In [20]:
```
import warnings, requests, zipfile, io
warnings.simplefilter('ignore')
import pandas as pd
from scipy.io import arff
import boto3
```

In [21]:
```
f_zip = 'http://archive.ics.uci.edu/ml/machine-learning-databases/00212/vertebra
r = requests.get(f_zip, stream=True)
Vertebral_zip = zipfile.ZipFile(io.BytesIO(r.content))
Vertebral_zip.extractall()
```

In [22]:
```
data = arff.loadarff('column_2C_weka.arff')
df = pd.DataFrame(data[0])
```

In [23]:
```
class_mapper = {b'Abnormal':1,b'Normal':0}
df['class']=df['class'].replace(class_mapper)
```

In [24]:
```
df.shape
```

Out[24]:
```
(310, 7)
```

In [25]:
```
df.columns
```

Out[25]:
```
Index(['pelvic_incidence', 'pelvic_tilt', 'lumbar_lordosis_angle',
       'sacral_slope', 'pelvic_radius', 'degree_spondylolisthesis', 'class'],
      dtype='object')
```

In [26]:
```
df.columns
```

Out[26]:
```
Index(['pelvic_incidence', 'pelvic_tilt', 'lumbar_lordosis_angle',
       'sacral_slope', 'pelvic_radius', 'degree_spondylolisthesis', 'class'],
      dtype='object')
```

In [27]:
```
cols = df.columns.tolist()
cols = [cols[-1]] + cols[:-1]
df = df[cols]

print(df.columns)
```

```
Index(['class', 'pelvic_incidence', 'pelvic_tilt', 'lumbar_lordosis_angle',
       'sacral_slope', 'pelvic_radius', 'degree_spondylolisthesis'],
      dtype='object')
```

In [28]:
```
from sklearn.model_selection import train_test_split
train, test_and_validate = train_test_split(df, test_size=0.2, random_state=42,
```

In [29]:
```
test, validate = train_test_split(test_and_validate, test_size=0.5, random_state
```

In [30]:
```
print(train.shape)
print(test.shape)
print(validate.shape)
```

```
(248, 7)
(31, 7)
(31, 7)
```

In [31]:
```
print(train['class'].value_counts())
print(test['class'].value_counts())
print(validate['class'].value_counts())
```

```
class
1    168
0     80
Name: count, dtype: int64
class
1    21
0    10
Name: count, dtype: int64
class
1    21
0    10
Name: count, dtype: int64
```

In [32]:
```
bucket='c169682a4380819l111150392t1w164112440464-labbucket-divyesujtnis'

prefix='lab3'

train_file='vertebral_train.csv'
test_file='vertebral_test.csv'
validate_file='vertebral_validate.csv'

import os

s3_resource = boto3.Session().resource('s3')
def upload_s3_csv(filename, folder, dataframe):
    csv_buffer = io.StringIO()
    dataframe.to_csv(csv_buffer, header=False, index=False)
    s3_resource.Bucket(bucket).Object(os.path.join(prefix, folder, filename)).pu
```

```
INFO:botocore.credentials:Found credentials from IAM Role: BaseNotebookInstance
Ec2InstanceRole
```

In [33]:
```
upload_s3_csv(train_file, 'train', train)
upload_s3_csv(test_file, 'test', test)
upload_s3_csv(validate_file, 'validate', validate)
```

In [34]:
```
import boto3
from sagemaker.image_uris import retrieve
container = retrieve('xgboost',boto3.Session().region_name,'1.0-1')
```

```
INFO:sagemaker.image_uris:Defaulting to only available Python version: py3
INFO:sagemaker.image_uris:Defaulting to only supported image scope: cpu.
```

In [35]:
```
hyperparams={"num_round":"42",
             "eval_metric": "auc",
             "objective": "binary:logistic"}
```

In [36]:
```
import sagemaker
s3_output_location="s3://{}/{}/output/".format(bucket,prefix)
xgb_model=sagemaker.estimator.Estimator(container,
                                        sagemaker.get_execution_role(),
                                        instance_count=1,
                                        instance_type='ml.m4.xlarge',
                                        output_path=s3_output_location,
```

```
                                        hyperparameters=hyperparams,
                                        sagemaker_session=sagemaker.Session())
```

In [37]:
```
train_channel = sagemaker.inputs.TrainingInput(
    "s3://{}/{}/train/".format(bucket,prefix,train_file),
    content_type='text/csv')

validate_channel = sagemaker.inputs.TrainingInput(
    "s3://{}/{}/validate/".format(bucket,prefix,validate_file),
    content_type='text/csv')

data_channels = {'train': train_channel, 'validation': validate_channel}
```

In [38]:
```
xgb_model.fit(inputs=data_channels, logs=False)
```

```
INFO:sagemaker.telemetry.telemetry_logging:SageMaker Python SDK will collect te
lemetry to help us better understand our user's needs, diagnose issues, and del
iver additional features.
To opt out of telemetry, please disable via TelemetryOptOut parameter in SDK de
faults config. For more information, refer to https://sagemaker.readthedocs.io/
en/stable/overview.html#configuring-and-using-defaults-with-the-sagemaker-pytho
n-sdk.
INFO:sagemaker:Creating training-job with name: sagemaker-xgboost-2025-08-08-06
-16-22-602
2025-08-08 06:16:22 Starting - Starting the training job...
2025-08-08 06:16:46 Starting - Preparing the instances for training....
2025-08-08 06:17:12 Downloading - Downloading input data.....
2025-08-08 06:17:42 Downloading - Downloading the training image.........
2025-08-08 06:18:32 Training - Training image download completed. Training in p
rogress....
2025-08-08 06:18:53 Uploading - Uploading generated training model..
2025-08-08 06:19:06 Completed - Training job completed
```

In [ ]:
```

```