In [2]:
```
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("ml-20m/ratings.csv")
data.show()
```

```
+------+-------+------+----------+
|userId|movieId|rating| timestamp|
+------+-------+------+----------+
|     1|      2|   3.5|1112486027|
|     1|     29|   3.5|1112484676|
|     1|     32|   3.5|1112484819|
|     1|     47|   3.5|1112484727|
|     1|     50|   3.5|1112484580|
|     1|    112|   3.5|1094785740|
|     1|    151|   4.0|1094785734|
|     1|    223|   4.0|1112485573|
|     1|    253|   4.0|1112484940|
|     1|    260|   4.0|1112484826|
|     1|    293|   4.0|1112484703|
|     1|    296|   4.0|1112484767|
|     1|    318|   4.0|1112484798|
|     1|    337|   3.5|1094785709|
|     1|    367|   3.5|1112485980|
|     1|    541|   4.0|1112484603|
|     1|    589|   3.5|1112485557|
|     1|    593|   3.5|1112484661|
|     1|    653|   3.0|1094785691|
|     1|    919|   3.5|1094785621|
+------+-------+------+----------+
only showing top 20 rows
```

In [3]:
```
(training, test) = data.randomSplit([0.8, 0.2])
als = ALS(maxIter=5, regParam=0.01, implicitPrefs=True,userCol="userId",
 itemCol="movieId", ratingCol="rating",
          coldStartStrategy="drop")
model = als.fit(training)
```

In [4]:
```
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

                                                |
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
-----------------------------------------------+
|148   |[[17,0.7760857], [539,0.76959646], [1307,0.71515596], [62,0.698
13776], [597,0.6962341], [1035,0.6932603], [1393,0.6854005], [357,0.673
3391], [11,0.66974914], [708,0.6490281]]       |
|463   |[[590,0.9840783], [150,0.9808516], [457,0.92620337], [454,0.912
4953], [296,0.91238725], [339,0.911694], [356,0.9104003], [380,0.904721
5], [597,0.9004352], [592,0.8948771]]          |
|471   |[[1721,1.1610202], [2028,1.1132988], [1610,1.1045119], [1961,1.
0844686], [1580,1.0815619], [2396,1.0779964], [1307,1.0614403], [2268,
1.0577246], [1270,1.0558075], [3578,1.0534228]] |
|496   |[[1196,0.90157634], [1197,0.8774836], [1270,0.87396485], [1198,
0.87006056], [1097,0.8494378], [1210,0.8274293], [260,0.825099], [919,
0.8202874], [1136,0.81852674], [1214,0.8003226]]|
|833   |[[592,0.9677862], [590,0.9576383], [380,0.9545622], [150,0.9510
537], [457,0.94851166], [480,0.9285179], [349,0.9152355], [165,0.910314
44], [356,0.9014194], [153,0.88558245]]        |
+------+--------------------------------------------------------------------
--------------------------------------------------------------------------
-----------------------------------------------+
only showing top 5 rows
```