确定als的三个参数

因为项目结束已经没多少时间了，所以要加速。故采用比较简单的，范围比较小的测试方法。

先确定lamdbda的数量级

（理论上应该rank，iterations，lambda三重循环判断，这里数量级比较大，就先判断lambda数量级）

确定大概的范围，比如0.1-0.2，之后再三重循环判断最好的参数。

判断方法为将数据集分为训练集，测试集。训练集根据三个参数训练模型后，测试集测试，计算RMSE(均方根误差)，认为RMSE越小越好

设定rank=5，iterations=10，然后给定几个数量级的lambda，计算RMSE，评估其数量级

step1：

测试代码：设定rank=5，iterations=10，lambda range in 0.001,0.005,0.1,0.3,0.6,1.0,3.0,6.0,10.0

# 使用Spark MLlib中推荐算法ALS对电影评分数据MovieLens推荐

from pyspark.sql import SparkSession

from pyspark.mllib.recommendation import ALS, Rating, MatrixFactorizationModel

from pyspark.mllib.regression import LabeledPoint, LinearRegressionWithSGD

from pyspark.mllib.evaluation import RegressionMetrics

from pyspark.mllib.linalg import DenseVector

import numpy as np #arange

def alsModelEvaluate(model, testing\_rdd):

# 对测试数据集预测评分，针对测试数据集进行预测

predict\_rdd = model.predictAll(testing\_rdd.map(lambda r: (r[0], r[1])))

#print(predict\_rdd.take(5))

predict\_actual\_rdd = predict\_rdd.map(lambda r: ((r[0], r[1]), r[2])) \

.join(testing\_ratings.map(lambda r: ((r[0], r[1]), r[2])))

#print(predict\_actual\_rdd.take(5))

# 创建评估指标实例对象

metrics = RegressionMetrics(predict\_actual\_rdd.map(lambda pr: pr[1]))

print("MSE = %s" % metrics.meanSquaredError)

print("RMSE = %s" % metrics.rootMeanSquaredError)

# 返回均方根误差

return metrics.rootMeanSquaredError

def train\_model\_evaluate(training\_rdd, testing\_rdd, rank, iterations, lambda\_):

# 定义函数，训练模型与模型评估

# 使用超参数的值，训练数据和ALS算法训练模型

print(lambda\_)

model = ALS.train(training\_rdd, rank, iterations, lambda\_)

# 模型的评估

rmse\_value = alsModelEvaluate(model, testing\_rdd)

# 返回多元组

return (model, rmse\_value, rank, iterations, lambda\_)

if \_\_name\_\_ == "\_\_main\_\_":

# 构建SparkSession实例对象

spark = SparkSession.builder \

.appName("SparkSessionExample") \

.master("local") \

.getOrCreate()

# 获取SparkContext实例对象

sc = spark.sparkContext

# 读取数据

raw\_ratings\_rdd = sc.textFile("C:/Users/lenovo/Desktop/ml-100k/ml-100k/u.data")

# print(raw\_ratings\_rdd.count())

# print(raw\_ratings\_rdd.first())

# 获取评分数据前三个字段，构建Rating实例对象

ratings\_rdd = raw\_ratings\_rdd.map(lambda line: line.split('\t')[0:3])

# print(ratings\_rdd.first())

ratings\_datas = ratings\_rdd.map(lambda x: Rating(int(x[0]), int(x[1]), float(x[2])))

# print(ratings\_datas.first())

# 将数据集分为训练数据集和测试数据集

training\_ratings, testing\_ratings = ratings\_datas.randomSplit([0.7, 0.3])

# 怎么评价模型的好坏，ALS模型评估指标(类似回归算法模型预测值，连续值)，使用回归模型中

# RMSE（均方根误差）评估模型

# 找到最佳模型

'''

如何找到最佳模型？？

-a. 模型的评估

计算RMSE

-b. 模型的优化，两个方向

1、数据

2、超参数的调整，选择合适的超参数的值，得到最优模型

交叉验证

训练数据集、验证数据集、测试数据集

K-Folds交叉验证

'''

# ALS算法的超参数的调整

# 定义一个函数，用于对模型进行评估

# 使用三层for循环，设置不同参数的值，分别使用ALS算法训练模型，评估获取RMSE的值

metrix\_list = [train\_model\_evaluate(training\_ratings, testing\_ratings, 5, 10, param\_lambda)

for param\_lambda in [0.001,0.005,0.1,0.3,0.6,1.0,3.0,6.0,10.0]

]

print(type(metrix\_list))

print("\n")

#sorted(metrix\_list, key=lambda k: k[1], reverse=False)

print(metrix\_list)

print("\n")

metrix\_list.sort(key=lambda s: s[1])

print(metrix\_list)

print("\n")

model, rmse\_value, rank, iterations, lambda\_ = metrix\_list[0]

print("The best parameters, rank=%s, iterations=%s, lambda\_=%s" % (rank ,iterations ,lambda\_))

结果：

0.001

MSE = 1.45551234218971

RMSE = 1.206446162159634

0.005

MSE = 1.1167293871452393

RMSE = 1.0567541753621035

0.1

MSE = 0.8660313737704743

RMSE = 0.9306080666803154

0.3

MSE = 0.9581848081322016

RMSE = 0.9788691476046232

0.6

MSE = 1.2537802656498287

RMSE = 1.1197232986992047

1.0

MSE = 1.895263148482993

RMSE = 1.3766855663088042

3.0

MSE = 11.880223928905217

RMSE = 3.446770071952177

6.0

MSE = 13.741151257362338

RMSE = 3.7069058873084892

10.0

MSE = 13.741151257408834

RMSE = 3.7069058873147607

<class 'list'>

[(<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169786D8>, 1.206446162159634, 5, 10, 0.001), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169D4438>, 1.0567541753621035, 5, 10, 0.005), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169D7898>, 0.9306080666803154, 5, 10, 0.1), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16978B70>, 0.9788691476046232, 5, 10, 0.3), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169F6BA8>, 1.1197232986992047, 5, 10, 0.6), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169F6DA0>, 1.3766855663088042, 5, 10, 1.0), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169D4320>, 3.446770071952177, 5, 10, 3.0), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16978748>, 3.7069058873084892, 5, 10, 6.0), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16A05320>, 3.7069058873147607, 5, 10, 10.0)]

[(<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169D7898>, 0.9306080666803154, 5, 10, 0.1), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16978B70>, 0.9788691476046232, 5, 10, 0.3), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169D4438>, 1.0567541753621035, 5, 10, 0.005), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169F6BA8>, 1.1197232986992047, 5, 10, 0.6), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169786D8>, 1.206446162159634, 5, 10, 0.001), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169F6DA0>, 1.3766855663088042, 5, 10, 1.0), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169D4320>, 3.446770071952177, 5, 10, 3.0), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16978748>, 3.7069058873084892, 5, 10, 6.0), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16A05320>, 3.7069058873147607, 5, 10, 10.0)]

The best parameters, rank=5, iterations=10, lambda\_=0.1

Step 2：

确定最好的lambda在0.1，0.3附近。数量级设为0.1继续判断

测试代码：设定rank=5，iterations=10，lambda range in

0.1~0.5

# 使用Spark MLlib中推荐算法ALS对电影评分数据MovieLens推荐

from pyspark.sql import SparkSession

from pyspark.mllib.recommendation import ALS, Rating, MatrixFactorizationModel

from pyspark.mllib.regression import LabeledPoint, LinearRegressionWithSGD

from pyspark.mllib.evaluation import RegressionMetrics

from pyspark.mllib.linalg import DenseVector

import numpy as np #arange

def alsModelEvaluate(model, testing\_rdd):

# 对测试数据集预测评分，针对测试数据集进行预测

predict\_rdd = model.predictAll(testing\_rdd.map(lambda r: (r[0], r[1])))

#print(predict\_rdd.take(5))

predict\_actual\_rdd = predict\_rdd.map(lambda r: ((r[0], r[1]), r[2])) \

.join(testing\_ratings.map(lambda r: ((r[0], r[1]), r[2])))

#print(predict\_actual\_rdd.take(5))

# 创建评估指标实例对象

metrics = RegressionMetrics(predict\_actual\_rdd.map(lambda pr: pr[1]))

print("MSE = %s" % metrics.meanSquaredError)

print("RMSE = %s" % metrics.rootMeanSquaredError)

# 返回均方根误差

return metrics.rootMeanSquaredError

def train\_model\_evaluate(training\_rdd, testing\_rdd, rank, iterations, lambda\_):

# 定义函数，训练模型与模型评估

# 使用超参数的值，训练数据和ALS算法训练模型

print(lambda\_)

model = ALS.train(training\_rdd, rank, iterations, lambda\_)

# 模型的评估

rmse\_value = alsModelEvaluate(model, testing\_rdd)

# 返回多元组

return (model, rmse\_value, rank, iterations, lambda\_)

if \_\_name\_\_ == "\_\_main\_\_":

# 构建SparkSession实例对象

spark = SparkSession.builder \

.appName("SparkSessionExample") \

.master("local") \

.getOrCreate()

# 获取SparkContext实例对象

sc = spark.sparkContext

# 读取数据

raw\_ratings\_rdd = sc.textFile("C:/Users/lenovo/Desktop/ml-100k/ml-100k/u.data")

# print(raw\_ratings\_rdd.count())

# print(raw\_ratings\_rdd.first())

# 获取评分数据前三个字段，构建Rating实例对象

ratings\_rdd = raw\_ratings\_rdd.map(lambda line: line.split('\t')[0:3])

# print(ratings\_rdd.first())

ratings\_datas = ratings\_rdd.map(lambda x: Rating(int(x[0]), int(x[1]), float(x[2])))

# print(ratings\_datas.first())

# 查看评分数据中有多少电影

# print(ratings\_datas.map(lambda x: x[1]).distinct().count())

# 查看评分数据中有多少用户

# print(ratings\_datas.map(lambda x: x[0]).distinct().count())

# 将数据集分为训练数据集和测试数据集

training\_ratings, testing\_ratings = ratings\_datas.randomSplit([0.7, 0.3])

# 怎么评价模型的好坏，ALS模型评估指标(类似回归算法模型预测值，连续值)，使用回归模型中

# RMSE（均方根误差）评估模型

# 找到最佳模型

'''

如何找到最佳模型？？

-a. 模型的评估

计算RMSE

-b. 模型的优化，两个方向

1、数据

2、超参数的调整，选择合适的超参数的值，得到最优模型

交叉验证

训练数据集、验证数据集、测试数据集

K-Folds交叉验证

'''

# ALS算法的超参数的调整

# 定义一个函数，用于对模型进行评估

# 使用三层for循环，设置不同参数的值，分别使用ALS算法训练模型，评估获取RMSE的值

#metrix\_list是tuple构成的list，形如metrix

metrix\_list = [train\_model\_evaluate(training\_ratings, testing\_ratings, 5, 10, param\_lambda)

#for param\_lambda in np.arange(0,0.5,0.1) 0不行。因为精度问题会到负数？

for param\_lambda in np.arange(0.1,0.5,0.1)

]

print(type(metrix\_list))

print("\n")

#sorted(metrix\_list, key=lambda k: k[1], reverse=False)

print(metrix\_list)

print("\n")

metrix\_list.sort(key=lambda s: s[1])

print(metrix\_list)

print("\n")

model, rmse\_value, rank, iterations, lambda\_ = metrix\_list[0]

print("The best parameters, rank=%s, iterations=%s, lambda\_=%s" % (rank ,iterations ,lambda\_))

结果：

0.1

MSE = 0.8559010486923162

RMSE = 0.925149203476021

0.2

MSE = 0.8840839540126963

RMSE = 0.9402573871088152

0.30000000000000004

MSE = 0.9596912035392299

RMSE = 0.9796383024051427

0.4

MSE = 1.0451959455152031

RMSE = 1.0223482506050485

<class 'list'>

[(<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169C5D68>, 0.925149203476021, 5, 10, 0.1), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E13DED208>, 0.9402573871088152, 5, 10, 0.2), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169D7240>, 0.9796383024051427, 5, 10, 0.30000000000000004), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16A0AC88>, 1.0223482506050485, 5, 10, 0.4)]

[(<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169C5D68>, 0.925149203476021, 5, 10, 0.1), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E13DED208>, 0.9402573871088152, 5, 10, 0.2), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169D7240>, 0.9796383024051427, 5, 10, 0.30000000000000004), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16A0AC88>, 1.0223482506050485, 5, 10, 0.4)]

The best parameters, rank=5, iterations=10, lambda\_=0.1

Step3：

0.1附近比较好。继续提高精度

测试代码：设定rank=5，iterations=10，lambda range in

0.01~0.15

# 使用Spark MLlib中推荐算法ALS对电影评分数据MovieLens推荐

from pyspark.sql import SparkSession

from pyspark.mllib.recommendation import ALS, Rating, MatrixFactorizationModel

from pyspark.mllib.regression import LabeledPoint, LinearRegressionWithSGD

from pyspark.mllib.evaluation import RegressionMetrics

from pyspark.mllib.linalg import DenseVector

import numpy as np #arange

def alsModelEvaluate(model, testing\_rdd):

# 对测试数据集预测评分，针对测试数据集进行预测

predict\_rdd = model.predictAll(testing\_rdd.map(lambda r: (r[0], r[1])))

#print(predict\_rdd.take(5))

predict\_actual\_rdd = predict\_rdd.map(lambda r: ((r[0], r[1]), r[2])) \

.join(testing\_ratings.map(lambda r: ((r[0], r[1]), r[2])))

#print(predict\_actual\_rdd.take(5))

# 创建评估指标实例对象

metrics = RegressionMetrics(predict\_actual\_rdd.map(lambda pr: pr[1]))

print("MSE = %s" % metrics.meanSquaredError)

print("RMSE = %s" % metrics.rootMeanSquaredError)

# 返回均方根误差

return metrics.rootMeanSquaredError

def train\_model\_evaluate(training\_rdd, testing\_rdd, rank, iterations, lambda\_):

# 定义函数，训练模型与模型评估

# 使用超参数的值，训练数据和ALS算法训练模型

print(lambda\_)

model = ALS.train(training\_rdd, rank, iterations, lambda\_)

# 模型的评估

rmse\_value = alsModelEvaluate(model, testing\_rdd)

# 返回多元组

return (model, rmse\_value, rank, iterations, lambda\_)

if \_\_name\_\_ == "\_\_main\_\_":

# 构建SparkSession实例对象

spark = SparkSession.builder \

.appName("SparkSessionExample") \

.master("local") \

.getOrCreate()

# 获取SparkContext实例对象

sc = spark.sparkContext

# 读取数据

raw\_ratings\_rdd = sc.textFile("C:/Users/lenovo/Desktop/ml-100k/ml-100k/u.data")

# print(raw\_ratings\_rdd.count())

# print(raw\_ratings\_rdd.first())

# 获取评分数据前三个字段，构建Rating实例对象

ratings\_rdd = raw\_ratings\_rdd.map(lambda line: line.split('\t')[0:3])

# print(ratings\_rdd.first())

ratings\_datas = ratings\_rdd.map(lambda x: Rating(int(x[0]), int(x[1]), float(x[2])))

# print(ratings\_datas.first())

# 查看评分数据中有多少电影

# print(ratings\_datas.map(lambda x: x[1]).distinct().count())

# 查看评分数据中有多少用户

# print(ratings\_datas.map(lambda x: x[0]).distinct().count())

# 将数据集分为训练数据集和测试数据集

training\_ratings, testing\_ratings = ratings\_datas.randomSplit([0.7, 0.3])

# 怎么评价模型的好坏，ALS模型评估指标(类似回归算法模型预测值，连续值)，使用回归模型中

# RMSE（均方根误差）评估模型

# 找到最佳模型

'''

如何找到最佳模型？？

-a. 模型的评估

计算RMSE

-b. 模型的优化，两个方向

1、数据

2、超参数的调整，选择合适的超参数的值，得到最优模型

交叉验证

训练数据集、验证数据集、测试数据集

K-Folds交叉验证

'''

# ALS算法的超参数的调整

# 定义一个函数，用于对模型进行评估

# 使用三层for循环，设置不同参数的值，分别使用ALS算法训练模型，评估获取RMSE的值

#metrix\_list是tuple构成的list，形如metrix

metrix\_list = [train\_model\_evaluate(training\_ratings, testing\_ratings, 5, 10, param\_lambda)

#for param\_lambda in np.arange(0,0.5,0.1) 0不行。因为精度问题会到负数？

for param\_lambda in np.arange(0.01,0.15,0.01)

]

print(type(metrix\_list))

print("\n")

#sorted(metrix\_list, key=lambda k: k[1], reverse=False)

print(metrix\_list)

print("\n")

metrix\_list.sort(key=lambda s: s[1])

print(metrix\_list)

print("\n")

model, rmse\_value, rank, iterations, lambda\_ = metrix\_list[0]

print("The best parameters, rank=%s, iterations=%s, lambda\_=%s" % (rank ,iterations ,lambda\_))

结果：

0.01

MSE = 1.0407685560127773

RMSE = 1.0201806487151073

0.02

MSE = 0.9845788753295515

RMSE = 0.9922594798385912

0.03

MSE = 0.961870553750992

RMSE = 0.9807499955396339

0.04

MSE = 0.9209323411107041

RMSE = 0.95965219799191

0.05

MSE = 0.9134677432325272

RMSE = 0.9557550644555995

0.060000000000000005

MSE = 0.8962691750998641

RMSE = 0.9467149386694308

0.06999999999999999

MSE = 0.894590124606308

RMSE = 0.9458277457371972

0.08

MSE = 0.8800128038484955

RMSE = 0.9380899764140407

0.09

MSE = 0.876466188642602

RMSE = 0.9361977294581535

0.09999999999999999

MSE = 0.8715220478026119

RMSE = 0.9335534520329363

0.11

MSE = 0.8682564585632947

RMSE = 0.9318028002551263

0.12

MSE = 0.8693271321803075

RMSE = 0.9323771405286101

0.13

MSE = 0.8784523478345924

RMSE = 0.937257887581957

0.14

MSE = 0.8691752269925992

RMSE = 0.9322956757341521

<class 'list'>

[(<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E13DED4E0>, 1.0201806487151073, 5, 10, 0.01), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E158D84E0>, 0.9922594798385912, 5, 10, 0.02), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169F60F0>, 0.9807499955396339, 5, 10, 0.03), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16768828>, 0.95965219799191, 5, 10, 0.04), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169C55F8>, 0.9557550644555995, 5, 10, 0.05), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16A05400>, 0.9467149386694308, 5, 10, 0.060000000000000005), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E1695DC18>, 0.9458277457371972, 5, 10, 0.06999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16A0A0B8>, 0.9380899764140407, 5, 10, 0.08), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169F64A8>, 0.9361977294581535, 5, 10, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16A0A978>, 0.9335534520329363, 5, 10, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169C5940>, 0.9318028002551263, 5, 10, 0.11), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E13DE44A8>, 0.9323771405286101, 5, 10, 0.12), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169F6E48>, 0.937257887581957, 5, 10, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16978358>, 0.9322956757341521, 5, 10, 0.14)]

[(<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169C5940>, 0.9318028002551263, 5, 10, 0.11), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16978358>, 0.9322956757341521, 5, 10, 0.14), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E13DE44A8>, 0.9323771405286101, 5, 10, 0.12), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16A0A978>, 0.9335534520329363, 5, 10, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169F64A8>, 0.9361977294581535, 5, 10, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169F6E48>, 0.937257887581957, 5, 10, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16A0A0B8>, 0.9380899764140407, 5, 10, 0.08), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E1695DC18>, 0.9458277457371972, 5, 10, 0.06999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16A05400>, 0.9467149386694308, 5, 10, 0.060000000000000005), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169C55F8>, 0.9557550644555995, 5, 10, 0.05), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E16768828>, 0.95965219799191, 5, 10, 0.04), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E169F60F0>, 0.9807499955396339, 5, 10, 0.03), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E158D84E0>, 0.9922594798385912, 5, 10, 0.02), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x0000023E13DED4E0>, 1.0201806487151073, 5, 10, 0.01)]

The best parameters, rank=5, iterations=10, lambda\_=0.11

Step 4：

得知lambda在0.09-0.14附近比较好。

Step 5：

认为lambda在0.09~0.14附近比较好，之后同时测试iterations，rank，lambda，找到最好的参数

设定rank range in 1~9，iterations range in 1~9，lambda range in 0.09~0.14

测试代码：

# 使用Spark MLlib中推荐算法ALS对电影评分数据MovieLens推荐

from pyspark.sql import SparkSession

from pyspark.mllib.recommendation import ALS, Rating, MatrixFactorizationModel

from pyspark.mllib.regression import LabeledPoint, LinearRegressionWithSGD

from pyspark.mllib.evaluation import RegressionMetrics

from pyspark.mllib.linalg import DenseVector

import numpy as np #arange

def alsModelEvaluate(model, testing\_rdd):

# 对测试数据集预测评分，针对测试数据集进行预测

predict\_rdd = model.predictAll(testing\_rdd.map(lambda r: (r[0], r[1])))

#print(predict\_rdd.take(5))

predict\_actual\_rdd = predict\_rdd.map(lambda r: ((r[0], r[1]), r[2])) \

.join(testing\_ratings.map(lambda r: ((r[0], r[1]), r[2])))

#print(predict\_actual\_rdd.take(5))

# 创建评估指标实例对象

metrics = RegressionMetrics(predict\_actual\_rdd.map(lambda pr: pr[1]))

print("MSE = %s" % metrics.meanSquaredError)

print("RMSE = %s" % metrics.rootMeanSquaredError)

# 返回均方根误差

return metrics.rootMeanSquaredError

def train\_model\_evaluate(training\_rdd, testing\_rdd, rank, iterations, lambda\_):

# 定义函数，训练模型与模型评估

# 使用超参数的值，训练数据和ALS算法训练模型

print(lambda\_,rank,iterations)

model = ALS.train(training\_rdd, rank, iterations, lambda\_)

# 模型的评估

rmse\_value = alsModelEvaluate(model, testing\_rdd)

# 返回多元组

return (model, rmse\_value, rank, iterations, lambda\_)

if \_\_name\_\_ == "\_\_main\_\_":

# 构建SparkSession实例对象

spark = SparkSession.builder \

.appName("SparkSessionExample") \

.master("local") \

.getOrCreate()

# 获取SparkContext实例对象

sc = spark.sparkContext

# 读取数据

raw\_ratings\_rdd = sc.textFile("C:/Users/lenovo/Desktop/ml-100k/ml-100k/u.data")

# print(raw\_ratings\_rdd.count())

# print(raw\_ratings\_rdd.first())

# 获取评分数据前三个字段，构建Rating实例对象

ratings\_rdd = raw\_ratings\_rdd.map(lambda line: line.split('\t')[0:3])

# print(ratings\_rdd.first())

ratings\_datas = ratings\_rdd.map(lambda x: Rating(int(x[0]), int(x[1]), float(x[2])))

# print(ratings\_datas.first())

# 将数据集分为训练数据集和测试数据集

training\_ratings, testing\_ratings = ratings\_datas.randomSplit([0.7, 0.3])

'''

# 使用ALS算法来训练模型

# help(ALS)

# 采用显示评分函数训练模型

alsModel = ALS.train(training\_ratings, 10, iterations=10, lambda\_=0.01)

# 用户特征因子矩阵

user\_feature\_matrix = alsModel.userFeatures()

print(type(user\_feature\_matrix))

print(user\_feature\_matrix.take(10))

# 物品因子矩阵

item\_feature\_matrix = alsModel.productFeatures()

print(type(item\_feature\_matrix))

print(item\_feature\_matrix.take(10))

# 预测某个用户对某个电影的评分

# 假设用户196，对电影242的评分，实际评分为3分

predictRating = alsModel.predict(196, 242)

print(predictRating)

# 为用户推荐（10部电影）

rmdMovies = alsModel.recommendProducts(196, 10)

print(rmdMovies)

# 为电影推荐（10个用户）

rmdUsers = alsModel.recommendUsers(242, 10)

print(rmdUsers)

'''

# 怎么评价模型的好坏，ALS模型评估指标(类似回归算法模型预测值，连续值)，使用回归模型中

# RMSE（均方根误差）评估模型

# 找到最佳模型

'''

如何找到最佳模型？？

-a. 模型的评估

计算RMSE

-b. 模型的优化，两个方向

1、数据

2、超参数的调整，选择合适的超参数的值，得到最优模型

交叉验证

训练数据集、验证数据集、测试数据集

K-Folds交叉验证

'''

# ALS算法的超参数的调整

# 定义一个函数，用于对模型进行评估

# 使用三层for循环，设置不同参数的值，分别使用ALS算法训练模型，评估获取RMSE的值

metrix\_list = [train\_model\_evaluate(training\_ratings, testing\_ratings, param\_rank, param\_iterations, param\_lambda)

for param\_rank in range(1, 10)

for param\_iterations in range(1, 10)

for param\_lambda in np.arange(0.09,0.14,0.01)

]

print(type(metrix\_list))

#sorted(metrix\_list, key=lambda k: k[1], reverse=False)

print("\n")

print("\n")

metrix\_list.sort(key=lambda s: s[1])

print(metrix\_list)

print("\n")

model, rmse\_value, rank, iterations, lambda\_ = metrix\_list[0]

print("The best parameters, rank=%s, iterations=%s, lambda\_=%s" % (rank ,iterations ,lambda\_))

结果：

0.09 1 1

MSE = 12.27100051871956

RMSE = 3.5029987894259342

0.09999999999999999 1 1

MSE = 11.43314385272269

RMSE = 3.3812932219378267

0.10999999999999999 1 1

MSE = 13.113870294950777

RMSE = 3.6213078155482417

0.11999999999999998 1 1

MSE = 13.02550049258728

RMSE = 3.6090858250514466

0.12999999999999998 1 1

MSE = 13.136027439836564

RMSE = 3.6243657982930704

0.13999999999999996 1 1

MSE = 11.46240966140817

RMSE = 3.3856180619509004

0.09 1 2

MSE = 1.4186608097107707

RMSE = 1.1910754844722355

0.09999999999999999 1 2

MSE = 1.2867665584351804

RMSE = 1.1343573327815095

0.10999999999999999 1 2

MSE = 1.1245018909006443

RMSE = 1.060425334901352

0.11999999999999998 1 2

MSE = 1.3621004430353527

RMSE = 1.1670905890441208

0.12999999999999998 1 2

MSE = 3.6107491953386215

RMSE = 1.9001971464399743

0.13999999999999996 1 2

MSE = 1.1576957370279248

RMSE = 1.0759627024334648

0.09 1 3

MSE = 0.9683012247983237

RMSE = 0.9840229798121198

0.09999999999999999 1 3

MSE = 0.9352130880908451

RMSE = 0.9670641592422113

0.10999999999999999 1 3

MSE = 0.9263143337730251

RMSE = 0.9624522501262206

0.11999999999999998 1 3

MSE = 0.9284253288239623

RMSE = 0.9635483012407642

0.12999999999999998 1 3

MSE = 0.9229007057317915

RMSE = 0.9606772120393985

0.13999999999999996 1 3

MSE = 0.9419270267799724

RMSE = 0.9705292508626272

0.09 1 4

MSE = 0.9095329046688458

RMSE = 0.9536943455158188

0.09999999999999999 1 4

MSE = 2.2038331542873

RMSE = 1.484531291110868

0.10999999999999999 1 4

MSE = 0.9582315501555264

RMSE = 0.9788930228352465

0.11999999999999998 1 4

MSE = 0.9323895164353483

RMSE = 0.9656031878755105

0.12999999999999998 1 4

MSE = 0.9144179402155393

RMSE = 0.9562520275615312

0.13999999999999996 1 4

MSE = 0.926077955555045

RMSE = 0.9623294423195442

0.09 1 5

MSE = 0.9034559115178579

RMSE = 0.9505029781741128

0.09999999999999999 1 5

MSE = 0.9119814232656362

RMSE = 0.9549771846832972

0.10999999999999999 1 5

MSE = 0.9047617576798102

RMSE = 0.9511896538965351

0.11999999999999998 1 5

MSE = 0.912366358793027

RMSE = 0.9551787051609908

0.12999999999999998 1 5

MSE = 0.9436685149655907

RMSE = 0.9714260213549927

0.13999999999999996 1 5

MSE = 0.9086747460358696

RMSE = 0.9532443265164864

0.09 1 6

MSE = 0.9077759135186125

RMSE = 0.9527727501973451

0.09999999999999999 1 6

MSE = 0.9085709944357315

RMSE = 0.9531899047072055

0.10999999999999999 1 6

MSE = 0.9041013081000747

RMSE = 0.9508424202253887

0.11999999999999998 1 6

MSE = 0.9098487648123839

RMSE = 0.9538599293462242

0.12999999999999998 1 6

MSE = 0.912817245977212

RMSE = 0.9554146984305883

0.13999999999999996 1 6

MSE = 0.916084461541021

RMSE = 0.9571230127528128

0.09 1 7

MSE = 0.927142237006517

RMSE = 0.9628822550065594

0.09999999999999999 1 7

MSE = 0.9043644554953387

RMSE = 0.9509807860810536

0.10999999999999999 1 7

MSE = 0.9194844880173344

RMSE = 0.9588975378096111

0.11999999999999998 1 7

MSE = 0.9209749885710742

RMSE = 0.959674418003874

0.12999999999999998 1 7

MSE = 0.9192601504834883

RMSE = 0.958780553872203

0.13999999999999996 1 7

MSE = 0.9149160771587348

RMSE = 0.9565124553076844

0.09 1 8

MSE = 0.9007567013633625

RMSE = 0.949082030892674

0.09999999999999999 1 8

MSE = 0.917370461075376

RMSE = 0.9577945818782732

0.10999999999999999 1 8

MSE = 0.9076379468180968

RMSE = 0.9527003447139593

0.11999999999999998 1 8

MSE = 0.9141832215292993

RMSE = 0.9561292912202299

0.12999999999999998 1 8

MSE = 0.917854422131532

RMSE = 0.958047192016934

0.13999999999999996 1 8

MSE = 0.9267032942848638

RMSE = 0.9626542963519478

0.09 1 9

MSE = 0.906446965038836

RMSE = 0.9520750837191551

0.09999999999999999 1 9

MSE = 0.9024576238631844

RMSE = 0.9499776965082835

0.10999999999999999 1 9

MSE = 0.9085393741048563

RMSE = 0.9531733179778252

0.11999999999999998 1 9

MSE = 0.9109437048111357

RMSE = 0.9544337089662832

0.12999999999999998 1 9

MSE = 0.9107327765616705

RMSE = 0.9543232034073522

0.13999999999999996 1 9

MSE = 0.9129997681831131

RMSE = 0.9555102135420181

0.09 2 1

MSE = 11.359480676403908

RMSE = 3.3703828679252315

0.09999999999999999 2 1

MSE = 13.026921154501546

RMSE = 3.609282637104158

0.10999999999999999 2 1

MSE = 12.496207016723261

RMSE = 3.534997456395586

0.11999999999999998 2 1

MSE = 10.954453312684029

RMSE = 3.309751246345264

0.12999999999999998 2 1

MSE = 11.149977116844957

RMSE = 3.3391581449288914

0.13999999999999996 2 1

MSE = 13.085007378034748

RMSE = 3.6173204693577743

0.09 2 2

MSE = 1.3353406847180265

RMSE = 1.155569420120672

0.09999999999999999 2 2

MSE = 1.1276251542918074

RMSE = 1.0618969602987889

0.10999999999999999 2 2

MSE = 1.034497446785883

RMSE = 1.0171024760494307

0.11999999999999998 2 2

MSE = 1.611628140786018

RMSE = 1.2694991692734652

0.12999999999999998 2 2

MSE = 1.0193913110445736

RMSE = 1.0096491029286232

0.13999999999999996 2 2

MSE = 1.1241959523205436

RMSE = 1.0602810723202332

0.09 2 3

MSE = 0.9293764877328389

RMSE = 0.9640417458454996

0.09999999999999999 2 3

MSE = 0.89806403345569

RMSE = 0.9476624047917539

0.10999999999999999 2 3

MSE = 1.0963158211309356

RMSE = 1.0470510117138208

0.11999999999999998 2 3

MSE = 0.9065512172373603

RMSE = 0.9521298321328664

0.12999999999999998 2 3

MSE = 0.9019429505229548

RMSE = 0.9497067708103143

0.13999999999999996 2 3

MSE = 0.9051367222767587

RMSE = 0.951386736441474

0.09 2 4

MSE = 0.880637306999809

RMSE = 0.9384227762580195

0.09999999999999999 2 4

MSE = 0.8690131585387213

RMSE = 0.9322087526615063

0.10999999999999999 2 4

MSE = 0.9155765092043521

RMSE = 0.9568576222220064

0.11999999999999998 2 4

MSE = 0.8973394048581689

RMSE = 0.9472800034087961

0.12999999999999998 2 4

MSE = 0.9066636969637719

RMSE = 0.9521888977318376

0.13999999999999996 2 4

MSE = 0.8757034921082618

RMSE = 0.9357903034912586

0.09 2 5

MSE = 0.8724844709622833

RMSE = 0.934068772073172

0.09999999999999999 2 5

MSE = 0.8854905790827907

RMSE = 0.9410050898283127

0.10999999999999999 2 5

MSE = 0.890068335905524

RMSE = 0.9434343304679579

0.11999999999999998 2 5

MSE = 0.8714243480268945

RMSE = 0.9335011237416345

0.12999999999999998 2 5

MSE = 0.8706929808406042

RMSE = 0.933109308088074

0.13999999999999996 2 5

MSE = 0.9056850735595215

RMSE = 0.9516748780752393

0.09 2 6

MSE = 0.9106312299624929

RMSE = 0.9542699984608617

0.09999999999999999 2 6

MSE = 0.8873897161454871

RMSE = 0.9420136496598588

0.10999999999999999 2 6

MSE = 0.882721390271071

RMSE = 0.9395325381651617

0.11999999999999998 2 6

MSE = 0.8742685740515894

RMSE = 0.935023301341517

0.12999999999999998 2 6

MSE = 0.8854514298159666

RMSE = 0.9409842877625357

0.13999999999999996 2 6

MSE = 0.8967813542835388

RMSE = 0.9469854034163033

0.09 2 7

MSE = 0.8709446683033074

RMSE = 0.9332441632838148

0.09999999999999999 2 7

MSE = 0.871171394225478

RMSE = 0.9333656273001905

0.10999999999999999 2 7

MSE = 0.8860868148650234

RMSE = 0.9413218444639556

0.11999999999999998 2 7

MSE = 0.8629431425636302

RMSE = 0.92894733034959

0.12999999999999998 2 7

MSE = 0.8758499221461906

RMSE = 0.9358685389231709

0.13999999999999996 2 7

MSE = 0.8657617794695947

RMSE = 0.9304632069402823

0.09 2 8

MSE = 0.8651269175552944

RMSE = 0.9301219906847136

0.09999999999999999 2 8

MSE = 0.8691063279605779

RMSE = 0.9322587237245774

0.10999999999999999 2 8

MSE = 0.8720032020711707

RMSE = 0.9338111169134637

0.11999999999999998 2 8

MSE = 0.8626319479288282

RMSE = 0.9287798167105206

0.12999999999999998 2 8

MSE = 0.922164149474107

RMSE = 0.9602937828988101

0.13999999999999996 2 8

MSE = 0.8744952989501243

RMSE = 0.9351445337219934

0.09 2 9

MSE = 0.8658622110811923

RMSE = 0.9305171739850868

0.09999999999999999 2 9

MSE = 0.8659658498168075

RMSE = 0.9305728611005198

0.10999999999999999 2 9

MSE = 0.8645691590206476

RMSE = 0.9298221114926487

0.11999999999999998 2 9

MSE = 0.862608656110372

RMSE = 0.928767277691442

0.12999999999999998 2 9

MSE = 0.866598563349727

RMSE = 0.9309127581839917

0.13999999999999996 2 9

MSE = 0.8829792965556273

RMSE = 0.9396697805908346

0.09 3 1

MSE = 11.203700647302156

RMSE = 3.347192950414146

0.09999999999999999 3 1

MSE = 12.324132911645066

RMSE = 3.5105744418321434

0.10999999999999999 3 1

MSE = 9.66931347441636

RMSE = 3.1095519732618007

0.11999999999999998 3 1

MSE = 8.950507833498618

RMSE = 2.9917399341350874

0.12999999999999998 3 1

MSE = 9.660375939492168

RMSE = 3.10811453127007

0.13999999999999996 3 1

MSE = 12.207060313707784

RMSE = 3.493860374100228

0.09 3 2

MSE = 1.167810152894349

RMSE = 1.0806526513613655

0.09999999999999999 3 2

MSE = 1.0956581801519654

RMSE = 1.0467369202201504

0.10999999999999999 3 2

MSE = 1.0002277134915472

RMSE = 1.0001138502648321

0.11999999999999998 3 2

MSE = 1.0288287098937081

RMSE = 1.0143119391457975

0.12999999999999998 3 2

MSE = 1.0688821672677602

RMSE = 1.033867577239832

0.13999999999999996 3 2

MSE = 1.1549451317723183

RMSE = 1.0746837356973065

0.09 3 3

MSE = 0.8987571000407324

RMSE = 0.9480280059369198

0.09999999999999999 3 3

MSE = 0.9151448555970977

RMSE = 0.9566320377224974

0.10999999999999999 3 3

MSE = 0.8940633322833578

RMSE = 0.9455492225597554

0.11999999999999998 3 3

MSE = 0.8896759812298429

RMSE = 0.9432263679678611

0.12999999999999998 3 3

MSE = 0.8899993113179371

RMSE = 0.9433977482048265

0.13999999999999996 3 3

MSE = 0.9195663954671283

RMSE = 0.9589402460357622

0.09 3 4

MSE = 0.8972415234008456

RMSE = 0.9472283375199697

0.09999999999999999 3 4

MSE = 0.8896911958639443

RMSE = 0.943234433141594

0.10999999999999999 3 4

MSE = 0.8879792909912588

RMSE = 0.9423265309813041

0.11999999999999998 3 4

MSE = 0.8859739987092691

RMSE = 0.9412619182296015

0.12999999999999998 3 4

MSE = 0.9042610601767678

RMSE = 0.9509264220625946

0.13999999999999996 3 4

MSE = 0.8760316998209086

RMSE = 0.9359656509834688

0.09 3 5

MSE = 0.8742584967393761

RMSE = 0.9350179125232714

0.09999999999999999 3 5

MSE = 0.8753555759290517

RMSE = 0.9356043907170657

0.10999999999999999 3 5

MSE = 0.8830393566797058

RMSE = 0.9397017381487095

0.11999999999999998 3 5

MSE = 0.872013569525964

RMSE = 0.9338166680489077

0.12999999999999998 3 5

MSE = 0.8714306296599521

RMSE = 0.9335044882912733

0.13999999999999996 3 5

MSE = 0.8751070226037995

RMSE = 0.9354715509323623

0.09 3 6

MSE = 0.8766821225799021

RMSE = 0.9363130473190588

0.09999999999999999 3 6

MSE = 0.8795804124009897

RMSE = 0.9378594843583924

0.10999999999999999 3 6

MSE = 0.874896917009218

RMSE = 0.9353592448942909

0.11999999999999998 3 6

MSE = 0.8851173675671057

RMSE = 0.9408067642013984

0.12999999999999998 3 6

MSE = 0.8825734828837598

RMSE = 0.9394538215813271

0.13999999999999996 3 6

MSE = 0.9078170443639915

RMSE = 0.9527943347669483

0.09 3 7

MSE = 0.8710454928610801

RMSE = 0.933298180037377

0.09999999999999999 3 7

MSE = 0.8716539292724724

RMSE = 0.9336240834899625

0.10999999999999999 3 7

MSE = 0.8628263914777976

RMSE = 0.9288844876935978

0.11999999999999998 3 7

MSE = 0.8739437367672975

RMSE = 0.9348495797545707

0.12999999999999998 3 7

MSE = 0.8675934650277384

RMSE = 0.9314469738142577

0.13999999999999996 3 7

MSE = 0.8675211349148313

RMSE = 0.9314081462574993

0.09 3 8

MSE = 0.8755862696350989

RMSE = 0.935727668520654

0.09999999999999999 3 8

MSE = 0.8731194166423342

RMSE = 0.934408591913802

0.10999999999999999 3 8

MSE = 0.8762687909587734

RMSE = 0.9360922983118564

0.11999999999999998 3 8

MSE = 0.8720121949344937

RMSE = 0.9338159320414777

0.12999999999999998 3 8

MSE = 0.8643470839633219

RMSE = 0.9297026857890225

0.13999999999999996 3 8

MSE = 0.8728186898116941

RMSE = 0.9342476597839001

0.09 3 9

MSE = 0.8711124227138352

RMSE = 0.9333340359773853

0.09999999999999999 3 9

MSE = 0.8597500540017546

RMSE = 0.927227077905814

0.10999999999999999 3 9

MSE = 0.8551677641285562

RMSE = 0.9247528124469566

0.11999999999999998 3 9

MSE = 0.8651671585240138

RMSE = 0.9301436225250452

0.12999999999999998 3 9

MSE = 0.8710804201467238

RMSE = 0.9333168916004487

0.13999999999999996 3 9

MSE = 0.8617355994932844

RMSE = 0.9282971504282906

0.09 4 1

MSE = 11.199491598199069

RMSE = 3.346564148227114

0.09999999999999999 4 1

MSE = 9.554163261299617

RMSE = 3.0909809545352456

0.10999999999999999 4 1

MSE = 8.612929474243568

RMSE = 2.9347792888467046

0.11999999999999998 4 1

MSE = 9.739383564229216

RMSE = 3.1207985459220557

0.12999999999999998 4 1

MSE = 10.525987815716643

RMSE = 3.2443778780710244

0.13999999999999996 4 1

MSE = 10.416151433516568

RMSE = 3.227406301276083

0.09 4 2

MSE = 1.0360016878754312

RMSE = 1.0178416811446813

0.09999999999999999 4 2

MSE = 1.0358346422547404

RMSE = 1.0177596190922198

0.10999999999999999 4 2

MSE = 0.9817887820184833

RMSE = 0.990852553117003

0.11999999999999998 4 2

MSE = 1.0874875244660465

RMSE = 1.0428266991528585

0.12999999999999998 4 2

MSE = 1.0188953361963167

RMSE = 1.0094034556094589

0.13999999999999996 4 2

MSE = 1.0101710023804686

RMSE = 1.0050726353754085

0.09 4 3

MSE = 0.9018051978666868

RMSE = 0.9496342442575914

0.09999999999999999 4 3

MSE = 0.8767421094941046

RMSE = 0.9363450803491759

0.10999999999999999 4 3

MSE = 0.8985376137422845

RMSE = 0.9479122394727713

0.11999999999999998 4 3

MSE = 0.8962895376895593

RMSE = 0.9467256929488918

0.12999999999999998 4 3

MSE = 0.9109644552517189

RMSE = 0.9544445794553599

0.13999999999999996 4 3

MSE = 0.9242351356089865

RMSE = 0.9613714867880088

0.09 4 4

MSE = 0.8856416130153947

RMSE = 0.9410853377964161

0.09999999999999999 4 4

MSE = 0.8917059959124207

RMSE = 0.94430185635337

0.10999999999999999 4 4

MSE = 0.8928358226881071

RMSE = 0.9448999008826846

0.11999999999999998 4 4

MSE = 0.8878354199027207

RMSE = 0.9422501896538523

0.12999999999999998 4 4

MSE = 0.8955300909684853

RMSE = 0.9463245167322283

0.13999999999999996 4 4

MSE = 0.9055269273631623

RMSE = 0.9515917860948372

0.09 4 5

MSE = 0.8766903409150454

RMSE = 0.936317435977268

0.09999999999999999 4 5

MSE = 0.8785437485088053

RMSE = 0.9373066459322719

0.10999999999999999 4 5

MSE = 0.8717799407911314

RMSE = 0.9336915661989945

0.11999999999999998 4 5

MSE = 0.8600825541517492

RMSE = 0.927406358697065

0.12999999999999998 4 5

MSE = 0.8625313899560398

RMSE = 0.9287256806808132

0.13999999999999996 4 5

MSE = 0.8753340781208774

RMSE = 0.9355929019188193

0.09 4 6

MSE = 0.8636931976684141

RMSE = 0.9293509550586442

0.09999999999999999 4 6

MSE = 0.8628561201057903

RMSE = 0.9289004898834914

0.10999999999999999 4 6

MSE = 0.8746875657114519

RMSE = 0.9352473286310161

0.11999999999999998 4 6

MSE = 0.8625655578609306

RMSE = 0.9287440755455351

0.12999999999999998 4 6

MSE = 0.873981980817492

RMSE = 0.9348700341852294

0.13999999999999996 4 6

MSE = 0.8707400727695638

RMSE = 0.9331345416227843

0.09 4 7

MSE = 0.8553897307952235

RMSE = 0.9248728187135913

0.09999999999999999 4 7

MSE = 0.8621648778248165

RMSE = 0.9285283398070392

0.10999999999999999 4 7

MSE = 0.8682931842996421

RMSE = 0.9318225068647151

0.11999999999999998 4 7

MSE = 0.8802995783942509

RMSE = 0.9382428141980363

0.12999999999999998 4 7

MSE = 0.8775379556096444

RMSE = 0.9367699587463533

0.13999999999999996 4 7

MSE = 0.8719727797353999

RMSE = 0.9337948274302016

0.09 4 8

MSE = 0.8751518955127651

RMSE = 0.9354955347369462

0.09999999999999999 4 8

MSE = 0.8625855817834063

RMSE = 0.9287548555907561

0.10999999999999999 4 8

MSE = 0.8728144398227169

RMSE = 0.9342453852295536

0.11999999999999998 4 8

MSE = 0.8619128816520599

RMSE = 0.928392633346506

0.12999999999999998 4 8

MSE = 0.8678415677484469

RMSE = 0.931580145638821

0.13999999999999996 4 8

MSE = 0.8716835631125637

RMSE = 0.9336399536826623

0.09 4 9

MSE = 0.8624718878061972

RMSE = 0.9286936458306352

0.09999999999999999 4 9

MSE = 0.8647996198653713

RMSE = 0.929946030619719

0.10999999999999999 4 9

MSE = 0.8587587656917339

RMSE = 0.9266923792131528

0.11999999999999998 4 9

MSE = 0.8551123678275513

RMSE = 0.9247228600113394

0.12999999999999998 4 9

MSE = 0.8724596874966117

RMSE = 0.9340555055758795

0.13999999999999996 4 9

MSE = 0.8599865091721242

RMSE = 0.9273545757541309

0.09 5 1

MSE = 5.637881007395718

RMSE = 2.37442224707311

0.09999999999999999 5 1

MSE = 7.205857883367729

RMSE = 2.6843729031875823

0.10999999999999999 5 1

MSE = 8.329871552216842

RMSE = 2.886151685587028

0.11999999999999998 5 1

MSE = 10.40889088122431

RMSE = 3.2262812774499854

0.12999999999999998 5 1

MSE = 12.122816107982313

RMSE = 3.481783466555942

0.13999999999999996 5 1

MSE = 11.026381108109007

RMSE = 3.3205995103458363

0.09 5 2

MSE = 0.9791384527997916

RMSE = 0.9895142509331494

0.09999999999999999 5 2

MSE = 0.9901623857625915

RMSE = 0.995069035676717

0.10999999999999999 5 2

MSE = 0.9665487550510502

RMSE = 0.9831321147491064

0.11999999999999998 5 2

MSE = 0.9893514136809598

RMSE = 0.9946614568188313

0.12999999999999998 5 2

MSE = 1.016050014697498

RMSE = 1.0079930628221099

0.13999999999999996 5 2

MSE = 1.026518484128189

RMSE = 1.0131724848850707

0.09 5 3

MSE = 0.8888759097178295

RMSE = 0.942802158312034

0.09999999999999999 5 3

MSE = 0.8997752257298757

RMSE = 0.9485648242106998

0.10999999999999999 5 3

MSE = 0.8962262546803477

RMSE = 0.9466922703182633

0.11999999999999998 5 3

MSE = 0.8797480788828355

RMSE = 0.9379488679468809

0.12999999999999998 5 3

MSE = 0.9106351651654776

RMSE = 0.954272060350442

0.13999999999999996 5 3

MSE = 0.9147093861298441

RMSE = 0.9564044051183809

0.09 5 4

MSE = 0.8732076909083621

RMSE = 0.9344558260872272

0.09999999999999999 5 4

MSE = 0.8766208063529595

RMSE = 0.9362803033028941

0.10999999999999999 5 4

MSE = 0.9054715778565096

RMSE = 0.9515627030608701

0.11999999999999998 5 4

MSE = 0.8819256999303036

RMSE = 0.9391089925723763

0.12999999999999998 5 4

MSE = 0.8920453005670158

RMSE = 0.9444814982661205

0.13999999999999996 5 4

MSE = 0.8792056142811826

RMSE = 0.9376596473567489

0.09 5 5

MSE = 0.8839609720396948

RMSE = 0.9401919867982788

0.09999999999999999 5 5

MSE = 0.8704824833806526

RMSE = 0.9329965077001374

0.10999999999999999 5 5

MSE = 0.8675822025212002

RMSE = 0.9314409280900213

0.11999999999999998 5 5

MSE = 0.8820715526711801

RMSE = 0.9391866442146524

0.12999999999999998 5 5

MSE = 0.8709826410855405

RMSE = 0.933264507567678

0.13999999999999996 5 5

MSE = 0.8846557359617621

RMSE = 0.940561394041751

0.09 5 6

MSE = 0.8768488012776758

RMSE = 0.9364020510857908

0.09999999999999999 5 6

MSE = 0.8615813980992656

RMSE = 0.9282140906597279

0.10999999999999999 5 6

MSE = 0.875434678062675

RMSE = 0.9356466630425585

0.11999999999999998 5 6

MSE = 0.8531046096791276

RMSE = 0.923636622097201

0.12999999999999998 5 6

MSE = 0.8740395375962555

RMSE = 0.9349008169834143

0.13999999999999996 5 6

MSE = 0.8800627580397948

RMSE = 0.9381166015159282

0.09 5 7

MSE = 0.8707565007530093

RMSE = 0.9331433441615544

0.09999999999999999 5 7

MSE = 0.8672365213428886

RMSE = 0.9312553470143883

0.10999999999999999 5 7

MSE = 0.8694830883906688

RMSE = 0.9324607704298711

0.11999999999999998 5 7

MSE = 0.8592520480328005

RMSE = 0.9269584931553303

0.12999999999999998 5 7

MSE = 0.8525970708018452

RMSE = 0.9233618309210345

0.13999999999999996 5 7

MSE = 0.8720540793540474

RMSE = 0.9338383582580272

0.09 5 8

MSE = 0.8708221828173612

RMSE = 0.9331785374821697

0.09999999999999999 5 8

MSE = 0.8595173915834372

RMSE = 0.9271016080146972

0.10999999999999999 5 8

MSE = 0.864784052151539

RMSE = 0.9299376603576924

0.11999999999999998 5 8

MSE = 0.865213518142143

RMSE = 0.9301685428685185

0.12999999999999998 5 8

MSE = 0.8685065981665263

RMSE = 0.9319370140554168

0.13999999999999996 5 8

MSE = 0.8738806719753539

RMSE = 0.9348158492320046

0.09 5 9

MSE = 0.8716199855996253

RMSE = 0.9336059048654445

0.09999999999999999 5 9

MSE = 0.8635395780483777

RMSE = 0.9292683025092257

0.10999999999999999 5 9

MSE = 0.872212895073642

RMSE = 0.9339233882249882

0.11999999999999998 5 9

MSE = 0.8677079391873294

RMSE = 0.9315084214258771

0.12999999999999998 5 9

MSE = 0.8626736850116451

RMSE = 0.9288022852101759

0.13999999999999996 5 9

MSE = 0.8679788091755234

RMSE = 0.9316538032850633

0.09 6 1

MSE = 6.908769460057759

RMSE = 2.628453815469802

0.09999999999999999 6 1

MSE = 11.847200634931225

RMSE = 3.4419762687925703

0.10999999999999999 6 1

MSE = 7.647208173501799

RMSE = 2.7653585976328277

0.11999999999999998 6 1

MSE = 9.418650779257923

RMSE = 3.068982042837319

0.12999999999999998 6 1

MSE = 12.484964854557978

RMSE = 3.533406975506498

0.13999999999999996 6 1

MSE = 9.037439646908537

RMSE = 3.0062334651368205

0.09 6 2

MSE = 0.9640752058816882

RMSE = 0.981873314578662

0.09999999999999999 6 2

MSE = 1.0324319636593378

RMSE = 1.0160865925989466

0.10999999999999999 6 2

MSE = 0.9950329822044459

RMSE = 0.9975133995112275

0.11999999999999998 6 2

MSE = 0.9681099840517997

RMSE = 0.9839258021069474

0.12999999999999998 6 2

MSE = 0.9539034536392813

RMSE = 0.9766798112172081

0.13999999999999996 6 2

MSE = 0.9556110155808747

RMSE = 0.9775535870635812

0.09 6 3

MSE = 0.8871834324903403

RMSE = 0.9419041524966011

0.09999999999999999 6 3

MSE = 0.9013441612692807

RMSE = 0.9493914689259013

0.10999999999999999 6 3

MSE = 0.8950706264886896

RMSE = 0.9460817229440011

0.11999999999999998 6 3

MSE = 0.8995387352178527

RMSE = 0.9484401590073317

0.12999999999999998 6 3

MSE = 0.9104630489679953

RMSE = 0.9541818741560727

0.13999999999999996 6 3

MSE = 0.8798946062153864

RMSE = 0.9380269752066762

0.09 6 4

MSE = 0.8847539426035161

RMSE = 0.9406135989892535

0.09999999999999999 6 4

MSE = 0.8929875916773462

RMSE = 0.9449802070294098

0.10999999999999999 6 4

MSE = 0.8767690399184808

RMSE = 0.9363594608474252

0.11999999999999998 6 4

MSE = 0.8941312038840041

RMSE = 0.9455851119196009

0.12999999999999998 6 4

MSE = 0.8886177530438903

RMSE = 0.942665239119323

0.13999999999999996 6 4

MSE = 0.8768808755643985

RMSE = 0.9364191772728699

0.09 6 5

MSE = 0.8658644220631204

RMSE = 0.9305183620236198

0.09999999999999999 6 5

MSE = 0.8932530146964796

RMSE = 0.9451206349966546

0.10999999999999999 6 5

MSE = 0.8829219940584008

RMSE = 0.9396392893330934

0.11999999999999998 6 5

MSE = 0.8565899016079316

RMSE = 0.9255214214743663

0.12999999999999998 6 5

MSE = 0.8671424831001894

RMSE = 0.9312048556038512

0.13999999999999996 6 5

MSE = 0.8765444990409643

RMSE = 0.9362395521665191

0.09 6 6

MSE = 0.8716564891828139

RMSE = 0.9336254544424193

0.09999999999999999 6 6

MSE = 0.8752639614560225

RMSE = 0.9355554293872824

0.10999999999999999 6 6

MSE = 0.8640981349673028

RMSE = 0.929568789798422

0.11999999999999998 6 6

MSE = 0.8744404081613997

RMSE = 0.935115184435265

0.12999999999999998 6 6

MSE = 0.8692353300475849

RMSE = 0.9323279090789811

0.13999999999999996 6 6

MSE = 0.8712434364656659

RMSE = 0.9334042192242683

0.09 6 7

MSE = 0.869788628931638

RMSE = 0.9326245916399792

0.09999999999999999 6 7

MSE = 0.8753900279469824

RMSE = 0.9356228021734947

0.10999999999999999 6 7

MSE = 0.8587865756013022

RMSE = 0.9267073840222178

0.11999999999999998 6 7

MSE = 0.8704389993759114

RMSE = 0.9329732039967232

0.12999999999999998 6 7

MSE = 0.871115586736985

RMSE = 0.9333357309869718

0.13999999999999996 6 7

MSE = 0.872975484769712

RMSE = 0.9343315711083041

0.09 6 8

MSE = 0.8810690106184347

RMSE = 0.9386527636024062

0.09999999999999999 6 8

MSE = 0.8640593717904144

RMSE = 0.9295479394794086

0.10999999999999999 6 8

MSE = 0.8633113106728962

RMSE = 0.9291454733640455

0.11999999999999998 6 8

MSE = 0.8702103926174619

RMSE = 0.9328506807723634

0.12999999999999998 6 8

MSE = 0.8679934147323856

RMSE = 0.9316616417629232

0.13999999999999996 6 8

MSE = 0.8689716599938407

RMSE = 0.9321864942133847

0.09 6 9

MSE = 0.8775171151686215

RMSE = 0.936758835116393

0.09999999999999999 6 9

MSE = 0.8690807544637273

RMSE = 0.9322450077440626

0.10999999999999999 6 9

MSE = 0.8692125799714314

RMSE = 0.9323157083152849

0.11999999999999998 6 9

MSE = 0.8621575538183371

RMSE = 0.9285243959198579

0.12999999999999998 6 9

MSE = 0.8611973753859135

RMSE = 0.9280072065377043

0.13999999999999996 6 9

MSE = 0.8650060151675918

RMSE = 0.9300569956554231

0.09 7 1

MSE = 8.775330989844397

RMSE = 2.962318515933828

0.09999999999999999 7 1

MSE = 8.22983476943538

RMSE = 2.8687688595345877

0.10999999999999999 7 1

MSE = 10.858630366131955

RMSE = 3.2952435973888115

0.11999999999999998 7 1

MSE = 8.486787467990464

RMSE = 2.9132091356424215

0.12999999999999998 7 1

MSE = 6.796484237859035

RMSE = 2.607006758307127

0.13999999999999996 7 1

MSE = 7.31747955324769

RMSE = 2.7050840196281687

0.09 7 2

MSE = 0.9699143282360078

RMSE = 0.984842285970707

0.09999999999999999 7 2

MSE = 0.9467646706909301

RMSE = 0.9730183300898961

0.10999999999999999 7 2

MSE = 0.9606358289078026

RMSE = 0.980120313485953

0.11999999999999998 7 2

MSE = 0.9693660330601266

RMSE = 0.984563879623931

0.12999999999999998 7 2

MSE = 0.9868553627514379

RMSE = 0.9934059405658081

0.13999999999999996 7 2

MSE = 0.9874157984893824

RMSE = 0.9936879784365827

0.09 7 3

MSE = 0.890092670590573

RMSE = 0.9434472272419762

0.09999999999999999 7 3

MSE = 0.8883976622277964

RMSE = 0.9425484933030217

0.10999999999999999 7 3

MSE = 0.8891164598529183

RMSE = 0.9429297215874142

0.11999999999999998 7 3

MSE = 0.9031362620560284

RMSE = 0.9503348157654903

0.12999999999999998 7 3

MSE = 0.9000199633710031

RMSE = 0.9486938196125255

0.13999999999999996 7 3

MSE = 0.9136447731443604

RMSE = 0.9558476725631341

0.09 7 4

MSE = 0.8752870841329198

RMSE = 0.9355677870325163

0.09999999999999999 7 4

MSE = 0.8869612309451984

RMSE = 0.941786191736319

0.10999999999999999 7 4

MSE = 0.8815154983016306

RMSE = 0.9388905677988413

0.11999999999999998 7 4

MSE = 0.875802884084847

RMSE = 0.9358434078866223

0.12999999999999998 7 4

MSE = 0.8903621923634469

RMSE = 0.9435900552482772

0.13999999999999996 7 4

MSE = 0.8779817134128606

RMSE = 0.9370067840804892

0.09 7 5

MSE = 0.8707169822376085

RMSE = 0.9331221689776792

0.09999999999999999 7 5

MSE = 0.8885022024133877

RMSE = 0.9426039478027809

0.10999999999999999 7 5

MSE = 0.8751935705701648

RMSE = 0.9355178087936994

0.11999999999999998 7 5

MSE = 0.8742783201488626

RMSE = 0.9350285130138346

0.12999999999999998 7 5

MSE = 0.8770571563672642

RMSE = 0.9365132974855532

0.13999999999999996 7 5

MSE = 0.8721142632964068

RMSE = 0.9338705816634374

0.09 7 6

MSE = 0.880617264804272

RMSE = 0.9384120975372557

0.09999999999999999 7 6

MSE = 0.8836926503877982

RMSE = 0.9400492808293607

0.10999999999999999 7 6

MSE = 0.877971953221807

RMSE = 0.937001575890781

0.11999999999999998 7 6

MSE = 0.867658569964227

RMSE = 0.9314819214371404

0.12999999999999998 7 6

MSE = 0.8748218744389682

RMSE = 0.9353191297300447

0.13999999999999996 7 6

MSE = 0.872530501597635

RMSE = 0.9340934116016636

0.09 7 7

MSE = 0.8708505274127001

RMSE = 0.9331937244820606

0.09999999999999999 7 7

MSE = 0.8792145330679944

RMSE = 0.9376644032211068

0.10999999999999999 7 7

MSE = 0.8652527921851215

RMSE = 0.9301896538798534

0.11999999999999998 7 7

MSE = 0.8597221862665432

RMSE = 0.9272120503242736

0.12999999999999998 7 7

MSE = 0.8652670315861488

RMSE = 0.9301973078794352

0.13999999999999996 7 7

MSE = 0.8756521476998125

RMSE = 0.9357628693744011

0.09 7 8

MSE = 0.8827310009895968

RMSE = 0.939537652779066

0.09999999999999999 7 8

MSE = 0.8713742844400757

RMSE = 0.9334743083985095

0.10999999999999999 7 8

MSE = 0.8693897382641683

RMSE = 0.9324107132933257

0.11999999999999998 7 8

MSE = 0.8668850605280221

RMSE = 0.9310666251821198

0.12999999999999998 7 8

MSE = 0.8663304486247344

RMSE = 0.9307687406787651

0.13999999999999996 7 8

MSE = 0.8674214894238236

RMSE = 0.9313546528706579

0.09 7 9

MSE = 0.8765227996441912

RMSE = 0.9362279635025815

0.09999999999999999 7 9

MSE = 0.8657442041851992

RMSE = 0.9304537625186966

0.10999999999999999 7 9

MSE = 0.8610935264405428

RMSE = 0.9279512521897596

0.11999999999999998 7 9

MSE = 0.8591979269925957

RMSE = 0.9269292998889375

0.12999999999999998 7 9

MSE = 0.8603782569827351

RMSE = 0.9275657696264643

0.13999999999999996 7 9

MSE = 0.8634226124431743

RMSE = 0.9292053661291321

0.09 8 1

MSE = 9.575281830409663

RMSE = 3.094395228539765

0.09999999999999999 8 1

MSE = 6.5677544963910375

RMSE = 2.562763058964101

0.10999999999999999 8 1

MSE = 8.709544428849648

RMSE = 2.9511937294677297

0.11999999999999998 8 1

MSE = 5.848444527698413

RMSE = 2.4183557487885055

0.12999999999999998 8 1

MSE = 7.531864750686837

RMSE = 2.744424302232954

0.13999999999999996 8 1

MSE = 4.891374573444049

RMSE = 2.2116452187102813

0.09 8 2

MSE = 1.016236706620716

RMSE = 1.0080856643265572

0.09999999999999999 8 2

MSE = 0.9966763606100223

RMSE = 0.9983367971832062

0.10999999999999999 8 2

MSE = 0.9638244078310102

RMSE = 0.9817455922136906

0.11999999999999998 8 2

MSE = 0.9671034611231346

RMSE = 0.9834141859476782

0.12999999999999998 8 2

MSE = 0.9726688055842432

RMSE = 0.9862397302807483

0.13999999999999996 8 2

MSE = 0.9830070800814029

RMSE = 0.9914671351494223

0.09 8 3

MSE = 0.9017494880771905

RMSE = 0.9496049115696436

0.09999999999999999 8 3

MSE = 0.8916505224019724

RMSE = 0.9442724831328997

0.10999999999999999 8 3

MSE = 0.8904551902237402

RMSE = 0.9436393327027759

0.11999999999999998 8 3

MSE = 0.9024770382312286

RMSE = 0.9499879147816717

0.12999999999999998 8 3

MSE = 0.9055943340627992

RMSE = 0.9516272033011662

0.13999999999999996 8 3

MSE = 0.9094514433394671

RMSE = 0.9536516362589995

0.09 8 4

MSE = 0.8798163408400658

RMSE = 0.9379852561954616

0.09999999999999999 8 4

MSE = 0.8764444135986017

RMSE = 0.9361860998746999

0.10999999999999999 8 4

MSE = 0.8851071854681144

RMSE = 0.9408013528200916

0.11999999999999998 8 4

MSE = 0.888422467482317

RMSE = 0.9425616518203555

0.12999999999999998 8 4

MSE = 0.8934350303722123

RMSE = 0.9452169223898884

0.13999999999999996 8 4

MSE = 0.8938377573563003

RMSE = 0.9454299325472514

0.09 8 5

MSE = 0.8822344244785225

RMSE = 0.9392733491793124

0.09999999999999999 8 5

MSE = 0.8798592561679995

RMSE = 0.9380081322504616

0.10999999999999999 8 5

MSE = 0.8715173099514859

RMSE = 0.933550914493412

0.11999999999999998 8 5

MSE = 0.8635242184510188

RMSE = 0.9292600381222786

0.12999999999999998 8 5

MSE = 0.8725587636443166

RMSE = 0.93410853954148

0.13999999999999996 8 5

MSE = 0.8712777095069059

RMSE = 0.9334225782071622

0.09 8 6

MSE = 0.8912755371699977

RMSE = 0.9440739045064204

0.09999999999999999 8 6

MSE = 0.8783279024784615

RMSE = 0.9371914972290676

0.10999999999999999 8 6

MSE = 0.8735487133858979

RMSE = 0.9346382794353641

0.11999999999999998 8 6

MSE = 0.8741074226647373

RMSE = 0.9349371223054186

0.12999999999999998 8 6

MSE = 0.8713065020414033

RMSE = 0.9334380011770483

0.13999999999999996 8 6

MSE = 0.8725217364118995

RMSE = 0.9340887197755358

0.09 8 7

MSE = 0.8761852293382094

RMSE = 0.9360476640311697

0.09999999999999999 8 7

MSE = 0.8718618332415472

RMSE = 0.9337354192926105

0.10999999999999999 8 7

MSE = 0.8726762939569721

RMSE = 0.9341714478386567

0.11999999999999998 8 7

MSE = 0.8691785630407816

RMSE = 0.9322974648902471

0.12999999999999998 8 7

MSE = 0.8706276979868989

RMSE = 0.9330743260785278

0.13999999999999996 8 7

MSE = 0.8676250549376041

RMSE = 0.9314639310985714

0.09 8 8

MSE = 0.8798360886298919

RMSE = 0.9379957828422748

0.09999999999999999 8 8

MSE = 0.8737829564625056

RMSE = 0.9347635831922987

0.10999999999999999 8 8

MSE = 0.8747416722436171

RMSE = 0.9352762545064518

0.11999999999999998 8 8

MSE = 0.8652526552783671

RMSE = 0.9301895802890758

0.12999999999999998 8 8

MSE = 0.8622417783459451

RMSE = 0.9285697487781653

0.13999999999999996 8 8

MSE = 0.8619887900800387

RMSE = 0.9284335140870555

0.09 8 9

MSE = 0.8801024906925273

RMSE = 0.9381377780968675

0.09999999999999999 8 9

MSE = 0.8718293631070251

RMSE = 0.9337180319063272

0.10999999999999999 8 9

MSE = 0.8688952412352157

RMSE = 0.932145504326023

0.11999999999999998 8 9

MSE = 0.8728377055360728

RMSE = 0.9342578367538978

0.12999999999999998 8 9

MSE = 0.8599046358453871

RMSE = 0.9273104312178242

0.13999999999999996 8 9

MSE = 0.8651549145475819

RMSE = 0.9301370407351713

0.09 9 1

MSE = 7.751882441796876

RMSE = 2.78422025741443

0.09999999999999999 9 1

MSE = 8.258704281625356

RMSE = 2.8737961447578977

0.10999999999999999 9 1

MSE = 6.173690506793647

RMSE = 2.484691229668919

0.11999999999999998 9 1

MSE = 9.015475805820016

RMSE = 3.002578193123372

0.12999999999999998 9 1

MSE = 7.779710473245582

RMSE = 2.7892132355281807

0.13999999999999996 9 1

MSE = 8.487066348845298

RMSE = 2.9132570001366678

0.09 9 2

MSE = 0.9522436102986336

RMSE = 0.9758297035336819

0.09999999999999999 9 2

MSE = 0.9425741743674975

RMSE = 0.9708625929386184

0.10999999999999999 9 2

MSE = 0.9517813558721516

RMSE = 0.9755928227863054

0.11999999999999998 9 2

MSE = 0.9606890886436998

RMSE = 0.980147483108384

0.12999999999999998 9 2

MSE = 0.983801647897573

RMSE = 0.9918677572628183

0.13999999999999996 9 2

MSE = 0.9738726330905837

RMSE = 0.9868498533670579

0.09 9 3

MSE = 0.8911002333299617

RMSE = 0.9439810555990844

0.09999999999999999 9 3

MSE = 0.8878889218382925

RMSE = 0.9422785797407752

0.10999999999999999 9 3

MSE = 0.8898625257443963

RMSE = 0.9433252491820604

0.11999999999999998 9 3

MSE = 0.9010633126757445

RMSE = 0.9492435476081702

0.12999999999999998 9 3

MSE = 0.8957912074250851

RMSE = 0.9464624701619632

0.13999999999999996 9 3

MSE = 0.9087446699963426

RMSE = 0.953281002641059

0.09 9 4

MSE = 0.8764329849689138

RMSE = 0.9361799960311659

0.09999999999999999 9 4

MSE = 0.884801078640068

RMSE = 0.9406386546597306

0.10999999999999999 9 4

MSE = 0.8901248382114054

RMSE = 0.9434642750053684

0.11999999999999998 9 4

MSE = 0.8811236932176857

RMSE = 0.9386818913868988

0.12999999999999998 9 4

MSE = 0.8887565552253213

RMSE = 0.942738858446665

0.13999999999999996 9 4

MSE = 0.8907620868621124

RMSE = 0.9438019320080419

0.09 9 5

MSE = 0.8812834064382191

RMSE = 0.9387669606660746

0.09999999999999999 9 5

MSE = 0.880295867567409

RMSE = 0.9382408366551783

0.10999999999999999 9 5

MSE = 0.8725384421304152

RMSE = 0.9340976619874473

0.11999999999999998 9 5

MSE = 0.8797581598414062

RMSE = 0.9379542418697226

0.12999999999999998 9 5

MSE = 0.8889997398762575

RMSE = 0.9428678273630178

0.13999999999999996 9 5

MSE = 0.8782547847338702

RMSE = 0.9371524874500788

0.09 9 6

MSE = 0.8790533153009722

RMSE = 0.9375784315463812

0.09999999999999999 9 6

MSE = 0.8754746276489525

RMSE = 0.9356680114490141

0.10999999999999999 9 6

MSE = 0.8723834438460023

RMSE = 0.934014691450837

0.11999999999999998 9 6

MSE = 0.8733433807081049

RMSE = 0.9345284269127959

0.12999999999999998 9 6

MSE = 0.8750244318028911

RMSE = 0.9354274059502913

0.13999999999999996 9 6

MSE = 0.8698880996972931

RMSE = 0.9326779185213366

0.09 9 7

MSE = 0.8817290943025516

RMSE = 0.9390043100553648

0.09999999999999999 9 7

MSE = 0.8751253835791035

RMSE = 0.935481364634862

0.10999999999999999 9 7

MSE = 0.8727103674356729

RMSE = 0.9341896849332436

0.11999999999999998 9 7

MSE = 0.8775901588649708

RMSE = 0.9367978217657057

0.12999999999999998 9 7

MSE = 0.8663892292306695

RMSE = 0.9308003165183548

0.13999999999999996 9 7

MSE = 0.8739904599717725

RMSE = 0.9348745691116924

0.09 9 8

MSE = 0.8794232612595797

RMSE = 0.9377756987998674

0.09999999999999999 9 8

MSE = 0.8686132274752186

RMSE = 0.9319942207305895

0.10999999999999999 9 8

MSE = 0.8679683639157997

RMSE = 0.9316481975057965

0.11999999999999998 9 8

MSE = 0.8687598000692148

RMSE = 0.9320728512671179

0.12999999999999998 9 8

MSE = 0.8567151252113182

RMSE = 0.9255890693019868

0.13999999999999996 9 8

MSE = 0.8650388685911456

RMSE = 0.9300746575362354

0.09 9 9

MSE = 0.8732561447005208

RMSE = 0.9344817519355425

0.09999999999999999 9 9

MSE = 0.8709050682331649

RMSE = 0.9332229466923565

0.10999999999999999 9 9

MSE = 0.8690369381305955

RMSE = 0.9322215070092491

0.11999999999999998 9 9

MSE = 0.8687770232915842

RMSE = 0.9320820904252931

0.12999999999999998 9 9

MSE = 0.8641472784490486

RMSE = 0.9295952229056734

0.13999999999999996 9 9

MSE = 0.8589563319841278

RMSE = 0.9267989706425702

<class 'list'>

[(<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DB898>, 0.9233618309210345, 5, 7, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D60B8>, 0.923636622097201, 5, 6, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6208>, 0.9247228600113394, 4, 9, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B3358>, 0.9247528124469566, 3, 9, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C1C18>, 0.9248728187135913, 4, 7, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9940>, 0.9255214214743663, 6, 5, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942F28>, 0.9255890693019868, 9, 8, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C61D0>, 0.9266923792131528, 4, 9, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336904B70>, 0.9267073840222178, 6, 7, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336935BE0>, 0.9267989706425702, 9, 9, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D6358>, 0.9269292998889375, 7, 9, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DBC50>, 0.9269584931553303, 5, 7, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DB080>, 0.9271016080146972, 5, 8, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DF748>, 0.9272120503242736, 7, 7, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870518>, 0.927227077905814, 3, 9, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369179E8>, 0.9273104312178242, 8, 9, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687B4E0>, 0.9273545757541309, 4, 9, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ABC88>, 0.927406358697065, 4, 5, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692C4A8>, 0.9275657696264643, 7, 9, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692C278>, 0.9279512521897596, 7, 9, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369048D0>, 0.9280072065377043, 6, 9, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D62E8>, 0.9282140906597279, 5, 6, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6A20>, 0.9282971504282906, 3, 9, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C16D8>, 0.928392633346506, 4, 8, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693ADA0>, 0.9284335140870555, 8, 8, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336900D30>, 0.9285243959198579, 6, 9, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C17B8>, 0.9285283398070392, 4, 7, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D6CF8>, 0.9285697487781653, 8, 8, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893400>, 0.9286936458306352, 4, 9, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368938D0>, 0.9287256806808132, 4, 5, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687B5F8>, 0.9287440755455351, 4, 6, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C1208>, 0.9287548555907561, 4, 8, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899048>, 0.928767277691442, 2, 9, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836198>, 0.9287798167105206, 2, 8, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5630>, 0.9288022852101759, 5, 9, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AA2E8>, 0.9288844876935978, 3, 7, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ADC50>, 0.9289004898834914, 4, 6, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368217B8>, 0.92894733034959, 2, 7, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336900470>, 0.9291454733640455, 6, 8, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D6710>, 0.9292053661291321, 7, 9, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336933CC0>, 0.9292600381222786, 8, 5, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DB400>, 0.9292683025092257, 5, 9, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368936A0>, 0.9293509550586442, 4, 6, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336900BA8>, 0.9295479394794086, 6, 8, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336904240>, 0.929568789798422, 6, 6, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369003C8>, 0.9295952229056734, 9, 9, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B3C50>, 0.9297026857890225, 3, 8, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D048>, 0.9298221114926487, 2, 9, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893B70>, 0.9299376603576924, 5, 8, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D710>, 0.929946030619719, 4, 9, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336900DA0>, 0.9300569956554231, 6, 9, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369470B8>, 0.9300746575362354, 9, 8, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870C50>, 0.9301219906847136, 2, 8, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336933550>, 0.9301370407351713, 8, 9, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6DD8>, 0.9301436225250452, 3, 9, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DBB70>, 0.9301685428685185, 5, 8, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693A2B0>, 0.9301895802890758, 8, 8, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33690BC18>, 0.9301896538798534, 7, 7, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33690BB38>, 0.9301973078794352, 7, 7, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DBDD8>, 0.9304537625186966, 7, 9, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368995C0>, 0.9304632069402823, 2, 7, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D898>, 0.9305171739850868, 2, 9, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E8390>, 0.9305183620236198, 6, 5, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368994E0>, 0.9305728611005198, 2, 9, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692CBE0>, 0.9307687406787651, 7, 8, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336947898>, 0.9308003165183548, 9, 7, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D198>, 0.9309127581839917, 2, 9, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DF128>, 0.9310666251821198, 7, 8, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336904BE0>, 0.9312048556038512, 6, 5, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C1CF8>, 0.9312553470143883, 5, 7, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692CC18>, 0.9313546528706579, 7, 8, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AD048>, 0.9314081462574993, 3, 7, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D6470>, 0.9314409280900213, 5, 5, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AADA0>, 0.9314469738142577, 3, 7, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693A8D0>, 0.9314639310985714, 8, 7, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33690B208>, 0.9314819214371404, 7, 6, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DBDA0>, 0.9315084214258771, 5, 9, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893898>, 0.931580145638821, 4, 8, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336917668>, 0.9316481975057965, 9, 8, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9BE0>, 0.9316538032850633, 5, 9, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899128>, 0.9316616417629232, 6, 8, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AAEF0>, 0.9318225068647151, 4, 7, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5978>, 0.9319370140554168, 5, 8, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336947358>, 0.9319942207305895, 9, 8, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336947F98>, 0.9320728512671179, 9, 8, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336947908>, 0.9320820904252931, 9, 9, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369171D0>, 0.932145504326023, 8, 9, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369001D0>, 0.9321864942133847, 6, 8, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336876DD8>, 0.9322087526615063, 2, 4, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942550>, 0.9322215070092491, 9, 9, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369002E8>, 0.9322450077440626, 6, 9, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899F28>, 0.9322587237245774, 2, 8, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693A208>, 0.9322974648902471, 8, 7, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336904D30>, 0.9323157083152849, 6, 9, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369044A8>, 0.9323279090789811, 6, 6, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B33C8>, 0.9324107132933257, 7, 8, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336821A20>, 0.9324607704298711, 5, 7, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336904B00>, 0.9326245916399792, 6, 7, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33694AE80>, 0.9326779185213366, 9, 6, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369005F8>, 0.9328506807723634, 6, 8, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9DD8>, 0.9329732039967232, 6, 7, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D6BA8>, 0.9329965077001374, 5, 5, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692CE80>, 0.9330743260785278, 8, 7, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861048>, 0.933109308088074, 2, 5, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E83C8>, 0.9331221689776792, 7, 5, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C1BE0>, 0.9331345416227843, 4, 6, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687B978>, 0.9331433441615544, 5, 7, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6A20>, 0.9331785374821697, 5, 8, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33690B860>, 0.9331937244820606, 7, 7, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369477F0>, 0.9332229466923565, 9, 9, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336831A58>, 0.9332441632838148, 2, 7, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5A58>, 0.933264507567678, 5, 5, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AA320>, 0.933298180037377, 3, 7, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B3940>, 0.9333168916004487, 3, 9, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B3748>, 0.9333340359773853, 3, 9, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899320>, 0.9333357309869718, 6, 7, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688BA20>, 0.9333656273001905, 2, 7, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6E10>, 0.9334042192242683, 6, 6, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369330B8>, 0.9334225782071622, 8, 5, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692CEF0>, 0.9334380011770483, 8, 6, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DFCF8>, 0.9334743083985095, 7, 8, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336821D68>, 0.9335011237416345, 2, 5, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D5C0>, 0.9335044882912733, 3, 5, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336900710>, 0.933550914493412, 8, 5, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5BA8>, 0.9336059048654445, 5, 9, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A62B0>, 0.9336240834899625, 3, 7, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369047B8>, 0.9336254544424193, 6, 6, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688DC18>, 0.9336399536826623, 4, 8, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AACF8>, 0.9336915661989945, 4, 5, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693A400>, 0.9337180319063272, 8, 9, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693A198>, 0.9337354192926105, 8, 7, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C19B0>, 0.9337948274302016, 4, 7, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D335830470>, 0.9338111169134637, 2, 8, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ADFD0>, 0.9338159320414777, 3, 8, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ADB70>, 0.9338166680489077, 3, 5, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DB748>, 0.9338383582580272, 5, 7, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9550>, 0.9338705816634374, 7, 5, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5048>, 0.9339233882249882, 5, 9, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942240>, 0.934014691450837, 9, 6, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C69E8>, 0.9340555055758795, 4, 9, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336831198>, 0.934068772073172, 2, 5, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693ABE0>, 0.9340887197755358, 8, 6, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DBAC8>, 0.9340934116016636, 7, 6, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33694A470>, 0.9340976619874473, 9, 5, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336900E80>, 0.93410853954148, 8, 5, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33690BDA0>, 0.9341714478386567, 8, 7, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942860>, 0.9341896849332436, 9, 7, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870438>, 0.9342453852295536, 4, 8, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B3898>, 0.9342476597839001, 3, 8, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336917208>, 0.9342578367538978, 8, 9, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9710>, 0.9343315711083041, 6, 7, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ADC88>, 0.934408591913802, 3, 8, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5C50>, 0.9344558260872272, 5, 4, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942B00>, 0.9344817519355425, 9, 9, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33694A588>, 0.9345284269127959, 9, 6, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692C898>, 0.9346382794353641, 8, 6, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693A470>, 0.9347635831922987, 8, 8, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DBE48>, 0.9348158492320046, 5, 8, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AD2E8>, 0.9348495797545707, 3, 7, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368933C8>, 0.9348700341852294, 4, 6, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336947748>, 0.9348745691116924, 9, 7, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5710>, 0.9349008169834143, 5, 6, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336933AC8>, 0.9349371223054186, 8, 6, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D668>, 0.9350179125232714, 3, 5, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688B550>, 0.935023301341517, 2, 6, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DFD30>, 0.9350285130138346, 7, 5, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6B70>, 0.935115184435265, 6, 6, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899AC8>, 0.9351445337219934, 2, 8, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893DA0>, 0.9352473286310161, 4, 6, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D6B38>, 0.9352762545064518, 8, 8, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33690B9E8>, 0.9353191297300447, 7, 6, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AABA8>, 0.9353592448942909, 3, 6, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369425F8>, 0.9354274059502913, 9, 6, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ADCF8>, 0.9354715509323623, 3, 5, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33694A8D0>, 0.935481364634862, 9, 7, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AAFD0>, 0.9354955347369462, 4, 8, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336904FD0>, 0.9355178087936994, 7, 5, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DB780>, 0.9355554293872824, 6, 6, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DFBA8>, 0.9355677870325163, 7, 4, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ABAC8>, 0.9355929019188193, 4, 5, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AD208>, 0.9356043907170657, 3, 5, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6668>, 0.9356228021734947, 6, 7, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5198>, 0.9356466630425585, 5, 6, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33694A2E8>, 0.9356680114490141, 9, 6, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687BE48>, 0.935727668520654, 3, 8, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DFA58>, 0.9357628693744011, 7, 7, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687BE10>, 0.9357903034912586, 2, 4, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DFA90>, 0.9358434078866223, 7, 4, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870F60>, 0.9358685389231709, 2, 7, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AD9B0>, 0.9359656509834688, 3, 4, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693ABA8>, 0.9360476640311697, 8, 7, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687B1D0>, 0.9360922983118564, 3, 8, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942E80>, 0.9361799960311659, 9, 4, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DF390>, 0.9361860998746999, 8, 4, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692C7B8>, 0.9362279635025815, 7, 9, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336904C18>, 0.9362395521665191, 6, 5, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D358>, 0.9362803033028941, 5, 4, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686DE80>, 0.9363130473190588, 3, 6, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AA0F0>, 0.936317435977268, 4, 5, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3358186A0>, 0.9363450803491759, 4, 3, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E8C88>, 0.9363594608474252, 6, 4, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E59E8>, 0.9364020510857908, 5, 6, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9F98>, 0.9364191772728699, 6, 4, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686DF60>, 0.9365132974855532, 7, 5, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AB9B0>, 0.936758835116393, 6, 9, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AA9E8>, 0.9367699587463533, 4, 7, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336947C50>, 0.9367978217657057, 9, 7, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33690B1D0>, 0.937001575890781, 7, 6, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DF7B8>, 0.9370067840804892, 7, 4, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33694A320>, 0.9371524874500788, 9, 5, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369337F0>, 0.9371914972290676, 8, 6, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AB550>, 0.9373066459322719, 4, 5, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942128>, 0.9375784315463812, 9, 6, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C1470>, 0.9376596473567489, 5, 4, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DFAC8>, 0.9376644032211068, 7, 7, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336917438>, 0.9377756987998674, 9, 8, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ADF98>, 0.9378594843583924, 3, 6, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368998D0>, 0.9379488679468809, 5, 3, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33694A5F8>, 0.9379542418697226, 9, 5, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692C358>, 0.9379852561954616, 8, 4, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6048>, 0.9379957828422748, 8, 8, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336933358>, 0.9380081322504616, 8, 5, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DB0F0>, 0.9380269752066762, 6, 3, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D6518>, 0.9381166015159282, 5, 6, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E86D8>, 0.9381377780968675, 8, 9, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33694ABA8>, 0.9382408366551783, 9, 5, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C1198>, 0.9382428141980363, 4, 7, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D6390>, 0.9384120975372557, 7, 6, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836438>, 0.9384227762580195, 2, 4, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6D30>, 0.9386527636024062, 6, 8, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693AF98>, 0.9386818913868988, 9, 4, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899588>, 0.9387669606660746, 9, 5, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E8E80>, 0.9388905677988413, 7, 4, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942358>, 0.9390043100553648, 9, 7, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C10B8>, 0.9391089925723763, 5, 4, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D65F8>, 0.9391866442146524, 5, 5, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336900668>, 0.9392733491793124, 8, 5, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AA5F8>, 0.9394538215813271, 3, 6, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336831208>, 0.9395325381651617, 2, 6, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B34A8>, 0.939537652779066, 7, 8, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E86A0>, 0.9396392893330934, 6, 5, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899E10>, 0.9396697805908346, 2, 9, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688DB70>, 0.9397017381487095, 3, 5, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9CC0>, 0.9400492808293607, 7, 6, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B3278>, 0.9401919867982788, 5, 5, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D6320>, 0.940561394041751, 5, 5, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E8FD0>, 0.9406135989892535, 6, 4, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6E80>, 0.9406386546597306, 9, 4, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692C438>, 0.9408013528200916, 8, 4, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AA470>, 0.9408067642013984, 3, 6, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368313C8>, 0.9409842877625357, 2, 6, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687B860>, 0.9410050898283127, 2, 5, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893630>, 0.9410853377964161, 4, 4, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AD940>, 0.9412619182296015, 3, 4, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336831D68>, 0.9413218444639556, 2, 7, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DF198>, 0.941786191736319, 7, 4, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686D400>, 0.9419041524966011, 6, 3, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368214E0>, 0.9420136496598588, 2, 6, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AB1D0>, 0.9422501896538523, 4, 4, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368D6C88>, 0.9422785797407752, 9, 3, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870EB8>, 0.9423265309813041, 3, 4, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368EC208>, 0.9425484933030217, 7, 3, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336933C50>, 0.9425616518203555, 8, 4, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DFC50>, 0.9426039478027809, 7, 5, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E8D30>, 0.942665239119323, 6, 4, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336876D68>, 0.942738858446665, 9, 4, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E57B8>, 0.942802158312034, 5, 3, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942208>, 0.9428678273630178, 9, 5, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AB128>, 0.9429297215874142, 7, 3, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6828>, 0.9432263679678611, 3, 3, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AD7B8>, 0.943234433141594, 3, 4, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942278>, 0.9433252491820604, 9, 3, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686DAC8>, 0.9433977482048265, 3, 3, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836AC8>, 0.9434343304679579, 2, 5, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336904A20>, 0.9434472272419762, 7, 3, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942908>, 0.9434642750053684, 9, 4, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E85C0>, 0.9435900552482772, 7, 4, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692CF60>, 0.9436393327027759, 8, 3, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693ACC0>, 0.9438019320080419, 9, 4, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369427B8>, 0.9439810555990844, 9, 3, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B3080>, 0.9440739045064204, 8, 6, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336921F98>, 0.9442724831328997, 8, 3, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ADE10>, 0.94430185635337, 4, 4, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368935C0>, 0.9444814982661205, 5, 4, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893B00>, 0.9448999008826846, 4, 4, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336821978>, 0.9449802070294098, 6, 4, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9320>, 0.9451206349966546, 6, 5, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336933898>, 0.9452169223898884, 8, 4, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336933748>, 0.9454299325472514, 8, 4, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D7B8>, 0.9455492225597554, 3, 3, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9630>, 0.9455851119196009, 6, 4, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E81D0>, 0.9460817229440011, 6, 3, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AB208>, 0.9463245167322283, 4, 4, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369429B0>, 0.9464624701619632, 9, 3, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5240>, 0.9466922703182633, 5, 3, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6BA8>, 0.9467256929488918, 4, 3, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688B470>, 0.9469854034163033, 2, 6, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ADC18>, 0.9472283375199697, 3, 4, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687B198>, 0.9472800034087961, 2, 4, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D3C8>, 0.9476624047917539, 2, 3, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368937F0>, 0.9479122394727713, 4, 3, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D6D8>, 0.9480280059369198, 3, 3, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E8208>, 0.9484401590073317, 6, 3, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899080>, 0.9485648242106998, 5, 3, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AB898>, 0.9486938196125255, 7, 3, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870DA0>, 0.949082030892674, 1, 8, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ECE10>, 0.9492435476081702, 9, 3, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F90F0>, 0.9493914689259013, 6, 3, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B3390>, 0.9496049115696436, 8, 3, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368937B8>, 0.9496342442575914, 4, 3, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687B710>, 0.9497067708103143, 2, 3, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688B668>, 0.9499776965082835, 1, 9, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369210B8>, 0.9499879147816717, 8, 3, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368EC860>, 0.9503348157654903, 7, 3, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870EF0>, 0.9505029781741128, 1, 5, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336831BE0>, 0.9508424202253887, 1, 6, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368214A8>, 0.9509264220625946, 3, 4, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861320>, 0.9509807860810536, 1, 7, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870470>, 0.9511896538965351, 1, 5, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336876C88>, 0.951386736441474, 2, 3, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688BCC0>, 0.9515627030608701, 5, 4, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AB9E8>, 0.9515917860948372, 4, 4, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692C4E0>, 0.9516272033011662, 8, 3, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368990B8>, 0.9516748780752393, 2, 5, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836D68>, 0.9520750837191551, 1, 9, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688DA20>, 0.9521298321328664, 2, 3, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861F28>, 0.9521888977318376, 2, 4, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688B908>, 0.9527003447139593, 1, 8, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836E80>, 0.9527727501973451, 1, 6, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6390>, 0.9527943347669483, 3, 6, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836DD8>, 0.9531733179778252, 1, 9, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686D710>, 0.9531899047072055, 1, 6, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686D160>, 0.9532443265164864, 1, 5, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ECFD0>, 0.953281002641059, 9, 3, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336921BE0>, 0.9536516362589995, 8, 3, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336821DA0>, 0.9536943455158188, 1, 4, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368767F0>, 0.9538599293462242, 1, 6, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E89E8>, 0.9541818741560727, 6, 3, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686D588>, 0.9542699984608617, 2, 6, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E59B0>, 0.954272060350442, 5, 3, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688DFD0>, 0.9543232034073522, 1, 9, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688B748>, 0.9544337089662832, 1, 9, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893AC8>, 0.9544445794553599, 4, 3, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336876B70>, 0.9549771846832972, 1, 5, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336876668>, 0.9551787051609908, 1, 5, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336831FD0>, 0.9554146984305883, 1, 6, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688B588>, 0.9555102135420181, 1, 9, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DFBE0>, 0.9558476725631341, 7, 3, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861710>, 0.9561292912202299, 1, 8, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870DD8>, 0.9562520275615312, 1, 4, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899DA0>, 0.9564044051183809, 5, 3, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861278>, 0.9565124553076844, 1, 7, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A67B8>, 0.9566320377224974, 3, 3, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368612B0>, 0.9568576222220064, 2, 4, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861898>, 0.9571230127528128, 1, 6, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861AC8>, 0.9577945818782732, 1, 8, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688BB70>, 0.958047192016934, 1, 8, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870978>, 0.958780553872203, 1, 7, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336821C88>, 0.9588975378096111, 1, 7, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ADBE0>, 0.9589402460357622, 3, 3, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861D68>, 0.959674418003874, 1, 7, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688DE10>, 0.9602937828988101, 2, 8, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368216D8>, 0.9606772120393985, 1, 3, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A69E8>, 0.9613714867880088, 4, 3, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D335830160>, 0.9623294423195442, 1, 4, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686D2B0>, 0.9624522501262206, 1, 3, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688B198>, 0.9626542963519478, 1, 8, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368702B0>, 0.9628822550065594, 1, 7, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686D5F8>, 0.9635483012407642, 1, 3, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687B7B8>, 0.9640417458454996, 2, 3, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686D6A0>, 0.9656031878755105, 1, 4, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686D128>, 0.9670641592422113, 1, 3, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686DA20>, 0.9705292508626272, 1, 3, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693A898>, 0.9708625929386184, 9, 2, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368705C0>, 0.9714260213549927, 1, 5, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368EC358>, 0.9730183300898961, 7, 2, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369009B0>, 0.9755928227863054, 9, 2, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369000B8>, 0.9758297035336819, 9, 2, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AAF98>, 0.9766798112172081, 6, 2, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F99B0>, 0.9775535870635812, 6, 2, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368760B8>, 0.9788930228352465, 1, 4, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686D3C8>, 0.980120313485953, 7, 2, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693A518>, 0.980147483108384, 9, 2, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336921470>, 0.9817455922136906, 8, 2, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5E10>, 0.981873314578662, 6, 2, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688B4A8>, 0.9831321147491064, 5, 2, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369215F8>, 0.9834141859476782, 8, 2, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9AC8>, 0.9839258021069474, 6, 2, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336870A90>, 0.9840229798121198, 1, 3, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ECCC0>, 0.984563879623931, 7, 2, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E8048>, 0.984842285970707, 7, 2, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33690B278>, 0.9862397302807483, 8, 2, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942C18>, 0.9868498533670579, 9, 2, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6898>, 0.9895142509331494, 5, 2, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893198>, 0.990852553117003, 4, 2, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336921898>, 0.9914671351494223, 8, 2, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336942BE0>, 0.9918677572628183, 9, 2, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336904F28>, 0.9934059405658081, 7, 2, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368EC0B8>, 0.9936879784365827, 7, 2, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C1FD0>, 0.9946614568188313, 5, 2, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C1DD8>, 0.995069035676717, 5, 2, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368703C8>, 0.9975133995112275, 6, 2, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336921BA8>, 0.9983367971832062, 8, 2, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336831550>, 1.0001138502648321, 3, 2, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368618D0>, 1.0050726353754085, 4, 2, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5BE0>, 1.0079930628221099, 5, 2, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33690BFD0>, 1.0080856643265572, 8, 2, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893860>, 1.0094034556094589, 4, 2, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687B5C0>, 1.0096491029286232, 2, 2, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368E5C18>, 1.0131724848850707, 5, 2, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6F28>, 1.0143119391457975, 3, 2, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F97F0>, 1.0160865925989466, 6, 2, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687B550>, 1.0171024760494307, 2, 2, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893BA8>, 1.0177596190922198, 4, 2, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336893BE0>, 1.0178416811446813, 4, 2, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688DB00>, 1.033867577239832, 3, 2, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688B2E8>, 1.0428266991528585, 4, 2, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6208>, 1.0467369202201504, 3, 2, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687BD30>, 1.0470510117138208, 2, 3, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688DA90>, 1.0602810723202332, 2, 2, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368317B8>, 1.060425334901352, 1, 2, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688BDA0>, 1.0618969602987889, 2, 2, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A64E0>, 1.0746837356973065, 3, 2, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686DB00>, 1.0759627024334648, 1, 2, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861FD0>, 1.0806526513613655, 3, 2, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836048>, 1.1343573327815095, 1, 2, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33687BCC0>, 1.155569420120672, 2, 2, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836518>, 1.1670905890441208, 1, 2, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336831A20>, 1.1910754844722355, 1, 2, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368700B8>, 1.2694991692734652, 2, 2, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33686D7F0>, 1.484531291110868, 1, 4, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336831128>, 1.9001971464399743, 1, 2, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DF7F0>, 2.2116452187102813, 8, 1, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6860>, 2.37442224707311, 5, 1, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DFDA0>, 2.4183557487885055, 8, 1, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336917C88>, 2.484691229668919, 9, 1, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368DF978>, 2.562763058964101, 8, 1, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368EC898>, 2.607006758307127, 7, 1, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9BA8>, 2.628453815469802, 6, 1, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688D588>, 2.6843729031875823, 5, 1, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368EC748>, 2.7050840196281687, 7, 1, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B30B8>, 2.744424302232954, 8, 1, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6D68>, 2.7653585976328277, 6, 1, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336917630>, 2.78422025741443, 9, 1, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336917DA0>, 2.7892132355281807, 9, 1, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F96A0>, 2.8687688595345877, 7, 1, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899D68>, 2.8737961447578977, 9, 1, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6C88>, 2.886151685587028, 5, 1, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ECC50>, 2.9132091356424215, 7, 1, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693AC88>, 2.9132570001366678, 9, 1, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B31D0>, 2.9347792888467046, 4, 1, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692C7F0>, 2.9511937294677297, 8, 1, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3369045F8>, 2.962318515933828, 7, 1, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6198>, 2.9917399341350874, 3, 1, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33693AA20>, 3.002578193123372, 9, 1, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F97B8>, 3.0062334651368205, 6, 1, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9860>, 3.068982042837319, 6, 1, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861EF0>, 3.0909809545352456, 4, 1, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33692C550>, 3.094395228539765, 8, 1, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6278>, 3.10811453127007, 3, 1, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688DA58>, 3.1095519732618007, 3, 1, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368AD710>, 3.1207985459220557, 4, 1, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C14A8>, 3.2262812774499854, 5, 1, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368ADE80>, 3.227406301276083, 4, 1, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B3320>, 3.2443778780710244, 4, 1, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336904860>, 3.2952435973888115, 7, 1, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861550>, 3.309751246345264, 2, 1, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C1B00>, 3.3205995103458363, 5, 1, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368611D0>, 3.3391581449288914, 2, 1, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368B3E48>, 3.346564148227114, 4, 1, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D33688DCF8>, 3.347192950414146, 3, 1, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836A58>, 3.3703828679252315, 2, 1, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836588>, 3.3812932219378267, 1, 1, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836860>, 3.3856180619509004, 1, 1, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368F9198>, 3.4419762687925703, 6, 1, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6DA0>, 3.481783466555942, 5, 1, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368A6CF8>, 3.493860374100228, 3, 1, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D335818A90>, 3.5029987894259342, 1, 1, 0.09), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336899A90>, 3.5105744418321434, 3, 1, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368C6630>, 3.533406975506498, 6, 1, 0.12999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368367F0>, 3.534997456395586, 2, 1, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D335818780>, 3.6090858250514466, 1, 1, 0.11999999999999998), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336861D30>, 3.609282637104158, 2, 1, 0.09999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336836080>, 3.6173204693577743, 2, 1, 0.13999999999999996), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D3368312B0>, 3.6213078155482417, 1, 1, 0.10999999999999999), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000002D336831438>, 3.6243657982930704, 1, 1, 0.12999999999999998)]

The best parameters, rank=5, iterations=7, lambda\_=0.12999999999999998

**说明rank=5，iterations=7，lambda=0.13时有最小误差**

MSE = 0.877081075894607

RMSE = 0.9365260679204862

Step 5：

前面考虑问题思想出了问题。虽然说因为空间有限，为了防止

java.lang.StackOverflowError

超过栈内存，所以限制rank，iterations，但是限制<10还是出了偏差。

但是如果扩大范围，需要更多的时间，空间来计算。但是因为项目结束已经没多少时间了，所以要加速。故采用比较简单的，范围比较小的测试方法。

我们确定lambda为0.13不再改变。Rank和iterations为5,10,15,20，在这里面大概找到合适的参数。

代码：

# 使用Spark MLlib中推荐算法ALS对电影评分数据MovieLens推荐

from pyspark.sql import SparkSession

from pyspark.mllib.recommendation import ALS, Rating, MatrixFactorizationModel

from pyspark.mllib.regression import LabeledPoint, LinearRegressionWithSGD

from pyspark.mllib.evaluation import RegressionMetrics

from pyspark.mllib.linalg import DenseVector

import numpy as np #arange

def alsModelEvaluate(model, testing\_rdd):

# 对测试数据集预测评分，针对测试数据集进行预测

predict\_rdd = model.predictAll(testing\_rdd.map(lambda r: (r[0], r[1])))

#print(predict\_rdd.take(5))

predict\_actual\_rdd = predict\_rdd.map(lambda r: ((r[0], r[1]), r[2])) \

.join(testing\_ratings.map(lambda r: ((r[0], r[1]), r[2])))

#print(predict\_actual\_rdd.take(5))

# 创建评估指标实例对象

metrics = RegressionMetrics(predict\_actual\_rdd.map(lambda pr: pr[1]))

print("MSE = %s" % metrics.meanSquaredError)

print("RMSE = %s" % metrics.rootMeanSquaredError)

# 返回均方根误差

return metrics.rootMeanSquaredError

def train\_model\_evaluate(training\_rdd, testing\_rdd, rank, iterations, lambda\_):

# 定义函数，训练模型与模型评估

# 使用超参数的值，训练数据和ALS算法训练模型

print(lambda\_,rank,iterations)

model = ALS.train(training\_rdd, rank, iterations, lambda\_)

# 模型的评估

rmse\_value = alsModelEvaluate(model, testing\_rdd)

# 返回多元组

return (model, rmse\_value, rank, iterations, lambda\_)

if \_\_name\_\_ == "\_\_main\_\_":

# 构建SparkSession实例对象

# local 127.0.0.1

spark = SparkSession.builder \

.appName("SparkSessionExample") \

.master("local") \

.getOrCreate()

# 获取SparkContext实例对象

sc = spark.sparkContext

# 读取数据

raw\_ratings\_rdd = sc.textFile("C:/Users/lenovo/Desktop/ml-100k/ml-100k/u.data")

# print(raw\_ratings\_rdd.count())

# print(raw\_ratings\_rdd.first())

# 获取评分数据前三个字段，构建Rating实例对象

ratings\_rdd = raw\_ratings\_rdd.map(lambda line: line.split('\t')[0:3])

# print(ratings\_rdd.first())

ratings\_datas = ratings\_rdd.map(lambda x: Rating(int(x[0]), int(x[1]), float(x[2])))

# print(ratings\_datas.first())

# 将数据集分为训练数据集和测试数据集

training\_ratings, testing\_ratings = ratings\_datas.randomSplit([0.7, 0.3])

'''

# 使用ALS算法来训练模型

# help(ALS)

# 采用显示评分函数训练模型

alsModel = ALS.train(training\_ratings, 10, iterations=10, lambda\_=0.01)

# 用户特征因子矩阵

user\_feature\_matrix = alsModel.userFeatures()

print(type(user\_feature\_matrix))

print(user\_feature\_matrix.take(10))

# 物品因子矩阵

item\_feature\_matrix = alsModel.productFeatures()

print(type(item\_feature\_matrix))

print(item\_feature\_matrix.take(10))

# 预测某个用户对某个电影的评分

# 假设用户196，对电影242的评分，实际评分为3分

predictRating = alsModel.predict(196, 242)

print(predictRating)

# 为用户推荐（10部电影）

rmdMovies = alsModel.recommendProducts(196, 10)

print(rmdMovies)

# 为电影推荐（10个用户）

rmdUsers = alsModel.recommendUsers(242, 10)

print(rmdUsers)

'''

# 怎么评价模型的好坏，ALS模型评估指标(类似回归算法模型预测值，连续值)，使用回归模型中

# RMSE（均方根误差）评估模型

# 找到最佳模型

'''

如何找到最佳模型？？

-a. 模型的评估

计算RMSE

-b. 模型的优化，两个方向

1、数据

2、超参数的调整，选择合适的超参数的值，得到最优模型

交叉验证

训练数据集、验证数据集、测试数据集

K-Folds交叉验证

'''

# ALS算法的超参数的调整

# 定义一个函数，用于对模型进行评估

# 使用三层for循环，设置不同参数的值，分别使用ALS算法训练模型，评估获取RMSE的值

metrix\_list = [train\_model\_evaluate(training\_ratings, testing\_ratings,param\_rank,param\_iterations, 0.13)

for param\_rank in [5,10,15,20]

for param\_iterations in [5,10,15,20]

#for param\_rank in range(1, 10)

#for param\_iterations in range(1, 10)

#for param\_lambda in np.arange(0.09,0.14,0.01)

]

print(type(metrix\_list))

#sorted(metrix\_list, key=lambda k: k[1], reverse=False)

print("\n")

print("\n")

metrix\_list.sort(key=lambda s: s[1])

print(metrix\_list)

print("\n")

model, rmse\_value, rank, iterations, lambda\_ = metrix\_list[0]

print("The best parameters, rank=%s, iterations=%s, lambda\_=%s" % (rank ,iterations ,lambda\_))

结果：

0.13 5 5

MSE = 0.8897023325962911

RMSE = 0.9432403366037158

0.13 5 10

MSE = 0.8666731501751669

RMSE = 0.930952818447405

0.13 5 15

MSE = 0.8631021193759038

RMSE = 0.929032894668377

0.13 5 20

MSE = 0.867435721781092

RMSE = 0.9313622935147696

0.13 10 5

MSE = 0.8963179691667347

RMSE = 0.9467407085188292

0.13 10 10

MSE = 0.8681726606918821

RMSE = 0.931757833716402

0.13 10 15

MSE = 0.8657303052657761

RMSE = 0.9304462935955928

0.13 10 20

MSE = 0.8649543696623578

RMSE = 0.9300292305418996

0.13 15 5

MSE = 0.8885943316414998

RMSE = 0.9426528160683019

0.13 15 10

MSE = 0.8726161927369172

RMSE = 0.9341392790890003

0.13 15 15

MSE = 0.8648388541624733

RMSE = 0.9299671253127572

0.13 15 20

MSE = 0.8644500932006702

RMSE = 0.9297580831596304

0.13 20 5

MSE = 0.8812356350764649

RMSE = 0.9387415166468696

0.13 20 10

MSE = 0.870184560353846

RMSE = 0.9328368347968716

0.13 20 15

MSE = 0.8640731043612488

RMSE = 0.9295553261432311

0.13 20 20

MSE = 0.8648719431347017

RMSE = 0.9299849155414843

<class 'list'>

[(<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D8F320>, 0.929032894668377, 5, 15, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCF080>, 0.9295553261432311, 20, 15, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD23C8>, 0.9297580831596304, 15, 20, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCE978>, 0.9299671253127572, 15, 15, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD2A20>, 0.9299849155414843, 20, 20, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D81898>, 0.9300292305418996, 10, 20, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCE630>, 0.9304462935955928, 10, 15, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D994E0>, 0.930952818447405, 5, 10, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CBFFC596A0>, 0.9313622935147696, 5, 20, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D814E0>, 0.931757833716402, 10, 10, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD2CF8>, 0.9328368347968716, 20, 10, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D81780>, 0.9341392790890003, 15, 10, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCF2E8>, 0.9387415166468696, 20, 5, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCEB00>, 0.9426528160683019, 15, 5, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CBFFC59B00>, 0.9432403366037158, 5, 5, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D991D0>, 0.9467407085188292, 10, 5, 0.13)]

The best parameters, rank=5, iterations=15, lambda\_=0.13

MSE = 0.8631021193759038

RMSE = 0.929032894668377

这里确实比之前的5,7,0.13要低。

这样下一步我们在rank=5, iterations=15附近继续寻找参数。

Step 6：

Rank在5~10，iterations在13~23之间，lambda为0.13，寻找合适值

代码：

需要改变的只有matrix\_list的生成部分

metrix\_list = [train\_model\_evaluate(training\_ratings, testing\_ratings,param\_rank,param\_iterations, 0.13)

for param\_rank in range(5,10)

for param\_iterations in range(12,23)

]

结果：

0.13 5 12

MSE = 0.8733132300157396

RMSE = 0.9345122952726409

0.13 5 13

MSE = 0.8630877609006291

RMSE = 0.9290251669899094

0.13 5 14

MSE = 0.8538969428700047

RMSE = 0.9240654429584545

0.13 5 15

MSE = 0.8567073758099158

RMSE = 0.925584883092802

0.13 5 16

MSE = 0.856407549279642

RMSE = 0.9254229029366207

0.13 5 17

MSE = 0.8571004840612247

RMSE = 0.92579721541017

0.13 5 18

MSE = 0.8583270549240297

RMSE = 0.9264594189299549

0.13 5 19

MSE = 0.8535052576646753

RMSE = 0.9238534827907915

0.13 5 20

MSE = 0.8589273318445636

RMSE = 0.9267833251869412

0.13 5 21

MSE = 0.853620720529332

RMSE = 0.9239159704915443

0.13 5 22

MSE = 0.8589204721855582

RMSE = 0.9267796243905874

0.13 6 12

MSE = 0.8602657830535533

RMSE = 0.9275051390981903

0.13 6 13

MSE = 0.8571710204979359

RMSE = 0.9258353095977362

0.13 6 14

MSE = 0.8607203417765782

RMSE = 0.9277501505128297

0.13 6 15

MSE = 0.8592591184479144

RMSE = 0.9269623069186332

0.13 6 16

MSE = 0.8571348463482114

RMSE = 0.9258157734388691

0.13 6 17

MSE = 0.8602972938827969

RMSE = 0.9275221258184609

0.13 6 18

MSE = 0.8557703529666193

RMSE = 0.9250785658346102

0.13 6 19

MSE = 0.8562261673625082

RMSE = 0.9253248982722275

0.13 6 20

MSE = 0.8547443026743563

RMSE = 0.9245238248278712

0.13 6 21

MSE = 0.854952447935999

RMSE = 0.924636386876484

0.13 6 22

MSE = 0.8522183391109492

RMSE = 0.9231567251073618

0.13 7 12

MSE = 0.8606206435058876

RMSE = 0.927696417749841

0.13 7 13

MSE = 0.8568773266116082

RMSE = 0.9256766857880824

0.13 7 14

MSE = 0.8592461284959658

RMSE = 0.9269553001606743

0.13 7 15

MSE = 0.8603625908550235

RMSE = 0.9275573248349794

0.13 7 16

MSE = 0.863486872747672

RMSE = 0.9292399435816736

0.13 7 17

MSE = 0.8522448382557218

RMSE = 0.9231710774584101

0.13 7 18

MSE = 0.8568234336387461

RMSE = 0.9256475752891843

0.13 7 19

MSE = 0.858711797042585

RMSE = 0.9266670367735031

0.13 7 20

MSE = 0.8558458907131599

RMSE = 0.9251193926802961

0.13 7 21

MSE = 0.8596179556871202

RMSE = 0.9271558421792532

0.13 7 22

MSE = 0.8552221631938739

RMSE = 0.9247822247393567

0.13 8 12

MSE = 0.8633390487907874

RMSE = 0.929160399926077

0.13 8 13

MSE = 0.8568280339677342

RMSE = 0.9256500602105172

0.13 8 14

MSE = 0.8553287956809245

RMSE = 0.9248398756979094

0.13 8 15

MSE = 0.8621868376605477

RMSE = 0.9285401648073969

0.13 8 16

MSE = 0.858682140218109

RMSE = 0.9266510347580199

0.13 8 17

MSE = 0.8558254860328158

RMSE = 0.9251083644810568

0.13 8 18

MSE = 0.8535041420915024

RMSE = 0.9238528790297199

0.13 8 19

MSE = 0.856292270020653

RMSE = 0.9253606162035712

0.13 8 20

MSE = 0.8569042026349766

RMSE = 0.9256912026345376

0.13 8 21

MSE = 0.856672042359578

RMSE = 0.925565795802534

0.13 8 22

MSE = 0.8555756487881273

RMSE = 0.9249733232845839

0.13 9 12

MSE = 0.8609606801610535

RMSE = 0.9278796690094322

0.13 9 13

MSE = 0.8592614681432613

RMSE = 0.9269635743346452

0.13 9 14

MSE = 0.8566465288487821

RMSE = 0.9255520130434497

0.13 9 15

MSE = 0.8616679022942227

RMSE = 0.9282606866038348

0.13 9 16

MSE = 0.8582052471875756

RMSE = 0.9263936782964225

0.13 9 17

MSE = 0.8561116176287094

RMSE = 0.9252629991676471

0.13 9 18

MSE = 0.8560889824952382

RMSE = 0.9252507673572815

0.13 9 19

MSE = 0.8572076215833223

RMSE = 0.9258550759073054

0.13 9 20

MSE = 0.8531760010431814

RMSE = 0.9236752681777192

0.13 9 21

MSE = 0.853753899066709

RMSE = 0.9239880405431171

0.13 9 22

MSE = 0.854163323779438

RMSE = 0.9242095670244049

<class 'list'>

[(<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D81BE0>, 0.9231567251073618, 6, 22, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD26A0>, 0.9231710774584101, 7, 17, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DE1240>, 0.9236752681777192, 9, 20, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99828>, 0.9238528790297199, 8, 18, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD2A58>, 0.9238534827907915, 5, 19, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D81AC8>, 0.9239159704915443, 5, 21, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D8F5C0>, 0.9239880405431171, 9, 21, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99198>, 0.9240654429584545, 5, 14, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81E01470>, 0.9242095670244049, 9, 22, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D81CF8>, 0.9245238248278712, 6, 20, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99BE0>, 0.924636386876484, 6, 21, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD2C18>, 0.9247822247393567, 7, 22, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CBFFC6C160>, 0.9248398756979094, 8, 14, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCE550>, 0.9249733232845839, 8, 22, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD2390>, 0.9250785658346102, 6, 18, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99550>, 0.9251083644810568, 8, 17, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99F60>, 0.9251193926802961, 7, 20, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DEF9B0>, 0.9252507673572815, 9, 18, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DEF7F0>, 0.9252629991676471, 9, 17, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99E80>, 0.9253248982722275, 6, 19, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD2F28>, 0.9253606162035712, 8, 19, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCE908>, 0.9254229029366207, 5, 16, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DE1FD0>, 0.9255520130434497, 9, 14, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99048>, 0.925565795802534, 8, 21, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD2080>, 0.925584883092802, 5, 15, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCEF60>, 0.9256475752891843, 7, 18, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DE1A90>, 0.9256500602105172, 8, 13, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D8FDA0>, 0.9256766857880824, 7, 13, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCF9E8>, 0.9256912026345376, 8, 20, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD2240>, 0.92579721541017, 5, 17, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD2F98>, 0.9258157734388691, 6, 16, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCEBA8>, 0.9258353095977362, 6, 13, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DEF940>, 0.9258550759073054, 9, 19, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CBFFC6C128>, 0.9263936782964225, 9, 16, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D81710>, 0.9264594189299549, 5, 18, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DC7160>, 0.9266510347580199, 8, 16, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DC7F60>, 0.9266670367735031, 7, 19, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DE1EB8>, 0.9267796243905874, 5, 22, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCE470>, 0.9267833251869412, 5, 20, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99B00>, 0.9269553001606743, 7, 14, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCEC88>, 0.9269623069186332, 6, 15, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DEF748>, 0.9269635743346452, 9, 13, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DC7B70>, 0.9271558421792532, 7, 21, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D81E80>, 0.9275051390981903, 6, 12, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCE198>, 0.9275221258184609, 6, 17, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD28D0>, 0.9275573248349794, 7, 15, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D996A0>, 0.927696417749841, 7, 12, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D81F60>, 0.9277501505128297, 6, 14, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99CF8>, 0.9278796690094322, 9, 12, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D8FF98>, 0.9282606866038348, 9, 15, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DE1C88>, 0.9285401648073969, 8, 15, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CBFFC62F28>, 0.9290251669899094, 5, 13, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DC77F0>, 0.929160399926077, 8, 12, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D81EB8>, 0.9292399435816736, 7, 16, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DD21D0>, 0.9345122952726409, 5, 12, 0.13)]

The best parameters, rank=6, iterations=22, lambda\_=0.13

MSE = 0.8522183391109492

RMSE = 0.9231567251073618

Step 7：

确定rank=6，lambda为0.13，接下来选择iterations

测试代码：

改变部分

metrix\_list = [train\_model\_evaluate(training\_ratings, testing\_ratings,6,param\_iterations, 0.13)

#for param\_rank in range(5,10)

for param\_iterations in range(15,25)

#for param\_rank in range(1, 10)

#for param\_iterations in range(1, 10)

#for param\_lambda in np.arange(0.09,0.14,0.01)

]

结果：

0.13 6 15

MSE = 0.8626580872441026

RMSE = 0.9287938884618603

0.13 6 16

MSE = 0.8722500740177478

RMSE = 0.933943292720574

0.13 6 17

MSE = 0.8609461562784312

RMSE = 0.9278718425938095

0.13 6 18

MSE = 0.8630460062266273

RMSE = 0.9290026944130072

0.13 6 19

MSE = 0.8688769391360727

RMSE = 0.9321356870842746

0.13 6 20

MSE = 0.8649496887405864

RMSE = 0.9300267139929833

0.13 6 21

MSE = 0.8598635156884178

RMSE = 0.9272882592206254

0.13 6 22

MSE = 0.8567812715543456

RMSE = 0.9256248006370322

0.13 6 23

MSE = 0.8631210822114702

RMSE = 0.929043100298081

0.13 6 24

MSE = 0.8649117829876415

RMSE = 0.9300063349180163

<class 'list'>

[(<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81DCF0B8>, 0.9256248006370322, 6, 22, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D8FFD0>, 0.9272882592206254, 6, 21, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81E01BA8>, 0.9278718425938095, 6, 17, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81E01B38>, 0.9287938884618603, 6, 15, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99908>, 0.9290026944130072, 6, 18, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81D99EB8>, 0.929043100298081, 6, 23, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81E12978>, 0.9300063349180163, 6, 24, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81E01668>, 0.9300267139929833, 6, 20, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81E01198>, 0.9321356870842746, 6, 19, 0.13), (<pyspark.mllib.recommendation.MatrixFactorizationModel object at 0x000001CB81E01748>, 0.933943292720574, 6, 16, 0.13)]

The best parameters, rank=6, iterations=22, lambda\_=0.13

综上，这次的参数选择为rank=6, iterations=22, lambda\_=0.13

MSE = 0.8567812715543456

RMSE = 0.9256248006370322