

```
01: from sagemaker import RandomCutForest
02: from sagemaker.amazon.common import RecordSerializer
03: import numpy as np
04: import io
05:
06: rcf_session = sagemaker.Session()
07: image_name = sagemaker.image_uris.retrieve("randomcutforest", region)
08:
09: rcf_prefix = 'rcf'
10: s3_output_path = "s3://{}/{}/output".format(s3_bucket, rcf_prefix)
11:
12: rcf_estimator = sagemaker.estimator.Estimator(
13:     image_name,
14:     sagemaker_session=sagemaker_session,
15:     role=role,
16:     instance_count=1,
17:     instance_type='ml.c4.xlarge',
18:     base_job_name='cqpub-randomcutforest',
19:     output_path=s3_output_path
20: )
21:
22: hyperparameters = {
23:     "num_samples_per_tree": 2,
24:     "num_trees": 100,
25:     "feature_dim": 120
26: }
27: rcf_estimator.set_hyperparameters(**hyperparameters)
28:
29: # シングリング処理によって特徴量の次元を増やす
30: def shingle(data, shingle_size):
31:     num_data = len(data)
32:     shingled_data = np.zeros((num_data-shingle_size, shingle_size))
33:
34:     for n in range(num_data - shingle_size):
35:         shingled_data[n] = data[n:(n+shingle_size)]
36:     return shingled_data
37:
38: shingle_size = 5 * 24
39: train_data = shingle(df['dust'], shingle_size)
40:
41: # RandomCutForestで学習するにあたり、RecordIOと呼ばれる形式でS3にアップロードする
42: def convert_and_upload_training_data(
43:     ndarray, bucket, prefix, filename='rcf_train.pbr'):
44:
```

```
45:     import boto3
46:     import os
47:
48:     serializer = RecordSerializer()
49:     buffer = serializer.serialize(ndarray)
50:
51:     s3 = boto3.resource('s3')
52:     s3.Bucket(bucket).put_object(Body=buffer,
53:                                 Key='rcf/train/rcf_train.pbr',
54:                                 ContentType='application/x-recordio-protobuf')
55:
56:     s3_object = os.path.join(prefix, 'train', filename)
57:     s3_path = 's3://{}/{}'.format(bucket, s3_object)
58:     return s3_path
59:
60: rcf_train_data = convert_and_upload_training_data(train_data, s3_bucket, rcf_prefix)
61:
62: # S3にアップロードしたRecordIO形式のデータを学習時のInputとする
63: rcf_train_input = sagemaker.inputs.TrainingInput(
64:     rcf_train_data,
65:     distribution='ShardedByS3Key',
66:     content_type='application/x-recordio-protobuf')
67:
68: rcf_estimator.fit({'train': rcf_train_input})
```