lab1_spark_bigdata

zahra jalilpour(zahja96), Zhixuan_Duan(zhidu838)

2021-05-16

Contents

Lab1 - Spark - Exercises

1)

```
#exe1
from pyspark import SparkContext
sc = SparkContext(appName = "exe 1")
Temp=sc.textFile("BDA/input/temperature-readings.csv")
# Definition for calculating the maximum value
def max_temp(a,b):
    if a>=b:
        return a
    else:
        return b
# Definition for calculating the minimum value
def min_temp(a,b):
    if a<=b:
        return a
    else:
        return b

# Split features of the file separated by a ';'
lines = Temp.map(lambda line: line.split(";"))
# Map (year, temperature)
year_temperature = lines.map(lambda x: (x[1][0:4], float(x[3])))
# Filter the data
filter_year_temperature = year_temperature.filter(lambda x: int(x[0])>=1950 and int(x[0])<=2014)
# Find max and min temperature (each year)
max_temp_year = filter_year_temperature.reduceByKey(max_temp)
min_temp_year = filter_year_temperature.reduceByKey(min_temp)
# Sort the result by descending temperature
Max_temp_sorted = max_temp_year.sortBy(ascending = False, keyfunc=lambda k: k[1])
Min_temp_sorted = min_temp_year.sortBy(ascending = False, keyfunc=lambda k: k[1])

Max_temp_sorted.saveAsTextFile("BDA/output/max_temperature")
Min_temp_sorted.saveAsTextFile("BDA/output/min_temperature")
```

min temperature

[('1990', -35.0),

('1952', -35.5),

('1974', -35.6),

('1954', -36.0),

('1992', -36.1),

('1975', -37.0),

('1972', -37.5),

('2000', -37.6),

('1995', -37.6),

(‘1957’, -37.8),
(‘1983’, -38.2),
(‘1989’, -38.2),
(‘1953’, -38.4),
(‘2009’, -38.5),
(‘1993’, -39.0),
(‘1984’, -39.2),
(‘1991’, -39.3),
(‘1973’, -39.3),
(‘2008’, -39.3),
(‘2005’, -39.4),
(‘1961’, -39.5),
(‘1964’, -39.5),
(‘1970’, -39.6),
(‘2004’, -39.7),
(‘1988’, -39.9),
(‘1960’, -40.0),
(‘1997’, -40.2),
(‘1994’, -40.5),
(‘2006’, -40.6),
(‘2013’, -40.7),
(‘2007’, -40.7),
(‘1963’, -41.0),
(‘1955’, -41.2),
(‘2003’, -41.5),
(‘1969’, -41.5),
(‘1996’, -41.7),
(‘2010’, -41.7),
(‘1962’, -42.0),
(‘1951’, -42.0),
(‘1950’, -42.0),
(‘2011’, -42.0),
(‘1968’, -42.0),
(‘1982’, -42.2),
(‘2002’, -42.2),
(‘1976’, -42.2),

(‘2014’, -42.5),
(‘1977’, -42.5),
(‘1998’, -42.7),
(‘2012’, -42.7),
(‘1958’, -43.0),
(‘1985’, -43.4),
(‘1959’, -43.6),
(‘2001’, -44.0),
(‘1965’, -44.0),
(‘1981’, -44.0),
(‘1979’, -44.0),
(‘1986’, -44.2),
(‘1971’, -44.3),
(‘1956’, -45.0),
(‘1980’, -45.0),
(‘1967’, -45.4),
(‘1987’, -47.3),
(‘1978’, -47.7),
(‘1999’, -49.0),
(‘1966’, -49.4)]

max temperature

[(‘1975’, 36.1),
(‘1992’, 35.4),
(‘1994’, 34.7),
(‘2014’, 34.4),
(‘2010’, 34.4),
(‘1989’, 33.9),
(‘1982’, 33.8),
(‘1968’, 33.7),
(‘1966’, 33.5),
(‘1983’, 33.3),
(‘2002’, 33.3),
(‘1986’, 33.2),
(‘1970’, 33.2),
(‘1956’, 33.0),

('2000', 33.0),
('1959', 32.8),
('2006', 32.7),
('1991', 32.7),
('1988', 32.6),
('2011', 32.5),
('1999', 32.4),
('1955', 32.2),
('2003', 32.2),
('1953', 32.2),
('1973', 32.2),
('2008', 32.2),
('2007', 32.2),
('2005', 32.1),
('1979', 32.0),
('1969', 32.0),
('2001', 31.9),
('1997', 31.8),
('1977', 31.8),
('2013', 31.6),
('2009', 31.5),
('2012', 31.3),
('1972', 31.2),
('1971', 31.2),
('1964', 31.2),
('1976', 31.1),
('1961', 31.0),
('1963', 31.0),
('1996', 30.8),
('1995', 30.8),
('1978', 30.8),
('1958', 30.8),
('1974', 30.6),
('1954', 30.5),
('1952', 30.4),
('1980', 30.4),

('2004', 30.2),

('1990', 30.2),

('1985', 29.8),

('1957', 29.8),

('1981', 29.7),

('1993', 29.7),

('1987', 29.6),

('1984', 29.5),

('1967', 29.5),

('1960', 29.4),

('1950', 29.4),

('1998', 29.2),

('1965', 28.5),

('1951', 28.5),

('1962', 27.4)]

2)

```
#exe2
from pyspark import SparkContext
sc = SparkContext(appName = "exercise 2")
Temp=sc.textFile("BDA/input/temperature-readings.csv")
lines = Temp.map(lambda line: line.split(";"))
#map(station, year, month, temperature)
year_temperature = lines.map(lambda x: (x[0], x[1][0:4], x[1][5:7], float(x[3])))

#filter 1950<year<2014 and temperature>10
filter_year = year_temperature.filter(lambda x: int(x[1])>=1950 and int(x[1])<=2014 and float(x[3])>10)
# map((year, month),1 )
filter_month = filter_year.map(lambda x: (( int(x[1]) , int(x[2])), 1))
# map((station, year, month),1) and return distint elements
filter_month_distinct = filter_year.map(lambda x: ((int(x[0]), int(x[1]), int(x[2])), 1)).distinct()
# Add a 1 to the value of each data point which have a temperature above 10 (distinct)
filter_month_distinct = filter_month_distinct.map(lambda x: ((x[0][1], x[0][2]), 1))
# Sum over all 1:s to count all instances
count_filter_month = filter_month.reduceByKey(lambda a,b: a+b)
count_filter_month_distinct = filter_month_distinct.reduceByKey(lambda a,b: a+b)
count_filter_month.saveAsTextFile("BDA/output/month_count")
count_filter_month_distinct.saveAsTextFile("BDA/output/month_count_distinct")
```

conth_filter_month

[((1970, 8), 54566),

((1982, 4), 4172),

((1992, 6), 60683),
((1993, 5), 34908),
((1996, 10), 22811),
((1997, 9), 74472),
((1955, 7), 25046),
((1958, 12), 4),
((1956, 2), 3),
((1959, 11), 30),
((1983, 3), 23),
((1957, 1), 3),
((1967, 4), 3483),
((1978, 7), 60998),
((1979, 8), 56629),
((2004, 5), 47957),
((2005, 6), 90724),
((2006, 11), 4144),
((1952, 9), 5347),
((1953, 10), 5488),
((1966, 3), 42),
((2007, 12), 5),
((1967, 5), 22220),
((1978, 6), 55893),
((1979, 9), 33944),
((2004, 4), 14334),
((2005, 7), 125294),
((2006, 10), 43877),
((1952, 8), 12018),
((1977, 3), 154),
((1953, 11), 120),
((1971, 10), 13326),
((1975, 6), 48426),
((2009, 8), 128349),
((2012, 7), 137477),
((1986, 5), 29765),
((1998, 9), 76535),
((2013, 4), 7169),
((1984, 3), 1),
. . . ..

count__filter__month__distinct

[((1982, 4), 246),
((1996, 10), 301),
((1997, 9), 340),
((1970, 8), 370),
((1992, 6), 310),
((1993, 5), 292),
((1955, 7), 124),
((1959, 11), 19),
((1983, 3), 17),
((1958, 12), 4),
((1956, 2), 2),
((1957, 1), 2),
((1979, 8), 340),
((2004, 5), 321),
((1978, 7), 343),
((2005, 6), 311),
((1967, 4), 279),
((2006, 11), 145),
((1952, 9), 114),
((1953, 10), 114),
((1966, 3), 33),
((2007, 12), 3),
((1979, 9), 351),
((1967, 5), 363),
((2005, 7), 307),
((1977, 3), 99),
((1978, 6), 354),
((2004, 4), 267),
((1952, 8), 115),
((2006, 10), 276),
((1953, 11), 42),
((1998, 9), 326),
((1986, 5), 317),
((2012, 7), 310),
((1971, 10), 347),

((1975, 6), 368),
((2009, 8), 311),
((2013, 4), 208),
((2008, 11), 106),
((1984, 3), 1),
((1967, 7), 351),
((2003, 3), 140),
((2004, 6), 319),
((2005, 5), 302),
((2006, 8), 309),
((1978, 4), 241),
((1953, 9), 117),
((1979, 11), 21),
((1952, 10), 62),
((1976, 2), 17),
((1954, 12), 3),
((1984, 5), 333),
((1985, 6), 324),
((1999, 8), 327),
((1974, 7), 362),
((1972, 9), 375),
((1973, 10), 349),
((2011, 4), 289),
((2010, 11), 49),
((1986, 3), 14),
((1971, 12), 27),
((2013, 2), 6),
((1984, 4), 313),
((1985, 7), 304),
((1999, 9), 328),
((2011, 5), 315),
((1972, 8), 376),
((1974, 6), 372),
((2010, 10), 277),
((1973, 11), 116),
((2013, 3), 9),

((1975, 1), 10),

((1993, 4), 278),

((1997, 8), 337),

((1970, 9), 369),

((1992, 7), 311),

((1982, 5), 330),

((1955, 6), 125),

((1959, 10), 126),

((1996, 11), 179),

((1956, 3), 61),

((2012, 2), 60),

((1999, 11), 225),

((2011, 7), 319),

((1972, 10), 378),

((1973, 9), 376),

((1984, 6), 324),

((1974, 4), 344),

((1985, 5), 325),

((2010, 8), 318),

((1975, 3), 30),

....

4)

```
# exe4
from pyspark import SparkContext
sc = SparkContext(appName = "exe 4")
Temp=sc.textFile("BDA/input/temperature-readings.csv")
Precipitation=sc.textFile("BDA/input/precipitation-readings.csv")

# split features of precipitation
lines_precipitation = Precipitation.map(lambda line: line.split(";"))
# map(station, temperature)
temperature = lines.map(lambda x: (x[0], float(x[3])))
# map(station, precipitation)
precipitation = lines_precipitation.map(lambda x: (x[0], float(x[3])))
# Finding the max temp and max precipitation for each station
maximum_temp = temperature.reduceByKey(max)
maximum_precipitation = precipitation.reduceByKey(max)
# Create a value pair containing the max temperature and max precipitation in each station
joined_maximum = maximum_temp.join(maximum_precipitation)
# Filter 25<temp<30 and 100<precipitation<200
```

```
final_station = joined_maximum.filter(lambda x: float(x[1][0])>=25 and float(x[1][0])<=30 and float(x[1
#final_station.collect()
final_station.saveAsTextFile("BDA/output/final_station")
```

empty result

5)

```
#exe5
from pyspark import SparkContext
sc = SparkContext(appName = "exe 5")
Precipitation=sc.textFile("BDA/input/precipitation-readings.csv")
Station=sc.textFile("BDA/input/stations-Ostergotland.csv")
lines_precipitation = Precipitation.map(lambda line: line.split(";"))
lines_stations = Station.map(lambda line: line.split(";"))
prec1 = lines_precipitation.map(lambda x: (x[0], x[1][0:4], x[1][5:7], float(x[3])))
precipit = prec1.filter(lambda x: int(x[1])>=1993 and int(x[1])<=2016)
stat = lines_stations.map(lambda x: (int(x[0]))).collect()
stations_distributed = sc.broadcast(stat)
precipitation_province = precipit.filter(lambda a: int(a[0]) in stations_distributed.value)
monthly_precipitation = precipitation_province.map(lambda x: ((x[0], x[1], x[2]), x[3]))
monthly_precipitation_avg_station = monthly_precipitation.reduceByKey(lambda a,b: a+b)
monthly_precipitation_avg = monthly_precipitation_avg_station.map(lambda x: ((x[0][1], x[0][2]), (x[1],
monthly_precipitation_avg = monthly_precipitation_avg.reduceByKey(lambda a,b: (a[0]+b[0], a[1]+b[1]))
monthly_precipitation_avg = monthly_precipitation_avg.mapValues(lambda x: x[0]/x[1])
monthly_precipitation_avg.saveAsTextFile("BDA/output/averge_monthly_precipitation")
```

((u'2012', u'09'), 72.75)

((u'1995', u'05'), 26.00000000000002)

((u'2015', u'04'), 15.337499999999999)

((u'2007', u'04'), 21.249999999999996)

((u'2007', u'06'), 108.95)

((u'2011', u'06'), 88.35000000000001)

((u'2011', u'10'), 43.75000000000001)

((u'2014', u'10'), 72.13749999999999)

((u'1996', u'09'), 57.466666666666676)

((u'1995', u'07'), 43.6)

((u'2002', u'05'), 72.13333333333334)

((u'2010', u'04'), 23.783333333333335)

((u'1999', u'01'), 61.933333333333394)

((u'2013', u'11'), 46.37500000000002)

((u'2010', u'03'), 23.88333333333334)

((u'1999', u'10'), 18.549999999999997)

((u'2003', u'11'), 54.45000000000001)

((u'2014', u'04'), 31.762500000000006)

((u'2006', u'09'), 19.26666666666667)

((u'2016', u'02'), 21.5625)

((u'2013', u'09'), 26.18750000000001)

((u'2016', u'05'), 29.250000000000007)

((u'2015', u'01'), 59.112500000000026)

((u'2009', u'07'), 113.16666666666663)

((u'2008', u'05'), 23.133333333333336)

((u'1998', u'07'), 85.16666666666664)

((u'1996', u'12'), 39.55000000000003)

....