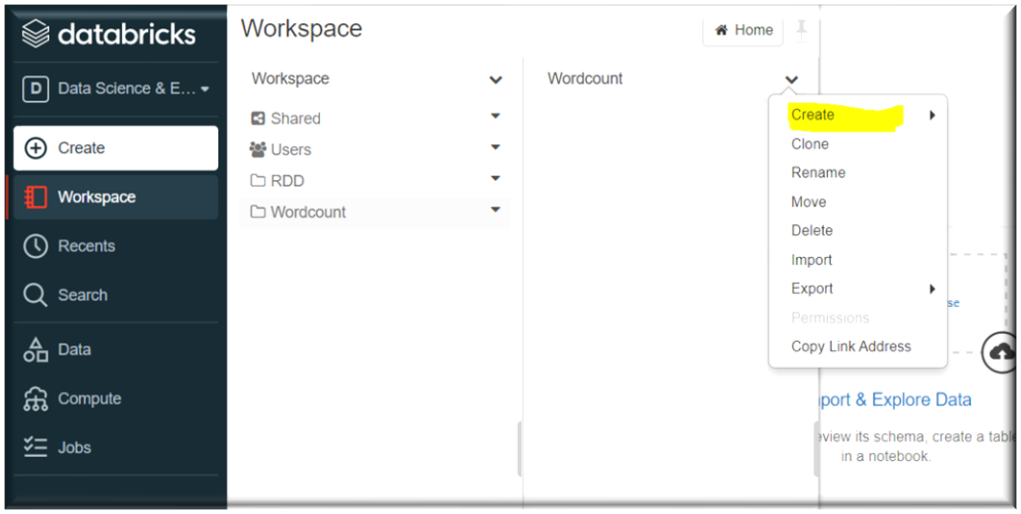
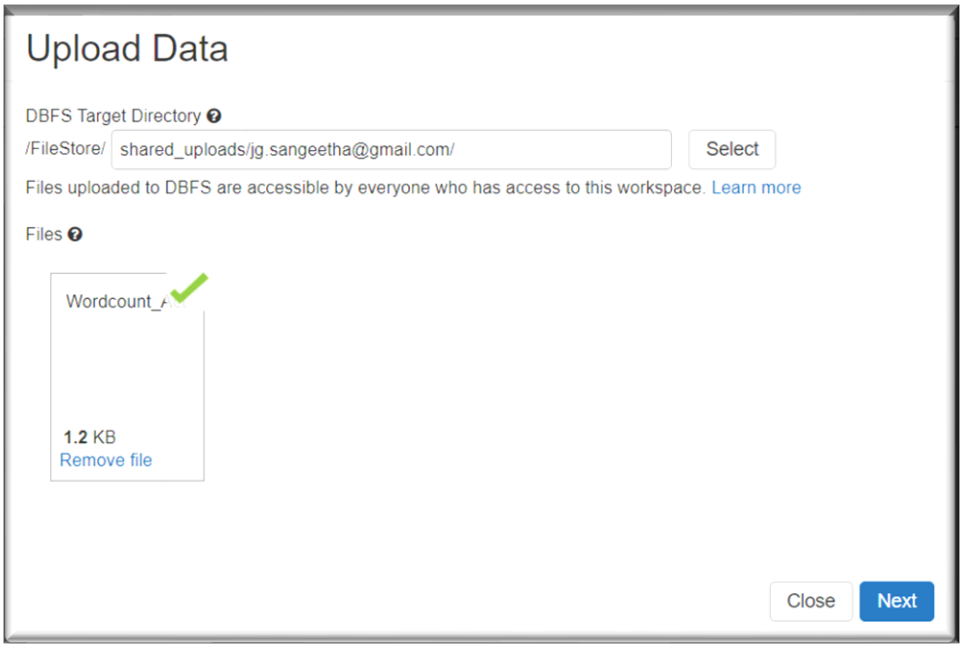
Spark RDD Example-Activity 2

1)Create Scala notebook in Wordcount folder



2) Upload Wordcount_Activity2 text file and copy the file path



3)Display 600 bytes of the file

```
//Display first 600 bytes of the file
dbutils.fs.head("dbfs:/FileStore/shared_uploads/jg.sangeetha@gmail.com/Wordcount_Activity2.txt",600)
```

```
[Truncated to first 600 bytes]
res0: String =
George Washington

January 8, 1790
Fellow-Citizens of the Senate and House of Representatives:
I embrace with great satisfaction the opportunity which now presents itself of congratulating you on the present favor
able prospects of our public affairs. The recent accession of the important state of North Carolina to the Constitutio
n of the United States (of which official information has been received), the rising credit and respectability of our
country, the general and increasing good will toward the government of the Union, and the concord, peace, and plenty w
ith which we are blessed a
```

4)Read the file into spark context as an RDD of strings

```
val SouAddress=sc.textFile("dbfs:/FileStore/shared_uploads/jg.sangeetha@gmail.com/Wordcount_Activity2.txt")
```

```
SouAddress: org.apache.spark.rdd.RDD[String] = dbfs:/FileStore/shared_uploads/jg.sangeetha@gmail.com/Wordcount_Activit
y2.txt MapPartitionsRDD[11] at textFile at command-3940489855021838:1
```

5) Perform some operations in RDD

Count-counts the number of elements in the RDD i.e., number of lines in the text file

take or collect- Displays the contents of RDD.

```
1   SouAddress.count()

▸ (1) Spark Jobs

res1: Long = 5
```

```
1   SouAddress.take(4).foreach(println)

▸ (1) Spark Jobs

George Washington

January 8, 1790
Fellow-Citizens of the Senate and House of Representatives:
```

```
1   SouAddress.collect

▸ (1) Spark Jobs

res6: Array[String] = Array("George Washington ", "", "January 8, 1790 ", "Fellow-Citizens of the Senate and House of
Representatives: ", I embrace with great satisfaction the opportunity which now presents itself of congratulating you
on the present favorable prospects of our public affairs. The recent accession of the important state of North Carolin
a to the Constitution of the United States (of which official information has been received), the rising credit and re
spectability of our country, the general and increasing good will toward the government of the Union, and the concord,
peace, and plenty with which we are blessed are circumstances auspicious in an eminent degree to our national prosperi
ty. In resuming your consultations for the general good you can not but derive encouragement from the reflection that
the measures of the last session have been as satisfactory to your constituents as the novelty and difficulty of the w
ork allowed you to hope. Still further to realize their expectations and to secure the blessings which a gracious Prov
idence has placed within our reach will in the course of the present important session call for the cool and deliberat
e exertion of your patriotism, firmness, and wisdom.)
```

6) Using **.cache()** in RDD, so that it will exist in memory after first action

```
SouAddress.cache()
```

```
res7: SouAddress.type = dbfs:/FileStore/shared_uploads/jg.sangeetha@gmail.com/Wordcount_Activity2.txt MapPartitionsRDD[11] at textFile at command-3940489855021838:1
```

7)Transforming lines to words

```
SouAddress.flatMap(line=>line.split(" ")).take(50)
```

```
res9: Array[String] = Array(George, Washington, "", January, 8,, 1790, Fellow-Citizens, of, the, Senate, and, House, of, Representatives:, I, embrace, with, great, satisfaction, th
e, opportunity, which, now, presents, itself, of, congratulating, you, on, the, present, favorable, prospects, of, our, public, affairs., The, recent, accession, of, the, importan
t, state, of, North, Carolina, to, the, Constitution)
```

8) Counting the words

flatMap→ Break each line by the white space character " " and find the words

Map→ word=>(word,1) to initialize each word with integer count 1 and transforming each word into (Key, value) pair

reduceByKey→Count all values with same key.

Collect→ to display the result.

```
SouAddress
.flatMap(line=>line.split(" "))
.map(word=>(word,1))
.reduceByKey(_+_)
.collect()
```

```
res10: Array[(String, Int)] = Array((patriotism,,1), (call,1), (satisfactory,1), (House,1), (favorable,1), (accession,1), (general,2), (Senate,1), (plenty,1), (have,1), (exertion,
1), (with,2), (session,2), (January,1), (national,1), (we,1), (States,1), (been,2), (eminent,1), (rising,1), (hope.,1), (satisfaction,1), (Representatives:,1), (from,1), (now,1),
(has,2), (affairs.,1), (realize,1), (further,1), (degree,1), (are,2), (received),,1), (8,,1), (Washington,1), (blessings,1), (presents,1), (congratulating,1), (can,1), (allowed,1),
(resuming,1), (their,1), (concord,,1), (country,,1), (last,1), (will,2), (Still,1), (our,4), (information,1), (as,2), ("",1), (important,2), (circumstances,1), (peace,,1), (respect
ability,1), (consultations,1), (blessed,1), (itself,1), (increasing,1), (Fellow-Citizens,1), (The,1), (novelty,1), (embrace,1), (auspicious,1), (secure,1), (on,1), (difficulty,1),
(wisdom.,1), (opportunity,1), (prosperity.,1), (George,1), (in,2), (which,4), (In,1), (good,2), (for,2), (derive,1), (great,1), ((of,1), (reflection,1), (placed,1), (present,2), (p
ublic,1), (Union,,1), (course,1), (your,3), (the,21), (government,1), (within,1), (cool,1), (North,1), (prospects,1), (not,1), (Providence,1), (you,3), (gracious,1), (that,1), (a,
1), (work,1), (state,1), (I,1), (to,6), (firmness,,1), (toward,1), (of,13), (credit,1), (expectations,1), (reach,1), (Constitution,1), (an,1), (1790,1), (but,1), (and,9), (officia
l,1), (constituents,1), (deliberate,1), (encouragement,1), (United,1), (recent,1), (measures,1), (Carolina,1))
```

Some words have punctuation marks in the end which means same words are counted as different words. We can use regular expression to remove those marks.

```
val SouAddress_Wordcount=SouAddress
.flatMap(line =>line.replaceAll("\\s+", " ")
        //replace multiple whitespace characters (including space, tab, new line, etc.) with one whitespace " "
                .replaceAll("""([,?.!:;])""", "")
        // replace the following punctions characters: , ? . ! : ; . with the empty string ""
                .toLowerCase()
        // converting to lower-case
                .split(" "))
 .map(x => (x, 1))
 .reduceByKey(_+_)
 .collect()
```

```
▸ (1) Spark Jobs
SouAddress_Wordcount: Array[(String, Int)] = Array((call,1), (satisfactory,1), (country,1), (states,1), (favorable,1), (accession,1), (general,2), (plenty,1), (have,1), (exertion,
1), (with,2), (january,1), (still,1), (washington,1), (we,1), (house,1), (been,2), (session,2), (national,1), (eminent,1), (rising,1), (satisfaction,1), (patriotism,1), (from,1),
(now,1), (has,2), (realize,1), (further,1), (degree,1), (affairs,1), (8,1), (are,2), (blessings,1), (presents,1), (congratulating,1), (hope,1), (can,1), (allowed,1), (resuming,1),
(their,1), (peace,1), (last,1), (will,2), (information,1), (our,4), (as,2), ("",1), (important,2), (circumstances,1), (respectability,1), (senate,1), (representatives,1), (concord,
1), (received),1), (consultations,1), (blessed,1), (itself,1), (increasing,1), (constitution,1), (novelty,1), (embrace,1), (auspicious,1), (secure,1), (on,1), (carolina,1), (i,1),
(difficulty,1), (opportunity,1), (george,1), (in,3), (north,1), (which,4), (united,1), (good,2), (for,2), (fellow-citizens,1), (derive,1), (great,1), ((of,1), (union,1), (reflectio
n,1), (placed,1), (present,2), (public,1), (course,1), (your,3), (the,22), (government,1), (within,1), (cool,1), (prospects,1), (not,1), (you,3), (gracious,1), (that,1), (a,1), (pr
ovidence,1), (work,1), (state,1), (to,6), (toward,1), (of,13), (wisdom,1), (credit,1), (expectations,1), (reach,1), (an,1), (firmness,1), (1790,1), (but,1), (and,9), (official,1),
(constituents,1), (deliberate,1), (encouragement,1), (recent,1), (measures,1), (prosperity,1))
```

9) Sorting in descending order

```
val top10 = SouAddress_Wordcount.sortBy(_._2).reverse.take(10)
```

```
top10: Array[(String, Int)] = Array((the,22), (of,13), (and,9), (to,6), (which,4), (our,4), (you,3), (your,3), (in,3), (present,2))
```