8)
```

[15]:

	0	1	2	3	4	5	6	7	8	\
0	842517	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	
1	84358402	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	
2	84501001	12.46	24.04	83.97	475.9	0.11860	0.23960	0.22730	0.08543	
3	84610002	15.78	17.89	103.60	781.0	0.09710	0.12920	0.09954	0.06606	
4	848406	14.68	20.13	94.74	684.5	0.09867	0.07200	0.07395	0.05259	
5	8511133	15.34	14.26	102.50	704.4	0.10730	0.21350	0.20770	0.09756	
6	854253	16.74	21.59	110.10	869.5	0.09610	0.13360	0.13480	0.06018	
7	858986	14.25	22.15	96.42	645.7	0.10490	0.20080	0.21350	0.08653	

	9	…	22	23	24	25	26	27	28	29	\
0	0.1812	…	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	
1	0.1809	…	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	
2	0.2030	…	40.68	97.65	711.4	0.1853	1.0580	1.1050	0.2210	0.4366	
3	0.1842	…	27.28	136.50	1299.0	0.1396	0.5609	0.3965	0.1810	0.3792	
4	0.1586	…	30.88	123.40	1138.0	0.1464	0.1871	0.2914	0.1609	0.3029	
5	0.2521	…	19.08	125.10	980.9	0.1390	0.5954	0.6305	0.2393	0.4667	
6	0.1896	…	29.02	133.50	1229.0	0.1563	0.3835	0.5409	0.1813	0.4863	
7	0.1949	…	29.51	119.10	959.5	0.1640	0.6247	0.6922	0.1785	0.2844	

	30	31
0	0.08902	0.902768
1	0.07678	0.933148

```
2  0.20750  0.903114
3  0.10480  0.987099
4  0.08216  0.989131
5  0.09946  0.944575
6  0.08633  0.993661
7  0.11320  0.993661

[8 rows x 32 columns]
```

3. Update the output filter to keep only ID and prediction results Let's change **output_filter** to "$[0,-1]", indicating that when presenting the output, we only want to keep column 0 (the 'ID') and the last column (the inference result i.e. the probability of a given tumor to be malignant)

[16]:
```
# start another transform job
sm_transformer.transform(
    input_location,
    split_type="Line",
    content_type="text/csv",
    input_filter="$[1:]",
    join_source="Input",
    output_filter="$[0,-1]",
)
sm_transformer.wait()
```

```
INFO:sagemaker:Creating transform job with name: sagemaker-
xgboost-2025-06-01-06-26-59-503

…[2025-06-01:06:31:35:INFO] No GPUs detected

(normal if no gpus installed)
[2025-06-01:06:31:35:INFO] No GPUs detected (normal if no gpus

installed)
[2025-06-01:06:31:35:INFO] No GPUs detected (normal if no gpus

installed)
[2025-06-01:06:31:35:INFO] No GPUs detected (normal if no gpus

installed)
[2025-06-01:06:31:35:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {

  worker_connections 2048;
}
```

```
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2025-06-01 06:31:35 +0000] [17] [INFO] Starting gunicorn 19.10.0
[2025-06-01 06:31:35 +0000] [17] [INFO] Listening at:
unix:/tmp/gunicorn.sock (17)
[2025-06-01 06:31:35 +0000] [17] [INFO] Using worker: gevent
[2025-06-01:06:31:35:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
```

```
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2025-06-01 06:31:35 +0000] [17] [INFO] Starting gunicorn 19.10.0
[2025-06-01 06:31:35 +0000] [17] [INFO] Listening at:
unix:/tmp/gunicorn.sock (17)
[2025-06-01 06:31:35 +0000] [17] [INFO] Using worker: gevent
/miniconda3/lib/python3.9/os.py:1023: RuntimeWarning: line buffering
(buffering=1) isn't supported in binary mode, the default buffer size will be
used
  return io.open(fd, *args, **kwargs)
[2025-06-01 06:31:35 +0000] [22] [INFO] Booting worker with pid: 22
[2025-06-01 06:31:35 +0000] [23] [INFO] Booting worker with pid: 23
[2025-06-01 06:31:35 +0000] [24] [INFO] Booting worker with pid: 24
[2025-06-01 06:31:35 +0000] [25] [INFO] Booting worker with pid: 25
```

```
/miniconda3/lib/python3.9/os.py:1023: RuntimeWarning: line buffering
(buffering=1) isn't supported in binary mode, the default buffer size will be
used
  return io.open(fd, *args, **kwargs)
[2025-06-01 06:31:35 +0000] [22] [INFO] Booting worker with pid: 22
[2025-06-01 06:31:35 +0000] [23] [INFO] Booting worker with pid: 23
[2025-06-01 06:31:35 +0000] [24] [INFO] Booting worker with pid: 24
[2025-06-01 06:31:35 +0000] [25] [INFO] Booting worker with pid: 25
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
```

```
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:40:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2025-06-01:06:31:40:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2025-06-01:06:31:40:INFO] Determined delimiter of CSV input is ','
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
[2025-06-01:06:31:40:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2025-06-01:06:31:40:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2025-06-01:06:31:40:INFO] Determined delimiter of CSV input is ','
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
```

```
2025-06-01T06:31:40.404:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD

[2025-06-01:06:31:35:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:35:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:35:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:35:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:35:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
```

```
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2025-06-01 06:31:35 +0000] [17] [INFO] Starting gunicorn 19.10.0
[2025-06-01 06:31:35 +0000] [17] [INFO] Listening at:
unix:/tmp/gunicorn.sock (17)
[2025-06-01 06:31:35 +0000] [17] [INFO] Using worker: gevent
[2025-06-01:06:31:35:INFO] nginx config:
worker_processes auto;
daemon off;
pid /tmp/nginx.pid;
error_log  /dev/stderr;
worker_rlimit_nofile 4096;
events {
  worker_connections 2048;
}
```

```
http {
  include /etc/nginx/mime.types;
  default_type application/octet-stream;
  access_log /dev/stdout combined;
  upstream gunicorn {
    server unix:/tmp/gunicorn.sock;
  }
  server {
    listen 8080 deferred;
    client_max_body_size 0;
    keepalive_timeout 3;
    location ~ ^/(ping|invocations|execution-parameters) {
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_redirect off;
      proxy_read_timeout 60s;
      proxy_pass http://gunicorn;
    }
    location / {
      return 404 "{}";
    }
  }
}
[2025-06-01 06:31:35 +0000] [17] [INFO] Starting gunicorn 19.10.0
[2025-06-01 06:31:35 +0000] [17] [INFO] Listening at:
unix:/tmp/gunicorn.sock (17)
[2025-06-01 06:31:35 +0000] [17] [INFO] Using worker: gevent
/miniconda3/lib/python3.9/os.py:1023: RuntimeWarning: line buffering
(buffering=1) isn't supported in binary mode, the default buffer size will be
used
  return io.open(fd, *args, **kwargs)
[2025-06-01 06:31:35 +0000] [22] [INFO] Booting worker with pid: 22
[2025-06-01 06:31:35 +0000] [23] [INFO] Booting worker with pid: 23
[2025-06-01 06:31:35 +0000] [24] [INFO] Booting worker with pid: 24
[2025-06-01 06:31:35 +0000] [25] [INFO] Booting worker with pid: 25
```

```
/miniconda3/lib/python3.9/os.py:1023: RuntimeWarning: line buffering
(buffering=1) isn't supported in binary mode, the default buffer size will be
used
  return io.open(fd, *args, **kwargs)
[2025-06-01 06:31:35 +0000] [22] [INFO] Booting worker with pid: 22
[2025-06-01 06:31:35 +0000] [23] [INFO] Booting worker with pid: 23
[2025-06-01 06:31:35 +0000] [24] [INFO] Booting worker with pid: 24
[2025-06-01 06:31:35 +0000] [25] [INFO] Booting worker with pid: 25
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
```

```
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:37:INFO] No GPUs detected (normal if no gpus
installed)
[2025-06-01:06:31:37:INFO] Loading the model from /opt/ml/model/xgboost-
model
[2025-06-01:06:31:37:INFO] Model objective : binary:logistic
[2025-06-01:06:31:40:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2025-06-01:06:31:40:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2025-06-01:06:31:40:INFO] Determined delimiter of CSV input is ','
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
[2025-06-01:06:31:40:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "GET /ping HTTP/1.1" 200 0
"-" "Go-http-client/1.1"
[2025-06-01:06:31:40:INFO] No GPUs detected (normal if no gpus
installed)
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "GET /execution-parameters
HTTP/1.1" 200 84 "-" "Go-http-client/1.1"
[2025-06-01:06:31:40:INFO] Determined delimiter of CSV input is ','
/miniconda3/lib/python3.9/site-packages/xgboost/core.py:122: UserWarning:
ntree_limit is deprecated, use `iteration_range` or model slicing instead.
  warnings.warn(
169.254.255.130 - - [01/Jun/2025:06:31:40 +0000] "POST /invocations
HTTP/1.1" 200 1033 "-" "Go-http-client/1.1"
```

```
2025-06-01T06:31:40.404:[sagemaker logs]: MaxConcurrentTransforms=4,
MaxPayloadInMB=6, BatchStrategy=MULTI_RECORD
```

Now, let's inspect the output of the Batch Transform job in S3 again. It should show 2 columns: the ID and their corresponding probabilities of being malignant.

[17]:
```
output_df = get_csv_output_from_s3(sm_transformer.output_path, batch_file)
output_df.head(8)
```

[17]:

	0	1
0	842517	0.902768
1	84358402	0.933148
2	84501001	0.903114
3	84610002	0.987099
4	848406	0.989131
5	8511133	0.944575
6	854253	0.993661
7	858986	0.993661

create_model(role=role, image_uri=XGBOOST_IMAGE)In summary, we can use newly introduced 3 attributes - **input_filter**, **join_source**, **output_filter** to 1. Filter / select useful features from the input dataset. e.g. exclude ID columns. 2. Associate the prediction results with their corresponding input records. 3. Filter the original or joined results before saving to S3. e.g. keep ID and probability columns only.

1.5 Upload the Sagemaker Model created during our training job to the Sagemaker Model Registry

[18]:
```
import boto3
import sagemaker

sess = sagemaker.Session()
role = sagemaker.get_execution_role()
region = sess.boto_region_name

sm_client = boto3.client("sagemaker")

#  Automatically get the training job name
training_job_name = sm_estimator.latest_training_job.name

# Describe the training job
info = sm_client.describe_training_job(TrainingJobName=training_job_name)
model_data = info["ModelArtifacts"]["S3ModelArtifacts"]

# XGBoost image URI
image = sagemaker.image_uris.retrieve("xgboost", region=region, version="1.7-1")

# Create SageMaker model
```

```
primary_container = {
    "Image": image,
    "ModelDataUrl": model_data
}

create_model_response = sm_client.create_model(
    ModelName=training_job_name,
    ExecutionRoleArn=role,
    PrimaryContainer=primary_container
)

print("Model created. ARN:", create_model_response["ModelArn"])
```

```
INFO:sagemaker.image_uris:Ignoring unnecessary instance type: None.
```

```
Model created. ARN: arn:aws:sagemaker:us-
east-1:672518276407:model/xgb-2025-06-01-06-11-34
```

[19]:
```
# Inspect Training Job Details
info
```

[19]:
```
{'TrainingJobName': 'xgb-2025-06-01-06-11-34',
 'TrainingJobArn': 'arn:aws:sagemaker:us-east-1:672518276407:training-
job/xgb-2025-06-01-06-11-34',
 'ModelArtifacts': {'S3ModelArtifacts': 's3://sagemaker-us-
east-1-672518276407/DEMO-breast-cancer-prediction-xgboost-highlevel/output/xgb-
2025-06-01-06-11-34/xgb-2025-06-01-06-11-34/output/model.tar.gz'},
 'TrainingJobStatus': 'Completed',
 'SecondaryStatus': 'Completed',
 'HyperParameters': {'eta': '0.2',
  'gamma': '4',
  'max_depth': '5',
  'min_child_weight': '6',
  'num_round': '100',
  'objective': 'binary:logistic',
  'subsample': '0.8',
  'verbosity': '0'},
 'AlgorithmSpecification': {'TrainingImage': '683313688378.dkr.ecr.us-
east-1.amazonaws.com/sagemaker-xgboost:1.7-1',
  'TrainingInputMode': 'File',
  'MetricDefinitions': [{'Name': 'train:mae',
    'Regex': '.*\\[[0-9]+\\].*#011train-
mae:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:aucpr',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
aucpr:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:f1_binary',
    'Regex': '.*\\[[0-9]+\\].*#011validation-f1_binary:([-+]?[0-9]*\\.?[0-
```

```
9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:mae',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
mae:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:logloss',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
logloss:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:f1',
    'Regex': '.*\\[[0-9]+\\].*#011validation-f1:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-
+]?[0-9]+)?).*'},
   {'Name': 'train:accuracy',
    'Regex': '.*\\[[0-9]+\\].*#011train-
accuracy:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:mse',
    'Regex': '.*\\[[0-9]+\\].*#011train-
mse:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:recall',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
recall:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:poisson-nloglik',
    'Regex': '.*\\[[0-9]+\\].*#011validation-poisson-
nloglik:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:precision',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
precision:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:error',
    'Regex': '.*\\[[0-9]+\\].*#011train-
error:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:ndcg',
    'Regex': '.*\\[[0-9]+\\].*#011train-
ndcg:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:map',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
map:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:f1_binary',
    'Regex': '.*\\[[0-9]+\\].*#011train-f1_binary:([-+]?[0-9]*\\.?[0-
9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:auc',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
auc:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:auc',
    'Regex': '.*\\[[0-9]+\\].*#011train-
auc:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:error',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
error:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:poisson-nloglik',
```

```
    'Regex': '.*\\[[0-9]+\\].*#011train-poisson-
nloglik:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:rmse',
    'Regex': '.*\\[[0-9]+\\].*#011train-
rmse:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:logloss',
    'Regex': '.*\\[[0-9]+\\].*#011train-
logloss:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:accuracy',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
accuracy:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:aucpr',
    'Regex': '.*\\[[0-9]+\\].*#011train-
aucpr:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:balanced_accuracy',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
balanced_accuracy:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:rmse',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
rmse:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:mse',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
mse:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'validation:ndcg',
    'Regex': '.*\\[[0-9]+\\].*#011validation-
ndcg:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:f1',
    'Regex':
'.*\\[[0-9]+\\].*#011train-f1:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'},
   {'Name': 'train:map',
    'Regex': '.*\\[[0-9]+\\].*#011train-
map:([-+]?[0-9]*\\.?[0-9]+(?:[eE][-+]?[0-9]+)?).*'}],
  'EnableSageMakerMetricsTimeSeries': False},
 'RoleArn': 'arn:aws:iam::672518276407:role/LabRole',
 'InputDataConfig': [{'ChannelName': 'train',
   'DataSource': {'S3DataSource': {'S3DataType': 'S3Prefix',
     'S3Uri': 's3://sagemaker-us-east-1-672518276407/DEMO-breast-cancer-
prediction-xgboost-highlevel/train',
     'S3DataDistributionType': 'FullyReplicated'}},
   'ContentType': 'text/csv',
   'CompressionType': 'None',
   'RecordWrapperType': 'None'},
  {'ChannelName': 'validation',
   'DataSource': {'S3DataSource': {'S3DataType': 'S3Prefix',
     'S3Uri': 's3://sagemaker-us-east-1-672518276407/DEMO-breast-cancer-
prediction-xgboost-highlevel/validation',
     'S3DataDistributionType': 'FullyReplicated'}},
```

```
   'ContentType': 'text/csv',
   'CompressionType': 'None',
   'RecordWrapperType': 'None'}],
 'OutputDataConfig': {'KmsKeyId': '',
  'S3OutputPath': 's3://sagemaker-us-east-1-672518276407/DEMO-breast-cancer-
prediction-xgboost-highlevel/output/xgb-2025-06-01-06-11-34',
  'CompressionType': 'GZIP'},
 'ResourceConfig': {'InstanceType': 'ml.m5.xlarge',
  'InstanceCount': 1,
  'VolumeSizeInGB': 50},
 'StoppingCondition': {'MaxRuntimeInSeconds': 86400},
 'CreationTime': datetime.datetime(2025, 6, 1, 6, 11, 34, 476000,
tzinfo=tzlocal()),
 'TrainingStartTime': datetime.datetime(2025, 6, 1, 6, 12, 13, 258000,
tzinfo=tzlocal()),
 'TrainingEndTime': datetime.datetime(2025, 6, 1, 6, 13, 52, 726000,
tzinfo=tzlocal()),
 'LastModifiedTime': datetime.datetime(2025, 6, 1, 6, 13, 53, 52000,
tzinfo=tzlocal()),
 'SecondaryStatusTransitions': [{'Status': 'Starting',
   'StartTime': datetime.datetime(2025, 6, 1, 6, 11, 34, 476000,
tzinfo=tzlocal()),
   'EndTime': datetime.datetime(2025, 6, 1, 6, 12, 13, 258000,
tzinfo=tzlocal()),
   'StatusMessage': 'Preparing the instances for training'},
  {'Status': 'Downloading',
   'StartTime': datetime.datetime(2025, 6, 1, 6, 12, 13, 258000,
tzinfo=tzlocal()),
   'EndTime': datetime.datetime(2025, 6, 1, 6, 13, 34, 652000,
tzinfo=tzlocal()),
   'StatusMessage': 'Downloading the training image'},
  {'Status': 'Training',
   'StartTime': datetime.datetime(2025, 6, 1, 6, 13, 34, 652000,
tzinfo=tzlocal()),
   'EndTime': datetime.datetime(2025, 6, 1, 6, 13, 40, 5000, tzinfo=tzlocal()),
   'StatusMessage': 'Training image download completed. Training in progress.'},
  {'Status': 'Uploading',
   'StartTime': datetime.datetime(2025, 6, 1, 6, 13, 40, 5000,
tzinfo=tzlocal()),
   'EndTime': datetime.datetime(2025, 6, 1, 6, 13, 52, 726000,
tzinfo=tzlocal()),
   'StatusMessage': 'Uploading generated training model'},
  {'Status': 'Completed',
   'StartTime': datetime.datetime(2025, 6, 1, 6, 13, 52, 726000,
tzinfo=tzlocal()),
   'EndTime': datetime.datetime(2025, 6, 1, 6, 13, 52, 726000,
tzinfo=tzlocal()),
```

```
   'StatusMessage': 'Training job completed'}],
 'FinalMetricDataList': [{'MetricName': 'validation:logloss',
   'Value': 0.2112399935722351,
   'Timestamp': datetime.datetime(2025, 6, 1, 6, 13, 36, tzinfo=tzlocal())},
  {'MetricName': 'train:logloss',
   'Value': 0.06807000190019608,
   'Timestamp': datetime.datetime(2025, 6, 1, 6, 13, 36, tzinfo=tzlocal())}],
 'EnableNetworkIsolation': False,
 'EnableInterContainerTrafficEncryption': False,
 'EnableManagedSpotTraining': False,
 'TrainingTimeInSeconds': 99,
 'BillableTimeInSeconds': 99,
 'DebugHookConfig': {'S3OutputPath': 's3://sagemaker-us-
east-1-672518276407/DEMO-breast-cancer-prediction-xgboost-
highlevel/output/xgb-2025-06-01-06-11-34',
  'CollectionConfigurations': []},
 'ProfilerConfig': {'S3OutputPath': 's3://sagemaker-us-east-1-672518276407/DEMO-
breast-cancer-prediction-xgboost-highlevel/output/xgb-2025-06-01-06-11-34',
  'ProfilingIntervalInMilliseconds': 500,
  'DisableProfiler': False},
 'ProfilingStatus': 'Enabled',
 'ResponseMetadata': {'RequestId': '9d67c7a8-35e2-49fb-bafa-4abeff672f6f',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '9d67c7a8-35e2-49fb-bafa-4abeff672f6f',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '7396',
   'date': 'Sun, 01 Jun 2025 06:32:36 GMT'},
  'RetryAttempts': 0}}
```

[20]:
```python
import time
from time import gmtime, strftime
import boto3

sagemaker = boto3.client("sagemaker")

# Create Endpoint Configuration
endpoint_config_name = 'lab4-1-endpoint-config-' +␣
 ↪strftime("%Y-%m-%d-%H-%M-%S", gmtime())
instance_type = 'ml.m5.xlarge'

model_name = training_job_name
endpoint_config_response = sagemaker.create_endpoint_config(
    EndpointConfigName=endpoint_config_name,
    ProductionVariants=[
        {
            "VariantName": "variant1",
            "ModelName": model_name,
```

```
            "InstanceType": instance_type,
            "InitialInstanceCount": 1
        }
    ]
)

print(f"Created EndpointConfig:␣
 ↪{endpoint_config_response['EndpointConfigArn']}")
```

```
Created EndpointConfig: arn:aws:sagemaker:us-east-1:672518276407:endpoint-
config/lab4-1-endpoint-config-2025-06-01-06-32-37
```

[22]:
```
# Deploy our model to real-time endpoint

# Create Endpoint
endpoint_name = 'lab4-1-endpoint-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())

create_endpoint_response = sagemaker.create_endpoint(
    EndpointName=endpoint_name,
    EndpointConfigName=endpoint_config_name
)

print(f"Creating endpoint: {endpoint_name}...")
```

```
Creating endpoint: lab4-1-endpoint-2025-06-01-06-38-24…
```

[23]:
```
# Wait for endpoint to spin up
from time import sleep
sagemaker.describe_endpoint(EndpointName=endpoint_name)

while True:
    print("Getting Job Status")
    res = sagemaker.describe_endpoint(EndpointName=endpoint_name)
    state = res["EndpointStatus"]

    if state == "InService":
        print("Endpoint in Service")
        break
    elif state == "Creating":
        print("Endpoint still creating...")
        sleep(60)
    else:
        print("Endpoint Creation Error - Check Sagemaker Console")
        break
```

```
Getting Job Status
Endpoint still creating…
Getting Job Status
```

```
Endpoint still creating…
Getting Job Status
Endpoint still creating…
Getting Job Status
Endpoint in Service
```

[24]:
```
# Invoke Endpoint

sagemaker_runtime = boto3.client("sagemaker-runtime", region_name=region)

response = sagemaker_runtime.invoke_endpoint(
                            EndpointName=endpoint_name,
                            ContentType='text/csv',
                            Body=data_batch_noID.to_csv(header=None,␣
 ↪index=False).strip('\n').split('\n')[0]
                            )
print(response['Body'].read().decode('utf-8'))
```

```
0.9027683138847351
```

[25]:
```
# Examine Response Body

response
```

[25]:
```
{'ResponseMetadata': {'RequestId': 'ad00f738-3f31-4f8c-8ac3-08cb9ab62fdf',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': 'ad00f738-3f31-4f8c-8ac3-08cb9ab62fdf',
   'x-amzn-invoked-production-variant': 'variant1',
   'date': 'Sun, 01 Jun 2025 06:42:14 GMT',
   'content-type': 'text/csv; charset=utf-8',
   'content-length': '19',
   'connection': 'keep-alive'},
  'RetryAttempts': 0},
 'ContentType': 'text/csv; charset=utf-8',
 'InvokedProductionVariant': 'variant1',
 'Body': <botocore.response.StreamingBody at 0x7f3ec2a1e5c0>}
```

1.6 Part 1: Set Up Model Group

[27]:
```
import boto3
from time import gmtime, strftime

sagemaker = boto3.client("sagemaker")

# Create model package group
```

```
# Give your model group a meaningful name
model_package_group_name = "xgboost-breast-cancer-detection-v1"

# Create the model package group
response = sagemaker.create_model_package_group(
    ModelPackageGroupName=model_package_group_name,
    ModelPackageGroupDescription="XGBoost model to detect breast cancer from␣
 ↪diagnostic features."
)

print(" Model Package Group Created:")
print(response["ModelPackageGroupArn"])

# Describe the created model package group
describe_response = sagemaker.describe_model_package_group(
    ModelPackageGroupName=model_package_group_name
)

print("Model Package Group Description:")
for k, v in describe_response.items():
    print(f"{k}: {v}")
```

```
 Model Package Group Created:
arn:aws:sagemaker:us-east-1:672518276407:model-package-group/xgboost-breast-
cancer-detection-v1
Model Package Group Description:
ModelPackageGroupName: xgboost-breast-cancer-detection-v1
ModelPackageGroupArn: arn:aws:sagemaker:us-east-1:672518276407:model-package-
group/xgboost-breast-cancer-detection-v1
ModelPackageGroupDescription: XGBoost model to detect breast cancer from
diagnostic features.
CreationTime: 2025-06-01 06:43:00.873000+00:00
CreatedBy: {'UserProfileArn': 'arn:aws:sagemaker:us-east-1:672518276407:user-
profile/d-sgx5zmzwfkik/arupchak', 'UserProfileName': 'arupchak', 'DomainId':
'd-sgx5zmzwfkik', 'IamIdentity': {'Arn': 'arn:aws:sts::672518276407:assumed-
role/LabRole/SageMaker', 'PrincipalId': 'AROAZZFJWQU36L6GW2MC3:SageMaker'}}
ModelPackageGroupStatus: Completed
ResponseMetadata: {'RequestId': '6fe783e9-8361-46c8-9cf1-e77f283d7451',
'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid':
'6fe783e9-8361-46c8-9cf1-e77f283d7451', 'content-type': 'application/x-amz-
json-1.1', 'content-length': '647', 'date': 'Sun, 01 Jun 2025 06:43:00 GMT'},
'RetryAttempts': 0}
```

1.7 Part 2: Set Up Model Package

[42]:
```
import boto3

sagemaker = boto3.client("sagemaker")
s3 = boto3.client("s3")

# Parse S3 path
model_artifact_path = sm_estimator.model_data

model_package_response = sagemaker.create_model_package(
    ModelPackageGroupName=model_package_group_name,
    ModelPackageDescription="XGBoost model v1 for breast cancer classification",
    InferenceSpecification={
        "Containers": [
            {
                "Image": image,  # e.g. '683313688378.dkr.ecr.us-west-2.
 ↪amazonaws.com/sagemaker-xgboost:1.7-1'
                "ModelDataUrl": model_artifact_path,  # e.g. 's3://bucket/path/
 ↪to/model.tar.gz'
                "Environment": {
                    "SAGEMAKER_SUBMIT_DIRECTORY": model_artifact_path,
                    "SAGEMAKER_PROGRAM": "inference.py",
                }
            }
        ],
        "SupportedContentTypes": ["text/csv"],
        "SupportedResponseMIMETypes": ["text/csv"]
    },
    CertifyForMarketplace=False
)

model_package_arn = model_package_response["ModelPackageArn"]
print("Model Package Created:", model_package_arn)
```

```
Model Package Created: arn:aws:sagemaker:us-east-1:672518276407:model-
package/xgboost-breast-cancer-detection-v1/2
```

[43]:
```
# Describe the registered model package
description = sagemaker.
 ↪describe_model_package(ModelPackageName=model_package_arn)

print("Model Package Details:")
for k, v in description.items():
    print(f"{k}: {v}")
```

```
Model Package Details:
ModelPackageGroupName: xgboost-breast-cancer-detection-v1
```

```
ModelPackageVersion: 2
ModelPackageArn: arn:aws:sagemaker:us-east-1:672518276407:model-package/xgboost-
breast-cancer-detection-v1/2
ModelPackageDescription: XGBoost model v1 for breast cancer classification
CreationTime: 2025-06-01 07:35:19.834000+00:00
InferenceSpecification: {'Containers': [{'Image': '683313688378.dkr.ecr.us-
east-1.amazonaws.com/sagemaker-xgboost:1.7-1', 'ImageDigest':
'sha256:50f42bf4e288ce1e2431b1574b37d41eb7f70a3d67f6faf5789a8624f4feea21',
'ModelDataUrl': 's3://sagemaker-us-east-1-672518276407/DEMO-breast-cancer-
prediction-xgboost-highlevel/output/xgb-2025-06-01-06-11-34/xgb-2025-06-01-06-
11-34/output/model.tar.gz', 'Environment': {'SAGEMAKER_PROGRAM': 'inference.py',
'SAGEMAKER_SUBMIT_DIRECTORY': 's3://sagemaker-us-east-1-672518276407/DEMO-
breast-cancer-prediction-xgboost-highlevel/output/xgb-2025-06-01-06-11-34/xgb-
2025-06-01-06-11-34/output/model.tar.gz'}, 'ModelDataETag':
'1f049af644a82e84d8fd61ff9084614e'}], 'SupportedContentTypes': ['text/csv'],
'SupportedResponseMIMETypes': ['text/csv']}
ModelPackageStatus: Completed
ModelPackageStatusDetails: {'ValidationStatuses': [], 'ImageScanStatuses': []}
CertifyForMarketplace: False
CreatedBy: {'UserProfileArn': 'arn:aws:sagemaker:us-east-1:672518276407:user-
profile/d-sgx5zmzwfkik/arupchak', 'UserProfileName': 'arupchak', 'DomainId':
'd-sgx5zmzwfkik', 'IamIdentity': {'Arn': 'arn:aws:sts::672518276407:assumed-
role/LabRole/SageMaker', 'PrincipalId': 'AROAZZFJWQU36L6GW2MC3:SageMaker'}}
ResponseMetadata: {'RequestId': '027bef00-c86e-4e19-8a29-23bca39c9924',
'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid':
'027bef00-c86e-4e19-8a29-23bca39c9924', 'content-type': 'application/x-amz-
json-1.1', 'content-length': '1507', 'date': 'Sun, 01 Jun 2025 07:35:29 GMT'},
'RetryAttempts': 0}
```

1.8 Part 3: Write the Model Card

Get the accuracy matrics

[44]:
```python
import pandas as pd

# Assuming you already downloaded the file from S3 to "validation_data.csv"
data = pd.read_csv("validation_data.csv", header=None)

# Split features and labels
X_val = data.iloc[:, 1:]
y_val = data.iloc[:, 0]
```

[ ]:
```python
#### Load the model
```

[45]:
```python
import boto3

s3 = boto3.client("s3")
```

```
# Parse S3 path
model_artifact_path = sm_estimator.model_data  # or from describe_training_job
print("Model artifact S3 path:", model_artifact_path)

# Parse bucket and key
s3_uri = model_artifact_path.replace("s3://", "")
bucket = s3_uri.split("/")[0]
key = "/".join(s3_uri.split("/")[1:])

# Download model.tar.gz
s3.download_file(bucket, key, "model.tar.gz")
```

```
Model artifact S3 path: s3://sagemaker-us-east-1-672518276407/DEMO-breast-
cancer-prediction-xgboost-highlevel/output/xgb-2025-06-01-06-11-34/xgb-2025-06-
01-06-11-34/output/model.tar.gz
```

[36]:
```
import tarfile
import os

extract_path = "./model"
os.makedirs(extract_path, exist_ok=True)

with tarfile.open("model.tar.gz", "r:gz") as tar:
    tar.extractall(path=extract_path)

print("Extracted files:", os.listdir(extract_path))
```

```
Extracted files: ['xgboost-model']
```

```
/tmp/ipykernel_6893/4014615773.py:8: DeprecationWarning: Python 3.14 will, by
default, filter extracted tar archives and reject files or modify their
metadata. Use the filter argument to control this behavior.
  tar.extractall(path=extract_path)
```

Predict and evaluate

[46]:
```
import xgboost as xgb
import pandas as pd
from sklearn.metrics import accuracy_score, precision_score, recall_score

# Load your validation dataset
data = pd.read_csv("validation_data.csv", header=None)
X_val = data.iloc[:, 1:]  # Features
y_val = data.iloc[:, 0]   # Labels

# Load the trained XGBoost model
model = xgb.Booster()
model.load_model("./model/xgboost-model")
```

```
# Run predictions
dval = xgb.DMatrix(X_val)
y_pred_probs = model.predict(dval)
y_pred = (y_pred_probs > 0.5).astype(int)

# Compute evaluation metrics
accuracy = accuracy_score(y_val, y_pred)
precision = precision_score(y_val, y_pred)
recall = recall_score(y_val, y_pred)

# Print results
print("Accuracy:", round(accuracy, 4))
print("Precision:", round(precision, 4))
print("Recall:", round(recall, 4))
```

```
Accuracy: 0.9167
Precision: 0.9091
Recall: 0.8696
```

[47]:
```
import boto3
import json
from time import gmtime, strftime

# Initialize the SageMaker client
sagemaker = boto3.client("sagemaker")

# Define the model card name with a timestamp
model_card_name = "xgboost-breast-cancer-card-" + strftime("%Y-%m-%d-%H-%M-%S",␣
 ↪gmtime())

# Define the content of the model card following the JSON schema
model_card_content = {
    "model_overview": {
        "model_description": "XGBoost model for breast cancer detection using␣
 ↪diagnostic features.",
        "model_owner": "arupchak",
        "problem_type": "Binary classification",
        "algorithm_type": "XGBoost"
    },
    "intended_uses": {
        "intended_uses": "Assist medical professionals in early detection of␣
 ↪breast cancer.",
        "risk_rating": "High"
    },
    "training_details": {
        "objective_function": {
            "function": "Minimize",
```

```
            "facet": "Loss",
            "description": "Binary logistic loss function."
        },
        "training_observations": "Model trained on balanced dataset with 1000␣
 ↪samples."
    },
    "evaluation_details": [
        {
            "name": "Validation Evaluation",
            "evaluation_observation": "Achieved 96% accuracy on validation␣
 ↪dataset.",
            "datasets": ["validation_data.csv"],
            "metric_groups": [
                {
                    "name": "Binary Classification Metrics",
                    "metric_data": [
                        {
                            "name": "Accuracy",
                            "type": "number",
                            "value": round(accuracy, 4)
                        },
                        {
                            "name": "Precision",
                            "type": "number",
                            "value": round(precision, 4)
                        },
                        {
                            "name": "Recall",
                            "type": "number",
                            "value": round(recall, 4)
                        }
                    ]
                }
            ]
        }
    ]
}

# Create the model card
response = sagemaker.create_model_card(
    ModelCardName=model_card_name,
    Content=json.dumps(model_card_content),
    ModelCardStatus="Draft"
)

print("Model Card Created:")
print(response["ModelCardArn"])
```

```
Model Card Created:
arn:aws:sagemaker:us-east-1:672518276407:model-card/xgboost-breast-cancer-
card-2025-06-01-07-36-11
```

[48]:
```
# Describe the model card to retrieve its details
description = sagemaker.describe_model_card(ModelCardName=model_card_name)

print("Model Card Description:")
for key, value in description.items():
    print(f"{key}: {value}")
```

```
Model Card Description:
ModelCardArn: arn:aws:sagemaker:us-east-1:672518276407:model-card/xgboost-
breast-cancer-card-2025-06-01-07-36-11
ModelCardName: xgboost-breast-cancer-card-2025-06-01-07-36-11
ModelCardVersion: 1
Content: {"model_overview": {"model_description": "XGBoost model for breast
cancer detection using diagnostic features.", "model_owner": "arupchak",
"problem_type": "Binary classification", "algorithm_type": "XGBoost"},
"intended_uses": {"intended_uses": "Assist medical professionals in early
detection of breast cancer.", "risk_rating": "High"}, "training_details":
{"objective_function": {"function": "Minimize", "facet": "Loss", "description":
"Binary logistic loss function."}, "training_observations": "Model trained on
balanced dataset with 1000 samples."}, "evaluation_details": [{"name":
"Validation Evaluation", "evaluation_observation": "Achieved 96% accuracy on
validation dataset.", "datasets": ["validation_data.csv"], "metric_groups":
[{"name": "Binary Classification Metrics", "metric_data": [{"name": "Accuracy",
"type": "number", "value": 0.9167}, {"name": "Precision", "type": "number",
"value": 0.9091}, {"name": "Recall", "type": "number", "value": 0.8696}]}]}]}
ModelCardStatus: Draft
CreationTime: 2025-06-01 07:36:11.773000+00:00
CreatedBy: {'UserProfileArn': 'arn:aws:sagemaker:us-east-1:672518276407:user-
profile/d-sgx5zmzwfkik/arupchak', 'UserProfileName': 'arupchak', 'DomainId':
'd-sgx5zmzwfkik'}
LastModifiedTime: 2025-06-01 07:36:11.773000+00:00
LastModifiedBy: {'UserProfileArn': 'arn:aws:sagemaker:us-
east-1:672518276407:user-profile/d-sgx5zmzwfkik/arupchak', 'UserProfileName':
'arupchak', 'DomainId': 'd-sgx5zmzwfkik'}
ResponseMetadata: {'RequestId': 'edb83169-dc3a-457a-9975-a3329f19b568',
'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid':
'edb83169-dc3a-457a-9975-a3329f19b568', 'content-type': 'application/x-amz-
json-1.1', 'content-length': '1725', 'date': 'Sun, 01 Jun 2025 07:36:17 GMT'},
'RetryAttempts': 0}
```

[ ]:
```
# Delete Endpoint

sagemaker.delete_endpoint(EndpointName=endpoint_name)
```

[ ]: