Aisha Muhammad Nawaz

Pyspark Clustering Lab 5 L20-0921

14th March 2024

```
# #Running on Colab
!pip install pyspark
!pip install -U -q PyDrive
!apt install openjdk-8-jdk-headless -qq
import os
os.environ['JAVA_HOME'] = '/usr/lib/jvm/java-8-openjdk-amd64'
```

```
Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.0/317.0 MB 3.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.1-py2.py3-none-any.whl size=317488491 sha256=7f0b644ccf1252aee5105123987315475e688970075d84542e97342ac72f0acd
  Stored in directory: /root/.cache/pip/wheels/80/1d/60/2c256ed38dddce2fdd93be545214a63e02fbd8d74fb0b7f3a6
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.1
The following additional packages will be installed:
  libxtst6 openjdk-8-jre-headless
Suggested packages:
  openjdk-8-demo openjdk-8-source libnss-mdns fonts-dejavu-extra fonts-nanum fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  libxtst6 openjdk-8-jdk-headless openjdk-8-jre-headless
0 upgraded, 3 newly installed, 0 to remove and 38 not upgraded.
Need to get 39.7 MB of archives.
After this operation, 144 MB of additional disk space will be used.
Selecting previously unselected package libxtst6:amd64.
(Reading database ... 121752 files and directories currently installed.)
Preparing to unpack .../libxtst6_2%3a1.2.3-1build4_amd64.deb ...
Unpacking libxtst6:amd64 (2:1.2.3-1build4) ...
Selecting previously unselected package openjdk-8-jre-headless:amd64.
Preparing to unpack .../openjdk-8-jre-headless_8u392-ga-1~22.04_amd64.deb ...
Unpacking openjdk-8-jre-headless:amd64 (8u392-ga-1~22.04) ...
Selecting previously unselected package openjdk-8-jdk-headless:amd64.
Preparing to unpack .../openjdk-8-jdk-headless_8u392-ga-1~22.04_amd64.deb ...
Unpacking openjdk-8-jdk-headless:amd64 (8u392-ga-1~22.04) ...
Setting up libxtst6:amd64 (2:1.2.3-1build4) ...
Setting up openjdk-8-jre-headless:amd64 (8u392-ga-1~22.04) ...
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/orbd to provide /usr/bin/orbd (orbd) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/servertool to provide /usr/bin/servertool (servertool) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/tnameserv to provide /usr/bin/tnameserv (tnameserv) in auto mode
Setting up openjdk-8-jdk-headless:amd64 (8u392-ga-1~22.04) ...
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/clhsdb to provide /usr/bin/clhsdb (clhsdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/extcheck to provide /usr/bin/extcheck (extcheck) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/hsdb to provide /usr/bin/hsdb (hsdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/idlj to provide /usr/bin/idlj (idlj) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/javah to provide /usr/bin/javah (javah) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jhat to provide /usr/bin/jhat (jhat) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/jsadebugd to provide /usr/bin/jsadebugd (jsadebugd) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/native2ascii to provide /usr/bin/native2ascii (native2ascii) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/schemagen to provide /usr/bin/schemagen (schemagen) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/wsgen to provide /usr/bin/wsgen (wsgen) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/wsimport to provide /usr/bin/wsimport (wsimport) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/xjc to provide /usr/bin/xjc (xjc) in auto mode
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link
```

```
#  Import the libraries we will need
import pyspark
from pyspark.sql import *
from pyspark.sql.functions import *
from pyspark import SparkContext, SparkConf

# Create Spark session and ContextRun PySpark.
# create the session
conf = SparkConf().set("spark.ui.port","4050")
# create the context
sc = pyspark.SparkContext(conf=conf)
spark = SparkSession.builder.appName("DataFrame").config('spark.ui.port', '4050').getOrCreate()
spark
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-6-39a620b45ddf> in <cell line: 11>()
      9 conf = SparkConf().set("spark.ui.port","4050")
     10 # create the context
---> 11 sc = pyspark.SparkContext(conf=conf)
     12 spark = SparkSession.builder.appName("DataFrame").config('spark.ui.port', '4050').getOrCreate()
     13 spark

2 frames
ValueError: Cannot run multiple SparkContexts at once; existing SparkContext(app=pyspark-shell, master=local[*]) created by __init__ at <ipython-input-2-39a620b45ddf>:11
```

```
# sampletTest.txt has content as follows:
# 1.0 2.0 3.0
# 1.5 2.5 3.5
# 5.0 6.0 7.0
# 6.0 7.0 8.0
# 1.8 2.8 3.8

myData=spark.read.text('sampleTest.txt').rdd.map(lambda r: r[0])
myData.collect()
```

```
['0.0 0.0 0.0',
 '0.1 0.1 0.1',
 '0.2 0.2 0.2',
 '9.0 9.0 9.0',
 '9.1 9.1 9.1',
 '9.2 9.2 9.2']
```

```
import numpy as np
def parseVector(line):
  return np.array([float(x) for x in line.split(' ')])
parsedData=myData.map(lambda x: parseVector(x))
parsedData.collect()
```

```
[array([0., 0., 0.]),
 array([0.1, 0.1, 0.1]),
 array([0.2, 0.2, 0.2]),
 array([9., 9., 9.]),
 array([9.1, 9.1, 9.1]),
 array([9.2, 9.2, 9.2])]
```

```
# THIS IMPLEMENTATION MIGHT BE WRONG
K=2
convergedDistance=0.2
kPoints=parsedData.takeSample(withReplacement=False , num=K ,seed=1)
tempD=1.0
def findNearestCluster(x,kPoints):
  bestCluster=0
  minDist=float('inf')
  for i in range(len(kPoints)):
    dist=np.sum((x-kPoints[i])**2)
    if(dist<minDist):
      bestCluster=i
      minDist=dist
  return bestCluster

while tempD>convergedDistance:
  closestFound=parsedData.map(lambda x: (findNearestCluster(x,kPoints),(x,1))) #Assign Nearest Cluster to each Point
  closestFound2=closestFound.reduceByKey(lambda p1,p2: (p1[0]+p2[0],p1[1]+p2[1])) # Sum points and their count in each cluster
  newPoints=closestFound2.map(lambda point: (point[0],point[1][0]/point[1][1])).collect() #Find avg of points in each cluster
  # Note above the key is cluster
  # Now finding distance between new cluster centeres and old ones
  tempD=0.0
  sum=0
  for (ik,p) in newPoints:
    sum=sum+np.sum((kPoints[ik]-p)**2)
  tempD=sum
  # For structing purposes, assigning new clusters their values
  for (ik,p) in newPoints:
    kPoints[ik]=p

print('Final Centres ',kPoints)
```

```
Final Centres  [array([0.2, 0.2, 0.2]), array([13.8, 13.8, 13.8])]
```

```
from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator


# Read data from file
data = spark.read.text("sampleTest.txt").rdd.map(lambda x: x[0])

# Save the data in LibSVM format
data.saveAsTextFile("path_to_save_data_in_libsvm_format")

# Load the data in LibSVM format
dataset = spark.read.format("libsvm").load("path_to_save_data_in_libsvm_format")


# Trains a k-means model.
kmeans = KMeans().setK(2).setSeed(1)
model = kmeans.fit(dataset)

# Make predictions
predictions = model.transform(dataset)

# Evaluate clustering by computing Silhouette score
evaluator = ClusteringEvaluator()

silhouette = evaluator.evaluate(predictions)
print("Silhouette with squared euclidean distance = " + str(silhouette))

# Shows the result.
centers = model.clusterCenters()
print("Cluster Centers: ")
for center in centers:
    print(center)
```

```
---------------------------------------------------------------------------
Py4JJavaError                             Traceback (most recent call last)
<ipython-input-49-9c5f822fa9c5> in <cell line: 12>()
     10
     11 # Load the data in LibSVM format
---> 12 dataset = spark.read.format("libsvm").load("path_to_save_data_in_libsvm_format")
     13
     14

 3 frames
/usr/local/lib/python3.10/dist-packages/py4j/protocol.py in get_return_value(answer, gateway_client, target_id, name)
    324             value = OUTPUT_CONVERTER[type](answer[2:], gateway_client)
    325             if answer[1] == REFERENCE_TYPE:
--> 326                 raise Py4JJavaError(
    327                     "An error occurred while calling {0}{1}{2}.\n".
    328                     format(target_id, ".", name), value)

Py4JJavaError: An error occurred while calling o1658.load.
: org.apache.spark.SparkException: Job aborted due to stage failure: Task 0 in stage 104.0 failed 1 times, most recent failure: Lost task 0.0 in stage 104.0 (TID 104)
(1a6ae519ca96 executor driver): java.lang.NumberFormatException: For input string: "0.1"
        at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
        at java.lang.Integer.parseInt(Integer.java:580)
        at java.lang.Integer.parseInt(Integer.java:615)
        at scala.collection.immutable.StringLike.toInt(StringLike.scala:310)
        at scala.collection.immutable.StringLike.toInt$(StringLike.scala:310)
        at scala.collection.immutable.StringOps.toInt(StringOps.scala:33)
        at org.apache.spark.mllib.util.MLUtils$.$anonfun$parseLibSVMRecord$2(MLUtils.scala:134)
        at scala.collection.TraversableLike.$anonfun$map$1(TraversableLike.scala:286)
        at scala.collection.IndexedSeqOptimized.foreach(IndexedSeqOptimized.scala:36)
        at scala.collection.IndexedSeqOptimized.foreach$(IndexedSeqOptimized.scala:33)
        at scala.collection.mutable.ArrayOps$ofRef.foreach(ArrayOps.scala:198)
        at scala.collection.TraversableLike.map(TraversableLike.scala:286)
        at scala.collection.TraversableLike.map$(TraversableLike.scala:279)
        at scala.collection.mutable.ArrayOps$ofRef.map(ArrayOps.scala:198)
        at org.apache.spark.mllib.util.MLUtils$.parseLibSVMRecord(MLUtils.scala:132)
        at org.apache.spark.mllib.util.MLUtils$.$anonfun$parseLibSVMFile$4(MLUtils.scala:126)
        at scala.collection.Iterator$$anon$10.next(Iterator.scala:461)
        at scala.collection.Iterator$$anon$10.next(Iterator.scala:461)
        at scala.collection.Iterator.foreach(Iterator.scala:943)
        at scala.collection.Iterator.foreach$(Iterator.scala:943)
        at scala.collection.AbstractIterator.foreach(Iterator.scala:1431)
        at scala.collection.TraversableOnce.reduceLeft(TraversableOnce.scala:237)
        at scala.collection.TraversableOnce.reduceLeft$(TraversableOnce.scala:220)
        at scala.collection.AbstractIterator.reduceLeft(Iterator.scala:1431)
        at org.apache.spark.rdd.RDD.$anonfun$reduce$2(RDD.scala:1125)
        at org.apache.spark.SparkContext.$anonfun$runJob$6(SparkContext.scala:2492)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:93)
        at org.apache.spark.TaskContext.runTaskWithListeners(TaskContext.scala:166)
        at org.apache.spark.scheduler.Task.run(Task.scala:141)
        at org.apache.spark.executor.Executor$TaskRunner.$anonfun$run$4(Executor.scala:620)
        at org.apache.spark.util.SparkErrorUtils.tryWithSafeFinally(SparkErrorUtils.scala:64)
        at org.apache.spark.util.SparkErrorUtils.tryWithSafeFinally$(SparkErrorUtils.scala:61)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:94)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:623)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:750)

Driver stacktrace:
        at org.apache.spark.scheduler.DAGScheduler.failJobAndIndependentStages(DAGScheduler.scala:2856)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$abortStage$2(DAGScheduler.scala:2792)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$abortStage$2$adapted(DAGScheduler.scala:2791)
        at scala.collection.mutable.ResizableArray.foreach(ResizableArray.scala:62)
        at scala.collection.mutable.ResizableArray.foreach$(ResizableArray.scala:55)
        at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:49)
        at org.apache.spark.scheduler.DAGScheduler.abortStage(DAGScheduler.scala:2791)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$handleTaskSetFailed$1(DAGScheduler.scala:1247)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$handleTaskSetFailed$1$adapted(DAGScheduler.scala:1247)
        at scala.Option.foreach(Option.scala:407)
        at org.apache.spark.scheduler.DAGScheduler.handleTaskSetFailed(DAGScheduler.scala:1247)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.doOnReceive(DAGScheduler.scala:3060)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:2994)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:2983)
        at org.apache.spark.util.EventLoop$$anon$1.run(EventLoop.scala:49)
        at org.apache.spark.scheduler.DAGScheduler.runJob(DAGScheduler.scala:989)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:2398)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:2493)
        at org.apache.spark.rdd.RDD.$anonfun$reduce$1(RDD.scala:1139)
        at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
        at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
        at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
        at org.apache.spark.rdd.RDD.reduce(RDD.scala:1121)
        at org.apache.spark.mllib.util.MLUtils$.computeNumFeatures(MLUtils.scala:94)
        at org.apache.spark.ml.source.libsvm.LibSVMFileFormat.$anonfun$inferSchema$1(LibSVMRelation.scala:106)
        at scala.runtime.java8.JFunction0$mcI$sp.apply(JFunction0$mcI$sp.java:23)
        at scala.Option.getOrElse(Option.scala:189)
        at org.apache.spark.ml.source.libsvm.LibSVMFileFormat.inferSchema(LibSVMRelation.scala:97)
        at org.apache.spark.sql.execution.datasources.DataSource.$anonfun$getOrInferFileFormatSchema$11(DataSource.scala:208)
        at scala.Option.orElse(Option.scala:447)
        at org.apache.spark.sql.execution.datasources.DataSource.getOrInferFileFormatSchema(DataSource.scala:205)
        at org.apache.spark.sql.execution.datasources.DataSource.resolveRelation(DataSource.scala:407)
        at org.apache.spark.sql.DataFrameReader.loadV1Source(DataFrameReader.scala:229)
        at org.apache.spark.sql.DataFrameReader.$anonfun$load$2(DataFrameReader.scala:211)
        at scala.Option.getOrElse(Option.scala:189)
        at org.apache.spark.sql.DataFrameReader.load(DataFrameReader.scala:211)
        at org.apache.spark.sql.DataFrameReader.load(DataFrameReader.scala:186)
```