CSP554—Big Data Technologies

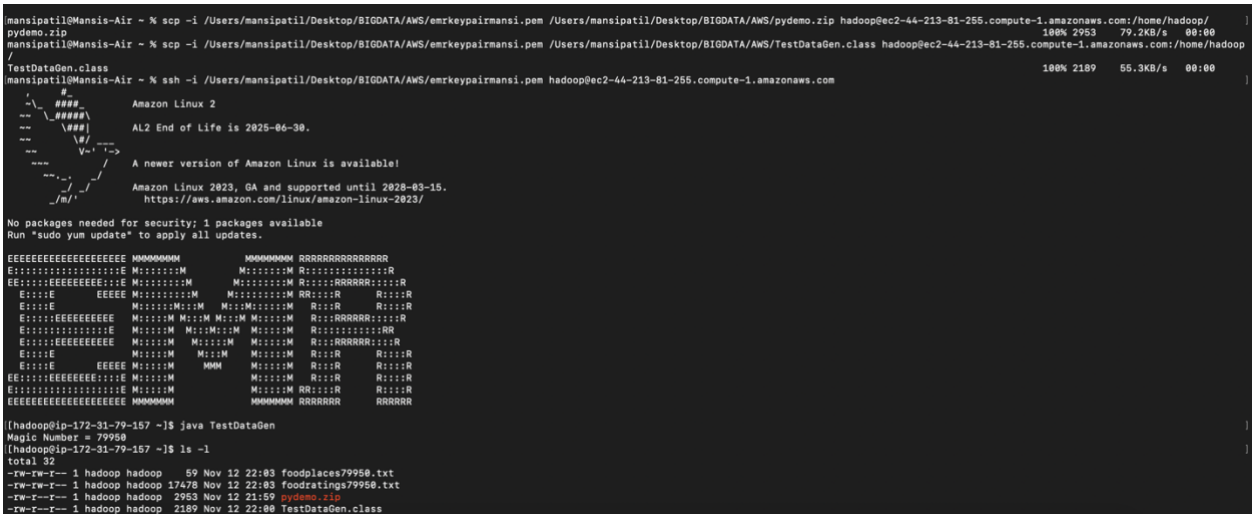
Assignment #06 (Modules 06)

Exercises

Exercise 1)

Use the TestDataGen program from previous assignments to generate a new foodratings<magic_number>.txt data file.

Magic Number Generation –



MAGIC NUMBER: 79950

Copy the file to HDFS, say into the /user/hadoop directory.

Read in the text file into an RDD named ex1RDD.

This RDD should now have records each consisting of a single string having 6 comma-separated parts something like the following:

u'Joe,44,33,41,1,5' u'Mel,13,33,30,50,6'

u'Mel,12,40,30,42,1'

u'Sam,15,28,28,39,2'

List the first five records of the RDD using the “take(5)” action and copy them and the “magic number to your assignment submission for this exercise.

ANS:

MAGIC NUMBER: 79950

Command –
ex1RDD = sc.textFile('/user/hadoop/foodratings79950.txt');
ex1RDD.take(5);

OUTPUT:

```
[[hadoop@ip-172-31-79-157 ~]$ pyspark
Python 3.7.16 (default, Aug 30 2023, 20:37:53)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-15)] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/11/12 22:08:01 WARN HiveConf: HiveConf of name hive.server2.thrift.url does not exist
23/11/12 22:08:05 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
23/11/12 22:08:31 WARN YarnSchedulerBackend$YarnSchedulerEndpoint: Attempted to request executors before the AM has registered!
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.4.1-amzn-1
      /_/

Using Python version 3.7.16 (default, Aug 30 2023 20:37:53)
Spark context Web UI available at http://ip-172-31-79-157.ec2.internal:4040
Spark context available as 'sc' (master = yarn, app id = application_1699826147264_0001).
SparkSession available as 'spark'.
```

```
>>> ex1RDD = sc.textFile('/user/hadoop/foodratings79950.txt');
>>> ex1RDD.take(5);

['Jill,36,23,24,21,2', 'Joy,10,41,30,41,3', 'Jill,41,36,42,44,4', 'Joy,33,32,41,12,1', 'Sam,32,33,2,22,2']
>>>
```

Exercise 2)

Create another RDD called ex2RDD where each record of this new RDD has 6 fields, each a string, by splitting apart each record on “,” boundaries from the ex1RDD.

The records of the new RDD should look something like:

u'Joe', u'44', u'33', u'41', u'1‘, u’5’ u‘Mel',

u'13', u'33', u'30', u'50, u’6’‘ u‘Mel',

u'12', u'40', u'30', u'42‘, u’1’ u'Sam',

u'15', u'28', u'28', u'39‘, u’3’

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

ANS:

Commands –
ex2RDD = ex1RDD.map(lambda line: line.split(","));
ex2RDD.take(5);

OUTPUT:

Exercise 3)

Create another RDD called ex3RDD from ex2RDD where each record of this new RDD has its third column converted from a string to an integer.

The records of the new RDD should look something like:

u'Joe', u'44', 33, u'41', u'1', u'1' u'Mel',

u'13', 33, u'30', u'50', u'2' u'Mel',

u'12', 40, u'30', u'42', u'3' u'Sam',

u'15', 28, u'28', u'39', u'4'

Hint: Use a lambda function something like the following:

lambda line : [line[0], line[1], int(line[2]), line[3], line[4], line[5]]

List the first five records of this RDD using the "take(5)" action and copy them to your assignment submission for this exercise.

ANS:

Commands –
ex3RDD = ex2RDD.map(lambda line: [line[0], line[1], int(line[2]), line[3], line[4], line[5]]);
ex3RDD.take(5);

OUTPUT:

Exercise 4)

Create another RDD called ex4RDD from ex3RDD where each record of this new RDD is allowed to have a value for its third field that is less than 25 (<25).

The records of the new RDD should look something like:

u'Joe', u'44', 21, u'41', u'1', u'6' u'Mel',

u'13', 3, u'30', u'50', u'1' u'Mel', u'12',

4, u'30', u'42‘, u’4’ u'Sam', u'15', 8,

u'28', u'39‘, u’5’

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

ANS:

Commands –
ex4RDD = ex3RDD.filter(lambda line: line[2] < 25)
ex4RDD.take(5)

OUTPUT:

```
>>> ex4RDD = ex3RDD.filter(lambda line: line[2] < 25)
>>> ex4RDD.take(5)
[['Jill', '36', 23, '24', '21', '2'], ['Mel', '30', 16, '46', '3', '1'], ['Jill', '15', 7, '11', '39', '4'], ['Mel', '12', 10, '23', '38', '4'], ['Joe', '3', 17, '37', '45', '2']]
```

Exercise 5)

Create another RDD called ex5RDD from ex4RDD where each record is a key value pair where the key is the first field of the record and the value is the entire record The records of the new RDD should look something like:

(u’Joe’, (u'Joe', u'44', 21, u'41', u'1‘, u’1’))

(u’Mel’, (u‘Mel', u'13', 3, u'30', u'50‘, u’6’))

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

ANS:

Commands –
ex5RDD = ex4RDD.map(lambda line: [line[0], line])
ex5RDD.take(5)

OUTPUT:

```
>>> ex5RDD = ex4RDD.map(lambda line: [line[0], line])
>>> ex5RDD.take(5)
[['Jill', ['Jill', '36', 23, '24', '21', '2']], ['Mel', ['Mel', '30', 16, '46', '3', '1']], ['Jill', ['Jill', '15', 7, '11', '39', '4']], ['Mel', ['Mel', '12', 10, '23', '38', '4']], ['Joe', ['Joe', '3', 17,
'37', '45', '2']]]
```

Exercise 6)

Create another RDD called ex6RDD from ex5RDD where the records are organized in ascending order by key .The records of the new RDD should look something like:

(u’Joe’, (u'Joe', u'44', 21, u'41', u'1‘, u’4’))

(u’Mel’ , (u‘Mel', u'13', 3, u'30', u'50‘, u’3’))

(u’Sam’ , (u‘Sam', u'23', 3, u'40', u'20‘, u’7’))

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

ANS:

Commands –
ex6RDD = ex5RDD.sortByKey(True)
ex6RDD.take(5)

OUTPUT: