import sys

from awsglue.transforms import \*

from awsglue.utils import getResolvedOptions

from pyspark.context import SparkContext

from awsglue.context import GlueContext

from awsglue.job import Job

## @params: [JOB\_NAME]

args = getResolvedOptions(sys.argv, ['JOB\_NAME'])

sc = SparkContext()

glueContext = GlueContext(sc)

spark = glueContext.spark\_session

job = Job(glueContext)

job.init(args['JOB\_NAME'], args)

node1 = glueContext.create\_dynamic\_frame.from\_options(

format\_options={"multiline":False},

connection\_type="s3",

format="json",

connection\_options={

"paths":["s3://stedi-supraja/accelerometer/trusted"],

"recurse":True,

},

transformation\_ctx="node1",

)

node2 = glueContext.create\_dynamic\_frame.from\_options(

format\_options={"multiline":False},

connection\_type="s3",

format="json",

connection\_options={

"paths":["s3://stedi-supraja/step\_trainer/trusted"],

"recurse":True,

},

transformation\_ctx="node2",

)

Join\_node = Join.apply(

frame1=node1,

frame2=node2,

keys1=["timestamp"],

keys2=["sensorReadingTime"],

transformation\_ctx="Join\_node",

)

DropFields\_node = DropFields.apply(

frame=Join\_node,

paths=["user"],

transformation\_ctx="DropFields\_node",

)

node3 = glueContext.write\_dynamic\_frame.from\_options(

frame = DropFields\_node,

connection\_type="s3",

format="json",

connection\_options={

"path":"s3://stedi-supraja/machine\_learning/curated/",

"partitionKeys":[],

},

transformation\_ctx="node3",

)

job.commit()