quiz6

CS 119

4/11/2022

Neal Zhao

Moving Averages

1

```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from pyspark import StorageLevel
import pandas as pd
import numpy as np
sc.stop()
def launch(n):
    ssc.start()
    ssc.awaitTermination(n)

sc = SparkContext('local[2]', 'NetworkWordCount')
ssc = StreamingContext(sc,1)
lines = ssc.socketTextStream('localhost', 9999,
StorageLevel.MEMORY_AND_DISK)
```

2

```
# q2
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from pyspark import StorageLevel
import pandas as pd
import numpy as np

sc.stop()

def launch(n):
    ssc.start()
    ssc.awaitTermination(n)

def calculate_sum(rdd):
    l = rdd.collect()
    if len(l) == 80:
        total10 = 0
        for i in range(60, len(l)):
            if i%2 == 1:
                total10 += float(l[i])

        total40 = 0
        for i in range(len(l)):
            if i%2 == 1:
```

```
                total40 += float(l[i])
        return sc.parallelize([(total40, total10)])

def dj_count(rdd):
    return sc.parallelize([len(rdd.collect())])


sc = SparkContext('local[2]', 'NetworkWordCount')
ssc = StreamingContext(sc,1)
lines = ssc.socketTextStream('localhost', 9999, StorageLevel.MEMORY_AND_DISK)
words =  lines.window(40,1).flatMap(lambda line:line.split(' '))
dj30sum = words.transform(calculate_sum)
dj30count = words.transform(dj_count)


launch(10)
```

3

```
def window_both(rdd):
    l = rdd.collect()
    if len(l) == 80:
        total10 = 0
        for i in range(60, len(l)):
            if i%2 == 1:
                total10 += float(l[i])
        total40 = 0
        for i in range(len(l)):
            if i%2 == 1:
                total40 += float(l[i])
        return sc.parallelize([total40/40, total10/10])

words =  lines.window(40,1).flatMap(lambda line:line.split(' '))
maBoth = words.transform(window_both)
maBoth.foreachRDD(lambda x: print('40 MA: ', x.collect()[0], '10 MA: ', x.collect()[1]))

launch(10)
```

4

```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from pyspark import StorageLevel
import pandas as pd
import numpy as np

sc.stop()


global records
records = [0,0,0]

```

```
def window_both(rdd):
    l = rdd.collect()
    if len(l) == 80:
        total10 = 0
        for i in range(60, len(l)):
            if i%2 == 1:
                total10 += float(l[i])
        total40 = 0
        for i in range(len(l)):
            if i%2 == 1:
                total40 += float(l[i])
        return sc.parallelize([l[78], total40/40, total10/10])


def store_record_and_check(rdd):
    global records
    if rdd is None:
        return
    l = rdd.collect()
    last_record = records
    current_record = l.copy()
    records = current_record
    if last_record[1] > last_record[2]: #40 above 10
        if current_record[1] < current_record[2]:# 40 below 10, buy
            return sc.parallelize([current_record[0], 'BUY DJ30'])
    elif last_record[1] < last_record[2]: #40 below 10
        if current_record[1] > current_record[2]:# 40 above 10, sell
            return sc.parallelize([current_record[0], 'sell DJ30'])









def launch(n):
    ssc.start()
    ssc.awaitTermination(n)


sc = SparkContext('local[2]', 'NetworkWordCount')
ssc = StreamingContext(sc,1)
lines = ssc.socketTextStream('localhost', 9999,
StorageLevel.MEMORY_AND_DISK)
words =  lines.window(40,1).flatMap(lambda line:line.split(' '))
maBoth = words.transform(window_both)
result = maBoth.transform(store_record_and_check)
result.foreachRDD(lambda x: print(x.collect()[0], x.collect()[1]))

launch(10)
```

printed result:

(The stream is not ended)

```
3/8/90 BUY DJ30
```

```
4/26/90 sell DJ30
5/14/90 BUY DJ30
8/2/90 sell DJ30
12/31/90 BUY DJ30
1/10/91 sell DJ30
1/25/91 BUY DJ30
3/29/91 sell DJ30
4/5/91 BUY DJ30
4/10/91 sell DJ30
4/16/91 BUY DJ30
5/16/91 sell DJ30
5/31/91 BUY DJ30
6/28/91 sell DJ30
7/15/91 BUY DJ30
7/17/91 sell DJ30
7/19/91 BUY DJ30
8/22/91 sell DJ30
8/23/91 BUY DJ30
9/23/91 sell DJ30
9/24/91 BUY DJ30
10/7/91 sell DJ30
10/21/91 BUY DJ30
11/19/91 sell DJ30
12/27/91 BUY DJ30
3/11/92 sell DJ30
3/25/92 BUY DJ30
4/3/92 sell DJ30
4/15/92 BUY DJ30
6/17/92 sell DJ30
```

```
8/3/92 BUY DJ30

8/19/92 sell DJ30

9/21/92 BUY DJ30

9/28/92 sell DJ30

11/5/92 BUY DJ30

1/14/93 sell DJ30

2/3/93 BUY DJ30

4/14/93 sell DJ30

4/19/93 BUY DJ30

4/28/93 sell DJ30

5/11/93 BUY DJ30
```

Bloom Filter

1

```
data = open('headline_words.txt').read().split()
n = len(data)*8
import numpy as np
filter_bit = np.zeros(n, int)
for word in data:
    position = hash(word) % n
    filter_bit[position] = 1
np.savetxt('arr.txt',filter_bit, fmt="%d", newline = ' ' )
```

2

```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from pyspark import StorageLevel
import pandas as pd
import numpy as np
import re

global word_l
word_l = np.array(open('bloom.txt').read().split(), int)
# print(word_l)

# takes in a RDD containning a line,
  # print it if it has unfamiliar words
def foo(rdd):
    global word_l # the bit array of familiar words
    line = rdd.collect()
```

```
    if len(line) == 0:
        return
    text = line[0]#get the string
    text = text.lower()
    text = re.sub("","",text)
    text = re.sub(r'[0-9\,$]+', '', text)
    text=re.sub("(\\d|\\W)+"," ",text)
    text = ' '.join([word for word in text.split() if len(word) > 3])
    words = text.split(' ')
    for word in words:
        pos = hash(word) % len(word_l)
        if word_l[pos] == 1:
            print(line[0])
            break
    return



def launch(n):
    ssc.start()
    ssc.awaitTermination(n)


sc.stop()
sc = SparkContext('local[2]', 'NetworkWordCount')
ssc = StreamingContext(sc,1)
lines = ssc.socketTextStream('localhost', 9999,
StorageLevel.MEMORY_AND_DISK)
lines.foreachRDD(foo)

launch(10)
```

Link:

https://tufts.zoom.us/rec/share/sZcsDgB2wPbVN-gJqywexBPO8_o73e3Q-o2cFUbkemr8RReLNfXha1m_Uj2lMDuH.3wCDwftW4TpRQozj?startTime=1649819763000