5e_transformations_sample

February 21, 2022

1 Transformation - sample

- “sample” transformation helps us in taking samples instead of working on full data.
- The sample method will return a new RDD, containing a statistical sample of the original RDD.
- We can pass the arguments insights as the sample operation:
    - “withReplacement = True” or False (to choose the sample with or without replacement)
    - “fraction = x” (x= .4 means we want to choose 40% of data in “rdd”)
    - “seed” for reproduce the results.

[1]:

```
# Import SparkContext and SparkConf
from pyspark import SparkContext, SparkConf

# Initialize spark
conf = SparkConf().setAppName("mapPartitions")
sc = SparkContext(conf=conf)
```

```
22/02/21 12:15:39 WARN Utils: Your hostname, ThinkCentre resolves to a loopback
address: 127.0.1.1; using 10.180.5.223 instead (on interface eno1)
22/02/21 12:15:39 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
22/02/21 12:15:40 WARN NativeCodeLoader: Unable to load native-hadoop library
for your platform… using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
22/02/21 12:15:41 WARN Utils: Service 'SparkUI' could not bind on port 4040.
Attempting port 4041.
22/02/21 12:15:41 WARN Utils: Service 'SparkUI' could not bind on port 4041.
Attempting port 4042.
22/02/21 12:15:41 WARN Utils: Service 'SparkUI' could not bind on port 4042.
Attempting port 4043.
22/02/21 12:15:41 WARN Utils: Service 'SparkUI' could not bind on port 4043.
Attempting port 4044.
22/02/21 12:15:41 WARN Utils: Service 'SparkUI' could not bind on port 4044.
Attempting port 4045.
```

[2]:

```
poemRDD = sc.textFile("5b_mydependence.txt")
poemRDD.take(6)
```

[2]:

```
['I like to be dependent, and so for ever',
 'with warmth and care of my mother',
 'my father , to love, kiss and embrace',
 'wear life happily in all their grace.',
 '',
 'I like to be dependent, and so for ever']
```

[3]:

```
# Question1: Convert all words in a rdd to lowercase and
#            split the lines of a document using space.

# Use flatmap to get a single list of words

def convert(lines):
    lines = lines.lower() # Convert all the words to lower case
    lines = lines.split() # Split each line into words
    return lines

wordsRDD = poemRDD.flatMap(convert)
wordsRDD.take(10)
```

[3]:

```
['i', 'like', 'to', 'be', 'dependent,', 'and', 'so', 'for', 'ever', 'with']
```

[4]:

```
# sample is based on RDD size, not the number of elements in RDD
sampleRDD = wordsRDD.sample(False, 0.2, 444) # 444 is seed
print(len(wordsRDD.collect()),len(sampleRDD.collect()))
```

```
125 23
```