In [1]:
```
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.recommendation import ALS
from pyspark.sql import Row
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
data = spark.read.format("csv").option("header",True).option("inferSchem
a",True).load("RCdata/rating_final.csv")
data.show()
```

```
+------+-------+------+-----------+--------------+
|userID|placeID|rating|food_rating|service_rating|
+------+-------+------+-----------+--------------+
|     1| 135085|     2|          2|             2|
|     2| 135038|     2|          2|             1|
|     3| 132825|     2|          2|             2|
|     4| 135060|     1|          2|             2|
|     5| 135104|     1|          1|             2|
|     6| 132740|     0|          0|             0|
|     7| 132663|     1|          1|             1|
|     8| 132732|     0|          0|             0|
|     9| 132630|     1|          1|             1|
|    10| 132584|     2|          2|             2|
|    11| 132733|     1|          1|             1|
|    12| 132732|     1|          2|             2|
|    13| 132630|     1|          0|             1|
|    14| 135104|     0|          0|             0|
|    15| 132560|     1|          0|             0|
|    16| 132584|     1|          2|             1|
|    17| 132732|     0|          0|             2|
|    18| 132630|     1|          2|             0|
|    19| 132613|     2|          2|             2|
|    20| 132667|     1|          2|             2|
+------+-------+------+-----------+--------------+
only showing top 20 rows
```

In [4]:
```
(training, test) = data.randomSplit([0.8, 0.2])
als = ALS(maxIter=5, regParam=0.01, implicitPrefs=True,userCol="userID",
 itemCol="placeID", ratingCol="rating",
          coldStartStrategy="drop")
model = als.fit(training)
```

In [7]:
```
predictions = model.transform(test)
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating",
                                predictionCol="prediction")
#rmse = evaluator.evaluate(predictions)
#print("Root-mean-square error = " + str(rmse))
userRecs = model.recommendForAllUsers(10)
movieRecs = model.recommendForAllItems(10)
userRecs.show(5,False)
```

```
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-----------+
|userID|recommendations


           |
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-----------+
|463   |[[132862,0.48100093], [135060,0.36393434], [135032,0.25766575],
[132754,0.18146174], [135051,0.17885107], [135072,0.1247983], [132723,
0.12071887], [135057,0.104022026], [132872,0.102057114], [135058,0.0910
4104]]     |
|833   |[[135038,0.09330819], [132825,0.090056315], [135039,0.0592330
1], [135057,0.057779774], [135079,0.049921643], [135045,0.048617616],
 [132921,0.04774813], [135075,0.035188816], [135058,0.031516954], [1350
59,0.03065281]]|
|496   |[[132560,0.0], [132630,0.0], [132660,0.0], [132740,0.0], [13283
0,0.0], [132870,0.0], [135000,0.0], [135030,0.0], [135040,0.0], [13505
0,0.0]]
             |
|148   |[[135051,0.051854], [132862,0.05014171], [135026,0.03888851],
 [132572,0.038523607], [135060,0.035370983], [135030,0.031751085], [135
072,0.03148463], [135038,0.030215243], [135076,0.028989723], [134976,0.
0251938]]    |
|1088  |[[135075,0.39391476], [135076,0.23548545], [135030,0.18134171],
[135057,0.15644243], [135066,0.15440075], [135041,0.13692331], [132754,
0.1352996], [135047,0.13422821], [132723,0.121487096], [135062,0.116676
53]]      |
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-----------+
only showing top 5 rows
```