Ian Pope 700717419 Big Data Analytics ICP 8

Ian Pope 700717419 Big Data Analytics ICP 8

```
[3] #pip install pyspark
```

Create spark context

```
[4] from pyspark import SparkContext
    sc = SparkContext()
```

1. Create RDD with first 15 natural numbers
2. Show RDD and the number of partitions

```
[5] nums = sc.parallelize([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])

    print(nums.collect())
    print(nums.getNumPartitions())
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
2
```

3. Get first element in the nums RDD

```
[6] #Gets the first element of nums
    print(nums.first())
```

```
1
```

4. Create new RDD with all entries of nums that is even

```
[7] #Gets a new RDD that contains all entries from nums that is even
    new_nums = nums.filter(lambda x: x % 2 == 0)
    print(new_nums.collect())
```

```
[2, 4, 6, 8, 10, 12, 14]
```

5. Apply transformation on RDD to create new RDD with each element being the square of the original

```
[8] #Apply map transformation to each element in the RDD and returns a new RDD with square of each element as an output.
    square = nums.map(lambda x: x * x)
    print(square.collect())
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225]
```

6. Aggregates the elements of the RDD into the sum

```
[9] #aggregates all elements in the RDD using reduce action
    sum = nums.reduce(lambda x, y: x + y)
    print(sum)
```

```
120
```

7. Save the RDD as a text file

```
[10] #saves the RDD data as a text file
     nums.saveAsTextFile('nums.txt')
```

nums.txt

8. Takes two RDDs and performs a union

```
[11] #take two new list RDDs and Combine them with union transformation
     nums1 = sc.parallelize([1,2,3,4,5,6,7,8,9,10])
     nums2 = sc.parallelize([11,12,13,14,15,16,17,18,19,20])
     union = nums1.union(nums2)
     print(union.collect())
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

9. Get the cartesian results of two RDDs nums1 and nums2

```
#Use cartesian transformation on defined list RDDs that returns a new list of ordered pairs
cartesian = nums1.cartesian(nums2)
print(cartesian.collect())
```

```
[(1, 11), (1, 12), (1, 13), (1, 14), (1, 15), (2, 11), (2, 12), (2, 13), (2, 14), (2, 15), (3, 11
```

10. Create an RDD from a dictionary

```
[13] #Create an RDD with Dictionary
     rdd = sc.parallelize([{"name": "Alice", "age": 25}, {"name": "Bob", "age": 30}, {"name": "Charlie
     print(rdd.collect())
```

```
[{'name': 'Alice', 'age': 25}, {'name': 'Bob', 'age': 30}, {'name': 'Charlie', 'age': 35}]
```

11. Get the the values in the RDD and their respective counts

```
#Get unique value in nums as the key and its count as the value
new_nums = sc.parallelize([1,2,1,1,4,6,7,2,3,4,5,2,2,4,1])
unique = new_nums.map(lambda x: (x, 1)).reduceByKey(lambda x, y: x + y)
print(unique.collect())
```

```
[(2, 4), (4, 3), (6, 1), (1, 4), (7, 1), (3, 1), (5, 1)]
```

+ Code + Text

12. Creates an RDD from text files

```
[15] #Create RDD by combining multiple .text files
     new_nums.saveAsTextFile('new_nums.txt')
     texts = sc.textFile("*.txt")
     print(texts.collect())
```

```
['2', '3', '4', '5', '2', '2', '4', '1', '1', '2', '1', '1', '4', '6', '7', '8', '9', '10', '11', '12', '13', '14',
```

new_nums.txt

nums.txt

13. Get the first 5 lines from the rdd

```
[16] #Inspect the First 5 Lines of an RDD
     texts.take(5)
```

```
['2', '3', '4', '5', '2']
```

14. Create a dataframe with pyspark

```
from pyspark.sql import SparkSession
from pyspark.sql import Row

spark = SparkSession.builder.getOrCreate()
df = spark.createDataFrame([
    Row(a=1, b=2., c='string1'),
    Row(a=2, b=3., c='string2'),
    Row(a=4, b=5., c='string3')
])
print(df)
```

```
DataFrame[a: bigint, b: double, c: string]
```

15. Show the difference between an RDD, Dataframe, and Dataset with an example

```python
#Create a dataframe
df = spark.createDataFrame([("Alice", 1), ("Bob", 2), ("Charlie", 3), ("David", 4)], ["Name", "Value"])

#Make the dataframe into a table so that it works with sql
df.createOrReplaceTempView("people")

# Perform an SQL query on the DataFrame
df_sql_result = spark.sql("SELECT Name, Value FROM people WHERE Value > 2")
df_sql_result.show()

#Use the same data for a RDD
rdd = spark.sparkContext.parallelize([("Alice", 1), ("Bob", 2), ("Charlie", 3), ("David", 4)])

#Trying to run a sql query on an RDD
rdd.createOrReplaceTempView("people")
rdd_sql_result = spark.sql("SELECT Name, Value FROM people WHERE Value > 2")
rdd_sql_result.show()
```

```
+-------+-----+
|   Name|Value|
+-------+-----+
|Charlie|    3|
|  David|    4|
+-------+-----+

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-19-543e86da27a5> in <cell line: 15>()
     13
     14 #Trying to run a sql query on an RDD
---> 15 rdd.createOrReplaceTempView("people")
     16 rdd_sql_result = spark.sql("SELECT Name, Value FROM people WHERE Value > 2")
     17 rdd_sql_result.show()

AttributeError: 'RDD' object has no attribute 'createOrReplaceTempView'
```