Vijay's Assignment – Spark 2

Task 1

1.Write a program to read a text file and print the number of rows of data in the document.

- Used flatMap to split the input RDD "test" by " " and performed the count() action

2. Write a program to read a text file and print the number of words in the document.

- Used flatMap to split the input RDD "test" by "," and performed the count() action

3. We have a document where the word separator is -, instead of space. Write a spark code, to obtain the count of the total number of words present in the document.

- Used flatMap to split the input RDD "test" by "-" and performed the count() action

Code:

```
package vksp1
import org.apache.spark.SparkConf
object vjsprk1 {
  def main(args: Array[String]) = {
    val conf = new SparkConf()
      .setAppName("WordCount")
      .setMaster("local")
    val sc = new SparkContext(conf)
    val test = sc.textFile("inp1.txt")
    val t1 = test.flatMap { line => line.split(" ") }
    println("Total number of lines in input file: ",t1.count())
    val t2 = test.flatMap { line => line.split(",") }
    println("Total number of words in input file: ",t2.count())
    val t3 = test.flatMap { line => line.split("-") }
    println("Total number of words in input file with - as the delimiter: ",t3.count())
    sc.stop
  }
}
```

Output:

```
19/02/09 11:47:05 INFO DAGScheduler: Job 0 finished: count at vksprk1.scala:13, took 0.486293 s
(Total number of lines in input file: ,22)
19/02/09 11:47:05 INFO SparkContext: Starting job: count at vksprk1.scala:15
19/02/09 11:47:06 INFO DAGScheduler: Job 1 finished: count at vksprk1.scala:15, took 0.040741 s
(Total number of words in input file: ,110)
19/02/09 11:47:06 INFO SparkContext: Starting job: count at vksprk1.scala:17
19/02/09 11:47:06 INFO DAGScheduler: Job 2 finished: count at vksprk1.scala:17, took 0.026706 s
(Total number of words in input file with - as the delimiter: ,44)
19/02/09 11:47:06 INFO SparkUI: Stopped Spark web UI at http://DESKTOP-NOTJQL9:4041
```

Task 2

Problem Statement 1:

1. Read the text file, and create a tupled rdd.

- RDD created by executing a flatmap to split by " " on the RDD test. Tuple RDD created by using collect action on the text file RDD

2. Find the count of total number of rows present.

- achieved using count on the earlier text file RDD

3. What is the distinct number of subjects present in the entire school

- achieved using map to split the file by ",", second map to filter out only the subjects, distinct transformation to remove duplicates and count action to find the distinct subjects

4. What is the count of the number of students in the school, whose name is Mathew and

marks is 55

- Achieved using filter operation on original map (split by ",") to select only the entries with "Mathew" in the first field (index 0) and 55 in fourth field(index 3)

Code:

```
package vksp1
import org.apache.spark.SparkConf
import org.apache.spark.SparkContext
import org.apache.spark.rdd.RDD.rddToPairRDDFunctions
object vjsprk2 {
  def main(args: Array[String]) = {
    val conf = new SparkConf()
      .setAppName("vjsprk2")
      .setMaster("local")
    val sc = new SparkContext(conf)
    val test = sc.textFile("inp1.txt")
    val t1 = test.flatMap { line => line.split(" ") }
    println("Tuple RDD: "+ t1.collect())
    println("Number of rows : "+t1.count())

    val t2 = test.map{ line => line.split(",")}
    val t3 = t2.map(line=>line(1))
    val t4 = t3.distinct()
    println("Distinct subjects in the entire school are : " + t4.collect().mkString(" "))
    println("Number of distinct subjects in entire school: "+ t4.count())

    val t5 = t2.filter(line => line(0).contains("Mathew")&&line(3).contains("55"))
    println("Number of students in school with name as Mathew and mark as 55 is:"+t5.count())
     sc.stop
  }
}
```

Output

```
19/02/09 12:20:04 INFO DAGScheduler: Job 0 finished: collect at vjsprk2.scala:13, took 0.550886 s
Tuple RDD: [Ljava.lang.String;@55ecbafe
19/02/09 12:20:04 INFO SparkContext: Starting job: count at vjsprk2.scala:14
19/02/09 12:20:04 INFO DAGScheduler: Job 1 finished: count at vjsprk2.scala:14, took 0.046915 s
Number of rows : 22
19/02/09 12:20:04 INFO SparkContext: Starting job: collect at vjsprk2.scala:19
```

```
19/02/09 12:20:04 INFO DAGScheduler: Job 2 finished: collect at vjsprk2.scala:19, took 0.476600 s
Distinct subjects in the entire school are : maths history science
19/02/09 12:20:04 INFO SparkContext: Starting job: count at vjsprk2.scala:20

19/02/09 12:20:05 INFO DAGScheduler: Job 3 finished: count at vjsprk2.scala:20, took 0.059315 s
Number of distinct subjects in entire school: 3
19/02/09 12:20:05 INFO SparkContext: Starting job: count at vjsprk2.scala:23

19/02/09 12:20:05 INFO DAGScheduler: Job 4 finished: count at vjsprk2.scala:23, took 0.046800 s
Number of students in school with name as Mathew and mark as 55 is:2
19/02/09 12:20:05 INFO SparkUI: Stopped Spark web UI at http://DESKTOP-NOTJQL9:4041
```

Problem Statement 2:

1. What is the count of students per grade in the school?

- From the text file RDD, using map only grade was extracted and numeric 1 was added to it. Using Reduce by key the numeric value was added to find the count of students per grade.

2. Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)

- this one was quite complex. A pair RDD was created in the form (name,grade),marks. Numeric one was added to it. Using ReduceByKey Marks and numeric 1 was added for each key(name&grade). Using mapValues the two values in the key section were divided to print the average per Key

3. What is the average score of students in each subject across all grades?

- A pair RDD was created in the form (name,Subject),marks. Numeric one was added to it. Using ReduceByKey Marks and numeric 1 was added for each key(name&Subject). Using mapValues the two values in the key section were divided to print the average per Key

4. What is the average score of students in each subject per grade?

- A pair RDD was created in the form (Subject, grade),marks. Numeric one was added to it. Using ReduceByKey Marks and numeric 1 was added for each key(Subject&grade). Using mapValues the two values in the key section were divided to print the average per Key

5. For all students in grade-2, how many have average score greater than 50?

- A pair RDD was created in the form (name,grade),marks. Numeric one was added to it. Using ReduceByKey Marks and numeric 1 was added for each key(name&grade). Using mapValues the two values in the key section were divided to print the average per Key. Once the average is calculated, data was filtered by grade-2 and average score > 50. Both count and the entries are printed.

```scala
object vjsprk3 {
  def main(args: Array[String]) = {
    val conf = new SparkConf()
      .setAppName("vjsprk3")
      .setMaster("local")
    val sc = new SparkContext(conf)
    val test = sc.textFile("inp1.txt")

    val t1 = test.map{ line => line.split(",")}
    val t2 = t1.map(line=>(line(2),1))
    val t3 = t2.reduceByKey(_+_)
    println("Count of students per grade in school:"+"\n"+t3.collect.mkString("\n"))

    val t4a = t1.map(line=>((line(0),line(2)),line(3).toInt))
    val t4b = t4a.mapValues(x=>(x,1))
    val t4c = t4b.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
    val final1 = t4c.mapValues{case(sum,count)=>(1.0*sum)/count}.foreach(println)

    val g4a = t1.map(line=>((line(0),line(1)),line(3).toInt))
    val g4b = g4a.mapValues(x=>(x,1))
    val g4c = g4b.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
    val finalg = g4c.mapValues{case(sum,count)=>(1.0*sum)/count}.foreach(println)

    val l4a = t1.map(line=>((line(1),line(2)),line(3).toInt))
    val l4b = l4a.mapValues(x=>(x,1))
    val l4c = l4b.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
    val finall = l4c.mapValues{case(sum,count)=>(1.0*sum)/count}.foreach(println)

    val t4a1 = t1.map(line=>((line(0),line(2)),line(3).toInt))
    val t4b1 = t4a1.mapValues(x=>(x,1))
    val t4c1 = t4b1.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
    val final11 = t4c1.mapValues{case(sum,count)=>(1.0*sum)/count}
    val final22 = final11.filter(x=>x._1._2=="grade-2"&&x._2>50).count()
    val final33 = final11.filter(x=>x._1._2=="grade-2"&&x._2>50).foreach(println)

    sc.stop()
```

Output

```
19/02/09 14:20:36 INFO DAGScheduler: Job 0 finished: collect at vjsprk3.scala:17, took 0.780725 s
Count of students per grade in school:
(grade-3,4)
(grade-1,9)
(grade-2,9)
19/02/09 14:20:36 INFO SparkContext: Starting job: foreach at vjsprk3.scala:22
```

Find the average of each student (Note - Mathew is grade-1, is different from Mathew in some other grade!)

```
19/02/09 14:20:36 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
((Lisa,grade-1),24.0)
((Mark,grade-2),17.5)
((Lisa,grade-2),61.0)
((Mathew,grade-3),45.0)
((Andrew,grade-2),77.0)
((Andrew,grade-1),43.666666666666664)
((Lisa,grade-3),86.0)
((John,grade-1),38.666666666666664)
((John,grade-2),74.0)
((Mark,grade-1),84.0)
((Andrew,grade-3),35.0)
((Mathew,grade-2),65.66666666666667)
19/02/09 14:20:36 INFO Executor: Finished task 0.0 in stage 3.0 (TID 3). 1138 bytes result sent to driver
```

3. What is the average score of students in each subject across all grades?

```
19/02/09 14:20:37 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 1 ms
((Lisa,history),92.0)
((Mark,maths),57.5)
((Andrew,science),35.0)
((Mark,science),44.0)
((Mathew,science),50.0)
((Andrew,maths),28.5)
((Mathew,history),71.0)
((John,maths),54.5)
((John,history),40.5)
((Lisa,science),24.0)
((Andrew,history),75.5)
19/02/09 14:20:37 INFO Executor: Finished task 0.0 in stage 5.0 (TID 5). 1052 bytes result sent to driver
```

4. What is the average score of students in each subject per grade?

```
19/02/09 14:20:37 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 2 ms
((history,grade-2),79.25)
((history,grade-3),86.0)
((maths,grade-1),46.0)
((science,grade-3),38.333333333333336)
((science,grade-1),50.0)
((science,grade-2),30.333333333333332)
((history,grade-1),51.666666666666664)
((maths,grade-2),48.5)
19/02/09 14:20:37 INFO Executor: Finished task 0.0 in stage 7.0 (TID 7). 1052 bytes result sent to driver
```

5. For all students in grade-2, how many have average score greater than 50?

```
19/02/09 14:20:37 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
((Lisa,grade-2),61.0)
((Andrew,grade-2),77.0)
((John,grade-2),74.0)
((Mathew,grade-2),65.66666666666667)
19/02/09 14:20:37 INFO Executor: Finished task 0.0 in stage 11.0 (TID 10). 1095 bytes result sent to driver
```

Problem Statement 3:

Are there any students in the college that satisfy the below criteria:

1. Average score per student_name across all grades is same as average score per

Student_name per grade

Hint - Use Intersection Property

- Two averages were calculated
    - A pair RDD was created in the form (Name, Marks). Numeric one was added to it. Using ReduceByKey Marks & numeric 1 was added for each key(Name). Using mapValues the two values in the key section were divided to print the average per Key
    - A pair RDD was created in the form (Name, Grade),Marks. Numeric one was added to it. Using ReduceByKey Marks & numeric 1 was added for each key(Name&Grade). Using mapValues the two values in the key section were divided to print the average per Key
    - Using map the output of second step was converted to the format name, average (in Double format)
    - Using map the output of first step was converted to name, average (in Double format)
    - Using Intersection, the common values acrosss both first and second RDDs were identified
    - Since there are no similarities, there was no output from intersect.

```
object vjsprk4 {
  def main(args: Array[String]) = {
    val conf = new SparkConf()
      .setAppName("vjsprk4")
      .setMaster("local")
    val sc = new SparkContext(conf)
    val test = sc.textFile("inp1.txt")

    val t1 = test.map{ line => line.split(",")}
    val t4a = t1.map(line=>((line(0),line(3).toInt)))
    val t4b = t4a.mapValues(x=>(x,1))
    val t4c = t4b.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
    val final1 = t4c.mapValues{case(sum,count)=>(1.0*sum)/count}
    final1.foreach(println)


    val t4a1 = t1.map(line=>((line(0),line(2)),line(3).toInt))
    val t4b1 = t4a1.mapValues(x=>(x,1))
    val t4c1 = t4b1.reduceByKey((x,y)=>(x._1+y._1,x._2+y._2))
    val final11 = t4c1.mapValues{case(sum,count)=>(1.0*sum)/count}
    val final22 = final11.map(x=>x._1._1+","+x._2.toDouble)
    final22.foreach(println)

    val final23 = final1.map(x=>x._1+","+x._2)
    final23.foreach(println)

    val inters = final22.intersection(final23)

    println("intersection output:")
    inters.foreach(println)


    sc.stop()
```

Output

Output of average score per Student across all grades

```
19/02/09 14:45:48 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 8 ms
(Mark,50.75)
(Andrew,46.333333333333336)
(Mathew,60.5)
(John,47.5)
(Lisa,58.0)
19/02/09 14:45:48 INFO Executor: Finished task 0.0 in stage 1.0 (TID 1). 1095 bytes result sent to driver
```

Output of average score per student per grade (converted output to remove the grade field)

```
19/02/09 14:45:48 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
Lisa,24.0
Mark,17.5
Lisa,61.0
Mathew,45.0
Andrew,77.0
Andrew,43.666666666666664
Lisa,86.0
John,38.666666666666664
John,74.0
Mark,84.0
Andrew,35.0
Mathew,65.66666666666667
19/02/09 14:45:48 INFO Executor: Finished task 0.0 in stage 3.0 (TID 3). 1095 bytes result sent to driver
```

Intersect output

```
19/02/09 14:45:48 INFO DAGScheduler: Job 2 finished: foreach at vjsprk4.scala:30, took 0.037586 s
intersection output:
19/02/09 14:45:48 INFO SparkContext: Starting job: foreach at vjsprk4.scala:35
19/02/09 14:45:48 INFO DAGScheduler: Registering RDD 14 (intersection at vjsprk4.scala:32)
19/02/09 14:45:48 INFO DAGScheduler: Registering RDD 13 (intersection at vjsprk4.scala:32)
```