README

实验过程

Spark 安装及环境配置

1. 解压 scala2.12.11, spark3.0.1

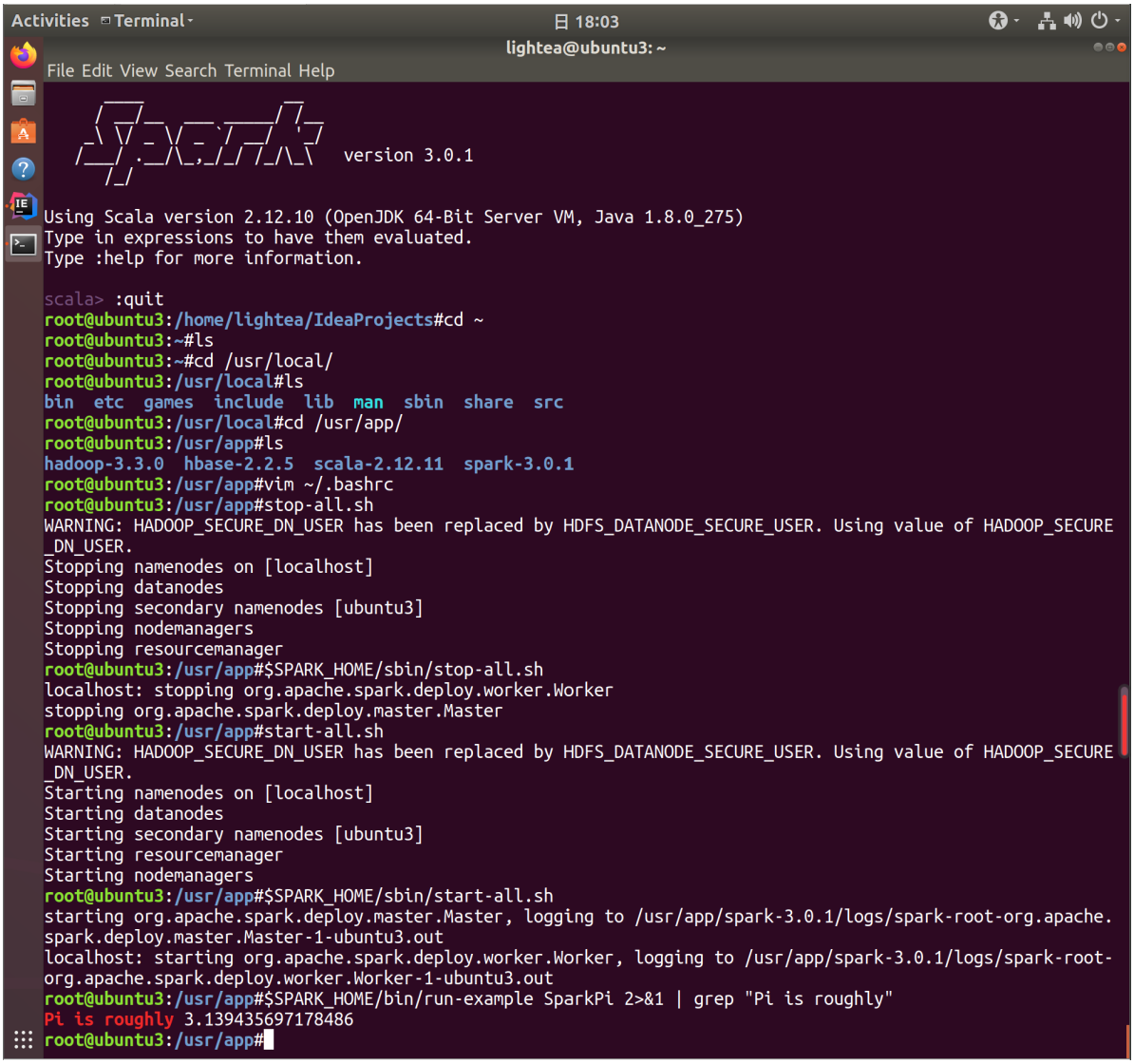
```
$ tar -zxvf scala-2.12.11.tgz -C /usr/app
$ tar -zxvf spark-3.0.1-bin-hadoop2.7.tgz -C /usr/app
```

2. 配置环境变量

```
# ~/.bashrc 中添加scala, spark的环境变量
## JAVA
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
## HADOOP
export HADOOP_HOME=/usr/app/hadoop-3.3.0
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
## SCALA
export SCALA_HOME=/usr/app/scala-2.12.11
export PATH=$PATH:$SCALA_HOME/bin
## SPARK
export SPARK_HOME=/usr/app/spark-3.0.1
export PATH=$PATH:$SPARK_HOME/bin
```

3. 测试安装是否成功

```
$ cd $SPARK_HOME
$ start-all.sh  # 启动hadoop,hdfs
$ sbin/start-all.sh # 启动spark
$ bin/run-example SparkPi 2>&1 | grep "Pi is roughly"
```



test1 - scala

> 统计双十一最热门的商品和最受年轻人(age<30)关注的商家（“添加购物车+购买+添加收藏夹”前100名）

1. IDEA中新建sbt项目管理scala代码
   build.sbt 文件内容

```
name := "ShoppingScala"

version := "0.1"

scalaVersion := "2.12.11"

idePackagePrefix := Some("zky.nju.edu.cn")

libraryDependencies += "org.apache.spark" %% "spark-core" % "3.0.1"
```

2. scala 代码

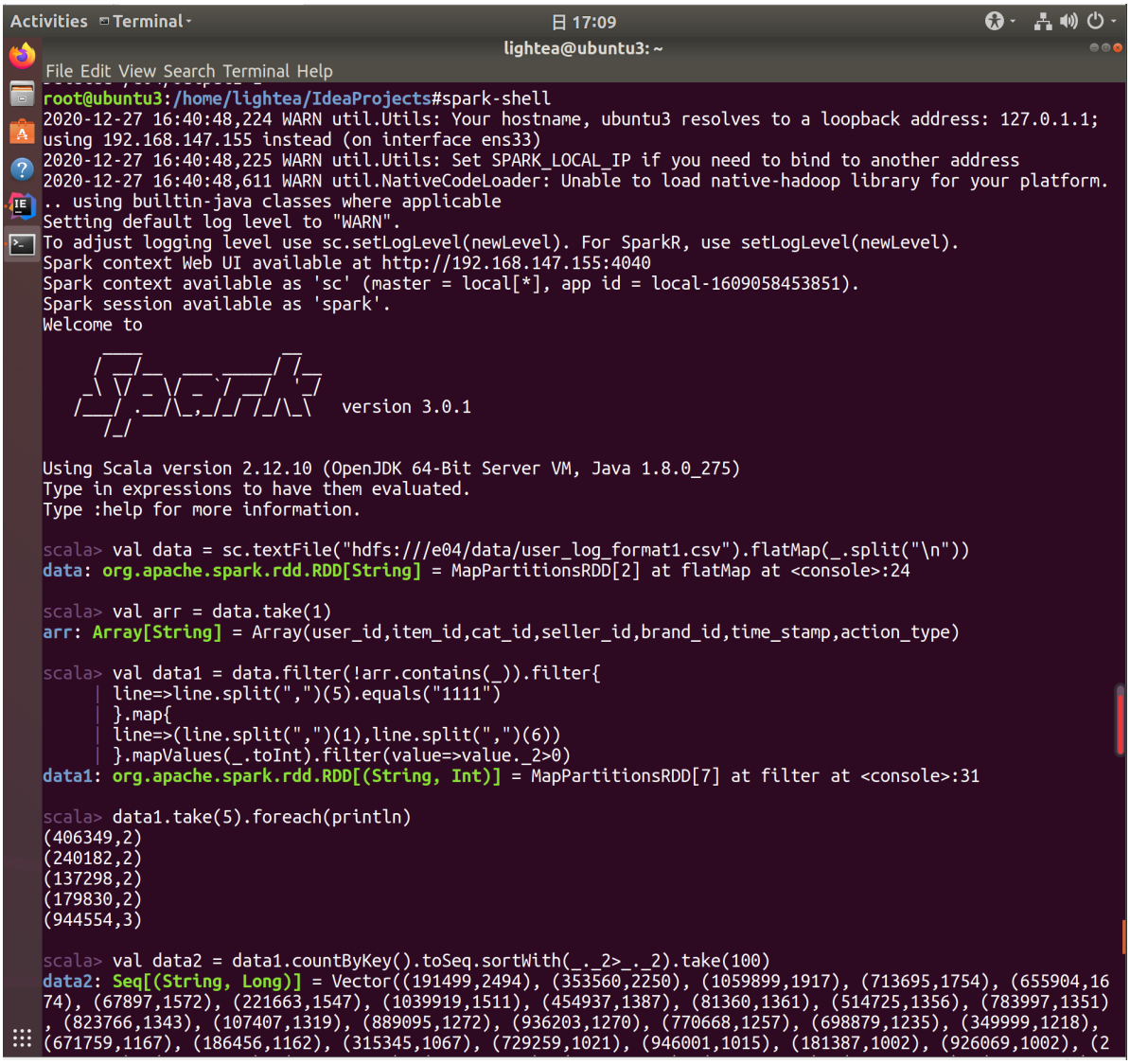
```
// 1) 统计双十一最热门的商品(不使用dataframe)

// 读取 user_log_format1.csv 文本文件
val data =
sc.textFile("hdfs:///e04/data/user_log_format1.csv").flatMap(_.split("\n"))
```

```
// 为去除表头(第一行)
val arr = data.take(1)
// 过滤选出日期为1111的数据行，再选择商品ID和动作编号作为(key,value)对，最后过滤掉动作编号为0的情况
val data1 = data.filter(!arr.contains(_)).filter(line=>line.split(",")(5).equals("1111")).map{
      line=>(line.split(",")(1),line.split(",")(6))
      }.mapValues(_.toInt).filter(value=>value._2>0)
// 计数每个key的rdd数量，再按value从大到小排序，取前100个
val data2 = data1.countByKey().toSeq.sortWith(_._2>_._2).take(100)
// 再次转化为rdd后保存到本地文本文件
sc.parallelize(data2).saveAsTextFile("hdfs:///e04/output1-1")
```

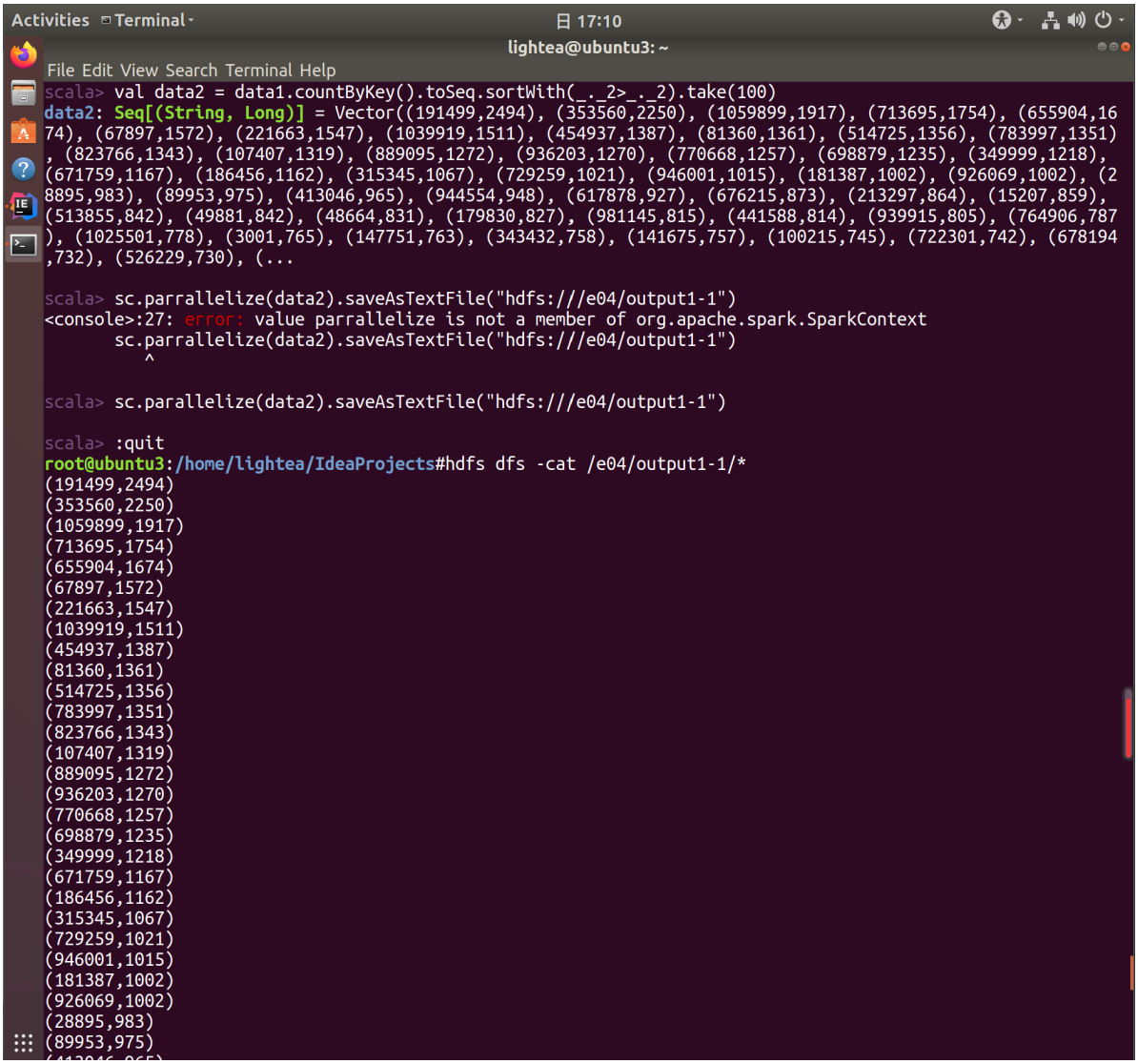
```
// 2) 统计双十一最受年轻人(age<30)关注的商家(使用dataframe)
/*
1. info 筛出age_range<4的user_id
2. time_stamp="1111", action_type!=0的数据条，
3. 用join筛选出1.中的user_id
4. countByKey 或者 groupBy("merchant_id").count()
5. 排序，取前100个
*/
val dflog =
spark.read.format("csv").option("header","true").load("hdfs:///e04/data/user_log_format1.csv")
val dfinfo =
spark.read.format("csv").option("header","true").load("hdfs:///e04/data/user_info_format1.csv")
val dfia = dfinfo.filter("age_range<4 and age_range>0").select("user_id","age_range")
val dfla = dflog.filter("time_stamp=1111 and action_type!=0").select("user_id","seller_id","action_type")
val dfjoin = dfia.join(dfla,"user_id")
val dfss = dfjoin.groupBy("seller_id").count()
val rddss = dfss.orderBy(dfss("count").desc).rdd.map(x=>(x(0),x(1))).take(100)
sc.parallelize(rddss).saveAsTextFile("hdfs:///e04/output1-2")
```

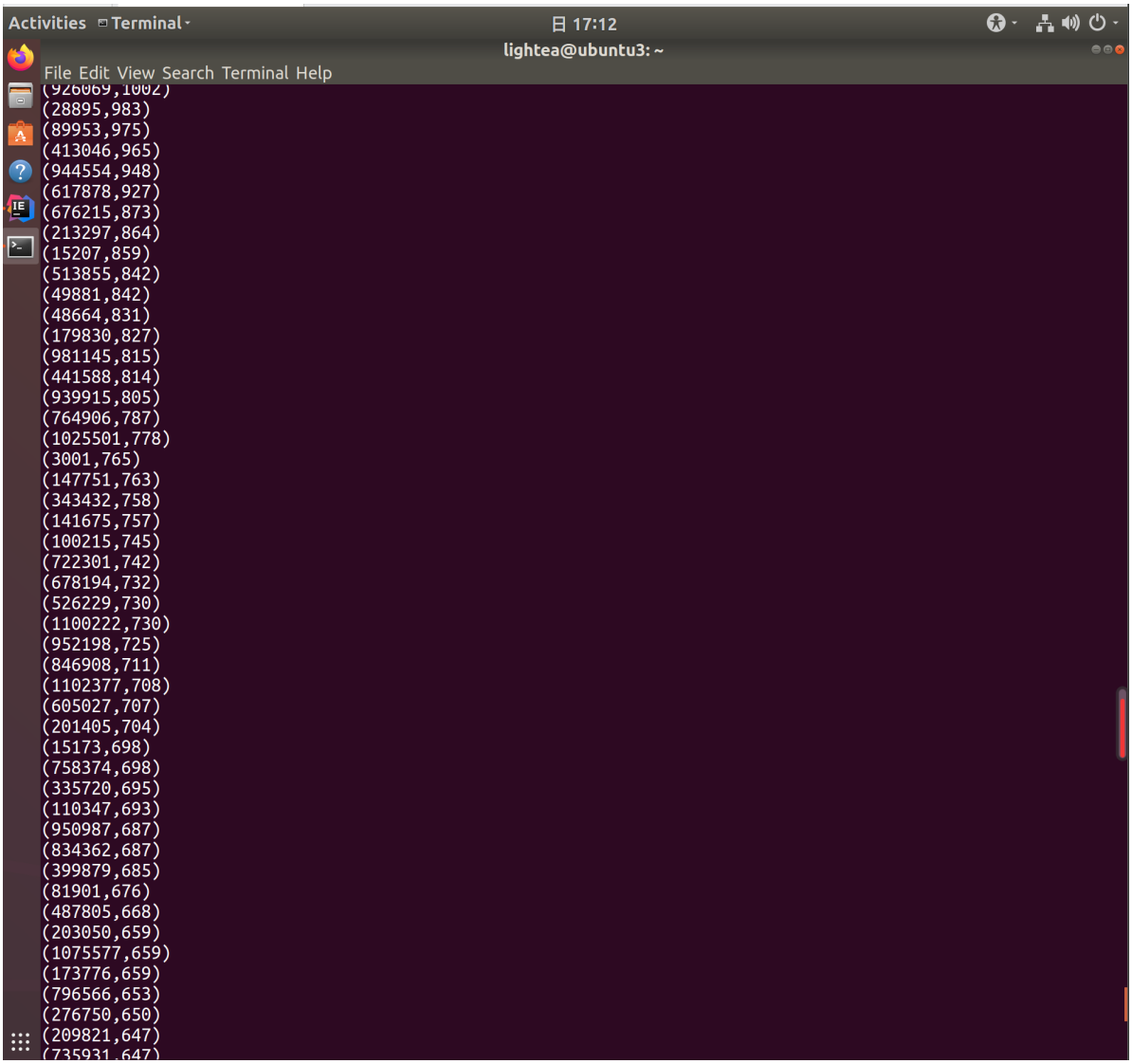
3. 运行结果

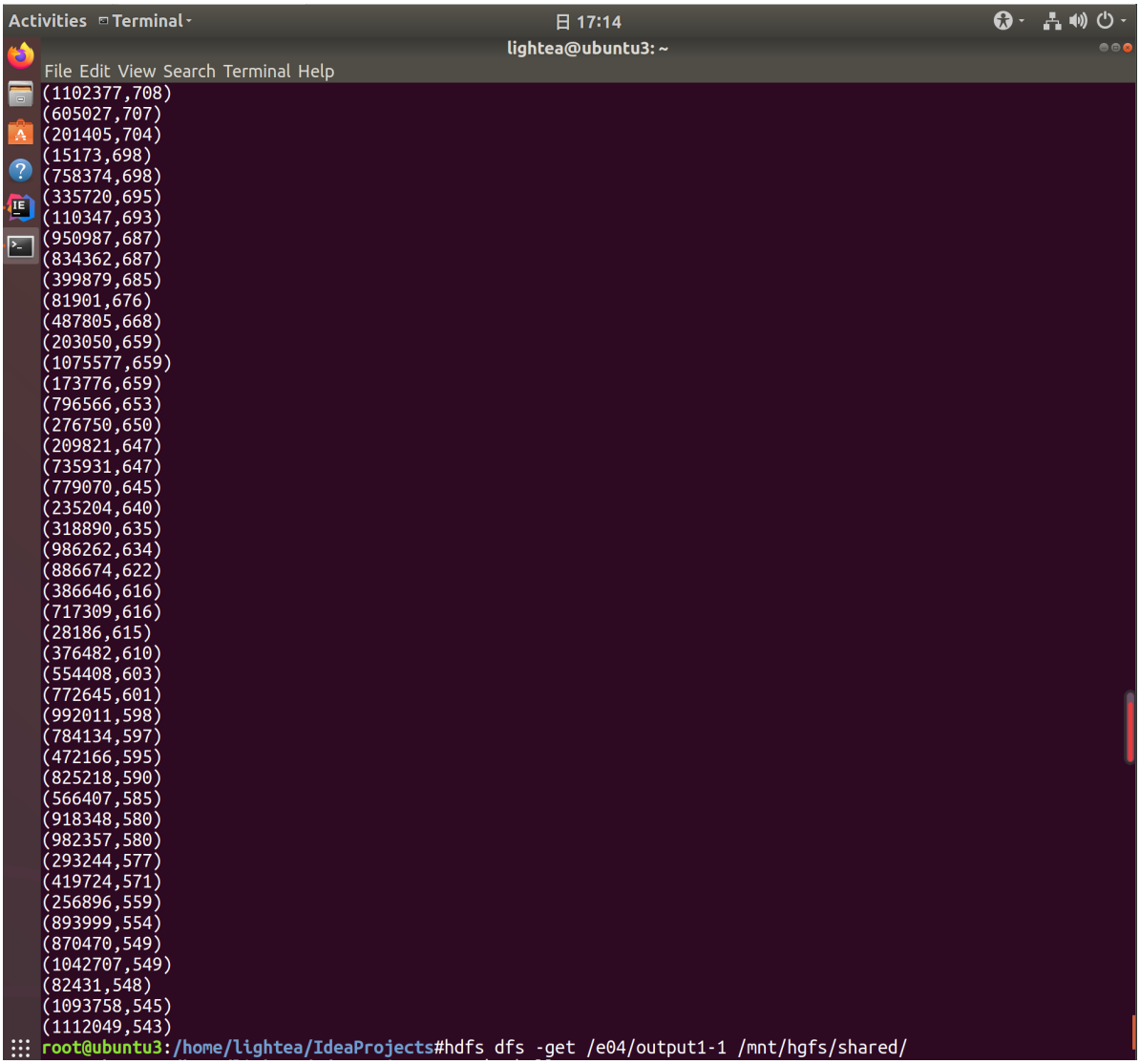
1-1


Activities Terminal 17:09
lightea@ubuntu3: ~
File Edit View Search Terminal Help
root@ubuntu3:/home/lightea/IdeaProjects#spark-shell
2020-12-27 16:40:48,224 WARN util.Utils: Your hostname, ubuntu3 resolves to a loopback address: 127.0.1.1; using 192.168.147.155 instead (on interface ens33)
2020-12-27 16:40:48,225 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
2020-12-27 16:40:48,611 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://192.168.147.155:4040
Spark context available as 'sc' (master = local[*], app id = local-1609058453851).
Spark session available as 'spark'.
Welcome to
version 3.0.1
Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_275)
Type in expressions to have them evaluated.
Type :help for more information.
scala> val data = sc.textFile("hdfs:///e04/data/user_log_format1.csv").flatMap(_.split("\n"))
data: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at <console>:24
scala> val arr = data.take(1)
arr: Array[String] = Array(user_id,item_id,cat_id,seller_id,brand_id,time_stamp,action_type)
scala> val data1 = data.filter(!arr.contains(_)).filter{
| line=>line.split(",")(5).equals("1111")
| }.map{
| line=>(line.split(",")(1),line.split(",")(6))
| }.mapValues(_.toInt).filter(value=>value._2>0)
data1: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[7] at filter at <console>:31
scala> data1.take(5).foreach(println)
(406349,2)
(240182,2)
(137298,2)
(179830,2)
(944554,3)
scala> val data2 = data1.countByKey().toSeq.sortWith(_._2>_._2).take(100)
data2: Seq[(String, Long)] = Vector((191499,2494), (353560,2250), (1059899,1917), (713695,1754), (655904,1674), (67897,1572), (221663,1547), (1039919,1511), (454937,1387), (81360,1361), (514725,1356), (783997,1351), (823766,1343), (107407,1319), (889095,1272), (936203,1270), (770668,1257), (698879,1235), (349999,1218), (671759,1167), (186456,1162), (315345,1067), (729259,1021), (946001,1015), (181387,1002), (926069,1002), (2

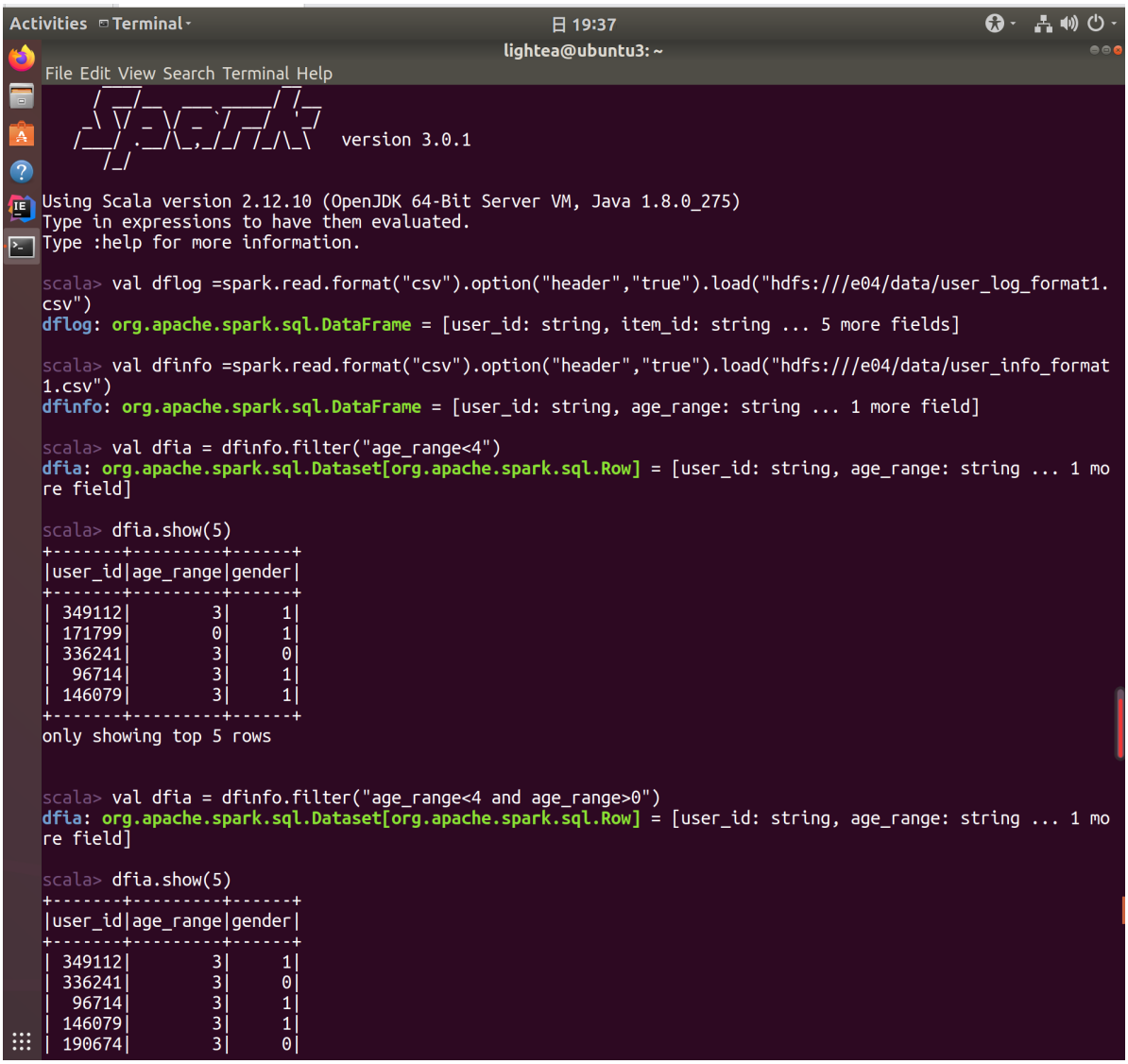
1-1 result

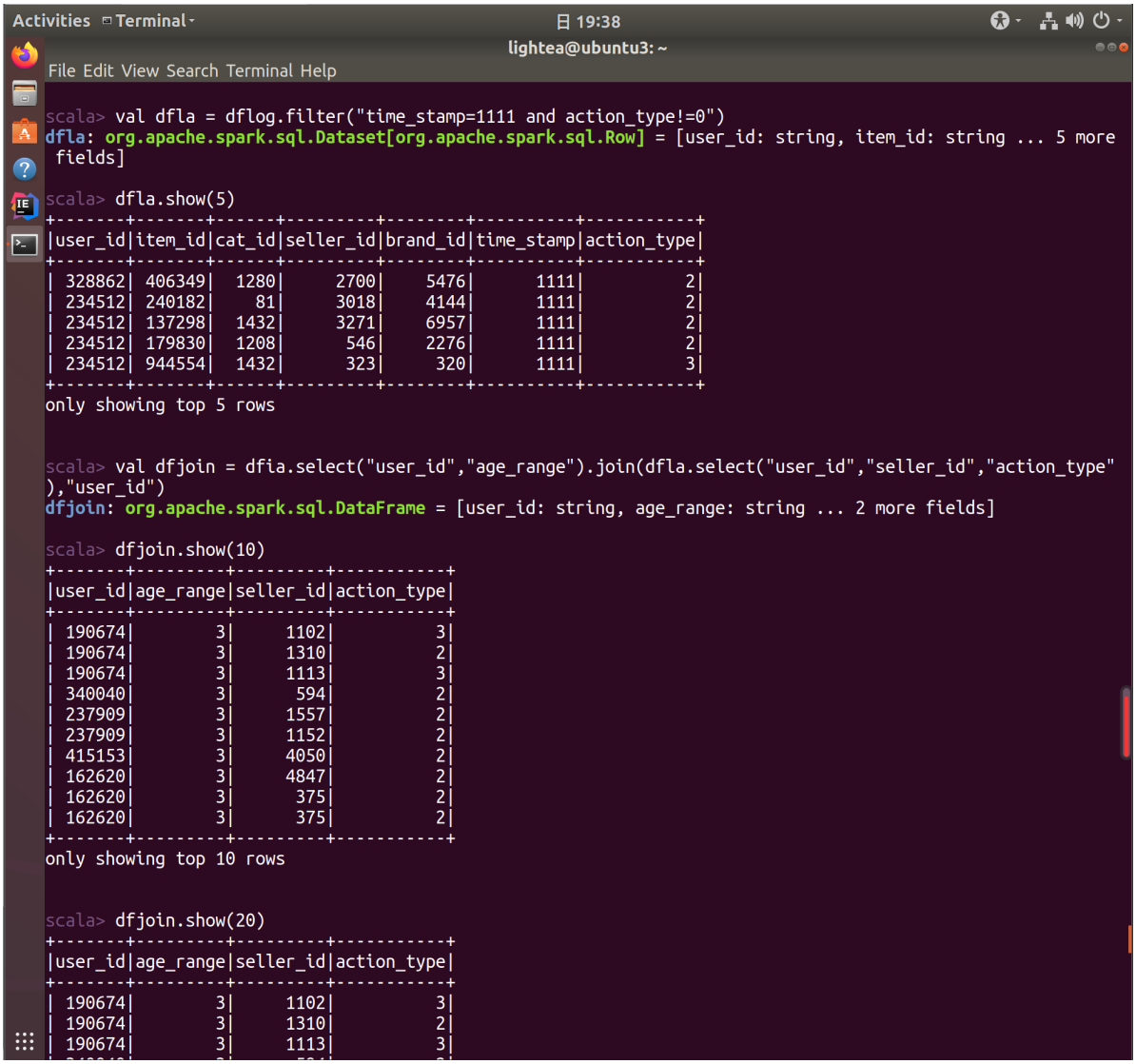

Activities
Terminal
17:10
lightea@ubuntu3: ~
File Edit View Search Terminal Help
scala> val data2 = data1.countByKey().toSeq.sortWith(_._2>_._2).take(100)
data2: Seq[(String, Long)] = Vector((191499,2494), (353560,2250), (1059899,1917), (713695,1754), (655904,1674), (67897,1572), (221663,1547), (1039919,1511), (454937,1387), (81360,1361), (514725,1356), (783997,1351), (823766,1343), (107407,1319), (889095,1272), (936203,1270), (770668,1257), (698879,1235), (349999,1218), (671759,1167), (186456,1162), (315345,1067), (729259,1021), (946001,1015), (181387,1002), (926069,1002), (28895,983), (89953,975), (413046,965), (944554,948), (617878,927), (676215,873), (213297,864), (15207,859), (513855,842), (49881,842), (48664,831), (179830,827), (981145,815), (441588,814), (939915,805), (764906,787), (1025501,778), (3001,765), (147751,763), (343432,758), (141675,757), (100215,745), (722301,742), (678194,732), (526229,730), (...
scala> sc.parrallelize(data2).saveAsTextFile("hdfs:///e04/output1-1")
<console>:27: error: value parrallelize is not a member of org.apache.spark.SparkContext
sc.parrallelize(data2).saveAsTextFile("hdfs:///e04/output1-1")
^
scala> sc.parallelize(data2).saveAsTextFile("hdfs:///e04/output1-1")
scala> :quit
root@ubuntu3:/home/lightea/IdeaProjects#hdfs dfs -cat /e04/output1-1/*
(191499,2494)
(353560,2250)
(1059899,1917)
(713695,1754)
(655904,1674)
(67897,1572)
(221663,1547)
(1039919,1511)
(454937,1387)
(81360,1361)
(514725,1356)
(783997,1351)
(823766,1343)
(107407,1319)
(889095,1272)
(936203,1270)
(770668,1257)
(698879,1235)
(349999,1218)
(671759,1167)
(186456,1162)
(315345,1067)
(729259,1021)
(946001,1015)
(181387,1002)
(926069,1002)
(28895,983)
(89953,975)

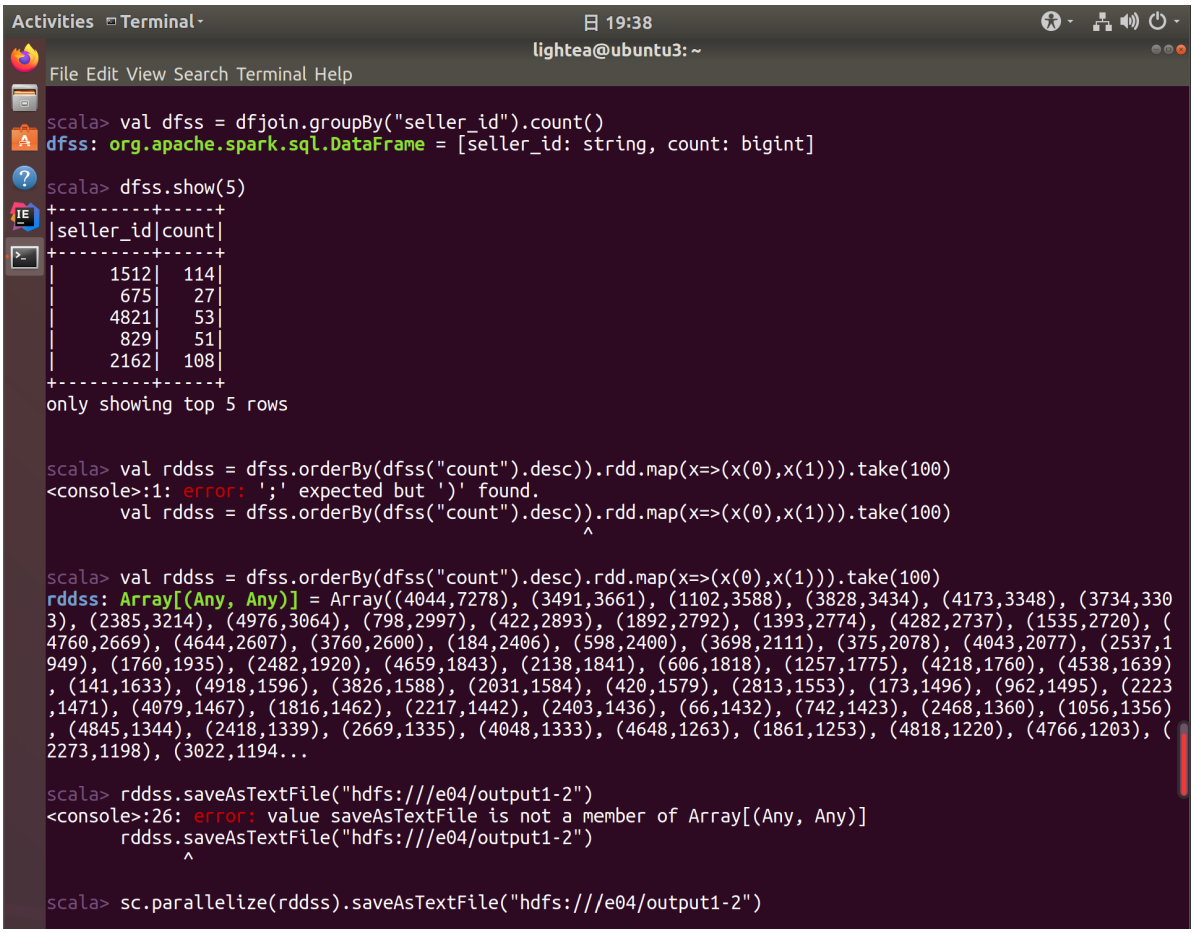

Activities
Terminal
17:12
lightea@ubuntu3: ~
File Edit View Search Terminal Help
(926069,1002)
(28895,983)
(89953,975)
(413046,965)
(944554,948)
(617878,927)
(676215,873)
(213297,864)
(15207,859)
(513855,842)
(49881,842)
(48664,831)
(179830,827)
(981145,815)
(441588,814)
(939915,805)
(764906,787)
(1025501,778)
(3001,765)
(147751,763)
(343432,758)
(141675,757)
(100215,745)
(722301,742)
(678194,732)
(526229,730)
(1100222,730)
(952198,725)
(846908,711)
(1102377,708)
(605027,707)
(201405,704)
(15173,698)
(758374,698)
(335720,695)
(110347,693)
(950987,687)
(834362,687)
(399879,685)
(81901,676)
(487805,668)
(203050,659)
(1075577,659)
(173776,659)
(796566,653)
(276750,650)
(209821,647)
(735931,647)


Activities
Terminal
17:14
lightea@ubuntu3: ~
File Edit View Search Terminal Help
(1102377,708)
(605027,707)
(201405,704)
(15173,698)
(758374,698)
(335720,695)
(110347,693)
(950987,687)
(834362,687)
(399879,685)
(81901,676)
(487805,668)
(203050,659)
(1075577,659)
(173776,659)
(796566,653)
(276750,650)
(209821,647)
(735931,647)
(779070,645)
(235204,640)
(318890,635)
(986262,634)
(886674,622)
(386646,616)
(717309,616)
(28186,615)
(376482,610)
(554408,603)
(772645,601)
(992011,598)
(784134,597)
(472166,595)
(825218,590)
(566407,585)
(918348,580)
(982357,580)
(293244,577)
(419724,571)
(256896,559)
(893999,554)
(870470,549)
(1042707,549)
(82431,548)
(1093758,545)
(1112049,543)
root@ubuntu3:/home/lightea/IdeaProjects#hdfs dfs -get /e04/output1-1 /mnt/hgfs/shared/

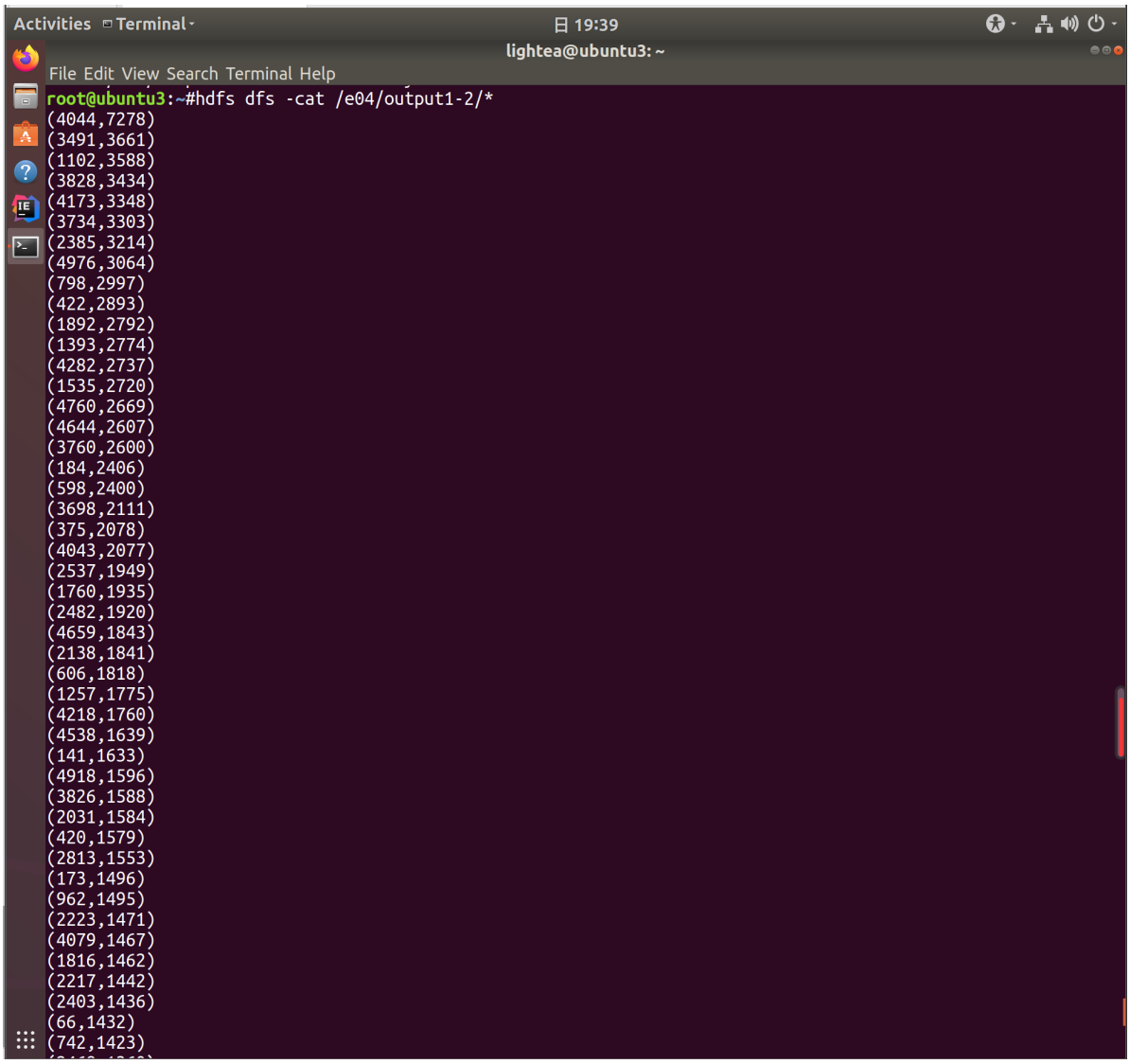
1-2

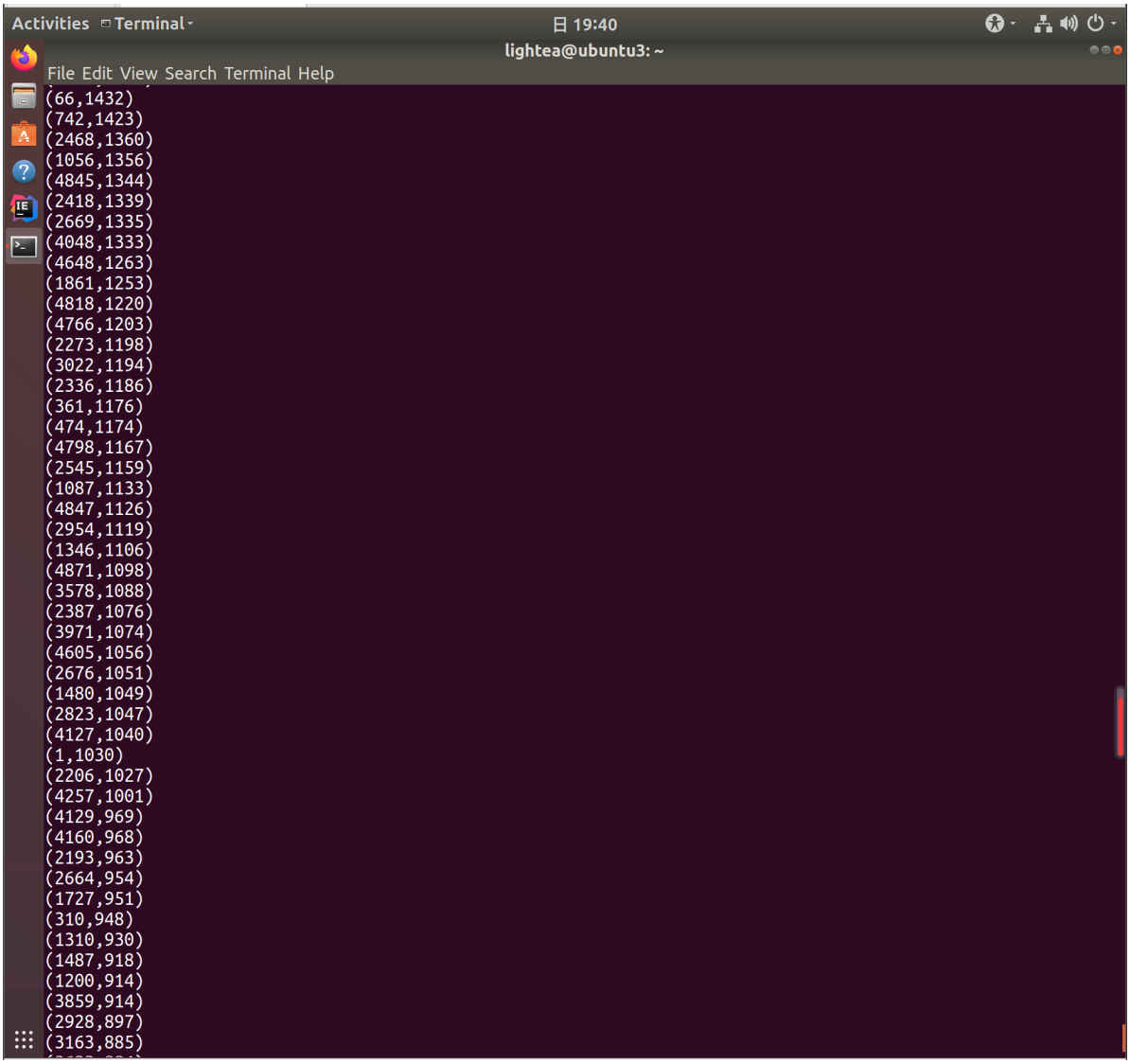

Activities
Terminal
19:37
lightea@ubuntu3: ~
File Edit View Search Terminal Help
version 3.0.1
Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_275)
Type in expressions to have them evaluated.
Type :help for more information.
scala> val dflog =spark.read.format("csv").option("header","true").load("hdfs:///e04/data/user_log_format1.csv")
dflog: org.apache.spark.sql.DataFrame = [user_id: string, item_id: string ... 5 more fields]
scala> val dfinfo =spark.read.format("csv").option("header","true").load("hdfs:///e04/data/user_info_format1.csv")
dfinfo: org.apache.spark.sql.DataFrame = [user_id: string, age_range: string ... 1 more field]
scala> val dfia = dfinfo.filter("age_range<4")
dfia: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [user_id: string, age_range: string ... 1 more field]
scala> dfia.show(5)
+-------+---------+------+
|user_id|age_range|gender|
+-------+---------+------+
349112	3	1
171799	0	1
336241	3	0
96714	3	1
146079	3	1
+-------+---------+------+
only showing top 5 rows
scala> val dfia = dfinfo.filter("age_range<4 and age_range>0")
dfia: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [user_id: string, age_range: string ... 1 more field]
scala> dfia.show(5)
+-------+---------+------+
|user_id|age_range|gender|
+-------+---------+------+
349112	3	1
336241	3	0
96714	3	1
146079	3	1
190674	3	0

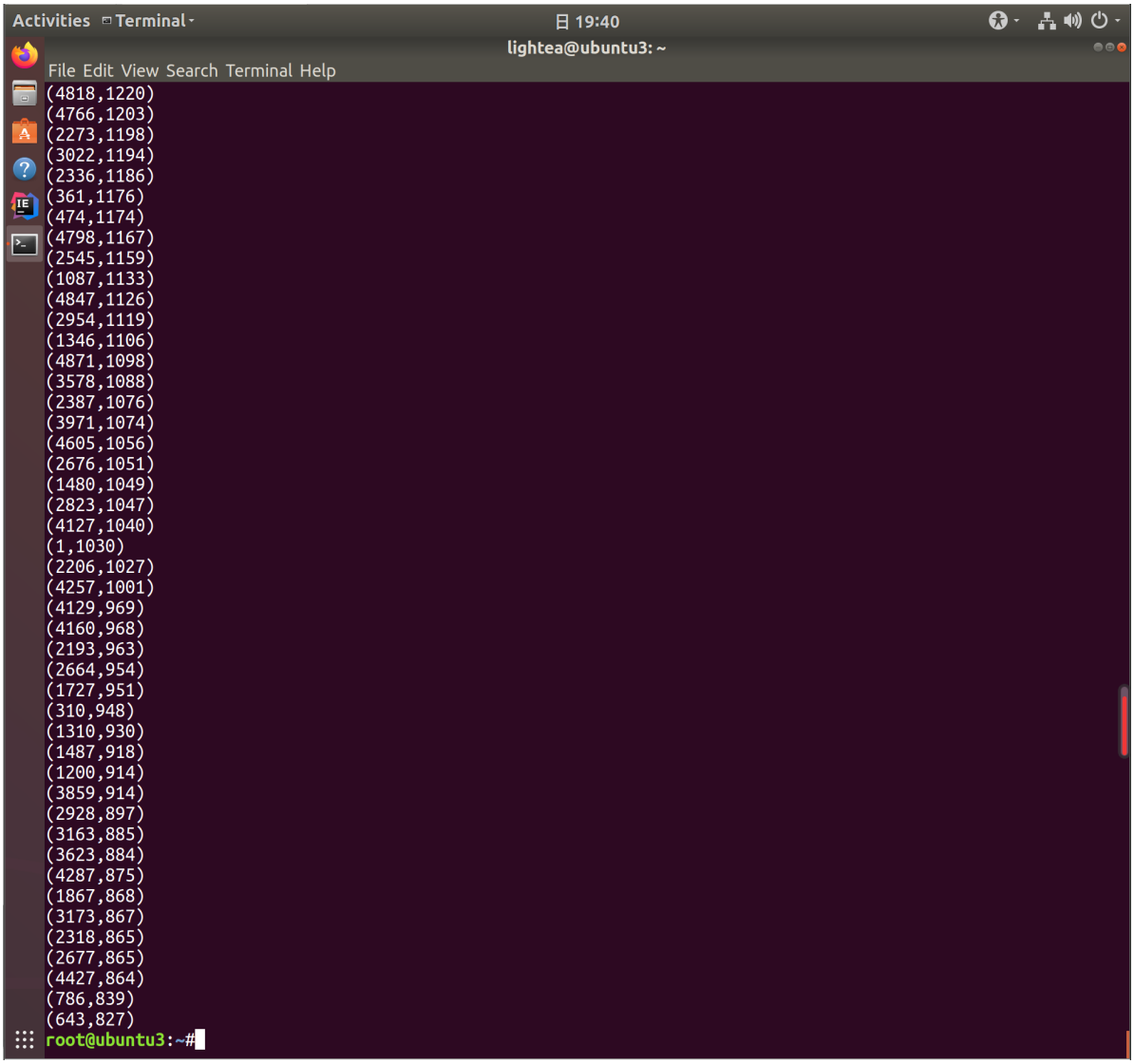

Activities Terminal 19:38
lightea@ubuntu3: ~
File Edit View Search Terminal Help
scala> val dfla = dflog.filter("time_stamp=1111 and action_type!=0")
dfla: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [user_id: string, item_id: string ... 5 more fields]
scala> dfla.show(5)
+-------+-------+------+---------+--------+----------+-----------+
|user_id|item_id|cat_id|seller_id|brand_id|time_stamp|action_type|
+-------+-------+------+---------+--------+----------+-----------+
328862	406349	1280	2700	5476	1111	2
234512	240182	81	3018	4144	1111	2
234512	137298	1432	3271	6957	1111	2
234512	179830	1208	546	2276	1111	2
234512	944554	1432	323	320	1111	3
+-------+-------+------+---------+--------+----------+-----------+
only showing top 5 rows
scala> val dfjoin = dfia.select("user_id","age_range").join(dfla.select("user_id","seller_id","action_type"),"user_id")
dfjoin: org.apache.spark.sql.DataFrame = [user_id: string, age_range: string ... 2 more fields]
scala> dfjoin.show(10)
+-------+---------+---------+-----------+
|user_id|age_range|seller_id|action_type|
+-------+---------+---------+-----------+
190674	3	1102	3
190674	3	1310	2
190674	3	1113	3
340040	3	594	2
237909	3	1557	2
237909	3	1152	2
415153	3	4050	2
162620	3	4847	2
162620	3	375	2
162620	3	375	2
+-------+---------+---------+-----------+
only showing top 10 rows
scala> dfjoin.show(20)
+-------+---------+---------+-----------+
|user_id|age_range|seller_id|action_type|
+-------+---------+---------+-----------+
190674	3	1102	3
190674	3	1310	2
190674	3	1113	3


Activities Terminal 19:38
lightea@ubuntu3: ~
File Edit View Search Terminal Help
scala> val dfss = dfjoin.groupBy("seller_id").count()
dfss: org.apache.spark.sql.DataFrame = [seller_id: string, count: bigint]
scala> dfss.show(5)
+---------+-----+
|seller_id|count|
+---------+-----+
1512	114
675	27
4821	53
829	51
2162	108
+---------+-----+
only showing top 5 rows
scala> val rddss = dfss.orderBy(dfss("count").desc)).rdd.map(x=>(x(0),x(1))).take(100)
<console>:1: error: ';' expected but ')' found.
val rddss = dfss.orderBy(dfss("count").desc)).rdd.map(x=>(x(0),x(1))).take(100)
^
scala> val rddss = dfss.orderBy(dfss("count").desc).rdd.map(x=>(x(0),x(1))).take(100)
rddss: Array[(Any, Any)] = Array((4044,7278), (3491,3661), (1102,3588), (3828,3434), (4173,3348), (3734,3303), (2385,3214), (4976,3064), (798,2997), (422,2893), (1892,2792), (1393,2774), (4282,2737), (1535,2720), (4760,2669), (4644,2607), (3760,2600), (184,2406), (598,2400), (3698,2111), (375,2078), (4043,2077), (2537,1949), (1760,1935), (2482,1920), (4659,1843), (2138,1841), (606,1818), (1257,1775), (4218,1760), (4538,1639), (141,1633), (4918,1596), (3826,1588), (2031,1584), (420,1579), (2813,1553), (173,1496), (962,1495), (2223,1471), (4079,1467), (1816,1462), (2217,1442), (2403,1436), (66,1432), (742,1423), (2468,1360), (1056,1356), (4845,1344), (2418,1339), (2669,1335), (4048,1333), (4648,1263), (1861,1253), (4818,1220), (4766,1203), (2273,1198), (3022,1194...
scala> rddss.saveAsTextFile("hdfs:///e04/output1-2")
<console>:26: error: value saveAsTextFile is not a member of Array[(Any, Any)]
rddss.saveAsTextFile("hdfs:///e04/output1-2")
^
scala> sc.parallelize(rddss).saveAsTextFile("hdfs:///e04/output1-2")

1-2 result


Activities
Terminal
19:39
lightea@ubuntu3: ~
File Edit View Search Terminal Help
root@ubuntu3:~#hdfs dfs -cat /e04/output1-2/*
(4044,7278)
(3491,3661)
(1102,3588)
(3828,3434)
(4173,3348)
(3734,3303)
(2385,3214)
(4976,3064)
(798,2997)
(422,2893)
(1892,2792)
(1393,2774)
(4282,2737)
(1535,2720)
(4760,2669)
(4644,2607)
(3760,2600)
(184,2406)
(598,2400)
(3698,2111)
(375,2078)
(4043,2077)
(2537,1949)
(1760,1935)
(2482,1920)
(4659,1843)
(2138,1841)
(606,1818)
(1257,1775)
(4218,1760)
(4538,1639)
(141,1633)
(4918,1596)
(3826,1588)
(2031,1584)
(420,1579)
(2813,1553)
(173,1496)
(962,1495)
(2223,1471)
(4079,1467)
(1816,1462)
(2217,1442)
(2403,1436)
(66,1432)
(742,1423)

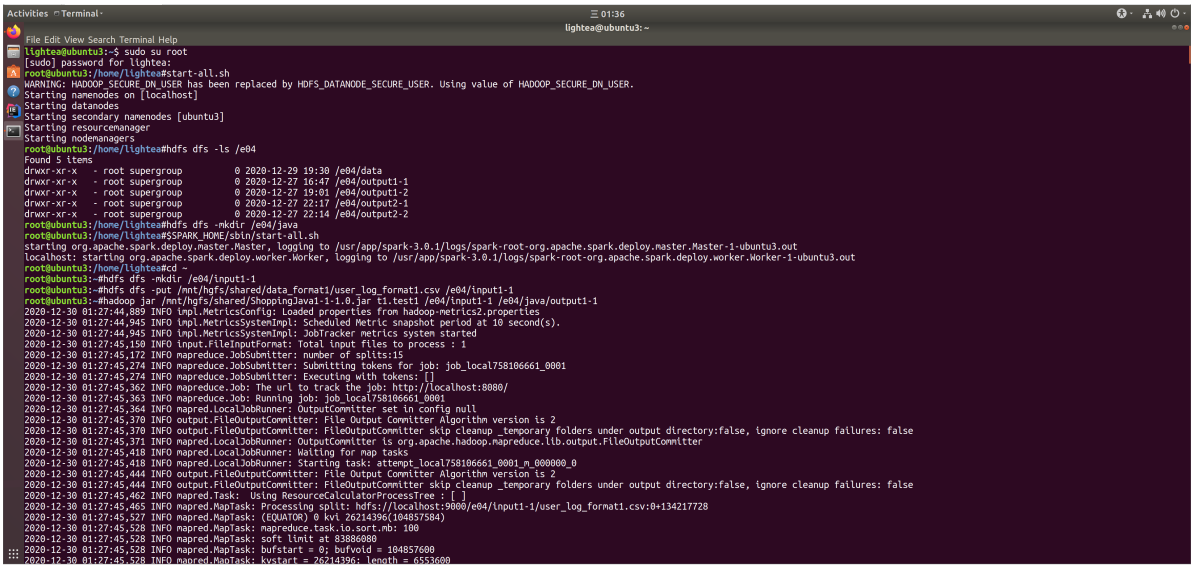

Activities
Terminal
19:40
lightea@ubuntu3: ~
File Edit View Search Terminal Help
(66,1432)
(742,1423)
(2468,1360)
(1056,1356)
(4845,1344)
(2418,1339)
(2669,1335)
(4048,1333)
(4648,1263)
(1861,1253)
(4818,1220)
(4766,1203)
(2273,1198)
(3022,1194)
(2336,1186)
(361,1176)
(474,1174)
(4798,1167)
(2545,1159)
(1087,1133)
(4847,1126)
(2954,1119)
(1346,1106)
(4871,1098)
(3578,1088)
(2387,1076)
(3971,1074)
(4605,1056)
(2676,1051)
(1480,1049)
(2823,1047)
(4127,1040)
(1,1030)
(2206,1027)
(4257,1001)
(4129,969)
(4160,968)
(2193,963)
(2664,954)
(1727,951)
(310,948)
(1310,930)
(1487,918)
(1200,914)
(3859,914)
(2928,897)
(3163,885)

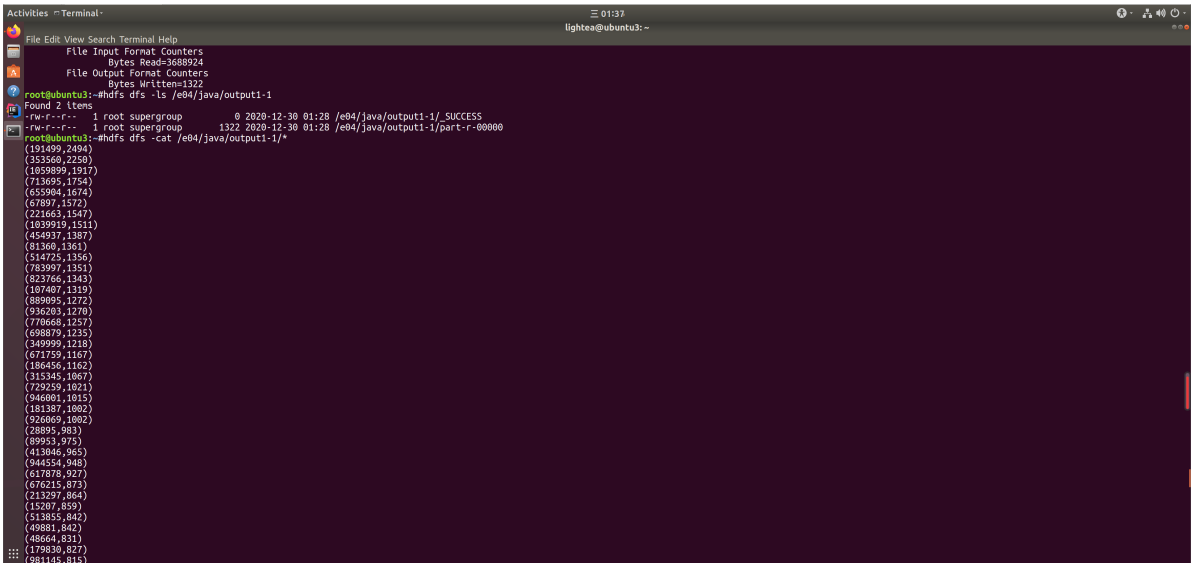


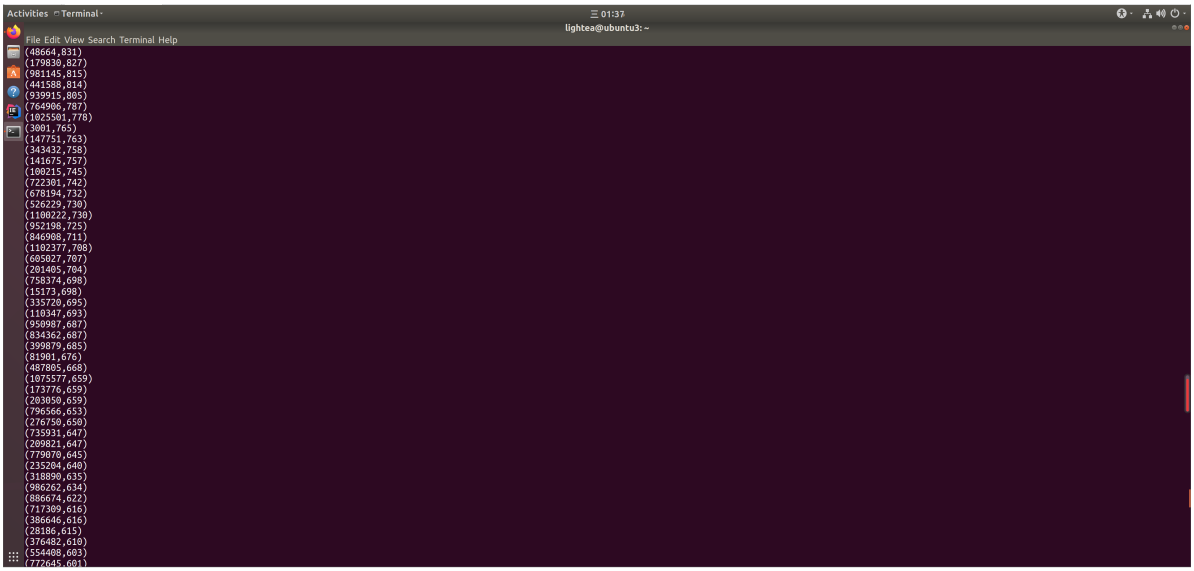
test1 - java

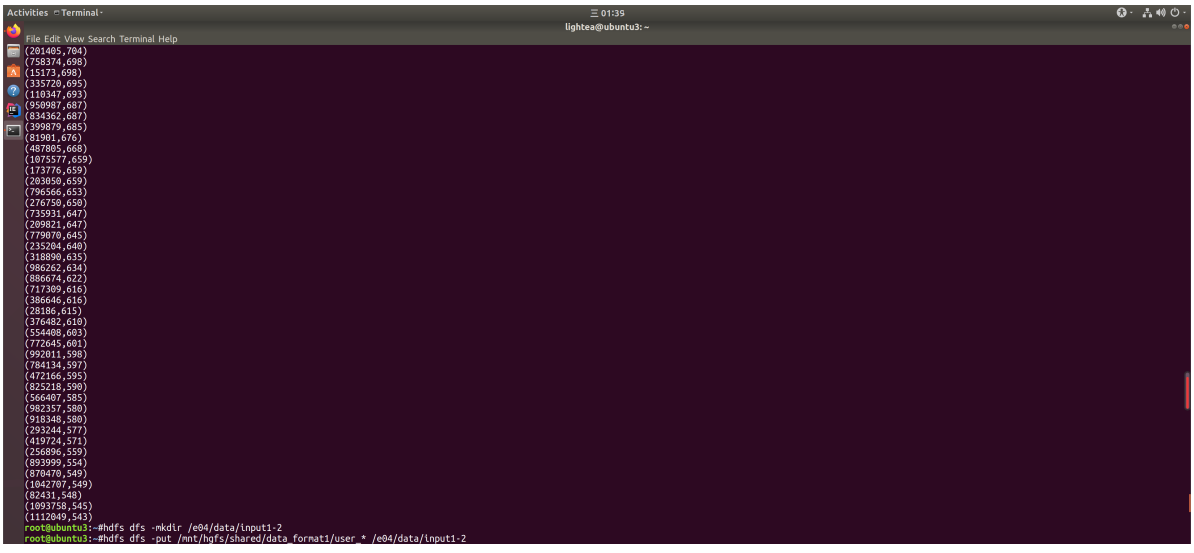
见 `ShoppingJava` 文件夹，运行结果：

1-1

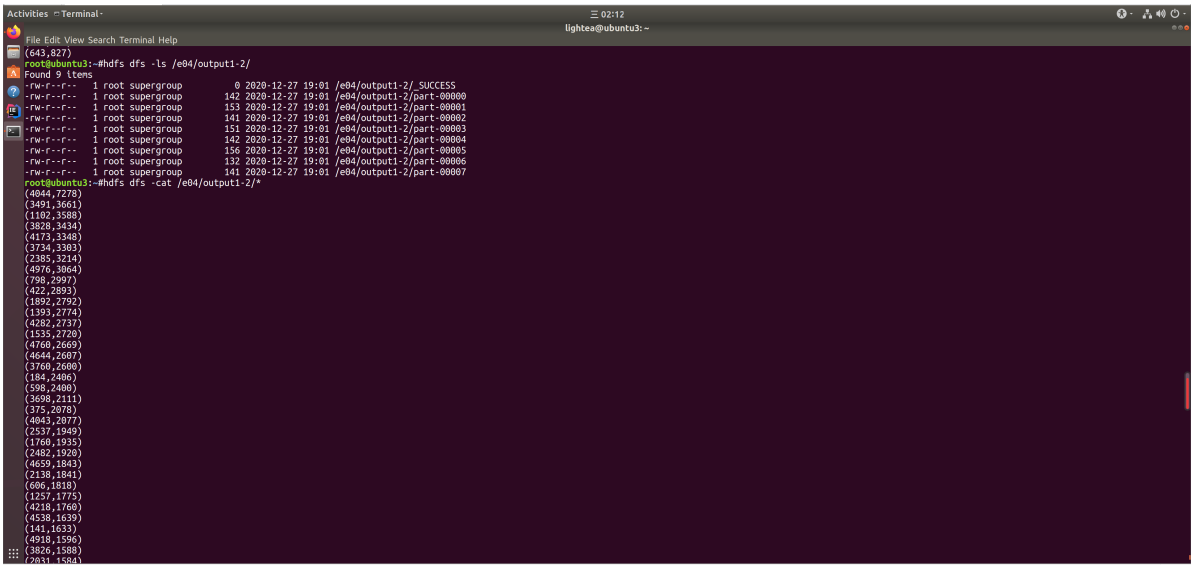


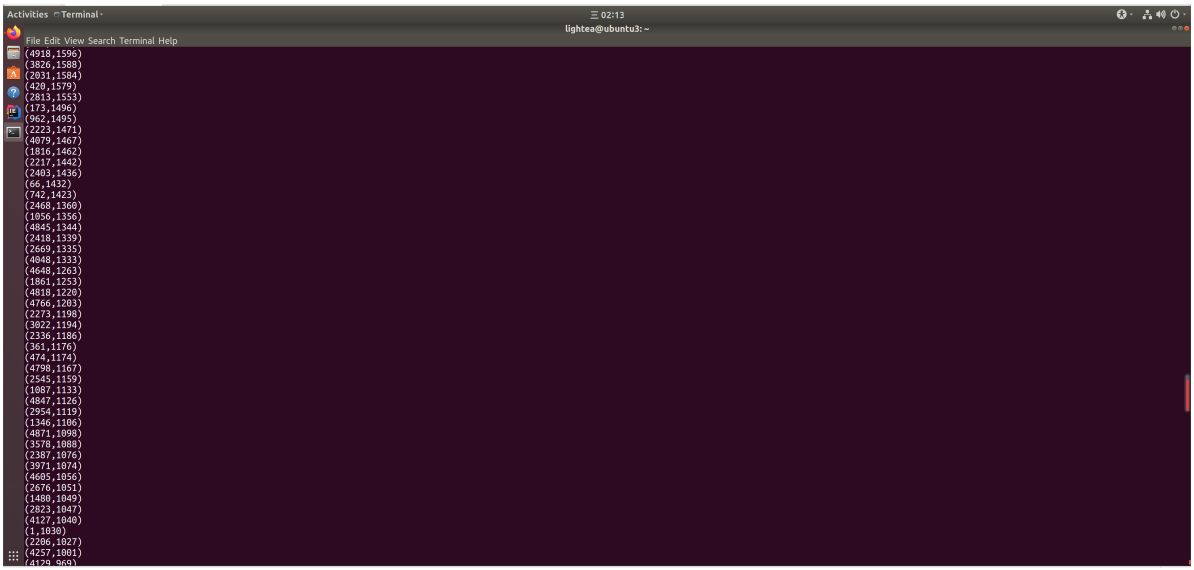

Activities Terminal 01:37
lightea@ubuntu3: ~
File Edit View Search Terminal Help
File Input Format Counters
Bytes Read=3688924
File Output Format Counters
Bytes Written=1322
root@ubuntu3:~#hdfs dfs -ls /e04/java/output1-1
Found 2 items
-rw-r--r-- 1 root supergroup 0 2020-12-30 01:28 /e04/java/output1-1/_SUCCESS
-rw-r--r-- 1 root supergroup 1322 2020-12-30 01:28 /e04/java/output1-1/part-r-00000
root@ubuntu3:~#hdfs dfs -cat /e04/java/output1-1/*
(191499,2494)
(353560,2250)
(1059899,1917)
(713695,1754)
(655904,1674)
(67897,1572)
(221663,1547)
(1039919,1511)
(454937,1387)
(81360,1361)
(514725,1356)
(783997,1351)
(823766,1343)
(107407,1319)
(889095,1272)
(936203,1270)
(770668,1257)
(698879,1235)
(349999,1218)
(671759,1167)
(186456,1162)
(315345,1067)
(729259,1021)
(946001,1015)
(181387,1002)
(926069,1002)
(28895,993)
(89953,975)
(413046,965)
(944554,948)
(617878,927)
(676215,873)
(213297,864)
(15207,859)
(513855,842)
(49881,842)
(48664,831)
(179830,827)
(981145,815)

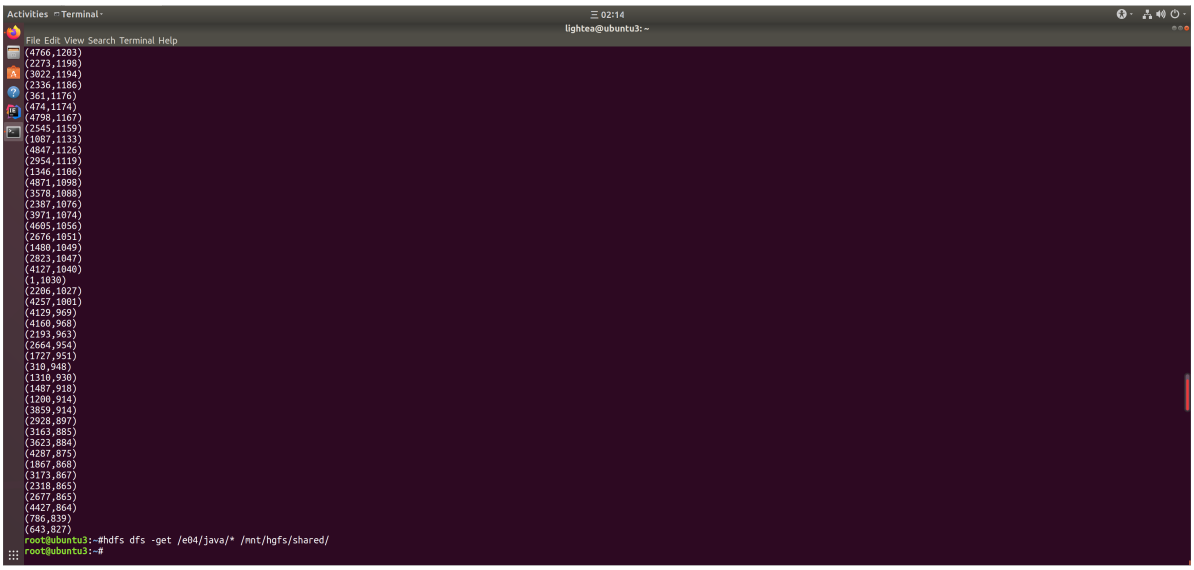

Activities Terminal 01:37
lightea@ubuntu3: ~
File Edit View Search Terminal Help
(48664,831)
(179830,827)
(981145,815)
(441588,814)
(939915,805)
(764906,787)
(1025501,778)
(3001,765)
(147751,763)
(343432,758)
(141675,757)
(100215,745)
(722301,742)
(678194,732)
(526229,730)
(110022,730)
(952198,725)
(846908,711)
(1102377,708)
(605027,707)
(201405,704)
(758374,698)
(15173,698)
(335720,695)
(110347,693)
(950987,687)
(834362,687)
(399879,685)
(81901,676)
(487805,668)
(1075577,659)
(173776,659)
(203050,659)
(796566,653)
(276750,650)
(735931,647)
(209821,647)
(779070,645)
(235204,640)
(318890,635)
(986262,634)
(886674,622)
(717309,616)
(386646,616)
(28186,615)
(376482,610)
(554408,603)
(772645,601)


Activities Terminal 01:39
lightea@ubuntu3: ~
File Edit View Search Terminal Help
(201405,704)
(758374,698)
(15173,698)
(335720,695)
(110347,693)
(950987,687)
(834362,687)
(399879,685)
(81901,676)
(487805,668)
(1075577,659)
(173776,659)
(203050,659)
(796566,653)
(276750,650)
(735931,647)
(209821,647)
(779070,645)
(235204,640)
(318890,635)
(986262,634)
(886674,622)
(717309,616)
(386646,616)
(28186,615)
(376482,610)
(554408,603)
(772645,601)
(992011,598)
(784134,597)
(472166,595)
(825218,590)
(566407,585)
(982357,580)
(918348,580)
(293244,577)
(419724,571)
(256896,559)
(893999,554)
(870470,549)
(1042707,549)
(82431,548)
(1093758,545)
(1112049,543)
root@ubuntu3:~#hdfs dfs -mkdir /e04/data/input1-2
root@ubuntu3:~#hdfs dfs -put /mnt/hgfs/shared/data_format1/user_* /e04/data/input1-2

1-2







test2

> 编写Spark程序统计双十一购买了了商品的男女女比例，以及购买了商品的买家年年龄段的比例；

1. 解决步骤

```
//2-1 双十一购买了商品的男女比例
/*TODO
1. 读入两张表
2. log 挑出user_id, time_stamp, action_type; 筛选出action_type=2,time_stamp=1111,
3. info 挑出user_id, gender; 筛选出gender=0/1
4. join by user_id
5. groupBy("gender").count()
6. ratio 计算
*/
```

```
//2-2 统计双十一购买了商品的商家年龄段的比例
/*TODO
1. info 挑出user_id, age_range; 筛选出 0<age_range<9
2. log 挑出user_id, time_stamp, action_type; 筛选出 time_stamp=1111 and
action_type=2
3. join by user_id
4. groupBy("age_range").count()
5. ratio 计算
*/
```

2. scala 代码

```
//2-1 双十一购买了商品的男女比例
val dflog
=spark.read.format("csv").option("header","true").load("hdfs:///e04/data/user_lo
g_format1.csv")
val dfinfo
=spark.read.format("csv").option("header","true").load("hdfs:///e04/data/user_in
fo_format1.csv")
val dflg =
dflog.select("user_id","time_stamp","action_type").filter("time_stamp=1111 and
action_type=2")
val dfig = dfinfo.select("user_id","gender").filter("gender=1 or gender=0")
val dfjoing = dfig.join(dflg,"user_id")
val dfcountg = dfjoing.groupBy("gender").count()
dfcountg.withColumn("ratio",dfcountg("count")/dfjoing.count).show
```

```
+------+------+------------------+
|gender| count|             ratio|
+------+------+------------------+
|     0|846054|0.7232596926427983|
|     1|323725|0.2767403073572017|
+------+------+------------------+
```

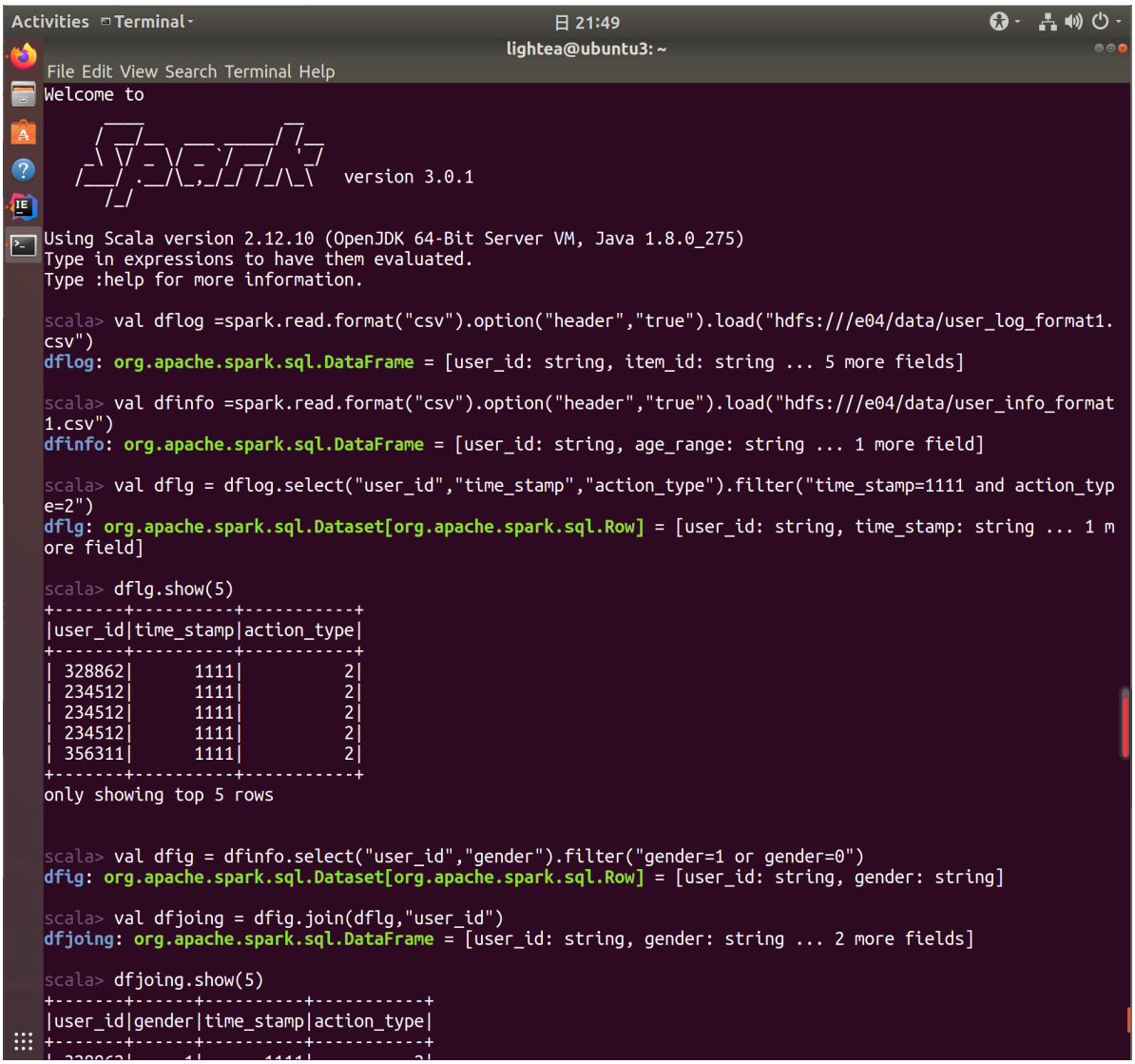
```
//2-2 统计双十一购买了商品的商家年龄段的比例
val dfia = dfinfo.select("user_id","age_range").filter("age_range>0 and
age_range<9")
val dfla =
dflog.select("user_id","time_stamp","action_type").filter("time_stamp=1111 and
action_type=2")
val dfjoina = dfia.join(dfla,"user_id")
val dfcounta = dfjoina.groupBy("age_range").count()
dfcounta.withColumn("ratio",dfcounta("count")/dfjoina.count).show
```

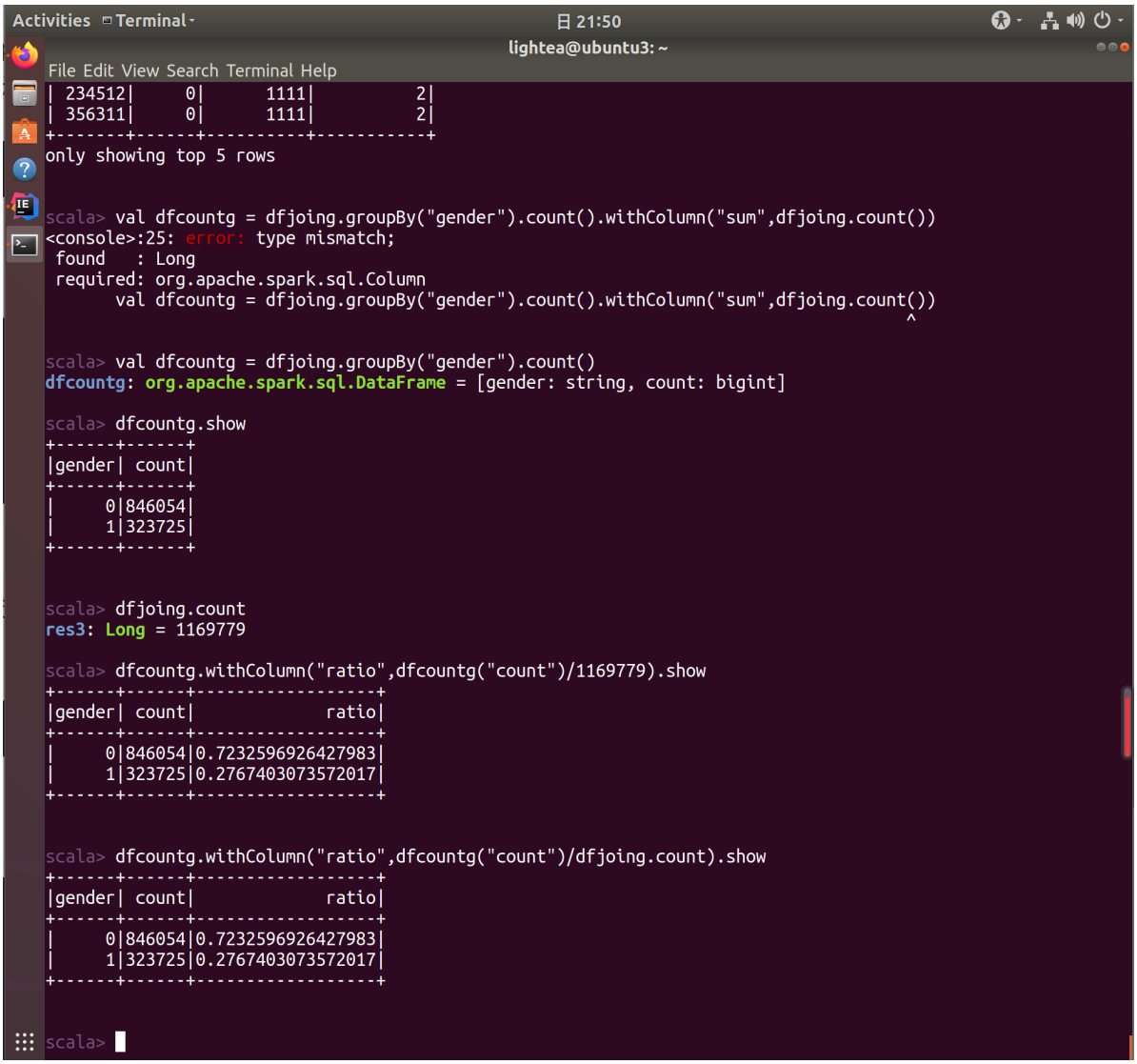
```
+---------+------+--------------------+
|age_range| count|               ratio|
+---------+------+--------------------+
|        7| 19363|0.019736191647869567|
|        3|327758| 0.33407502464093547|
|        8|  3476|0.003542994482672861|
|        5|133480| 0.13605261897214427|
|        6|103774| 0.10577408211878409|
|        1|    54|5.504076584129301E-5|
|        4|268549|  0.2737248634428407|
|        2|124637| 0.12703918392891178|
+---------+------+--------------------+
```

```
/*将答案保存到本地*/
val rddrg =
dfcountg.withColumn("ratio",dfcountg("count")/dfjoing.count).select("gender","ra
tio").rdd.map(x=>(x(0),x(1))).collect()
sc.parallelize(rddrg).saveAsTextFile("hdfs:///e04/output2-1")

val dfratioa =
dfcounta.withColumn("ratio",dfcounta("count")/dfjoina.count).select("age_range",
"ratio").orderBy("age_range")
val rddra = dfratioa.rdd.map(x=>(x(0),x(1)))
val rddra = dfratioa.rdd.map(x=>(x(0),x(1))).collect()
sc.parallelize(rddra).saveAsTextFile("hdfs:///e04/output2-2")
```

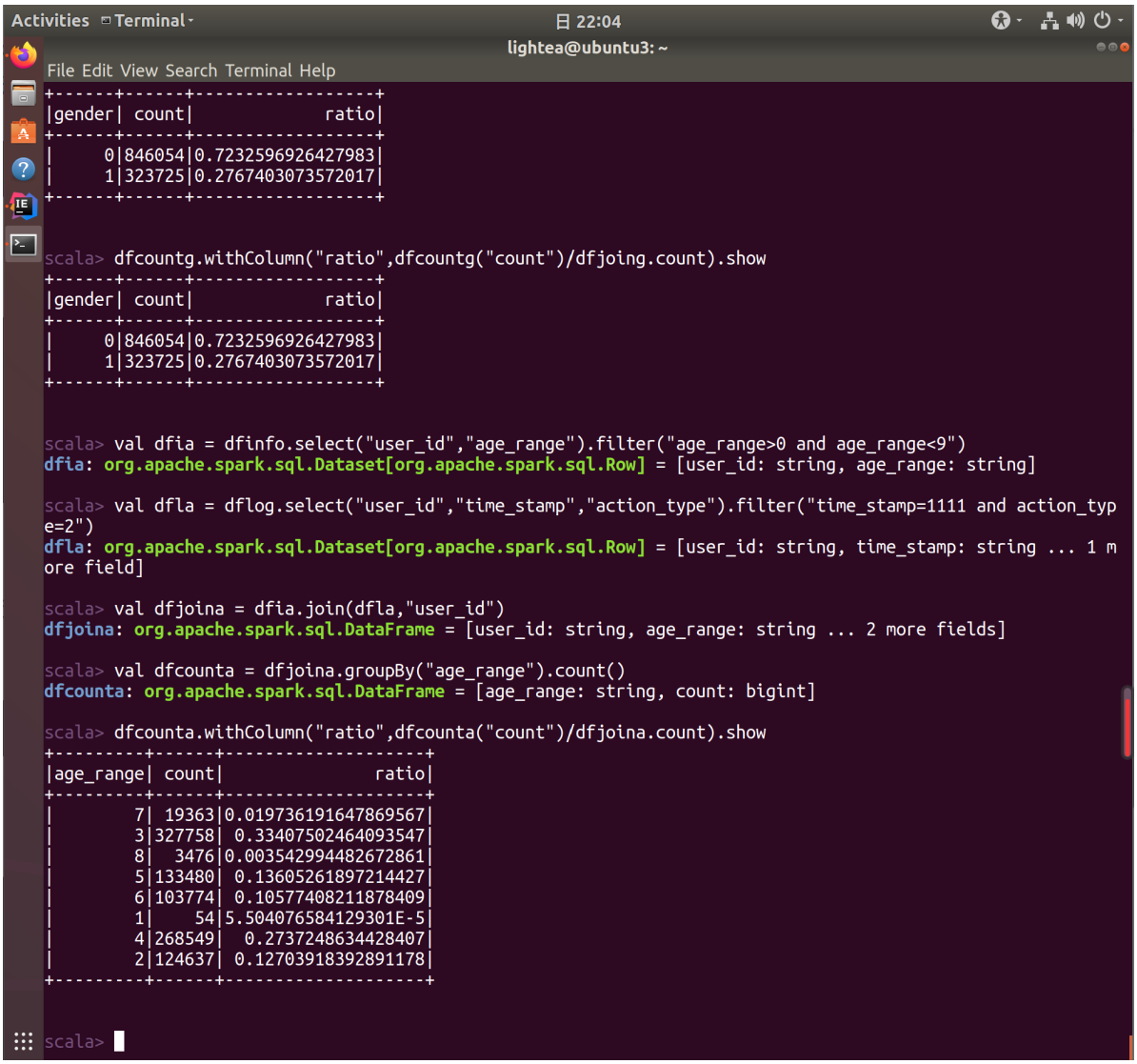
3. 运行结果

   2-1

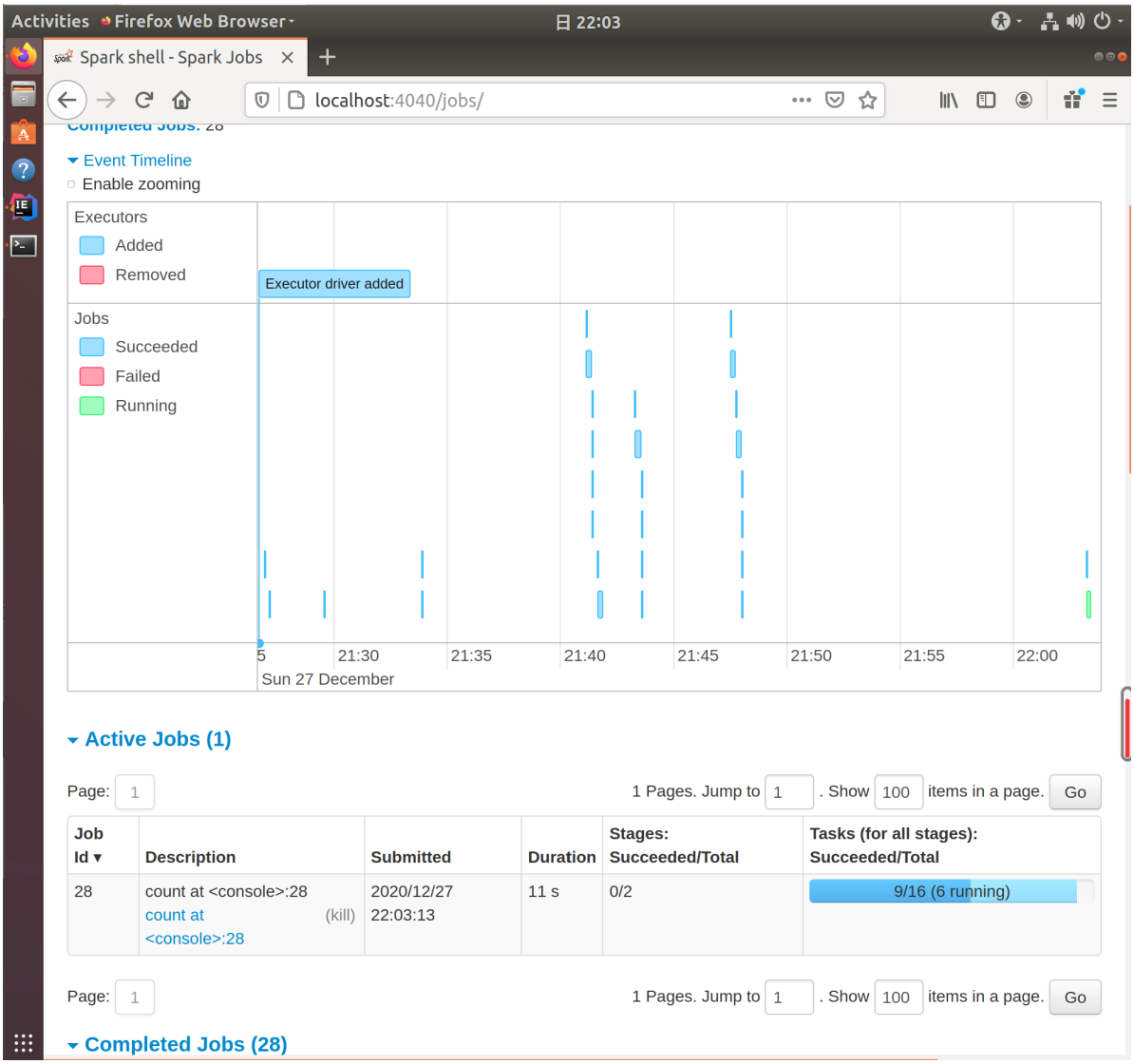

Activities
Terminal
21:49
lightea@ubuntu3: ~
File Edit View Search Terminal Help
Welcome to
version 3.0.1
Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_275)
Type in expressions to have them evaluated.
Type :help for more information.
scala> val dflog =spark.read.format("csv").option("header","true").load("hdfs:///e04/data/user_log_format1.csv")
dflog: org.apache.spark.sql.DataFrame = [user_id: string, item_id: string ... 5 more fields]
scala> val dfinfo =spark.read.format("csv").option("header","true").load("hdfs:///e04/data/user_info_format1.csv")
dfinfo: org.apache.spark.sql.DataFrame = [user_id: string, age_range: string ... 1 more field]
scala> val dflg = dflog.select("user_id","time_stamp","action_type").filter("time_stamp=1111 and action_type=2")
dflg: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [user_id: string, time_stamp: string ... 1 more field]
scala> dflg.show(5)
+-------+----------+-----------+
|user_id|time_stamp|action_type|
+-------+----------+-----------+
328862	1111	2
234512	1111	2
234512	1111	2
234512	1111	2
356311	1111	2
+-------+----------+-----------+
only showing top 5 rows
scala> val dfig = dfinfo.select("user_id","gender").filter("gender=1 or gender=0")
dfig: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [user_id: string, gender: string]
scala> val dfjoing = dfig.join(dflg,"user_id")
dfjoing: org.apache.spark.sql.DataFrame = [user_id: string, gender: string ... 2 more fields]
scala> dfjoing.show(5)
+-------+------+----------+-----------+
|user_id|gender|time_stamp|action_type|
+-------+------+----------+-----------+


Activities Terminal 21:50
lightea@ubuntu3: ~
File Edit View Search Terminal Help
| 234512| 0| 1111| 2|
| 356311| 0| 1111| 2|
+-------+------+----------+-----------+
only showing top 5 rows
scala> val dfcountg = dfjoing.groupBy("gender").count().withColumn("sum",dfjoing.count())
<console>:25: error: type mismatch;
found : Long
required: org.apache.spark.sql.Column
val dfcountg = dfjoing.groupBy("gender").count().withColumn("sum",dfjoing.count())
^
scala> val dfcountg = dfjoing.groupBy("gender").count()
dfcountg: org.apache.spark.sql.DataFrame = [gender: string, count: bigint]
scala> dfcountg.show
+------+------+
|gender| count|
+------+------+
| 0|846054|
| 1|323725|
+------+------+
scala> dfjoing.count
res3: Long = 1169779
scala> dfcountg.withColumn("ratio",dfcountg("count")/1169779).show
+------+------+------------------+
|gender| count| ratio|
+------+------+------------------+
| 0|846054|0.7232596926427983|
| 1|323725|0.2767403073572017|
+------+------+------------------+
scala> dfcountg.withColumn("ratio",dfcountg("count")/dfjoing.count).show
+------+------+------------------+
|gender| count| ratio|
+------+------+------------------+
| 0|846054|0.7232596926427983|
| 1|323725|0.2767403073572017|
+------+------+------------------+
scala>

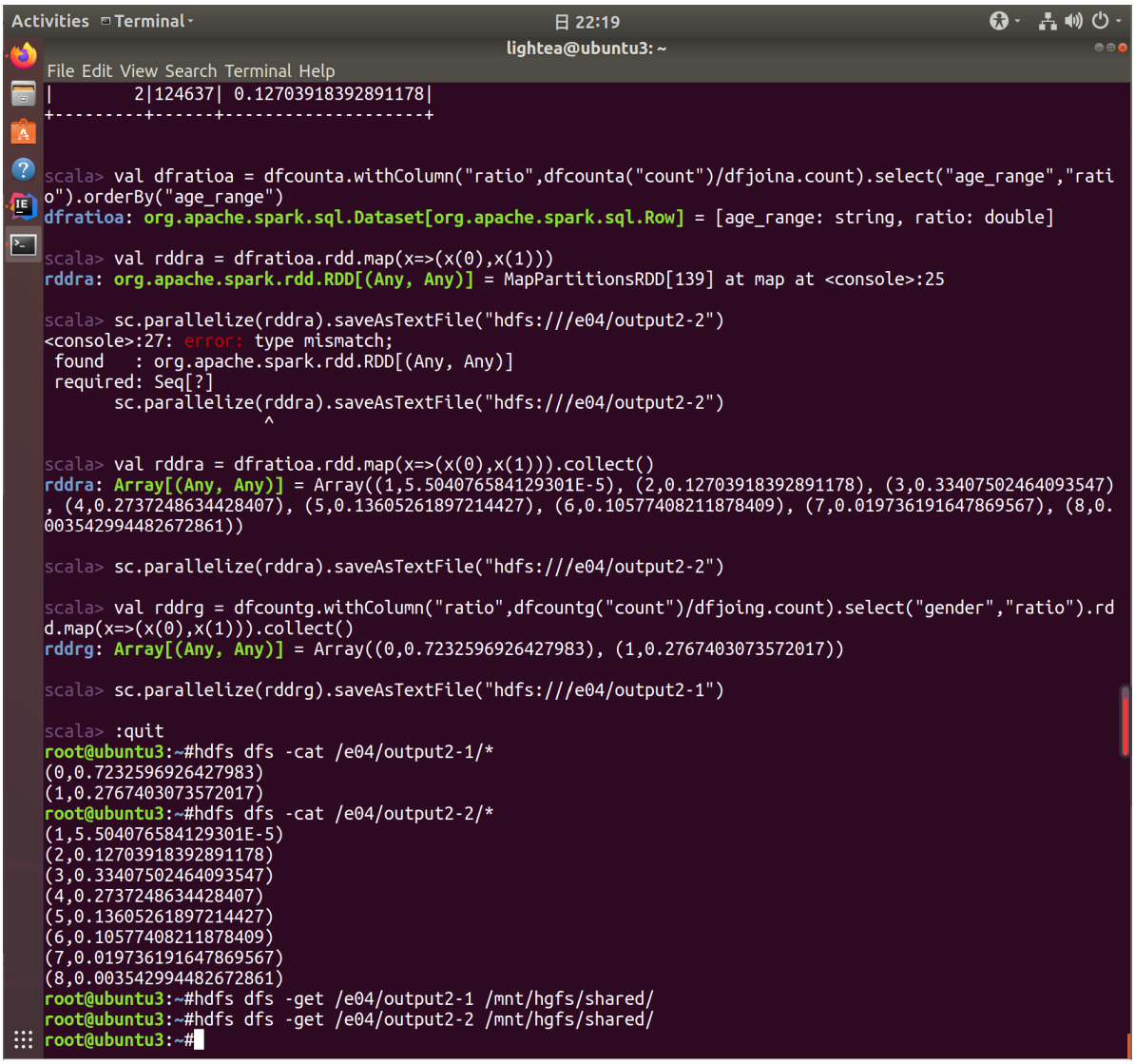
2-2


Activities
Terminal
22:04
lightea@ubuntu3: ~
File Edit View Search Terminal Help
+------+------+------------------+
|gender| count| ratio|
+------+------+------------------+
| 0|846054|0.7232596926427983|
| 1|323725|0.2767403073572017|
+------+------+------------------+
scala> dfcountg.withColumn("ratio",dfcountg("count")/dfjoing.count).show
+------+------+------------------+
|gender| count| ratio|
+------+------+------------------+
| 0|846054|0.7232596926427983|
| 1|323725|0.2767403073572017|
+------+------+------------------+
scala> val dfia = dfinfo.select("user_id","age_range").filter("age_range>0 and age_range<9")
dfia: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [user_id: string, age_range: string]
scala> val dfla = dflog.select("user_id","time_stamp","action_type").filter("time_stamp=1111 and action_typ
e=2")
dfla: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [user_id: string, time_stamp: string ... 1 m
ore field]
scala> val dfjoina = dfia.join(dfla,"user_id")
dfjoina: org.apache.spark.sql.DataFrame = [user_id: string, age_range: string ... 2 more fields]
scala> val dfcounta = dfjoina.groupBy("age_range").count()
dfcounta: org.apache.spark.sql.DataFrame = [age_range: string, count: bigint]
scala> dfcounta.withColumn("ratio",dfcounta("count")/dfjoina.count).show
+---------+------+--------------------+
|age_range| count| ratio|
+---------+------+--------------------+
7	19363	0.019736191647869567
3	327758	0.33407502464093547
8	3476	0.003542994482672861
5	133480	0.13605261897214427
6	103774	0.10577408211878409
1	54	5.504076584129301E-5
4	268549	0.2737248634428407
2	124637	0.12703918392891178
+---------+------+--------------------+
scala>

SparkUI


Activities
Firefox Web Browser
22:03
Spark shell - Spark Jobs
localhost:4040/jobs/
Completed Jobs: 28
Event Timeline
Enable zooming
Executors
Added
Removed
Executor driver added
Jobs
Succeeded
Failed
Running
5
21:30
21:35
21:40
21:45
21:50
21:55
22:00
Sun 27 December
Active Jobs (1)
Page: 1
1 Pages. Jump to 1 . Show 100 items in a page. Go
Job Id ▾
Description
Submitted
Duration
Stages: Succeeded/Total
Tasks (for all stages): Succeeded/Total
28
count at <console>:28
count at <console>:28
(kill)
2020/12/27 22:03:13
11 s
0/2
9/16 (6 running)
Page: 1
1 Pages. Jump to 1 . Show 100 items in a page. Go
Completed Jobs (28)

all result (save as text files)



test3

1. sql 代码

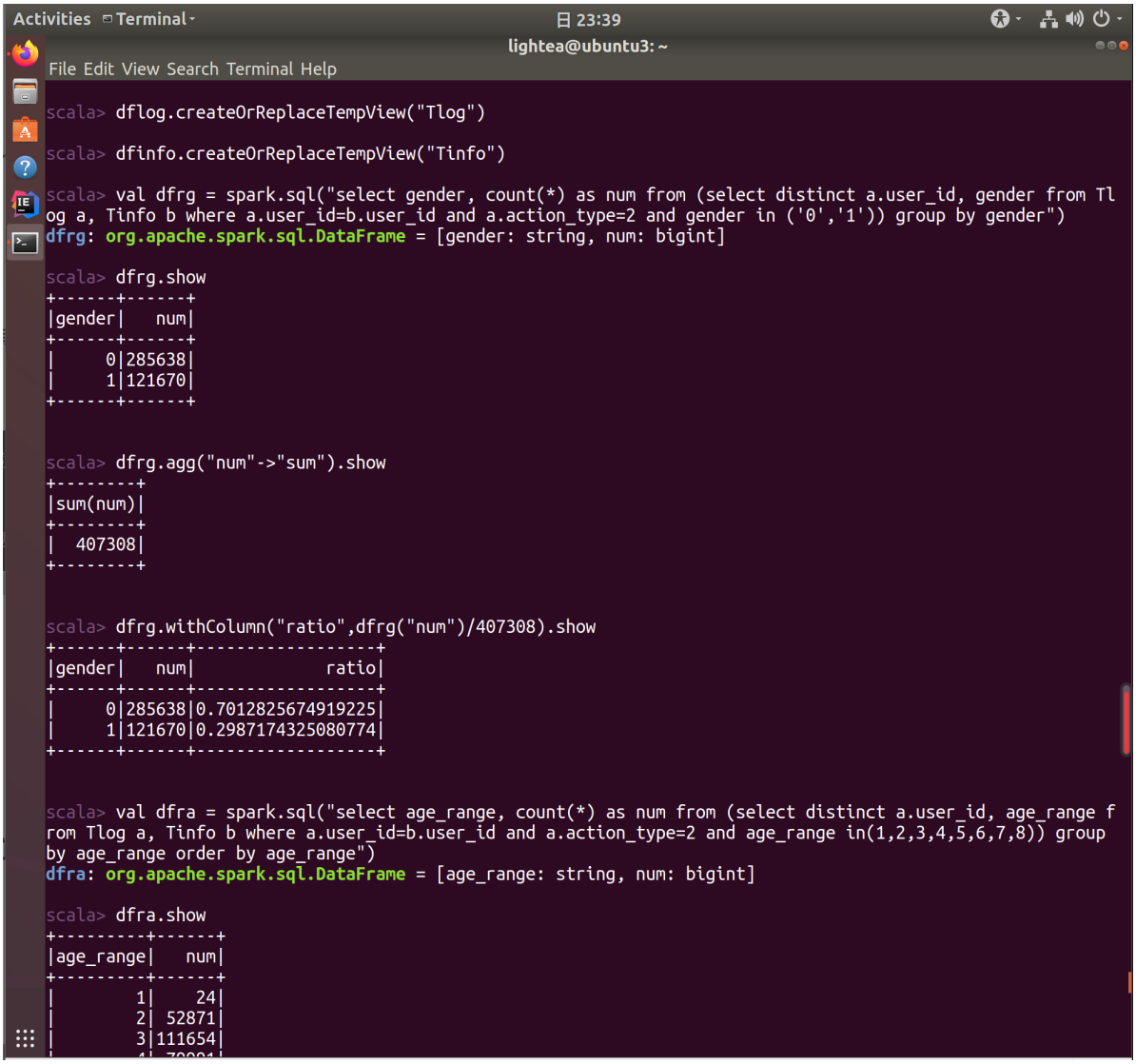
```
    // 以 dataframe 读入两个 csv 文件
val dflog
=spark.read.format("csv").option("header","true").load("hdfs:///e04/data/user_log_format1.csv")
val dfinfo
=spark.read.format("csv").option("header","true").load("hdfs:///e04/data/user_info_format1.csv")
    // 转换成sql可用的表格类型
dflog.createOrReplaceTempView("Tlog")
dfinfo.createOrReplaceTempView("Tinfo")
    // sql 语句
val dfrg = spark.sql("select gender, count(*) as num from (select distinct a.user_id, gender from Tlog a, Tinfo b where a.user_id=b.user_id and a.action_type=2 and gender in ('0','1')) group by gender")
val dfra = spark.sql("select age_range, count(*) as num from (select distinct a.user_id, age_range from Tlog a, Tinfo b where a.user_id=b.user_id and a.action_type=2 and age_range in(1,2,3,4,5,6,7,8)) group by age_range order by age_range")
```

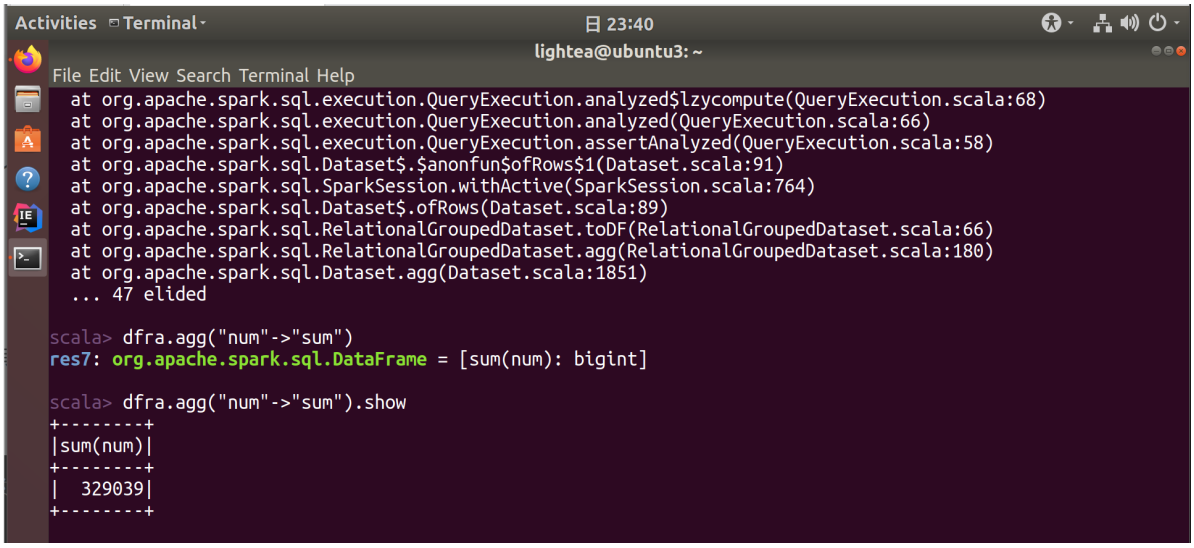
```
dfrg.show
```

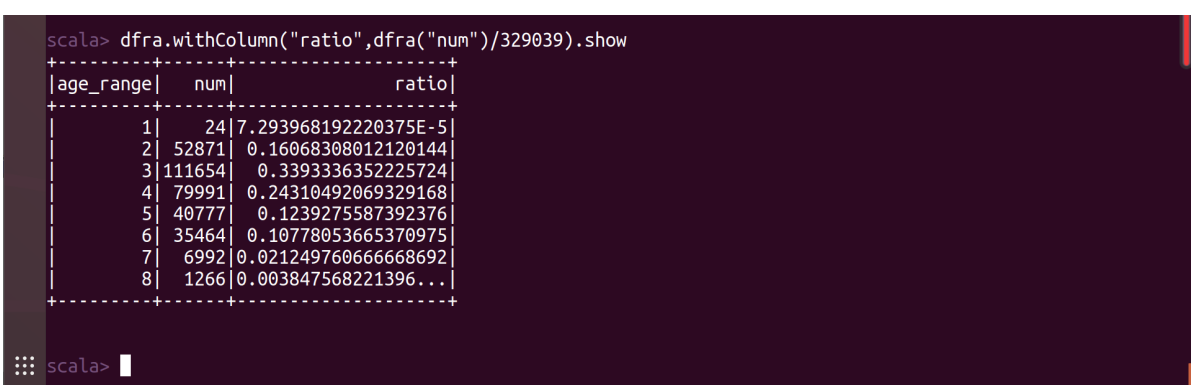
```
+------+------+
|gender|   num|
+------+------+
|     0|285638|
|     1|121670|
+------+------+
dfra.show
+---------+------+
|age_range|   num|
+---------+------+
|        1|    24|
|        2| 52871|
|        3|111654|
|        4| 79991|
|        5| 40777|
|        6| 35464|
|        7|  6992|
|        8|  1266|
+---------+------+
```

```
/*添加一列ratio计算*/
dfrg.agg("num"->"sum").show
+--------+
|sum(num)|
+--------+
|  407308|
+--------+
dfrg.withColumn("ratio",dfrg("num")/407308).show
+------+------+------------------+
|gender|   num|             ratio|
+------+------+------------------+
|     0|285638|0.7012825674919225|
|     1|121670|0.2987174325080774|
+------+------+------------------+
dfra.agg("num"->"sum").show
+--------+
|sum(num)|
+--------+
|  329039|
+--------+
dfra.withColumn("ratio",dfra("num")/329039).show
+---------+------+--------------------+
|age_range|   num|               ratio|
+---------+------+--------------------+
|        1|    24|7.293968192220375E-5|
|        2| 52871| 0.16068308012120144|
|        3|111654|  0.3393336352225724|
|        4| 79991| 0.24310492069329168|
|        5| 40777|  0.1239275587392376|
|        6| 35464| 0.10778053665370975|
|        7|  6992|0.021249760666668692|
|        8|  1266|0.003847568221396...|
+---------+------+--------------------+
```

2. 运行结果


Activities Terminal 日 23:39
lightea@ubuntu3: ~
File Edit View Search Terminal Help
scala> dflog.createOrReplaceTempView("Tlog")
scala> dfinfo.createOrReplaceTempView("Tinfo")
scala> val dfrg = spark.sql("select gender, count(*) as num from (select distinct a.user_id, gender from Tlog a, Tinfo b where a.user_id=b.user_id and a.action_type=2 and gender in ('0','1')) group by gender")
dfrg: org.apache.spark.sql.DataFrame = [gender: string, num: bigint]
scala> dfrg.show
+------+------+
|gender| num|
+------+------+
| 0|285638|
| 1|121670|
+------+------+
scala> dfrg.agg("num"->"sum").show
+--------+
|sum(num)|
+--------+
| 407308|
+--------+
scala> dfrg.withColumn("ratio",dfrg("num")/407308).show
+------+------+------------------+
|gender| num| ratio|
+------+------+------------------+
| 0|285638|0.7012825674919225|
| 1|121670|0.2987174325080774|
+------+------+------------------+
scala> val dfra = spark.sql("select age_range, count(*) as num from (select distinct a.user_id, age_range from Tlog a, Tinfo b where a.user_id=b.user_id and a.action_type=2 and age_range in(1,2,3,4,5,6,7,8)) group by age_range order by age_range")
dfra: org.apache.spark.sql.DataFrame = [age_range: string, num: bigint]
scala> dfra.show
+---------+------+
|age_range| num|
+---------+------+
1	24
2	52871
3	111654


Activities Terminal 日 23:40
lightea@ubuntu3: ~
File Edit View Search Terminal Help
at org.apache.spark.sql.execution.QueryExecution.analyzed$lzycompute(QueryExecution.scala:68)
at org.apache.spark.sql.execution.QueryExecution.analyzed(QueryExecution.scala:66)
at org.apache.spark.sql.execution.QueryExecution.assertAnalyzed(QueryExecution.scala:58)
at org.apache.spark.sql.Dataset$.$anonfun$ofRows$1(Dataset.scala:91)
at org.apache.spark.sql.SparkSession.withActive(SparkSession.scala:764)
at org.apache.spark.sql.Dataset$.ofRows(Dataset.scala:89)
at org.apache.spark.sql.RelationalGroupedDataset.toDF(RelationalGroupedDataset.scala:66)
at org.apache.spark.sql.RelationalGroupedDataset.agg(RelationalGroupedDataset.scala:180)
at org.apache.spark.sql.Dataset.agg(Dataset.scala:1851)
... 47 elided
scala> dfra.agg("num"->"sum")
res7: org.apache.spark.sql.DataFrame = [sum(num): bigint]
scala> dfra.agg("num"->"sum").show
+--------+
|sum(num)|
+--------+
| 329039|
+--------+


scala> dfra.withColumn("ratio",dfra("num")/329039).show
+---------+------+--------------------+
|age_range| num| ratio|
+---------+------+--------------------+
1	24	7.293968192220375E-5
2	52871	0.16068308012120144
3	111654	0.3393336352225724
4	79991	0.24310492069329168
5	40777	0.1239275587392376
6	35464	0.10778053665370975
7	6992	0.021249760666668692
8	1266	0.003847568221396...
+---------+------+--------------------+
scala>

test4

> 预测给定的商家中，哪些新消费者在未来会成为忠实客户，即需要预测这些新消费者在6个月内再次购买的概率。基于Spark MLlib编写程序预测回头客，评估实验结果的准确率。

1. pyspark 配置 参考

```
$ pip install pyspark
```

使用 jupyter notebook 编写的配置

```
# ~/.bashrc 中添加
export PYSPARK_DRIVER_PYTHON=jupyter
export PYSPARK_DRIVER_PYTHON_OPTS='notebook'
```

```
$ jupyter notebook --generate-config
$ vim ~/.jupyter/jupyter_notebook_config.py
            #修改
         c.NotebookApp.allow_root = True
```

```
$ start-all.sh  # 开启hadoop,hdfs
$ $SPARK-HOME/sbin/start-all.sh # 开启spark
$ pyspark   # 打开相应网址进行在线编辑
```

2. 数据处理及特征工程

特征建立 - 与参考网址一致

```
用户的年龄(age_range)
用户的性别(gender)
某用户在该商家日志的总条数(total_logs)
用户浏览的商品的数目，就是浏览了多少个商品(unique_item_ids)
浏览的商品的种类的数目，就是浏览了多少种商品(categories)
用户浏览的天数(browse_days)
用户单击的次数(one_clicks)
用户添加购物车的次数(shopping_carts)
用户购买的次数(purchase_times)
用户收藏的次数(favourite_times)
```

见 `dataDealing.py`

2. 随机森林预测

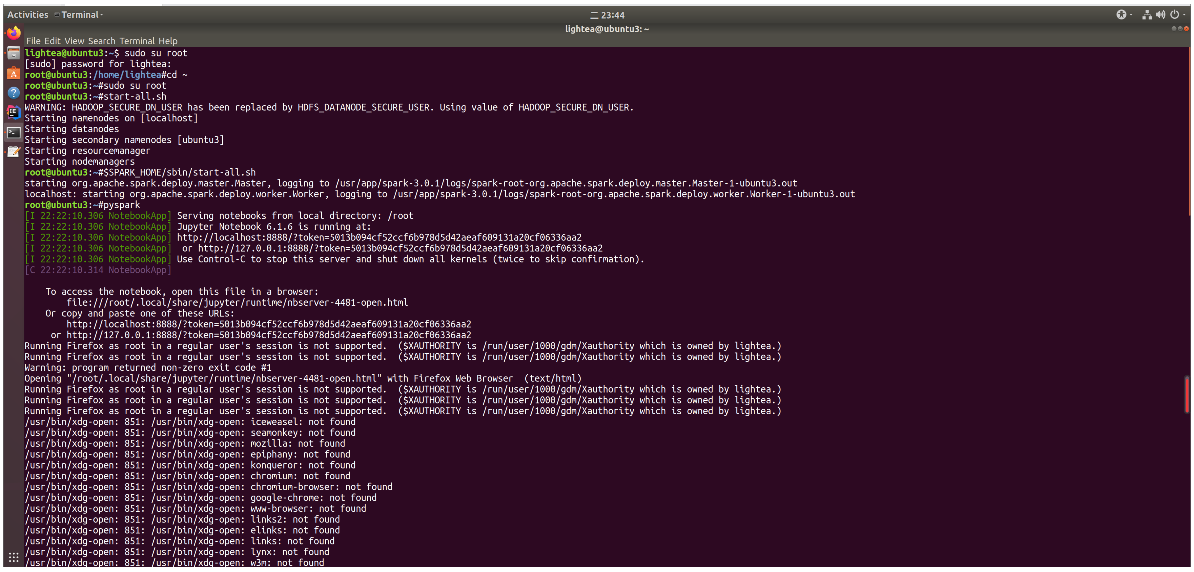
由 baseline data_train的结果呈现，发现其随机森林预测正确率最高，因此采取spark ml中的random forest 进行预测

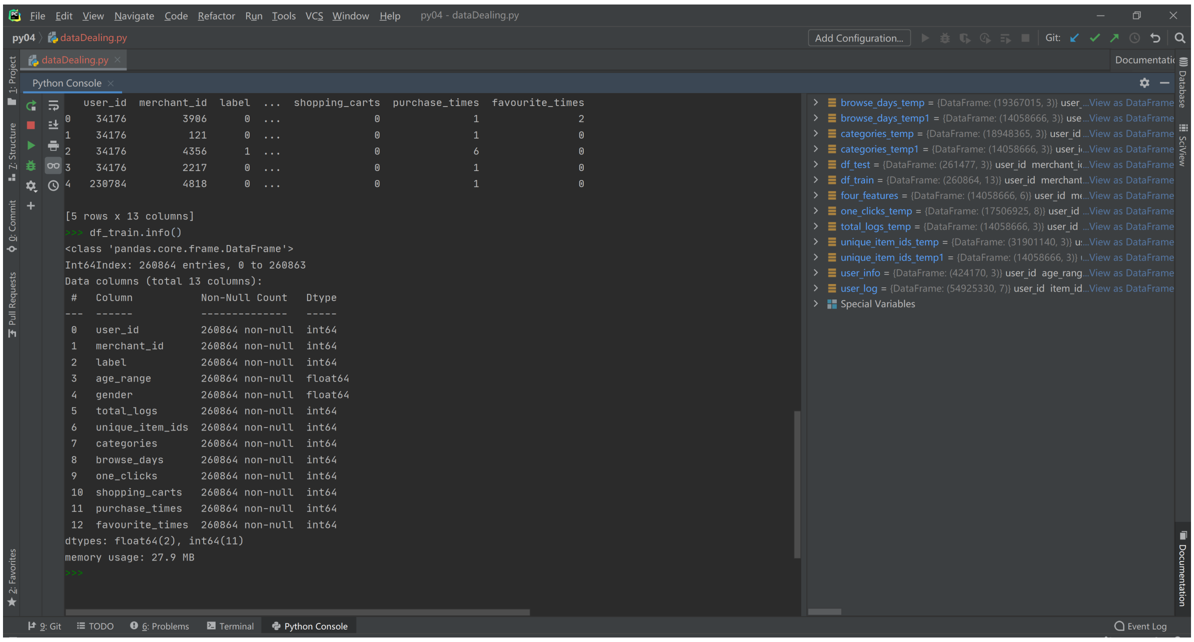
`pyspark-ml.py` 参数设定：

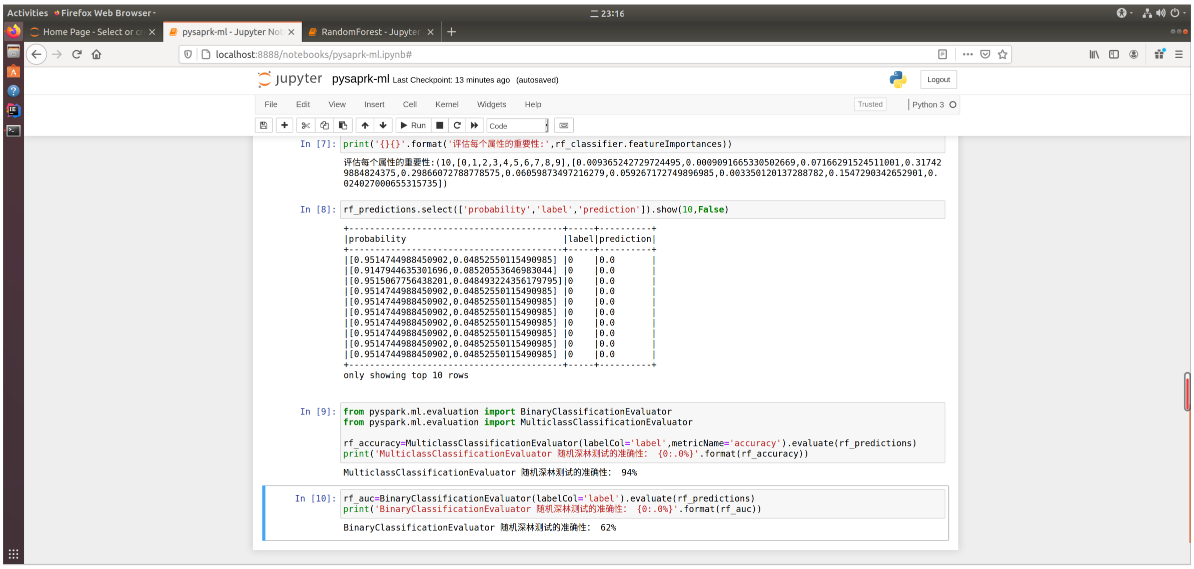
- 测试集与训练集划分比例 `randomSplit([0.7,0.3])`
- 随机森林基学习器数量 `numTree=200`

调用 `pyspark.ml.evaluation` 中的 `MulticlassClassificationEvaluator` 计算**Accuracy=0.94**
调用 `pyspark.ml.evaluation` 中的 `BinaryClassificationEvaluator` 计算**AUC=0.62**

3. 运行结果


Activities Terminal
23:44
lightea@ubuntu3: ~
File Edit View Search Terminal Help
lightea@ubuntu3:~$ sudo su root
[sudo] password for lightea:
root@ubuntu3:/home/lightea#cd ~
root@ubuntu3:~#sudo su root
root@ubuntu3:~#start-all.sh
WARNING: HADOOP_SECURE_DN_USER has been replaced by HDFS_DATANODE_SECURE_USER. Using value of HADOOP_SECURE_DN_USER.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu3]
Starting resourcemanager
Starting nodemanagers
root@ubuntu3:~#$SPARK_HOME/sbin/start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /usr/app/spark-3.0.1/logs/spark-root-org.apache.spark.deploy.master.Master-1-ubuntu3.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /usr/app/spark-3.0.1/logs/spark-root-org.apache.spark.deploy.worker.Worker-1-ubuntu3.out
root@ubuntu3:~#pyspark
[I 22:22:10.306 NotebookApp] Serving notebooks from local directory: /root
[I 22:22:10.306 NotebookApp] Jupyter Notebook 6.1.6 is running at:
[I 22:22:10.306 NotebookApp] http://localhost:8888/?token=5013b094cf52ccf6b978d5d42aeaf609131a20cf06336aa2
[I 22:22:10.306 NotebookApp] or http://127.0.0.1:8888/?token=5013b094cf52ccf6b978d5d42aeaf609131a20cf06336aa2
[I 22:22:10.306 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 22:22:10.314 NotebookApp]
To access the notebook, open this file in a browser:
file:///root/.local/share/jupyter/runtime/nbserver-4481-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=5013b094cf52ccf6b978d5d42aeaf609131a20cf06336aa2
or http://127.0.0.1:8888/?token=5013b094cf52ccf6b978d5d42aeaf609131a20cf06336aa2
Running Firefox as root in a regular user's session is not supported. ($XAUTHORITY is /run/user/1000/gdm/Xauthority which is owned by lightea.)
Running Firefox as root in a regular user's session is not supported. ($XAUTHORITY is /run/user/1000/gdm/Xauthority which is owned by lightea.)
Warning: program returned non-zero exit code #1
Opening "/root/.local/share/jupyter/runtime/nbserver-4481-open.html" with Firefox Web Browser (text/html)
Running Firefox as root in a regular user's session is not supported. ($XAUTHORITY is /run/user/1000/gdm/Xauthority which is owned by lightea.)
Running Firefox as root in a regular user's session is not supported. ($XAUTHORITY is /run/user/1000/gdm/Xauthority which is owned by lightea.)
Running Firefox as root in a regular user's session is not supported. ($XAUTHORITY is /run/user/1000/gdm/Xauthority which is owned by lightea.)
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: iceweasel: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: seamonkey: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: mozilla: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: epiphany: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: konqueror: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: chromium: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: chromium-browser: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: google-chrome: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: www-browser: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: links2: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: elinks: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: links: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: lynx: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: w3m: not found


File Edit View Navigate Code Refactor Run Tools VCS Window Help py04 - dataDealing.py
py04 dataDealing.py
Add Configuration...
Python Console
user_id merchant_id label ... shopping_carts purchase_times favourite_times
0 34176 3906 0 ... 0 1 2
1 34176 121 0 ... 0 1 0
2 34176 4356 1 ... 0 6 0
3 34176 2217 0 ... 0 1 0
4 230784 4818 0 ... 0 1 0
[5 rows x 13 columns]
>>> df_train.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 260864 entries, 0 to 260863
Data columns (total 13 columns):
Column Non-Null Count Dtype
0 user_id 260864 non-null int64
1 merchant_id 260864 non-null int64
2 label 260864 non-null int64
3 age_range 260864 non-null float64
4 gender 260864 non-null float64
5 total_logs 260864 non-null int64
6 unique_item_ids 260864 non-null int64
7 categories 260864 non-null int64
8 browse_days 260864 non-null int64
9 one_clicks 260864 non-null int64
10 shopping_carts 260864 non-null int64
11 purchase_times 260864 non-null int64
12 favourite_times 260864 non-null int64
dtypes: float64(2), int64(11)
memory usage: 27.9 MB
9: Git TODO 6: Problems Terminal Python Console Event Log


Activities Firefox Web Browser
23:16
localhost:8888/notebooks/pysaprk-ml.ipynb#
jupyter pysaprk-ml Last Checkpoint: 13 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
In [7]: print('{}{}'.format('评估每个属性的重要性:',rf_classifier.featureImportances))
评估每个属性的重要性:(10,[0,1,2,3,4,5,6,7,8,9],[0.009365242729724495,0.0009091665330502669,0.07166291524511001,0.317429884824375,0.29866072788778575,0.06059873497216279,0.059267172749896985,0.003350120137288782,0.1547290342652901,0.024027000655315735])
In [8]: rf_predictions.select(['probability','label','prediction']).show(10,False)
probability	label	prediction
[0.9514744988450902,0.04852550115490985]	0	0.0
[0.9147944635301696,0.08520553646983044]	0	0.0
[0.9515067756438201,0.048493224356179795]	0	0.0
[0.9514744988450902,0.04852550115490985]	0	0.0
[0.9514744988450902,0.04852550115490985]	0	0.0
[0.9514744988450902,0.04852550115490985]	0	0.0
[0.9514744988450902,0.04852550115490985]	0	0.0
[0.9514744988450902,0.04852550115490985]	0	0.0
[0.9514744988450902,0.04852550115490985]	0	0.0
[0.9514744988450902,0.04852550115490985]	0	0.0
only showing top 10 rows
In [9]: from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
rf_accuracy=MulticlassClassificationEvaluator(labelCol='label',metricName='accuracy').evaluate(rf_predictions)
print('MulticlassClassificationEvaluator 随机深林测试的准确性: {0:.0%}'.format(rf_accuracy))
MulticlassClassificationEvaluator 随机深林测试的准确性: 94%
In [10]: rf_auc=BinaryClassificationEvaluator(labelCol='label').evaluate(rf_predictions)
print('BinaryClassificationEvaluator 随机深林测试的准确性: {0:.0%}'.format(rf_auc))
BinaryClassificationEvaluator 随机深林测试的准确性: 62%