spark

May 4, 2023

[11]:
```
import os
import pyspark
import findspark
findspark.init()
from pyspark.sql import SparkSession

def init_spark(app_name: str):
  spark = SparkSession.builder.appName(app_name).getOrCreate()
  sc = spark.sparkContext
  return spark, sc

spark, sc = init_spark('demo')
sc
```

[11]:
```
<SparkContext master=local[*] appName=demo>
```

[12]:
```
data_file = 'Lab3_view_data.csv'
view_rdd = sc.textFile(data_file)
print('Type:', type(view_rdd))
print('Count (rows):', view_rdd.count())
```

```
Type: <class 'pyspark.rdd.RDD'>
Count (rows): 1048576
```

[13]:
```
view_rdd.take(1)
```

[13]:
```
['mso_code ,device_id ,event_date ,event_time,station_num,prog_code']
```

[14]:
```
# Split by comma:
csv_rdd = view_rdd.map(lambda row: row.split(','))
csv_rdd.take(2)
```

[14]:
```
[['mso_code ',
  'device_id ',
  'event_date ',
  'event_time',
  'station_num',
  'prog_code'],
```

```
['1540', '0000000057f6', '20151101', '192440', '11590', 'null']]
```

[15]:
```
# Split to header and data:
header = csv_rdd.first()
print(header)
data_rdd = csv_rdd.filter(lambda row: row != header)
data_rdd.first()
```

```
['mso_code ', 'device_id ', 'event_date ', 'event_time', 'station_num',
'prog_code']
```

[15]:
```
['1540', '0000000057f6', '20151101', '192440', '11590', 'null']
```

[16]:
```
#extracting the Data from each row and counting the unique dates
date_rdd = data_rdd.map(lambda row: row[2])
number_of_days = date_rdd.distinct().count()
```

[17]:
```
prime_time_rdd = data_rdd.filter(lambda row: 200000<=int(row[3])<230000).
 ↪map(lambda row: ((row[1], row[2]), 1))
device_day_counts_rdd = prime_time_rdd.reduceByKey(lambda x, y: x+y).map(lambda␣
 ↪row: (row[0][0],row[0][1], row[1]))
top_5 = device_day_counts_rdd.map(lambda row: (row[0], row[2])).
 ↪reduceByKey(lambda x, y:x+y).map(lambda row: (row[0], row[1] /␣
 ↪number_of_days)).takeOrdered(5, key = lambda row: -row[1])
```

[19]:
```
for item in top_5:
    print(f'Device ID:{item[0]:} ********* Avg:{item[1]:.4}')
```

```
Device ID:7.5E+14 ********* Avg:97.64
Device ID:7.46E+14 ********* Avg:11.71
Device ID:7.503E+14 ********* Avg:9.357
Device ID:8.00001E+11 ********* Avg:7.5
Device ID:8.4843E+14 ********* Avg:5.5
```

[20]:
```
prime_time_rdd = data_rdd.filter(lambda row: 200000<=int(row[3])<230000).
 ↪map(lambda row: (row[1], 1))
avg_score = prime_time_rdd.reduceByKey(lambda x, y: x+y).map(lambda row:␣
 ↪(row[0], row[1]/number_of_days)).takeOrdered(5, key = lambda row: -row[1])
```

[21]:
```
for item in avg_score:
    print(f'Device ID:{item[0]:} ********* Avg:{item[1]:.4}')
```

```
Device ID:7.5E+14 ********* Avg:97.64
Device ID:7.46E+14 ********* Avg:11.71
Device ID:7.503E+14 ********* Avg:9.357
Device ID:8.00001E+11 ********* Avg:7.5
Device ID:8.4843E+14 ********* Avg:5.5
```

[21]:
```

```