Programming Assignment: Simple Join in Spark

You have not submitted. You must earn 100/100 points to pass.

Deadline Pass this assignment by February 10, 11:59 PM PST

Instructions

My submission

Discussions

Make sure first you were able to complete the "Setup PySpark on the Cloudera VM" tutorial in lesson 1 of this module.

In this programming assignment we will implement in Spark the same code in the programming assignment in module 4 lesson 2 to perform a join of 2 different wordcount datasets.

Load datasets

First of all open the pyspark shell and load the datasets you created for the previous assignment from HDFS:

```
1  fileA = sc.textFile("input/join1_FileA.txt")
```

Let's make sure the file content is correct:

```
1  fileA.collect()
```

should return:

```
1  Out[]: [u'able,991', u'about,11', u'burger,15', u'actor,22']
2
```

Then load the second dataset:

```
fileB = sc.textFile("input/join1_FileB.txt")
```

same verification:

```
fileB.collect()
Out[29]:
[u'Jan-01 able,5',
 u'Feb-02 about,3',
 u'Mar-03 about,8 ',
 u'Apr-04 able,13',
 u'Feb-22 actor,3',
 u'Feb-23 burger,5',
 u'Mar-08 burger,2',
 u'Dec-15 able,100']
```

Mapper for fileA

First you need to create a map function for fileA that takes a line, splits it on the comma and turns the count to an integer.

You need to copy paste the following function into the pyspark console, then edit the 2 <ENTER_CODE_HERE> lines to perform the necessary operations:

```
def split_fileA(line):
    # split the input line in word and count on the comma
    <ENTER_CODE_HERE>
    # turn the count to an integer
    <ENTER_CODE_HERE>
    return (word, count)
```

You can test your function by defining a test variable:

```
test_line = "able,991"

```

and make sure that:

```
split_fileA(test_line)
```

returns:

```
Out[]: ('able', 991)
```

Now we can proceed on running the map transformation to the fileA RDD:

```
fileA_data = fileA.map(split_fileA)
```

If the mapper is implemented correctly, you should get this result:

```
fileA_data.collect()
Out[]: [(u'able', 991), (u'about', 11), (u'burger', 15), (u'actor', 22)]
```

Make sure that the key of each pair is a string (i.e. is delimited by ' ') and the value is an integer.

Mapper for fileB

The mapper for fileB is more complex because we need to extract

```
def split_fileB(line):
    # split the input line into word, date and count_string
    <ENTER_CODE_HERE>
    <ENTER_CODE_HERE>
    return (word, date + " " + count_string)
```

running:

```
fileB_data = fileB.map(split_fileB)
```

and then gathering the output back to the pyspark Driver console:

```
fileB_data.collect()
```

should give the result:

```
Out[]:
[(u'able', u'Jan-01 5'),
 (u'about', u'Feb-02 3'),
 (u'about', u'Mar-03 8 '),
 (u'able', u'Apr-04 13'),
 (u'actor', u'Feb-22 3'),
 (u'burger', u'Feb-23 5'),
 (u'burger', u'Mar-08 2'),
 (u'able', u'Dec-15 100')]
```

Run join

The goal is to join the two datasets using the words as keys and print for each word the wordcount for a specific date and then the total output from A.

Basically for each word in fileB, we would like to print the date and count from fileB but also the total count from fileA.

Spark implements the join transformation that given a RDD of (K, V) pairs to be joined with another RDD of (K, W) pairs, returns a dataset that contains (K, (V, W)) pairs.

```
fileB_joined_fileA = fileB_data.join(fileA_data)
```

Verify the result

You can inspect the full result with:

```
fileB_joined_fileA.collect()
```

You should try to make sure that this results agrees with what you were expecting.

Submit one line for grading

Finally, you need to create a text file with just one line for submission.

From the Cloudera VM, open the text editor from Applications > Accessories > gedit Text Editor.

Paste 1 single line of code from your pyspark console, the line related to the word actor:

```
1  (u'actor', ??????????)
```

do NOT copy the comma at the end, the line should start with open parenthesis (and end with closed parenthesis).

In gedit, click on the Save button and save it in the default folder (/home/cloudera) with the name spark_join1.txt

Open now the browser within the Cloudera VM, login to coursera, and upload that file for grading.

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.