final

November 2, 2019

[1]:

```
import org.apache.spark.sql.{ DataFrame, Row, SQLContext }
import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._
import org.apache.spark.sql._
import org.apache.hadoop.io.LongWritable
import org.apache.hadoop.io.Text
import org.apache.hadoop.conf.Configuration
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat
//import sqlContext.implicits._
import org.apache.spark.sql.{ DataFrame, Row, SQLContext }
import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._
import org.apache.spark.sql._
import org.apache.hadoop.io.LongWritable
import org.apache.hadoop.io.Text
import org.apache.hadoop.conf.Configuration
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat
import scala.io.Source
import scala.util.control.Breaks._
import scala.util.matching.Regex
import org.apache.spark.rdd.RDD
import java.io._
import org.apache.spark.sql.types.IntegerType
import org.apache.spark.sql.types.DoubleType
```

[2]:

```
val sqlContext = new org.apache.spark.sql.SQLContext(sc)
//val sqlContext =new SQLContext(sc)
val df =sqlContext.read.format("com.databricks.spark.csv").option("header",␣
 ↪"false").option("delimiter","#").load("/user/augment/movies")
```

```
sqlContext = org.apache.spark.sql.SQLContext@64f0f324
df = [_c0: string, _c1: string ... 1 more field]



warning: there was one deprecation warning; re-run with -deprecation for details
```

[2]:
```
[_c0: string, _c1: string … 1 more field]
```

[3]:
```
var movies=df.withColumn("movie_id", split(col("_c0"), "\n").getItem(0))
.withColumn("movie",col("_c1"))
.withColumn("genre1", split(col("_c2"), "\\|").getItem(0))
.withColumn("genre2", split(col("_c2"), "\\|").getItem(1))
.withColumn("genre3", split(col("_c2"), "\\|").getItem(2))
.drop("_c0").drop("_c1").drop("_c2")
```

```
movies = [movie_id: string, movie: string ... 3 more fields]
```

[3]:
```
[movie_id: string, movie: string … 3 more fields]
```

[4]:
```
movies.createOrReplaceTempView("movies")
```

[5]:
```
val s = Source.fromFile("/home/augment/Downloads/ratings.dat")
var x=""
var j=1
val pw = new PrintWriter(new File("/home/augment/Downloads/r.txt" ))
var pairs22:Map[Int,String]=Map()
for(y <- s.getLines())
{

        val h = new Regex("[\t#]+[0-9]*")

        var z=((h findAllMatchIn y).mkString("").toString)
        for(i<-z)
    {
        if(i=='#')
        pw.write("\n"+j+",")
        else if(i=='\t')
        {

         pw.write("\n"+j+",")
        }
        else

        pw.write(i)
    }

       j=j+1;

}
pw.close
```

```
s = empty iterator
```

```
j = 3001
pw = java.io.PrintWriter@6cba39b0
pairs22 = Map()
```

```
x: String = ""
```

[5]:
```
Map()
```

[6]:
```
val sqlContext =new SQLContext(sc)
val dff =sqlContext.read.format("com.databricks.spark.csv").option("header",␣
 ↪"false").option("delimiter",",").load("/user/augment/r.txt")
```

```
sqlContext = org.apache.spark.sql.SQLContext@177c12d7
dff = [_c0: string, _c1: string]
```

```
warning: there was one deprecation warning; re-run with -deprecation for details
```

[6]:
```
[_c0: string, _c1: string]
```

[7]:
```
var rating=dff.withColumn("user_id", col("_c0"))
.withColumn("movie_id",col("_c1"))
.drop("_c0")
.drop("_c1")
```

```
rating = [user_id: string, movie_id: string]
```

[7]:
```
[user_id: string, movie_id: string]
```

[8]:
```
rating.createOrReplaceTempView("rating")
```

[9]:
```
movies.show
```

```
+--------+--------------------+---------+----------+--------+
|movie_id|               movie|   genre1|    genre2|  genre3|
+--------+--------------------+---------+----------+--------+
|       1|    Toy Story (1995)|Animation|Children's|  Comedy|
|       2|      Jumanji (1995)|Adventure|Children's| Fantasy|
|       3|Grumpier Old Men ...|   Comedy|   Romance|    null|
|       4|Waiting to Exhale...|   Comedy|     Drama|    null|
|       5|Father of the Bri...|   Comedy|      null|    null|
|       6|         Heat (1995)|   Action|     Crime|Thriller|
|       7|      Sabrina (1995)|   Comedy|   Romance|    null|
|       8| Tom and Huck (1995)|Adventure|Children's|    null|
|       9| Sudden Death (1995)|   Action|      null|    null|
|      10|    GoldenEye (1995)|   Action| Adventure|Thriller|
|      11|American Presiden…|   Comedy|     Drama| Romance|
|      12|Dracula: Dead and…|   Comedy|    Horror|    null|
|      13|        Balto (1995)|Animation|Children's|    null|
|      14|        Nixon (1995)|    Drama|      null|    null|
|      15|Cutthroat Island …|   Action| Adventure| Romance|
|      16|       Casino (1995)|    Drama|  Thriller|    null|
|      17|Sense and Sensibi…|    Drama|   Romance|    null|
|      18|   Four Rooms (1995)| Thriller|      null|    null|
|      19|Ace Ventura: When…|   Comedy|      null|    null|
|      20|  Money Train (1995)|   Action|      null|    null|
+--------+--------------------+---------+----------+--------+
only showing top 20 rows
```

[10]:

```
rating.show
```

```
+-------+--------+
|user_id|movie_id|
+-------+--------+
|      1|       1|
|      1|    1097|
|      1|    1907|
|      1|    2321|
|      1|    2018|
|      1|     260|
|      1|     938|
|      1|    1246|
|      1|    2028|
|      1|     150|
|      1|    3408|
|      1|    2340|
|      1|     919|
|      1|     527|
|      1|     914|
|      1|    3186|
|      1|    1270|
|      1|    2355|
|      1|      48|
|      1|     531|
+-------+--------+
only showing top 20 rows
```

[11]:
```
var x=spark.sql("select user_id,rating.movie_id,movie from rating INNER JOIN␣
↪movies ON rating.movie_id=movies.movie_id")
```

```
x = [user_id: string, movie_id: string ... 1 more field]
```

[11]:
```
[user_id: string, movie_id: string … 1 more field]
```

[12]:
```
x.show
```

```
+-------+--------+--------------------+
|user_id|movie_id|               movie|
+-------+--------+--------------------+
|      1|       1|    Toy Story (1995)|
|      1|    1097|E.T. the Extra-Te…|
|      1|    1907|        Mulan (1998)|
|      1|    2321|Pleasantville (1998)|
|      1|    2018|        Bambi (1942)|
|      1|     260|Star Wars: Episod…|
|      1|     938|         Gigi (1958)|
|      1|    1246|Dead Poets Societ…|
|      1|    2028|Saving Private Ry…|
|      1|     150|    Apollo 13 (1995)|
|      1|    3408|Erin Brockovich (…|
|      1|    2340|Meet Joe Black (1…|
|      1|     919|Wizard of Oz, The…|
|      1|     527|Schindler's List …|
|      1|     914| My Fair Lady (1964)|
|      1|    3186|Girl, Interrupted…|
|      1|    1270|Back to the Futur…|
|      1|    2355|Bug's Life, A (1998)|
|      1|      48|   Pocahontas (1995)|
|      1|     531|Secret Garden, Th…|
+-------+--------+--------------------+
only showing top 20 rows
```

[13]:
```
val y=spark.sql("select r1.user_id as user_id1, r2.user_id as user_id2, r1.
↪movie_id,movies.movie from rating AS r1, rating AS r2 INNER JOIN movies ON r1.
↪movie_id=movies.movie_id where r1.movie_id= r2.movie_id")
```

```
y = [user_id1: string, user_id2: string ... 2 more fields]
```

[13]:
```
[user_id1: string, user_id2: string … 2 more fields]
```

[14]:

```
spark.sql("select DISTINCT(movie_id),count(user_id) AS total from rating GROUP␣
 ↪BY movie_id ORDER BY movie_id ASC").show
```

```
+--------+-----+
|movie_id|total|
+--------+-----+
|       1|  993|
|      10|  419|
|     100|   56|
|    1000|    3|
|    1002|    3|
|    1003|   62|
|    1004|   44|
|    1005|   73|
|    1006|   35|
|    1007|  111|
|    1008|   44|
|    1009|  135|
|     101|  129|
|    1010|  123|
|    1011|   60|
|    1012|  146|
|    1013|  125|
|    1014|   68|
|    1015|  124|
|    1016|   83|
+--------+-----+
only showing top 20 rows
```

[15]:

```
y.show
```

```
+--------+--------+--------+----------------+
|user_id1|user_id2|movie_id|           movie|
+--------+--------+--------+----------------+
|       1|    2999|       1|Toy Story (1995)|
|       1|    2996|       1|Toy Story (1995)|
|       1|    2995|       1|Toy Story (1995)|
|       1|    2994|       1|Toy Story (1995)|
|       1|    2989|       1|Toy Story (1995)|
|       1|    2987|       1|Toy Story (1995)|
|       1|    2985|       1|Toy Story (1995)|
|       1|    2980|       1|Toy Story (1995)|
|       1|    2979|       1|Toy Story (1995)|
|       1|    2972|       1|Toy Story (1995)|
|       1|    2970|       1|Toy Story (1995)|
|       1|    2968|       1|Toy Story (1995)|
|       1|    2967|       1|Toy Story (1995)|
|       1|    2966|       1|Toy Story (1995)|
|       1|    2955|       1|Toy Story (1995)|
|       1|    2951|       1|Toy Story (1995)|
|       1|    2950|       1|Toy Story (1995)|
|       1|    2941|       1|Toy Story (1995)|
|       1|    2939|       1|Toy Story (1995)|
|       1|    2938|       1|Toy Story (1995)|
+--------+--------+--------+----------------+
only showing top 20 rows
```

[16]:

```
y.createOrReplaceTempView("y")
```

[ ]:

```
val pw = new PrintWriter(new File("/home/augment/Downloads/pairs.txt" ))
for(i<-1 to 3000)
{


    for(j<- 1 to 3000)
    {
        if(i!=j)
        {
            if(j>i)
            {

                val b=spark.sql(s"select movie from y where user_id1=${i} and␣
 ↪user_id2=${j}")
                val d=spark.sql(s"select count(movie_id)from y where␣
 ↪user_id1=${i} and user_id2=${j}")
                pw.write(i+"\t"+j+"\t")
                //print(i+"\t"+j+"\t")
                d.collect().foreach {
                    row =>pw.write(row.mkString(""))
                    pw.write("\t")
                }
                b.collect().foreach {
                    row =>pw.write(row.mkString(""))
                    pw.write(",")
                }
                pw.write("\n")

        }
        }
    }
}
pw.close
```

[18]:

```
val sqlContext =new SQLContext(sc)
val dff =sqlContext.read.format("com.databricks.spark.csv").option("header",␣
↪"false").option("delimiter","\t").load("/user/augment/pairs.txt")
```

```
sqlContext = org.apache.spark.sql.SQLContext@2c835e42
dff = [_c0: string, _c1: string ... 2 more fields]
```

```
warning: there was one deprecation warning; re-run with -deprecation for details
```

```
[18]: [_c0: string, _c1: string … 2 more fields]
```

[19]:

```
dff.show
```

```
+---+---+---+--------------------+
|_c0|_c1|_c2|                 _c3|
+---+---+---+--------------------+
|  1|  2|  7|Pleasantville (19…|
|  1|  3|  6|Star Wars: Episod…|
|  1|  4|  4|E.T. the Extra-Te…|
|  1|  5| 10|Saving Private Ry…|
|  2|  3| 12|Dances with Wolve…|
|  2|  4|  8|Hustler, The (196…|
|  2|  5| 20|Like Water for Ch…|
|  3|  4|  5|Star Wars: Episod…|
|  3|  5|  6|Being John Malkov…|
|  4|  5|  3|Run Lola Run (Lol…|
+---+---+---+--------------------+
```

[20]:

```
var df=dff.withColumn("user_id1", col("_c0"))
.withColumn("user_id2",col("_c1"))
.withColumn("count",col("_c2"))
.withColumn("list",col("_c3"))
.drop("_c0")
.drop("_c1")
.drop("_c2")
.drop("_c3")
```

```
df = [user_id1: string, user_id2: string ... 2 more fields]
```

```
[20]: [user_id1: string, user_id2: string … 2 more fields]
```

[21]:

```
val df2 = df.selectExpr("user_id1",
                        "user_id2",
    "cast(count as int) count",
                        "list")
```

```
df2 = [user_id1: string, user_id2: string ... 2 more fields]
```

[21]:

```
[user_id1: string, user_id2: string … 2 more fields]
```

[23]:

```
df2.createOrReplaceTempView("answer")
```

[24]:

```
//spark.sql("select * from answer where count > 5 ORDER BY count desc").show
spark.sql("select * from answer where count > 50 ORDER BY count desc").show
```

```
+--------+--------+-----+--------------------+
|user_id1|user_id2|count|                list|
+--------+--------+-----+--------------------+
|       2|       5|   20|Like Water for Ch…|
|       2|       3|   12|Dances with Wolve…|
|       1|       5|   10|Saving Private Ry…|
|       2|       4|    8|Hustler, The (196…|
|       1|       2|    7|Pleasantville (19…|
|       1|       3|    6|Star Wars: Episod…|
|       3|       5|    6|Being John Malkov…|
+--------+--------+-----+--------------------+
```