

Spark Case Study, Working with Sensor Data

For this data analysis, you can download the necessary dataset from [this link](#).

In the above link there are two datasets; **building.csv** contains the details of the top 20 buildings all over the world and **HVAC.csv** contains the target temperature and the actual temperature along with the buildingId.

HVAC (heating, ventilating/ventilation, and **air conditioning**) is the technology of indoor and vehicular environmental comfort. Its goal is to provide thermal comfort and acceptable indoor air quality.

Through the HVAC sensors, we will get the temperature of the buildings.

Here are the columns that are present in the datasets:

Building.csv – BuildingID, BuildingMgr, BuildingAge, HVACproduct, Country

HVAC.csv – Date, Time, TargetTemp, ActualTemp, System, SystemAge, BuildingID

Objective-1

- Load HVAC.csv file into temporary table
- Add a new column, tempchange -set to 1, if there is a change of greater than +/-5 between actual and target temperature

Objective-2

- Load building.csv file into temporary table

Objective-3

- Figure out the number of times, temperature has changed by 5 degrees or more for each country:
- Join both the tables.
- Select temp change and country column
- Filter the rows where temp change is 1 and count the number of occurrence for each country

Solution

The following is the program output. Complete source code has been pasted later in the document

Objective-1

Date	Time	TargetTemp	ActualTemp	System	SystemAge	BuildingId	tempchange
6/1/13	0:00:01	66	58	13	20	4	1
6/2/13	1:00:01	69	68	3	20	17	0
6/3/13	2:00:01	70	73	17	20	18	0
6/4/13	3:00:01	67	63	2	23	15	0
6/5/13	4:00:01	68	74	16	9	3	1
6/6/13	5:00:01	67	56	13	28	4	1
6/7/13	6:00:01	70	58	12	24	2	1
6/8/13	7:00:01	70	73	20	26	16	0
6/9/13	8:00:01	66	69	16	9	9	0
6/10/13	9:00:01	65	57	6	5	12	1
6/11/13	10:00:01	67	70	10	17	15	0
6/12/13	11:00:01	69	62	2	11	7	1
6/13/13	12:00:01	69	73	14	2	15	0
6/14/13	13:00:01	65	61	3	2	6	0

6/15/13 14:00:01	67	59	19	22	20	1
6/16/13 15:00:01	65	56	19	11	8	1
6/17/13 16:00:01	67	57	15	7	6	1
6/18/13 17:00:01	66	57	12	5	13	1
6/19/13 18:00:01	69	58	8	22	4	1
6/20/13 19:00:01	67	55	17	5	7	1

+-----+-----+-----+-----+-----+-----+

only showing top 20 rows

Objective-2

buildid	buildmgr	buildAge	hvacproduct	Country
1	M1	25	AC1000	USA
2	M2	27	FN39TG	France
3	M3	28	JDNS77	Brazil
4	M4	17	GG1919	Finland
5	M5	3	ACMAX22	Hong Kong
6	M6	9	AC1000	Singapore
7	M7	13	FN39TG	South Africa
8	M8	25	JDNS77	Australia
9	M9	11	GG1919	Mexico
10	M10	23	ACMAX22	China
11	M11	14	AC1000	Belgium
12	M12	26	FN39TG	Finland
13	M13	25	JDNS77	Saudi Arabia
14	M14	17	GG1919	Germany
15	M15	19	ACMAX22	Israel
16	M16	23	AC1000	Turkey
17	M17	11	FN39TG	Egypt
18	M18	25	JDNS77	Indonesia
19	M19	14	GG1919	Canada
20	M20	19	ACMAX22	Argentina

Objective-3

Buildings data registered as building table

Date	Time	TargetTemp	ActualTemp	System	SystemAge	BuildingId	tempchange	country	hvacproduct
6/10/13 9:00:01	65	57	6	5	12	1	Finland	FN39TG	
6/18/13 23:13:19	66	75	1	13	12	1	Finland	FN39TG	
6/2/13 13:43:51	65	72	20	26	12	1	Finland	FN39TG	
6/13/13 0:13:20	67	77	8	19	12	1	Finland	FN39TG	
6/16/13 3:13:20	67	55	11	16	12	1	Finland	FN39TG	
6/30/13 17:13:20	65	57	17	9	12	1	Finland	FN39TG	

6/1/13 18:13:20	68	65	7	21	12	0 Finland	FN39TG
6/25/13 18:33:07	70	66	20	20	12	0 Finland	FN39TG
6/17/13 16:00:01	69	68	16	4	12	0 Finland	FN39TG
6/5/13 16:43:51	69	69	19	15	12	0 Finland	FN39TG
6/23/13 10:13:20	65	61	1	1	12	0 Finland	FN39TG
6/29/13 16:13:20	67	80	12	8	12	1 Finland	FN39TG
6/4/13 21:13:20	66	72	7	1	12	1 Finland	FN39TG
6/3/13 2:00:01	69	72	7	21	12	0 Finland	FN39TG
6/16/13 15:00:01	67	77	4	22	12	1 Finland	FN39TG
6/22/13 21:00:01	70	77	13	12	12	1 Finland	FN39TG
6/26/13 7:43:51	65	62	6	6	12	0 Finland	FN39TG
6/26/13 13:13:20	65	63	20	9	12	0 Finland	FN39TG
6/30/13 17:13:20	66	62	14	26	12	0 Finland	FN39TG
6/10/13 3:33:07	70	78	5	9	12	1 Finland	FN39TG

+-----+

only showing top 20 rows

+---+-----+

| _1| _2|

+---+-----+

1 Finland
1 Finland
1 Finland
1 Finland
1 Finland
1 Finland
0 Finland
0 Finland
0 Finland
0 Finland
0 Finland
1 Finland
1 Finland
0 Finland
1 Finland
1 Finland
0 Finland
0 Finland
0 Finland
1 Finland

+---+-----+

only showing top 20 rows

+---+-----+

| _1| _2|

+---+-----+

1 Finland
1 Finland

	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	
	1		Finland	

```
+---+-----+
```

only showing top 20 rows

```
+-----+-----+
|      _2|count|
+-----+-----+
| Singapore| 230|
|   Turkey| 243|
|  Germany| 196|
|   France| 251|
| Argentina| 230|
|   Belgium| 199|
|   Finland| 473|
|    China| 241|
| Hong Kong| 248|
|   Israel| 232|
|    USA| 213|
|   Mexico| 228|
| Indonesia| 243|
|Saudi Arabia| 233|
|   Canada| 232|
|   Brazil| 226|
| Australia| 225|
|    Egypt| 236|
|South Africa| 237|
+-----+-----+
```

Process finished with exit code 0

Source Code

```
package demo

import org.apache.spark.sql.SparkSession
import org.apache.spark.{SparkConf, SparkContext}

import org.apache.spark.graphx._
import org.apache.spark.rdd.RDD
import org.apache.spark.sql.SparkSession

object SparkSQLUseCase1
{
  case class
  hvac_cls(Date:String,Time:String,TargetTemp:Int,ActualTemp:Int,System:Int,SystemAge:In
t,BuildingId:Int)
  case class
  building(buildid:Int,buildmgr:String,buildAge:Int,hvacproduct:String,Country:String)

  def main(args: Array[String]): Unit = {

    case_study

  }

  def case_study=
  {

    // println("Case Study")
    val spark = SparkSession.builder().master(master="local")
      .appName(name="spark sql example")
      .config("spark.some.config.option","some-value").getOrCreate()

    spark.sparkContext.setLogLevel("ERROR")
    println("session object created")

    val data = spark.sparkContext.textFile("/Users/mmisra/Desktop/acad/spark-
files/HVAC.csv")
    // number of lines in the file
    println("HVAC Data->>" + data.count())

    val header = data.first()

    val data1 = data.filter(row => row != header)

    println("Header removed from the data !")

    //For implicit conversions like converting RDDs and sequences to DataFrames
    import spark.implicits._

    // second map is to populate case class with the data and then convert RDD to
dataframe
    val hvac = data1.map(x=>x.split(",")).map(x =>
hvac_cls(x(0),x(1),x(2).toInt,x(3).toInt,x(4).toInt,x(5).toInt,x(6).toInt)).toDF()

    hvac.show()

    // we can use create or replace temp view
```

```

hvac.registerTempTable("HVAC")

// mark the record as 1 if the tem is in thne range -5 to +5 and call the column
as tempchange
val hvac1 = spark.sql("select *,IF((targettemp - actualtemp) > 5, '1',
IF((targettemp - actualtemp) < -5, '1', 0)) AS tempchange from HVAC")

hvac1.show()

// now create another view for this table
hvac1.registerTempTable("HVAC1")

println("Data Frame Registered as HVAC1 table !")

//Now lets load the second data set

val data2 = spark.sparkContext.textFile("/Users/mmisra/Desktop/acad/spark-
files/building.csv");

val header1 = data2.first()

val data3 = data2.filter(row => row != header1)

println("Header removed from the building data")

//Now let us create the building dataframe

val build = data3.map(x=> x.split(",")).map(x =>
building(x(0).toInt,x(1),x(2).toInt,x(3),x(4))).toDF

build.show()

// we can use create or replace view also for this
build.registerTempTable("building")

println("Buildings data registered as building table")

//Now join the two tables
// h is the alias for hvac1 and b is alias for building
val build1 = spark.sql("select h.*, b.country, b.hvacproduct from building b join
hvac1 h on b.buildid = h.buildingid")

build1.show()

//Select temperature and country column from above

val tempCountry = build1.map(x => (new Integer(x(7).toString),x(8).toString))

tempCountry.show()

//Filter the values

val tempCountryOnes = tempCountry.filter(x=> {if(x._1==1) true else false})

tempCountryOnes.show()

// group by country
tempCountryOnes.groupBy("_2").count.show

//Save the output to the disk. does not work in IDEA
//tempCountryOnes.write.save("/Users/mmisra/Desktop/acad/spark-

```

```
files/sparksqlout.txt")
```

```
}
```

```
}
```