In [1]:
```
import findspark
findspark.init('/home/manojangane/spark/spark-2.2.0-bin-hadoop2.7')
```

In [2]:
```
from pyspark.sql import SparkSession
from pyspark.sql import SQLContext
```

In [3]:
```
spark = SparkSession \
       .builder \
       .appName("RecommendationSystem") \
       .getOrCreate()
```

In [4]:
```
sc = spark.sparkContext
sqlContext = SQLContext(sc)
```

In [5]:
```
# File location and type
file_song_data = "/home/manojangane/song_data.csv"
file_triplets_data ="/home/manojangane//Triplets.txt"
```

In [6]:
```
#Uploading Song CSV file with tab delimited
song_df = spark.read.csv(file_song_data,
                         inferSchema ='true',
                         header = 'true',
                         sep=',')


song_df.show()
```

```
+------------------+------------------+--------------------+---------
-----------+----+
|           song_id|             title|             release|
artist_name|year|
+------------------+------------------+--------------------+---------
-----------+----+
|SOQMMHC12AB0180CB8|       Silent Night|Monster Ballads X...|    Faste
r Pussy cat|2003|
|SOVFVAK12A8C1350D9|        Tanssi vaan|         Karkuteillä|    Karkk
iautomaatti|1995|
|SOGTUKN12AB017F4F1|  No One Could Ever|              Butter|      Hud
son Mohawke|2006|
|SOBNYVR12A8C13558C|      Si Vos Querés|             De Culo|
Yerba Brava|2003|
|SOHSBXH12A8C13B0DF|   Tangle Of Aspens|Rene Ablaze Prese...|
Der Mystic|   0|
|SOZVAPQ12A8C13B63C|"Symphony No. 1 G...|Berwald: Symphoni...|    David
Montgomery|   0|
|SOQVRHI12A6D4FB2D7|   We Have Got Love|Strictly The Best...|  Sasha /
Turbulence|   0|
|SOEYRFT12AB018936C|  2 Da Beat Ch'yall|             Da Bomb|
Kris Kross|1993|
|SOPMIYT12A6D4F851E|            Goodbye|           Danny Boy|        J
oseph Locke|   0|
|SOJCFMH12A8C13B0C2|Mama_ mama can't ...|March to cadence ...|The Sun H
arbor's ...|   0|
|SOYGNWH12AB018191E|      L'antarctique|Des cobras des ta...|    3 Gar
s Su'l Sofa|2007|
|SOLJTLX12AB01890ED|  El hijo del pueblo|32 Grandes Éxitos...|       Jo
rge Negrete|1997|
|SOQQESG12A58A7AA28|Cold Beer feat. P...|International Har...|        D
anny Diablo|   0|
|SOMPVQB12A8C1379BB|              Pilots|           The Loyal|
Tiger Lou|2005|
|SOGPCJI12A8C13CCA0|             N Gana|Afropea 3 - Telli...|     Wald
emar Bastos|   0|
|SOSDCFG12AB0184647|                006|          Lena 20 År|     Lena
Philipsson|1998|
|SOBARPM12A8C133DFF|(Looking For) The...|          Cover Girl|        S
hawn Colvin|1994|
|SOKOVRQ12A8C142811|  Ethos of Coercion|Descend Into Depr...|
Dying Fetus|2009|
|SOIMMJJ12AF72AD643|        Rock-N-Rule|I'm Only A Man (B...|
Emery|2007|
|SOVMBTP12A8C13A8F6|      La bola extra|       La bola extra|        L
os Ronaldos|   0|
+------------------+------------------+--------------------+---------
-----------+----+
only showing top 20 rows
```

In [7]:
```
from pyspark.sql.types import *

# Creating schema
schema = StructType([StructField('user_id', StringType()),
                     StructField('songid', StringType()),
                     StructField('play_count', IntegerType())])

#Uploading Triplets file with tab delimited
tri_df = spark.read.csv(file_triplets_data,
                        schema= schema,
                          sep='\t')


tri_df.show()
```

```
+--------------------+------------------+----------+
|             user_id|            songid|play_count|
+--------------------+------------------+----------+
|b80344d063b5ccb32...|SOAKIMP12A8C130995|         1|
|b80344d063b5ccb32...|SOBBMDR12A8C13253B|         2|
|b80344d063b5ccb32...|SOBXHDL12A81C204C0|         1|
|b80344d063b5ccb32...|SOBYHAJ12A6701BF1D|         1|
|b80344d063b5ccb32...|SODACBL12A8C13C273|         1|
|b80344d063b5ccb32...|SODDNQT12A6D4F5F7E|         5|
|b80344d063b5ccb32...|SODXRTY12AB0180F3B|         1|
|b80344d063b5ccb32...|SOFGUAY12AB017B0A8|         1|
|b80344d063b5ccb32...|SOFRQTD12A81C233C0|         1|
|b80344d063b5ccb32...|SOHQWYZ12A6D4FA701|         1|
|b80344d063b5ccb32...|SOIYTOA12A6D4F9A23|         1|
|b80344d063b5ccb32...|SOIZAZL12A6701C53B|         5|
|b80344d063b5ccb32...|SOJNNUA12A8AE48C7A|         1|
|b80344d063b5ccb32...|SOJPFQG12A58A7833A|         1|
|b80344d063b5ccb32...|SOKRIMP12A6D4F5DA3|         5|
|b80344d063b5ccb32...|SOLLGNU12AF72A4D4F|         1|
|b80344d063b5ccb32...|SOMGIYR12AB0187973|         6|
|b80344d063b5ccb32...|SOMLMKI12A81C204BC|         1|
|b80344d063b5ccb32...|SOMSQJY12A8C138539|         1|
|b80344d063b5ccb32...|SONSAEZ12A8C138D7A|         1|
+--------------------+------------------+----------+
only showing top 20 rows
```

In [8]:
```
MSD = tri_df.join(song_df, tri_df.songid == song_df.song_id,how='left')
MSD.show(5)
```

```
+--------------------+------------------+----------+------------------+
-----------------+------------------+-----------------+----+
|             user_id|            songid|play_count|           song_id|
title|            release|      artist_name|year|
+--------------------+------------------+----------+------------------+
-----------------+------------------+-----------------+----+
|5be795a4758aa4f13...|SOEYVHS12AB0181D31|         3|SOEYVHS12AB0181D31|
Monster|   The Fame Monster|        Lady GaGa|2009|
|8d4f1822e21f0a91f...|SOSAUVD12A67ADE6AE|         1|SOSAUVD12A67ADE6AE|
I Know It's Over|  The Queen Is Dead|       The Smiths|1986|
|5be795a4758aa4f13...|SOHNVHC12A6D4F95AB|         6|SOHNVHC12A6D4F95AB|
Elephant Gun| The Gulag Orkestar|           Beirut|2006|
|84cd5a870058d7d85...|SOZTCOW12A8C134269|         1|SOZTCOW12A8C134269|
Crazy On You|The Essential Heart|            Heart|   0|
|4f90d64332d78199f...|SOXKMJJ12AC468910D|         2|SOXKMJJ12AC468910D|
Hiatus (feat. NO)|Stand Up And Scream|Asking Alexandria|2009|
+--------------------+------------------+----------+------------------+
-----------------+------------------+-----------------+----+
only showing top 5 rows
```

In [9]:
```
MSD = MSD['user_id','song_id','play_count','title','release','artist_nam
e','year']
MSD.show(5)
```

```
+--------------------+------------------+----------+-----+------------
-----+-----------+----+
|             user_id|           song_id|play_count|title|           re
lease|artist_name|year|
+--------------------+------------------+----------+-----+------------
-----+-----------+----+
|79f93851e840f9d1f...|SOATHTW12A58A7EDB5|         1| Mutt|Enema Of The
State|  Blink-182|1998|
|043d81932e75d5749...|SOATHTW12A58A7EDB5|         5| Mutt|Enema Of The
State|  Blink-182|1998|
|ebacfcb5fa29a601f...|SOATHTW12A58A7EDB5|         1| Mutt|Enema Of The
State|  Blink-182|1998|
|417c73dd95669d191...|SOATHTW12A58A7EDB5|         1| Mutt|Enema Of The
State|  Blink-182|1998|
|52ab33fbb2fa3aeb2...|SOATHTW12A58A7EDB5|         1| Mutt|Enema Of The
State|  Blink-182|1998|
+--------------------+------------------+----------+-----+------------
-----+-----------+----+
only showing top 5 rows
```

In [10]:
```
MSD = MSD.withColumnRenamed('year', 'release_year')
MSD.show(5)
```

```
+--------------------+------------------+----------+-----+------------
-----+-----------+------------+
|             user_id|           song_id|play_count|title|           re
lease|artist_name|release_year|
+--------------------+------------------+----------+-----+------------
-----+-----------+------------+
|79f93851e840f9d1f...|SOATHTW12A58A7EDB5|         1| Mutt|Enema Of The
State|  Blink-182|        1998|
|043d81932e75d5749...|SOATHTW12A58A7EDB5|         5| Mutt|Enema Of The
State|  Blink-182|        1998|
|ebacfcb5fa29a601f...|SOATHTW12A58A7EDB5|         1| Mutt|Enema Of The
State|  Blink-182|        1998|
|417c73dd95669d191...|SOATHTW12A58A7EDB5|         1| Mutt|Enema Of The
State|  Blink-182|        1998|
|52ab33fbb2fa3aeb2...|SOATHTW12A58A7EDB5|         1| Mutt|Enema Of The
State|  Blink-182|        1998|
+--------------------+------------------+----------+-----+------------
-----+-----------+------------+
only showing top 5 rows
```

In [11]:
```
# Number of rows
MSD.count()
```

Out[11]: 2086946

In [12]:
```
# number of distinct user_Id
user=MSD.select("user_id").distinct()
user1,user2= user.randomSplit([0.05,0.95], seed=123)
usercount = user1.count()
print("Number of users: ", usercount)
```

```
Number of users:  3783
```

In [13]:
```
# number of distinct song_Id
songs= MSD.select("song_id").distinct()
songcount=songs.count()
print("Number of songs: ", songcount)
```

```
Number of songs:  10000
```

In [14]:
```
from pyspark.sql.functions import monotonically_increasing_id

# Creating new columns of unique integers for user_id and song_id
user_df = user1.withColumn("new_userid", monotonically_increasing_id())
user_df.show()
```

```
+--------------------+----------+
|             user_id|new_userid|
+--------------------+----------+
|126ef5859eb5e96f7...|         0|
|1d7b9780e492c062c...|         1|
|2c0815308dfd33b4b...|         2|
|3d24b9ffed6a82778...|         3|
|3dd54878cb47456b8...|         4|
|459ab8388e7806755...|         5|
|5e7105e485a04bf0b...|         6|
|6bb6fa9a23505dc55...|         7|
|739f32d4c2690554b...|         8|
|7a3943dfa7f83e321...|         9|
|7de4388c64742657d...|        10|
|9619f405e777e8331...|        11|
|a2c1d795852bd22c6...|        12|
|a8e8fd13a2909af99...|        13|
|aab99ec2a563f732e...|        14|
|c06d794619168b4bf...|        15|
|c2cfc654c54fddc9f...|        16|
|e43304402c7407bc4...|        17|
|e6bf98dccce485c26...|        18|
|e7fc73d0eb0d851bc...|        19|
+--------------------+----------+
only showing top 20 rows
```

In [15]:
```
songs_df = songs.select("song_id", monotonically_increasing_id().alias(
'new_songId'))
songs_df.show()
```

```
+------------------+----------+
|           song_id|new_songId|
+------------------+----------+
|SOATHTW12A58A7EDB5|         0|
|SOAZMXH12AB0186DDE|         1|
|SOBAQTV12A8C142277|         2|
|SOCKUUJ12A6D4FA41C|         3|
|SOCUVKX12A6D4F8ED7|         4|
|SODABFB12A58A81788|         5|
|SODASIJ12A6D4F5D89|         6|
|SODYTRD12A81C2329F|         7|
|SOECOOL12AB0181A2F|         8|
|SOECTGX12A6310E233|         9|
|SOERLLT12AC468DAF3|        10|
|SOGKEGN12AB0185355|        11|
|SOGXQYC12AB0183AE5|        12|
|SOHXDTJ12A81C219C2|        13|
|SOICVFJ12A8AE47FF0|        14|
|SOJGIUN12A6BD55B8E|        15|
|SOKOVZK12A6D4F707F|        16|
|SOKQHXV12AB0185B3D|        17|
|SOKUAGP12A8C133B94|        18|
|SOLIVXX12A6D4F7950|        19|
+------------------+----------+
only showing top 20 rows
```

In [16]:
```
#Cross Join user and Songs
crossjoin = user_df.crossJoin(songs_df)
crossjoin.show(5)
```

```
+--------------------+----------+------------------+----------+
|             user_id|new_userid|           song_id|new_songId|
+--------------------+----------+------------------+----------+
|126ef5859eb5e96f7...|         0|SOATHTW12A58A7EDB5|         0|
|126ef5859eb5e96f7...|         0|SOAZMXH12AB0186DDE|         1|
|126ef5859eb5e96f7...|         0|SOBAQTV12A8C142277|         2|
|126ef5859eb5e96f7...|         0|SOCKUUJ12A6D4FA41C|         3|
|126ef5859eb5e96f7...|         0|SOCUVKX12A6D4F8ED7|         4|
+--------------------+----------+------------------+----------+
only showing top 5 rows
```

In [17]:
```
crossjoin.count()
```

Out[17]: 37830000

In [18]:
```
df = crossjoin.join(MSD, ["user_id", "song_id"], "left").fillna(0)
```

In [19]:
```
model_df= df.select(df.new_userid.cast("int"),df.new_songId.cast("int"),
df.play_count.cast("int"))
```

In [20]:
```
# Set the ALS hyperparameters
from pyspark.ml.recommendation import ALS

model = ALS(userCol= "new_userid", itemCol= "new_songId", ratingCol= "pl
ay_count", rank = 10, maxIter = 10,alpha = 20, regParam = .05,  coldStar
tStrategy="drop", nonnegative = True, implicitPrefs = True)
```

In [21]:
```
# Split the dataframe into training and test data
(train_data, test_data) = model_df.select('new_userid','new_songId','pla
y_count').randomSplit([0.7, 0.3], seed=12345)
```

In [22]:
```
 #Expected percentile rank error metric function
def ROEM(predictions, userCol = "new_userid", itemCol = "new_songId", ra
tingCol = "play_count"):
  #Creates table that can be queried
  predictions.createOrReplaceTempView("predictions")

  #Sum of total number of plays of all songs
  denominator = predictions.groupBy().sum(ratingCol).collect()[0][0]

  #Calculating rankings of songs predictions by user
  spark.sql("SELECT " + userCol + " , " + ratingCol + " , PERCENT_RANK()
OVER (PARTITION BY " + userCol + " ORDER BY prediction DESC) AS rank FRO
M predictions").createOrReplaceTempView("rankings")

   #Multiplies the rank of each song by the number of plays and adds the
products together
  numerator = spark.sql('SELECT SUM(' + ratingCol + ' * rank) FROM ranki
ngs').collect()[0][0]

  performance = numerator/denominator

  return performance
```

In [23]:
```
train_data.cache()
```

Out[23]:
```
DataFrame[new_userid: int, new_songId: int, play_count: int]
```

In [24]:
```
# Fits model to fold within training data
fitted_model = model.fit(train_data)
```

```
---------------------------------------------------------------------------
Py4JJavaError                             Traceback (most recent call last)
<ipython-input-24-2013cb48fc1b> in <module>()
      1 # Fits model to fold within training data
----> 2 fitted_model = model.fit(train_data)

~/spark/spark-2.2.0-bin-hadoop2.7/python/pyspark/ml/base.py in fit(self, dataset, params)
     62                 return self.copy(params)._fit(dataset)
     63             else:
---> 64                 return self._fit(dataset)
     65         else:
     66             raise ValueError("Params must be either a param map or a list/tuple of param maps, "

~/spark/spark-2.2.0-bin-hadoop2.7/python/pyspark/ml/wrapper.py in _fit(self, dataset)
    263
    264     def _fit(self, dataset):
--> 265         java_model = self._fit_java(dataset)
    266         return self._create_model(java_model)
    267

~/spark/spark-2.2.0-bin-hadoop2.7/python/pyspark/ml/wrapper.py in _fit_java(self, dataset)
    260         """
    261         self._transfer_params_to_java()
--> 262         return self._java_obj.fit(dataset._jdf)
    263
    264     def _fit(self, dataset):

~/spark/spark-2.2.0-bin-hadoop2.7/python/lib/py4j-0.10.4-src.zip/py4j/java_gateway.py in __call__(self, *args)
   1131         answer = self.gateway_client.send_command(command)
   1132         return_value = get_return_value(
-> 1133             answer, self.gateway_client, self.target_id, self.name)
   1134
   1135         for temp_arg in temp_args:

~/spark/spark-2.2.0-bin-hadoop2.7/python/pyspark/sql/utils.py in deco(*a, **kw)
     61     def deco(*a, **kw):
     62         try:
---> 63             return f(*a, **kw)
     64         except py4j.protocol.Py4JJavaError as e:
     65             s = e.java_exception.toString()

~/spark/spark-2.2.0-bin-hadoop2.7/python/lib/py4j-0.10.4-src.zip/py4j/protocol.py in get_return_value(answer, gateway_client, target_id, name)
    317                 raise Py4JJavaError(
    318                     "An error occurred while calling {0}{1}{2}.\n".
--> 319                     format(target_id, ".", name), value)
    320             else:
```

```
    321                 raise Py4JError(

Py4JJavaError: An error occurred while calling o101.fit.
: org.apache.spark.SparkException: Job aborted due to stage failure: Ta
sk 0 in stage 76.0 failed 1 times, most recent failure: Lost task 0.0 i
n stage 76.0 (TID 83485, localhost, executor driver): java.lang.OutOfMe
moryError: Java heap space
        at scala.collection.mutable.ArrayBuilder$ofInt.mkArray(ArrayBui
lder.scala:323)
        at scala.collection.mutable.ArrayBuilder$ofInt.resize(ArrayBuil
der.scala:329)
        at scala.collection.mutable.ArrayBuilder$ofInt.ensureSize(Array
Builder.scala:341)
        at scala.collection.mutable.ArrayBuilder$ofInt.$plus$eq(ArrayBu
ilder.scala:346)
        at scala.collection.mutable.ArrayBuilder$ofInt.$plus$eq(ArrayBu
ilder.scala:316)
        at scala.collection.generic.Growable$$anonfun$$plus$plus$eq$1.a
pply(Growable.scala:59)
        at scala.collection.generic.Growable$$anonfun$$plus$plus$eq$1.a
pply(Growable.scala:59)
        at scala.collection.IndexedSeqOptimized$class.foreach(IndexedSe
qOptimized.scala:33)
        at scala.collection.mutable.ArrayOps$ofInt.foreach(ArrayOps.sca
la:234)
        at scala.collection.generic.Growable$class.$plus$plus$eq(Growab
le.scala:59)
        at scala.collection.mutable.ArrayBuilder$ofInt.$plus$plus$eq(Ar
rayBuilder.scala:359)
        at scala.collection.mutable.ArrayBuilder$ofInt.$plus$plus$eq(Ar
rayBuilder.scala:316)
        at org.apache.spark.ml.recommendation.ALS$UncompressedInBlockBu
ilder.add(ALS.scala:1158)
        at org.apache.spark.ml.recommendation.ALS$$anonfun$23$$anonfun
$apply$16.apply(ALS.scala:1376)
        at org.apache.spark.ml.recommendation.ALS$$anonfun$23$$anonfun
$apply$16.apply(ALS.scala:1375)
        at scala.collection.Iterator$class.foreach(Iterator.scala:893)
        at org.apache.spark.util.collection.CompactBuffer$$anon$1.forea
ch(CompactBuffer.scala:115)
        at scala.collection.IterableLike$class.foreach(IterableLike.sca
la:72)
        at org.apache.spark.util.collection.CompactBuffer.foreach(Compa
ctBuffer.scala:30)
        at org.apache.spark.ml.recommendation.ALS$$anonfun$23.apply(AL
S.scala:1375)
        at org.apache.spark.ml.recommendation.ALS$$anonfun$23.apply(AL
S.scala:1372)
        at org.apache.spark.rdd.PairRDDFunctions$$anonfun$mapValues$1
$$anonfun$apply$40$$anonfun$apply$41.apply(PairRDDFunctions.scala:760)
        at org.apache.spark.rdd.PairRDDFunctions$$anonfun$mapValues$1
$$anonfun$apply$40$$anonfun$apply$41.apply(PairRDDFunctions.scala:760)
        at scala.collection.Iterator$$anon$11.next(Iterator.scala:409)
        at org.apache.spark.storage.memory.MemoryStore.putIteratorAsVal
ues(MemoryStore.scala:216)
        at org.apache.spark.storage.BlockManager$$anonfun$doPutIterator
$1.apply(BlockManager.scala:1038)
```

```
        at org.apache.spark.storage.BlockManager$$anonfun$doPutIterator
$1.apply(BlockManager.scala:1029)
        at org.apache.spark.storage.BlockManager.doPut(BlockManager.sca
la:969)
        at org.apache.spark.storage.BlockManager.doPutIterator(BlockMan
ager.scala:1029)
        at org.apache.spark.storage.BlockManager.getOrElseUpdate(BlockM
anager.scala:760)
        at org.apache.spark.rdd.RDD.getOrCompute(RDD.scala:334)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:285)

Driver stacktrace:
        at org.apache.spark.scheduler.DAGScheduler.org$apache$spark$sch
eduler$DAGScheduler$$failJobAndIndependentStages(DAGScheduler.scala:149
9)
        at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage
$1.apply(DAGScheduler.scala:1487)
        at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage
$1.apply(DAGScheduler.scala:1486)
        at scala.collection.mutable.ResizableArray$class.foreach(Resiza
bleArray.scala:59)
        at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.sca
la:48)
        at org.apache.spark.scheduler.DAGScheduler.abortStage(DAGSchedu
ler.scala:1486)
        at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskS
etFailed$1.apply(DAGScheduler.scala:814)
        at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskS
etFailed$1.apply(DAGScheduler.scala:814)
        at scala.Option.foreach(Option.scala:257)
        at org.apache.spark.scheduler.DAGScheduler.handleTaskSetFailed
(DAGScheduler.scala:814)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.doOn
Receive(DAGScheduler.scala:1714)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onRe
ceive(DAGScheduler.scala:1669)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onRe
ceive(DAGScheduler.scala:1658)
        at org.apache.spark.util.EventLoop$$anon$1.run(EventLoop.scala:
48)
        at org.apache.spark.scheduler.DAGScheduler.runJob(DAGScheduler.
scala:630)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:202
2)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:204
3)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:206
2)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:208
7)
        at org.apache.spark.rdd.RDD.count(RDD.scala:1158)
        at org.apache.spark.ml.recommendation.ALS$.train(ALS.scala:857)
        at org.apache.spark.ml.recommendation.ALS.fit(ALS.scala:622)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAcce
ssorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMe
```

```
thodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:244)
        at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.jav
a:357)
        at py4j.Gateway.invoke(Gateway.java:280)
        at py4j.commands.AbstractCommand.invokeMethod(AbstractCommand.j
ava:132)
        at py4j.commands.CallCommand.execute(CallCommand.java:79)
        at py4j.GatewayConnection.run(GatewayConnection.java:214)
        at java.lang.Thread.run(Thread.java:748)
Caused by: java.lang.OutOfMemoryError: Java heap space
        at scala.collection.mutable.ArrayBuilder$ofInt.mkArray(ArrayBui
lder.scala:323)
        at scala.collection.mutable.ArrayBuilder$ofInt.resize(ArrayBuil
der.scala:329)
        at scala.collection.mutable.ArrayBuilder$ofInt.ensureSize(Array
Builder.scala:341)
        at scala.collection.mutable.ArrayBuilder$ofInt.$plus$eq(ArrayBu
ilder.scala:346)
        at scala.collection.mutable.ArrayBuilder$ofInt.$plus$eq(ArrayBu
ilder.scala:316)
        at scala.collection.generic.Growable$$anonfun$$plus$plus$eq$1.a
pply(Growable.scala:59)
        at scala.collection.generic.Growable$$anonfun$$plus$plus$eq$1.a
pply(Growable.scala:59)
        at scala.collection.IndexedSeqOptimized$class.foreach(IndexedSe
qOptimized.scala:33)
        at scala.collection.mutable.ArrayOps$ofInt.foreach(ArrayOps.sca
la:234)
        at scala.collection.generic.Growable$class.$plus$plus$eq(Growab
le.scala:59)
        at scala.collection.mutable.ArrayBuilder$ofInt.$plus$plus$eq(Ar
rayBuilder.scala:359)
        at scala.collection.mutable.ArrayBuilder$ofInt.$plus$plus$eq(Ar
rayBuilder.scala:316)
        at org.apache.spark.ml.recommendation.ALS$UncompressedInBlockBu
ilder.add(ALS.scala:1158)
        at org.apache.spark.ml.recommendation.ALS$$anonfun$23$$anonfun
$apply$16.apply(ALS.scala:1376)
        at org.apache.spark.ml.recommendation.ALS$$anonfun$23$$anonfun
$apply$16.apply(ALS.scala:1375)
        at scala.collection.Iterator$class.foreach(Iterator.scala:893)
        at org.apache.spark.util.collection.CompactBuffer$$anon$1.forea
ch(CompactBuffer.scala:115)
        at scala.collection.IterableLike$class.foreach(IterableLike.sca
la:72)
        at org.apache.spark.util.collection.CompactBuffer.foreach(Compa
ctBuffer.scala:30)
        at org.apache.spark.ml.recommendation.ALS$$anonfun$23.apply(AL
S.scala:1375)
        at org.apache.spark.ml.recommendation.ALS$$anonfun$23.apply(AL
S.scala:1372)
        at org.apache.spark.rdd.PairRDDFunctions$$anonfun$mapValues$1
$$anonfun$apply$40$$anonfun$apply$41.apply(PairRDDFunctions.scala:760)
        at org.apache.spark.rdd.PairRDDFunctions$$anonfun$mapValues$1
$$anonfun$apply$40$$anonfun$apply$41.apply(PairRDDFunctions.scala:760)
```

```
        at scala.collection.Iterator$$anon$11.next(Iterator.scala:409)
        at org.apache.spark.storage.memory.MemoryStore.putIteratorAsVal
ues(MemoryStore.scala:216)
        at org.apache.spark.storage.BlockManager$$anonfun$doPutIterator
$1.apply(BlockManager.scala:1038)
        at org.apache.spark.storage.BlockManager$$anonfun$doPutIterator
$1.apply(BlockManager.scala:1029)
        at org.apache.spark.storage.BlockManager.doPut(BlockManager.sca
la:969)
        at org.apache.spark.storage.BlockManager.doPutIterator(BlockMan
ager.scala:1029)
        at org.apache.spark.storage.BlockManager.getOrElseUpdate(BlockM
anager.scala:760)
        at org.apache.spark.rdd.RDD.getOrCompute(RDD.scala:334)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:285)
```

In [ ]:
```
# Generates predictions using fitted_model on respective CV test data
predictions = fitted_model.transform(test_data)
```

In [ ]:
```
# Generates and prints a ROEM metric CV test data
validation_performance = ROEM(predictions)
print(validation_performance)
```