LINKÖPING UNIVERSITY

TDDE31 Big Data Analytics

Lab 1 - Report

Lawrence Thanakumar Rajappa (lawra776)

Miaoling Shi (miash554)

May 17, 2020

Task 1

Highest Temperature	Lowest Temperature
(u'1975', 36.1)	(u'1990', -35.0)
(u'1992', 35.4)	(u'1952', -35.5)
(u'1994', 34.7)	(u'1974', -35.6)
(u'2010', 34.4)	(u'1954', -36.0)
(u'2014', 34.4)	(u'1992', -36.1)
(u'1989', 33.9)	(u'1975', -37.0)
(u'1982', 33.8)	(u'1972', -37.5)
(u'1968', 33.7)	(u'1995', -37.6)
(u'1966', 33.5)	(u'2000', -37.6)
(u'2002', 33.3)	(u'1957', -37.8)
.....	.....

Task 2

Monthwise temperature readings	Monthwise distinct temperature readings
(u'1977', u'03', 231)	(u'1964', u'07', 47541)
(u'2001', u'02', 27)	(u'1954', u'04', 279)
(u'1964', u'03', 79)	(u'1956', u'02', 3)
(u'1977', u'10', 15108)	(u'1988', u'01', 12)
(u'1979', u'09', 34870)	(u'1967', u'06', 46669)
(u'2003', u'08', 109213)	(u'2006', u'07', 125192)
(u'1997', u'09', 75582)	(u'2005', u'06', 91843)
(u'2008', u'05', 58438)	(u'1972', u'04', 1951)
(u'2001', u'11', 2373)	(u'1983', u'04', 3918)
(u'1994', u'04', 8480)	(u'1995', u'03', 119)
.....	.....

Task 3

Average monthly tempearture for each station in Sweden (Year, month, station number, avg. monthly temperature)

```
(u'1989', u'06', u'92400', 14.212903225806453)
(u'1982', u'09', u'107530', 10.811290322580644)
(u'2002', u'11', u'136360', -5.672580645161289)
(u'1964', u'04', u'53370', 7.787096774193548)
(u'1967', u'08', u'98170', 15.408064516129032)
(u'2002', u'08', u'181900', 15.598387096774193)
(u'1996', u'08', u'96190', 17.1)
(u'1973', u'10', u'97520', 3.962903225806453)
(u'2010', u'10', u'64130', 5.974193548387096)
(u'1999', u'10', u'104090', 3.980645161290322)
```

.....

Task 4

The resultset for this task was empty.

Task 5

Average monthly precipitation for the Östergotland region for the period 1993-2016

```
(u'2008-03', 42.200000000000024)
(u'2006-09', 19.266666666666666)
(u'2014-05', 58.000000000000014)
(u'2001-11', 26.38333333333334)
(u'2011-05', 37.85)
(u'1999-07', 29.08333333333334)
(u'2010-02', 52.75000000000005)
(u'2013-08', 54.075)
(u'2010-09', 43.08333333333335)
(u'2013-05', 47.92500000000001)
(u'1998-11', 28.96666666666668)
```

.....

Code Appendix

Task 1

```
from pyspark import SparkContext

def maximumTemperature(a,b):
    if(a>=b):
        return a
    else:
        return b

def minimumTemperature(a,b):
    if(a<=b):
        return a
    else:
        return b

sparkContxt = SparkContext(appName="Lab-1_Task_1") #Name of the job
temperatureData = sparkContxt.textFile("BDA/input/temperature-readings.csv")
readLines = temperatureData.map(lambda line: line.split(";"))
year_temperature = readLines.map(lambda x:(x[1][0:4],float(x[3])))
year_temperature = year_temperature.filter(lambda x:int(x[0])>=1950 and int(x[0])<=2014)
max_temperatures = year_temperature.reduceByKey(maximumTemperature)
min_temperatures = year_temperature.reduceByKey(minimumTemperature)
max_temperature_sorted = max_temperatures.sortBy(ascending=False, keyfunc = lambda x:
    x[1])
min_temperature_sorted = min_temperatures.sortBy(ascending=False, keyfunc = lambda x:
    x[1])
max_temperature_sorted.saveAsTextFile("BDA/output")
min_temperature_sorted.saveAsTextFile("BDA/output")
```

Task 2a

```
from pyspark import SparkContext

sparkContxt = SparkContext(appName="Lab-1_Task_2a") #Name of the job
temperatureData = sparkContxt.textFile("BDA/input/temperature-readings.csv")
readLines = temperatureData.map(lambda line: line.split(";"))
year_temperature = readLines.map(lambda x:((x[1][0:4],x[1][5:7]),float(x[3])))
```

```
year_temperature = year_temperature.filter(lambda x: int(x[0][0])>=1950 and
    int(x[0][0])<=2014 and x[1]>=10)
temperature_count = year_temperature.groupByKey()
temperature_count = temperature_count.map(lambda x:(x[0][0],x[0][1],len(x[1])))
temperature_count.saveAsTextFile("BDA/output")
```

Task 2b

```
from pyspark import SparkContext

sparkContxt = SparkContext(appName="Lab-1_Task_2") #Name of the job
temperatureData = sparkContxt.textFile("BDA/input/temperature-readings.csv")
readLines = temperatureData.map(lambda line: line.split(";"))
year_temperature = readLines.map(lambda x:((x[1][0:4],x[1][5:7]),float(x[3])))
year_temperature = year_temperature.filter(lambda x: int(x[0][0])>=1950 and
    int(x[0][0])<=2014 and x[1]>=10)
temperature_count = year_temperature.groupByKey()
temperature_count = temperature_count.distinct().map(lambda
    x:(x[0][0],x[0][1],len(x[1])))
temperature_count.saveAsTextFile("BDA/output")
```

Task 3

```
from pyspark import SparkContext

def minMaxData(temp):
    return max(temp)+min(temp)

sparkContxt = SparkContext(appName="Lab-1_Task_3") #Name of the job
temperatureData = sparkContxt.textFile("BDA/input/temperature-readings.csv")
readLines = temperatureData.map(lambda line: line.split(";"))
stationTemperature = readLines.map(lambda
    x:((x[1][0:4],x[1][5:7],x[1][8:10],x[0]),float(x[3])))
stationTemperature = stationTemperature.filter(lambda x:int(x[0][0])>=1960 and
    int(x[0][0])<=2014)
averageTemp = stationTemperature.groupByKey()
averageTemp = averageTemp.map(lambda x: ((x[0][0],x[0][1],x[0][3]),minMaxData(x[1])))
averageTemp = averageTemp.groupByKey()
averageTemp = averageTemp.map(lambda x:(x[0][0],x[0][1],x[0][2],sum(x[1])/62))
averageTemp.saveAsTextFile("BDA/output")
```

Task 4

```
from pyspark import SparkContext

sc = SparkContext(appName = "Lab-1_Task_4")
temperature_file = sc.textFile("BDA/input/temperature-readings.csv")
precipitation_file = sc.textFile("BDA/input/precipitation-readings.csv")
tlines = temperature_file.map(lambda line: line.split(";"))
plines = precipitation_file.map(lambda line: line.split(";"))

temperature = tlines.map(lambda x: (x[0], float(x[3])))
temperature = temperature.groupByKey()
temperature = temperature.map(lambda x: (x[0], max(x[1])))
temperature = temperature.filter(lambda x: x[1] >= 25 and x[1] <= 30)

precipitation = plines.map(lambda x: ((x[0], x[1][0:4], x[1][5:7], x[1][8:10]),
```

```
    float(x[3])))
precipitation = precipitation.groupByKey()
precipitation = precipitation.map(lambda x: (x[0][0], sum(x[1])))
precipitation = precipitation.filter(lambda x: x[1] >= 100 and x[1] <= 200)

tp = temperature.join(precipitation)
tp.saveAsTextFile("BDA/output")
```

Task 5

```
from pyspark import SparkContext

sc = SparkContext(appName = "Lab-1_Task_5")
precipitation_file = sc.textFile("BDA/input/precipitation-readings.csv")
ostergotland_file = sc.textFile("BDA/input/stations-Ostergotland.csv")

plines = precipitation_file.map(lambda line: line.split(";"))
olines = ostergotland_file.map(lambda line: line.split(";"))

precipLines= plines.filter(lambda x: int(x[1][0:4]) >= 1993 and int(x[1][0:4]) <= 2016)
stations = olines.map(lambda x: x[0])
stations = sc.broadcast(stations.collect())

precipData = precipLines.map(lambda x:((x[0],x[1][0:7]),float(x[3])))

#monthly precipitation
monthly_precipitation = precipData.reduceByKey(lambda x,y:x+y)

#filtering Ostergotland from monthly_precipitation and avg_precipitation
monthly_precipitation_filtered = monthly_precipitation.filter(lambda x:x[0][0] in
    stations.value)
monthly_precipitation_filtered = monthly_precipitation_filtered.map(lambda
    x:(x[0][1],(x[1],1)))
monthly_precipitation_agg = monthly_precipitation_filtered.reduceByKey(lambda x,y:
    (x[0] + y[0], x[1] + y[1]))
monthly_precipitation_avg = monthly_precipitation_agg.map(lambda x: (x[0], x[1][0] /
    x[1][1]))

monthly_precipitation_avg.saveAsTextFile("BDA/output/")
```