**Codes:**

1. Kinesis\_data\_stream.ipynb

import json

import datetime

import random

def getData(iotName, lowVal, highVal):

data = {}

data["iotName"] = iotName

data["iotValue"] = random.randint(lowVal, highVal)

return data

kinesis = boto3.client('kinesis')

while 1:

rnd = random.random()

if (rnd < 0.01):

data = json.dumps(getData("DemoSensor", 100, 120))

kinesis.put\_record(StreamName="demo", Data=data, PartitionKey="DemoSensor")

# kinesis.put\_record("demo", data, "DemoSensor")

print ('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* anomaly \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ' + data)

else:

data = json.dumps(getData("DemoSensor", 10, 20))

kinesis.put\_record(StreamName="demo", Data=data, PartitionKey="DemoSensor")

#kinesis.put\_record("demo", data, "DemoSensor")

print (data)

1. Kinesis\_sql\_analytics.sql

CREATE OR REPLACE STREAM "TEMP\_STREAM" (

"iotName" varchar (40),

"iotValue" integer,

"ANOMALY\_SCORE" DOUBLE);

-- Creates an output stream and defines a schema

CREATE OR REPLACE STREAM "DESTINATION\_SQL\_STREAM" (

"iotName" varchar(40),

"iotValue" integer,

"ANOMALY\_SCORE" DOUBLE,

"created" TimeStamp);

-- Compute an anomaly score for each record in the source stream

-- using Random Cut Forest

CREATE OR REPLACE PUMP "STREAM\_PUMP\_1" AS INSERT INTO "TEMP\_STREAM"

SELECT STREAM "iotName", "iotValue", ANOMALY\_SCORE FROM

TABLE(RANDOM\_CUT\_FOREST(

CURSOR(SELECT STREAM \* FROM "SOURCE\_SQL\_STREAM\_001")

)

);

-- Sort records by descending anomaly score, insert into output stream

CREATE OR REPLACE PUMP "OUTPUT\_PUMP" AS INSERT INTO "DESTINATION\_SQL\_STREAM"

SELECT STREAM "iotName", "iotValue", ANOMALY\_SCORE, ROWTIME FROM "TEMP\_STREAM"

ORDER BY FLOOR("TEMP\_STREAM".ROWTIME TO SECOND), ANOMALY\_SCORE DESC;