Case Study – III

The datasets are

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

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

The code will be as follows..

First, remove the header from the CSV file

```
val data = sc.textFile("/home/acadgild/Documents/HVAC.csv")
val header = data.first()
val data1 = data.filter(row => row != header)
```

Write a case class holding the schema of the HVAC dataset

```
case class
hvac_cls(Date:String,Time:String,TargetTemp:Int,ActualTemp:Int,System:Int,SystemAge:Int,Buil
dingId:Int)
```

Split each row of the dataset with the delimiter and we are mapping the columns to our case class and finally, we are converting it into a data frame.

```
case class
hvac_cls(Date:String,Time:String,TargetTemp:Int,ActualTemp:Int,System:Int,SystemAge:Int,Buil
dingId:Int)
```

Create a table HVAC for our dataframe

```
hvac.registerTempTable("HVAC")
```

Perform a SQL query on the table, which creates one new column tempchange, which will set to 1 if there is a temperature change of either +5 or -5 between the actual_temperature and the target_temperature.

```
val hvac1 = sqlContext.sql("select *,IF((targettemp - actualtemp) > 5, '1', IF((targettemp -
actualtemp) < -5, '1', 0)) AS tempchange from HVAC")
```

Register the table HVAC1.

```
hvac1.registerTempTable("HVAC1")
```

create a table for the building dataset.

Remove the header from the dataset

```
val data2 = sc.textFile("/home/acadgild/Documents/building.csv")
val header1 = data2.first()
val data3 = data2.filter(row => row != header1)
```

Define a case class holding the schema of the building dataset.

```
case class building(buildid:Int,buildmgr:String,buildAge:Int,hvacproduct:String,Country:String)
```

Map the dataset to the case class which we have built.

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

Register the build dataframe as a table building

```
build.registerTempTable("building")
```

Join both the tables building and hvac1

Join the two datasets using the buildingId.

```
val build1 = sqlContext.sql("select h.*, b.country, b.hvacproduct from building b join hvac1 h on
buildid = buildingid")
```

Take the details from the tempchange column and the country column, which are required to find the maximum temperature change areas

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

Filter the rows which have a change in temperature, which is identified by 1.

```
val test1 = test.filter(x=> {if(x._1==1) true else false})
```

We are taking the country and we are adding 1 to know how many times the temperature in that building has changed. We are applying reduceByKey operation on the data to count the number of times temperature has been changed and finally, we are sorting the in descending order and printing it out.

```
val test2 = test1.map(x=>(x._2,1)).reduceByKey(_+_).map(item =>
item.swap).sortByKey(false).collect
```

The output

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

From the above output, we can say that temperature in Finland is changing more frequently followed by France and Hong Kong.