TOPIC:

Project on NCDC (National Climate Data Center)

Course: CPSC-651-11-Big Data Systems & Analysis – Fall 2019

Submission to: Prof. Jeongkyu Lee

Submitted by: Srikanth Dabbiru (UB ID 1046112) & Aditya Lavu (UB ID 1066404)

- **<u>Testing the Project with Sample Data on VM:</u>**

1. To test the project and the codes, we have uploaded a sample of year 2015 from the NOAA dataset.

2. The idea was to find out if it was a hot or a cold day depending on the temperatures recorded.

3. Days of the year crossing 35.0 were considered to be the hottest days from the sample.

4. Snapshot of the sample .csv used in the VM is as below:

A1 23907 20150101 2.423 -98.08 30.62 2.2 -0.6 0.8 0.9 7.0 1.47 C 3.7 1.1 2.5 99.9 85.4 97.2 0.369 0.308 -99.000 -99.000 -99.000 7.0 8.1 -9999.0 -9999.0 -9999.0

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	23907	20150101	2.423	-98.08	30.62	2.2	-0.6	0.8	0.9	7.0	1.47 C	3.7	1.1	2.5	99.9	85.4	97.2
2	23907	20150102	2.423	-98.08	30.62	3.5	1.3	2.4	2.2	10.2	1.43 C	4.9	2.3	3.1	100.0	98.8	99.8
3	23907	20150103	2.423	-98.08	30.62	15.9	2.3	9.1	7.5	3.1	11.00 C	16.4	2.9	7.3	100.0	34.8	73.7
4	23907	20150104	2.423	-98.08	30.62	9.2	-1.3	3.9	4.2	0.0	13.24 C	12.4	-0.5	4.9	82.0	40.6	61.7
5	23907	20150105	2.423	-98.08	30.62	10.9	-3.7	3.6	2.6	0.0	13.37 C	14.7	-3.0	3.8	77.9	33.3	57.4
6	23907	20150106	2.423	-98.08	30.62	20.2	2.9	11.6	10.9	0.0	12.90 C	22.0	1.6	9.9	67.7	30.2	49.3
7	23907	20150107	2.423	-98.08	30.62	10.9	-3.4	3.8	4.5	0.0	12.68 C	12.4	-2.1	5.5	82.7	36.5	55.7
8	23907	20150108	2.423	-98.08	30.62	0.6	-7.9	-3.6	-3.3	0.0	4.98 C	3.9	-4.8	-0.5	57.7	37.6	48.1
9	23907	20150109	2.423	-98.08	30.62	2.0	0.1	1.0	0.8	0.0	2.52 C	4.1	1.2	2.5	87.8	48.9	64.4
10	23907	20150110	2.423	-98.08	30.62	0.5	-2.0	-0.8	-0.6	3.9	2.11 C	2.5	-0.1	1.4	99.9	47.7	85.8
11	23907	20150111	2.423	-98.08	30.62	10.9	0.0	5.4	4.4	2.6	6.38 C	12.7	1.3	5.8	100.0	77.8	97.1
12	23907	20150112	2.423	-98.08	30.62	6.5	1.4	4.0	4.3	0.0	1.55 C	6.9	2.7	5.1	100.0	89.4	97.8
13	23907	20150113	2.423	-98.08	30.62	3.0	-0.7	1.1	1.2	0.0	3.26 C	5.6	0.7	2.9	99.7	80.7	90.7
14	23907	20150114	2.423	-98.08	30.62	2.9	0.9	1.9	1.8	0.7	1.88 C	4.7	2.0	3.1	99.6	90.8	97.9
15	23907	20150115	2.423	-98.08	30.62	13.2	1.2	7.2	6.4	0.0	13.37 C	16.4	1.4	6.7	98.9	46.7	73.4
16	23907	20150116	2.423	-98.08	30.62	16.7	3.5	10.1	9.9	0.0	13.68 C	19.2	1.3	8.7	80.2	38.1	58.2
17	23907	20150117	2.423	-98.08	30.62	19.5	5.0	12.2	12.3	0.0	10.96 C	20.9	3.3	10.6	87.7	30.4	55.7
18	23907	20150118	2.423	-98.08	30.62	20.9	7.6	14.3	13.7	0.0	15.03 C	23.4	3.5	11.9	45.9	14.6	31.4
19	23907	20150119	2.423	-98.08	30.62	23.9	6.7	15.3	14.3	0.0	14.10 C	25.6	3.8	12.6	65.3	26.8	45.6
20	23907	20150120	2.423	-98.08	30.62	26.0	9.5	17.8	15.9	0.0	14.57 C	27.9	6.5	14.5	88.4	16.1	50.2
21	23907	20150121	2.423	-98.08	30.62	11.0	6.9	8.9	8.9	1.7	2.71 C	13.1	6.8	9.7	99.2	68.0	88.1
22	23907	20150122	2.423	-98.08	30.62	8.6	3.5	6.1	5.6	40.0	1.28 C	9.1	4.1	6.3	99.6	95.2	98.0
23	23907	20150123	2.423	-98.08	30.62	9.4	2.2	5.8	4.2	7.5	6.58 C	11.1	2.0	4.8	98.4	58.8	86.5
24	23907	20150124	2.423	-98.08	30.62	16.0	1.4	8.7	8.0	0.0	14.26 C	18.8	0.4	7.7	92.0	33.0	63.0
25	23907	20150125	2.423	-98.08	30.62	20.2	6.4	13.3	12.7	0.0	14.99 C	22.0	4.4	11.0	69.2	18.9	43.8
26	23907	20150126	2.423	-98.08	30.62	21.5	7.2	14.4	14.1	0.0	12.01 C	22.9	5.5	12.2	56.8	23.7	40.6
27	23907	20150127	2.423	-98.08	30.62	26.5	10.7	18.6	17.5	0.0	15.18 C	28.9	8.1	15.5	52.2	21.4	38.8
28	23907	20150128	2.423	-98.08	30.62	26.3	13.3	19.8	19.1	0.0	15.11 C	28.1	7.9	16.3	54.9	19.4	35.5
29	23907	20150129	2.423	-98.08	30.62	23.1	9.8	16.5	16.4	0.0	13.74 C	27.4	9.7	16.4	87.0	34.2	55.6
30	23907	20150130	2.423	-98.08	30.62	13.0	6.9	10.0	9.0	0.2	7.19 C	19.2	8.3	11.0	67.6	48.4	58.8
31	23907	20150131	2.423	-98.08	30.62	15.1	7.4	11.3	10.2	8.5	1.18 C	14.5	8.4	10.7	100.0	63.1	90.3
32	23907	20150201	2.423	-98.08	30.62	18.3	3.9	11.1	13.3	0.0	8.69 C	22.1	4.1	13.8	98.8	53.6	79.1
33	23907	20150202	2.423	-98.08	30.62	8.0	-1.9	3.1	3.3	0.0	12.48 C	15.2	-0.6	5.8	69.4	34.8	54.2
34	23907	20150203	2.423	-98.08	30.62	5.3	2.3	3.8	3.8	0.8	2.69 C	8.3	3.9	5.7	100.0	65.1	82.6
35	23907	20150204	2.423	-98.08	30.62	11.8	4.3	8.1	7.9	0.3	4.41 C	13.8	5.5	8.8	100.0	80.6	94.1

	J	K	L	M	N	O
1	0.308	-99.000	-99.000	-99.000	7.0	8.1
2	0.327	-99.000	-99.000	-99.000	7.1	7.9
3	0.397	-99.000	-99.000	-99.000	7.6	7.9
4	0.352	-99.000	-99.000	-99.000	7.3	7.9
5	0.340	-99.000	-99.000	-99.000	6.3	7.0
6	0.335	-99.000	-99.000	-99.000	8.0	8.0
7	0.328	-99.000	-99.000	-99.000	7.6	8.3
8	0.316	-99.000	-99.000	-99.000	4.7	6.1
9	0.312	-99.000	-99.000	-99.000	5.4	6.2
10	0.314	-99.000	-99.000	-99.000	5.1	6.0
11	0.362	-99.000	-99.000	-99.000	6.5	6.7
12	0.350	-99.000	-99.000	-99.000	7.3	7.5
13	0.337	-99.000	-99.000	-99.000	6.1	6.8
14	0.331	-99.000	-99.000	-99.000	6.1	6.7
15	0.333	-99.000	-99.000	-99.000	6.7	7.0
16	0.330	-99.000	-99.000	-99.000	7.3	7.4
17	0.327	-99.000	-99.000	-99.000	8.7	8.4
18	0.325	-99.000	-99.000	-99.000	9.5	9.2
19	0.321	-99.000	-99.000	-99.000	9.9	9.5
20	0.320	-99.000	-99.000	-99.000	10.9	10.4
21	0.317	-99.000	-99.000	-99.000	10.7	10.6
22	0.418	-99.000	-99.000	-99.000	9.0	9.3
23	0.409	-99.000	-99.000	-99.000	7.6	8.1
24	0.381	-99.000	-99.000	-99.000	7.7	7.9
25	0.357	-99.000	-99.000	-99.000	9.1	8.9
26	0.349	-99.000	-99.000	-99.000	10.0	9.7
27	0.344	-99.000	-99.000	-99.000	11.4	10.8
28	0.339	-99.000	-99.000	-99.000	12.1	11.5
29	0.334	-99.000	-99.000	-99.000	13.1	12.4
30	0.328	-99.000	-99.000	-99.000	11.6	11.8
31	0.337	-99.000	-99.000	-99.000	11.2	11.3
32	0.386	-99.000	-99.000	-99.000	12.9	12.6
33	0.354	-99.000	-99.000	-99.000	9.3	10.1
34	0.343	-99.000	-99.000	-99.000	8.7	9.3
35	0.344	-99.000	-99.000	-99.000	9.6	9.8

	M	N	O
1	-9999.0	-9999.0	-9999.0
2	-9999.0	-9999.0	-9999.0
3	-9999.0	-9999.0	-9999.0
4	-9999.0	-9999.0	-9999.0
5	-9999.0	-9999.0	-9999.0
6	-9999.0	-9999.0	-9999.0
7	-9999.0	-9999.0	-9999.0
8	-9999.0	-9999.0	-9999.0
9	-9999.0	-9999.0	-9999.0
10	-9999.0	-9999.0	-9999.0
11	-9999.0	-9999.0	-9999.0
12	-9999.0	-9999.0	-9999.0
13	-9999.0	-9999.0	-9999.0
14	-9999.0	-9999.0	-9999.0
15	-9999.0	-9999.0	-9999.0
16	-9999.0	-9999.0	-9999.0
17	-9999.0	-9999.0	-9999.0
18	-9999.0	-9999.0	-9999.0
19	-9999.0	-9999.0	-9999.0
20	-9999.0	-9999.0	-9999.0
21	-9999.0	-9999.0	-9999.0
22	-9999.0	-9999.0	-9999.0
23	-9999.0	-9999.0	-9999.0
24	-9999.0	-9999.0	-9999.0
25	-9999.0	-9999.0	-9999.0
26	-9999.0	-9999.0	-9999.0
27	-9999.0	-9999.0	-9999.0
28	-9999.0	-9999.0	-9999.0
29	-9999.0	-9999.0	-9999.0
30	-9999.0	-9999.0	-9999.0
31	-9999.0	-9999.0	-9999.0
32	-9999.0	-9999.0	-9999.0
33	-9999.0	-9999.0	-9999.0
34	-9999.0	-9999.0	-9999.0
35	-9999.0	-9999.0	-9999.0

2015 +

Ready

- **<u>Mapper-Reducer Code:</u>**

For the mapper, the max temperature class is static, this method takes the input as text data type. Leaving the first five tokens, the 6th token is taken as the temp_max and the 7th as temp_min.

Now temp_max value is set to be >35.0 and the temp_min is set to be <10.0 and are now passed to the reducer step.

If the temp values for the day are >35.0 output as Hot Day and if <10.0 output as a Cold Day.

For the Reducer method, it takes the input as key and the pairs would be the list of values from the Mapper.

Now **Aggregation** is applied, and it produce the next result.

For the main method, it is used for setting up all the configuration properties. This will be acting as the driver for our Map Reduce code.

- **<u>Below is the Complete Source Code used:</u>**

MyMaxMin.java 〉 main(String[] args)

```java
import java.io.IOException;
import java.util.Iterator;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;

public class MyMaxMin {


    //Mapper



    public static class MaxTemperatureMapper extends
            Mapper<LongWritable, Text, Text, Text> {


        @Override
        public void map(LongWritable arg0, Text Value, Context context)
                throws IOException, InterruptedException {

        //To Convert the record (single line) to String and storing it

            String line = Value.toString();

        //To Check if the line is not empty

            if (!(line.length() == 0)) {

                //date

                String date = line.substring(6, 14);

                //maximum temperature

                float temp_Max = Float
                        .parseFloat(line.substring(39, 45).trim());

                //minimum temperature

                float temp_Min = Float
                        .parseFloat(line.substring(47, 53).trim());

                //if maximum temperature is greater than 35.0 , its a h

                if (temp_Max > 35.0) {
                    // Hot day
                    context.write(new Text("Hot Day " + date),
                            new Text(String.valueOf(temp_Max)));
                }
```

MyMaxMin.java 〉 main(String[] args)

```java
                }

                //if minimum temperature is less than 10.0 , its a cold day

                if (temp_Min < 10) {
                    // Cold day
                    context.write(new Text("Cold Day " + date),
                            new Text(String.valueOf(temp_Min)));
                }
            }
        }

    }

//Reducer


    public static class MaxTemperatureReducer extends
            Reducer<Text, Text, Text, Text> {


        public void reduce(Text Key, Iterator<Text> Values, Context context)
                throws IOException, InterruptedException {


            //To put all the values in temperature variable of type String

            String temperature = Values.next().toString();
            context.write(Key, new Text(temperature));
        }

    }



    public static void main(String[] args) throws Exception {

        //reads the default configuration of cluster from the configuration xml files
        Configuration conf = new Configuration();

        //Initializing the job with the default configuration of the cluster
        Job job = new Job(conf, "weather example");

        //Assigning the driver class name
        job.setJarByClass(MyMaxMin.class);

        //Key type coming out of mapper
        job.setMapOutputKeyClass(Text.class);

        //value type coming out of mapper
        job.setMapOutputValueClass(Text.class);

        //Defining the mapper class name
        job.setMapperClass(MaxTemperatureMapper.class);

        //Defining the reducer class name
        job.setReducerClass(MaxTemperatureReducer.class);

        //Defining input Format class which is responsible to parse the dataset into a key value pair
        job.setInputFormatClass(TextInputFormat.class);
```

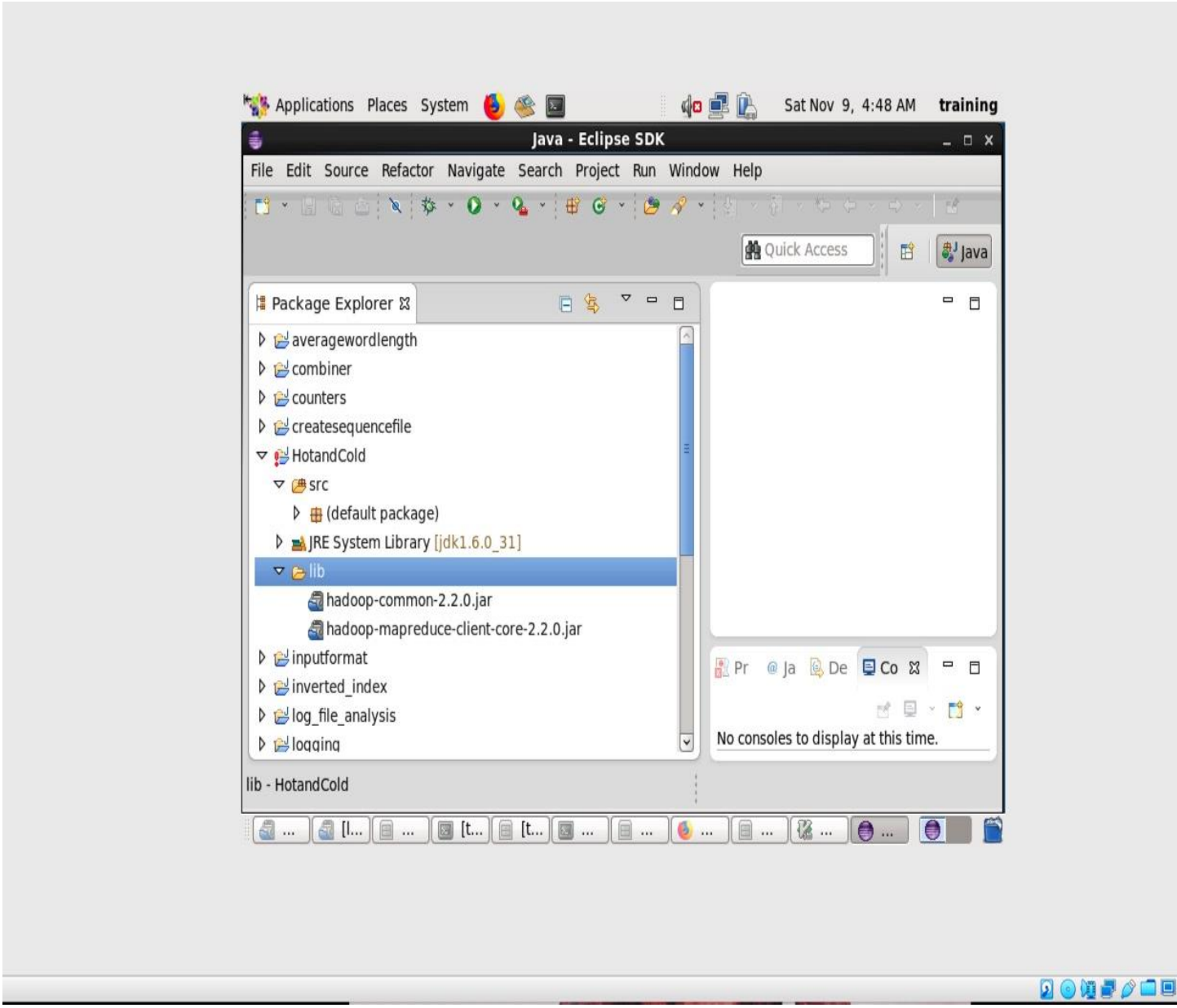
MyMaxMin.java 〉 main(String[] args)

```java
        job.setMapperClass(MaxTemperatureMapper.class);

        //Defining the reducer class name
        job.setReducerClass(MaxTemperatureReducer.class);

        //Defining input Format class which is responsible to parse the dataset into a key value pair
        job.setInputFormatClass(TextInputFormat.class);

        //Defining output Format class which is responsible to parse the dataset into a key value pair
        job.setOutputFormatClass(TextOutputFormat.class);

        //setting the second argument as a path in a path variable
        Path OutputPath = new Path(args[1]);

        //Configuring the input path from the filesystem into the job
        FileInputFormat.addInputPath(job, new Path(args[0]));

        //Configuring the output path from the filesystem into the job
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        //deleting the context path automatically from hdfs so that we don't have delete it
        OutputPath.getFileSystem(conf).delete(OutputPath);

        //exiting the job only if the flag value becomes false
        System.exit(job.waitForCompletion(true) ? 0 : 1);

    }
}
```

- **Eclipse IDE:**

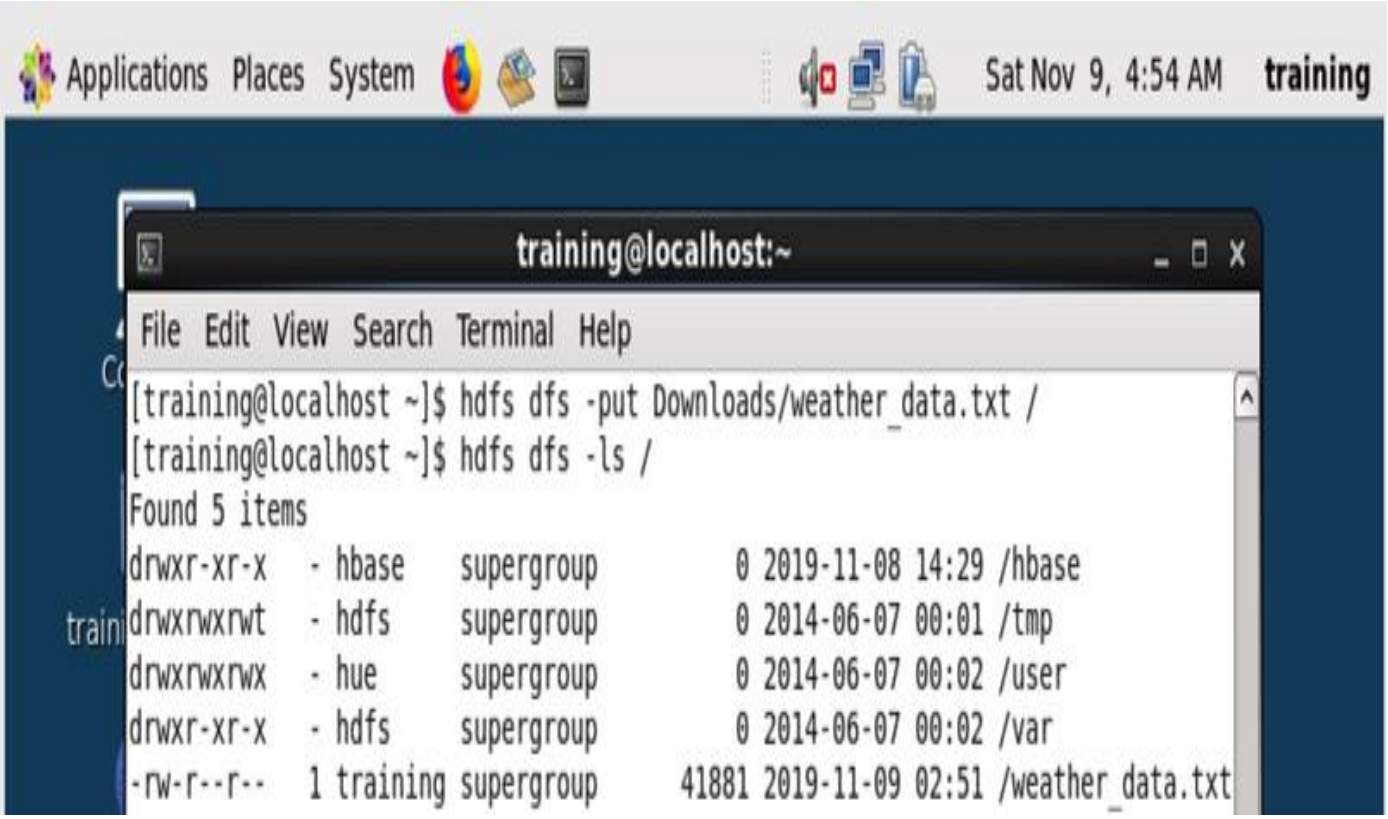
Now the project is created in the Eclipse IDE to analyze the sample dataset.

The jar file is then exported after having no issues with the project files.



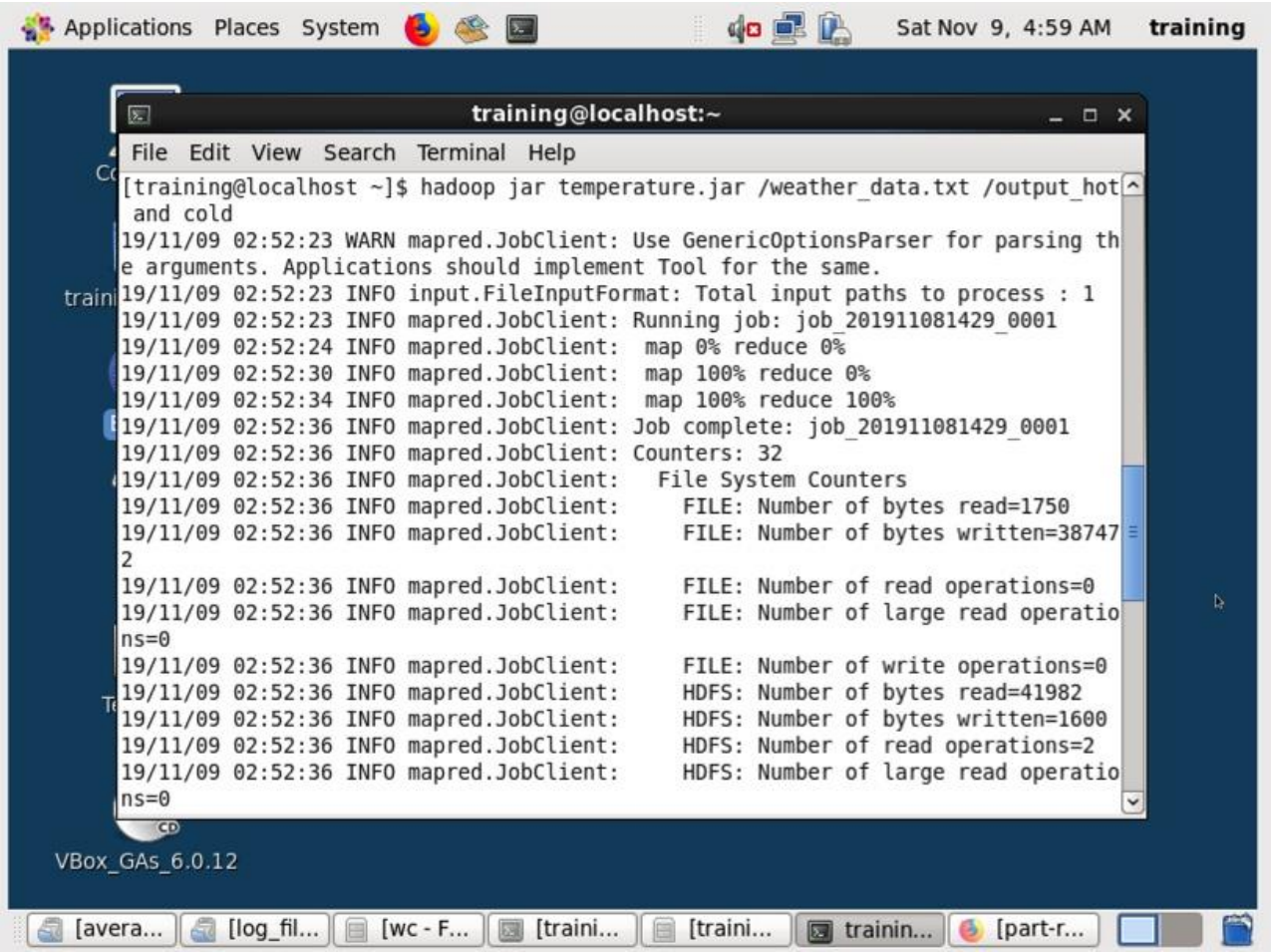
The next step was to send the sample dataset onto HDFS.

Hdfs dfs -put Downloads/Noaa_Weather_data.txt /

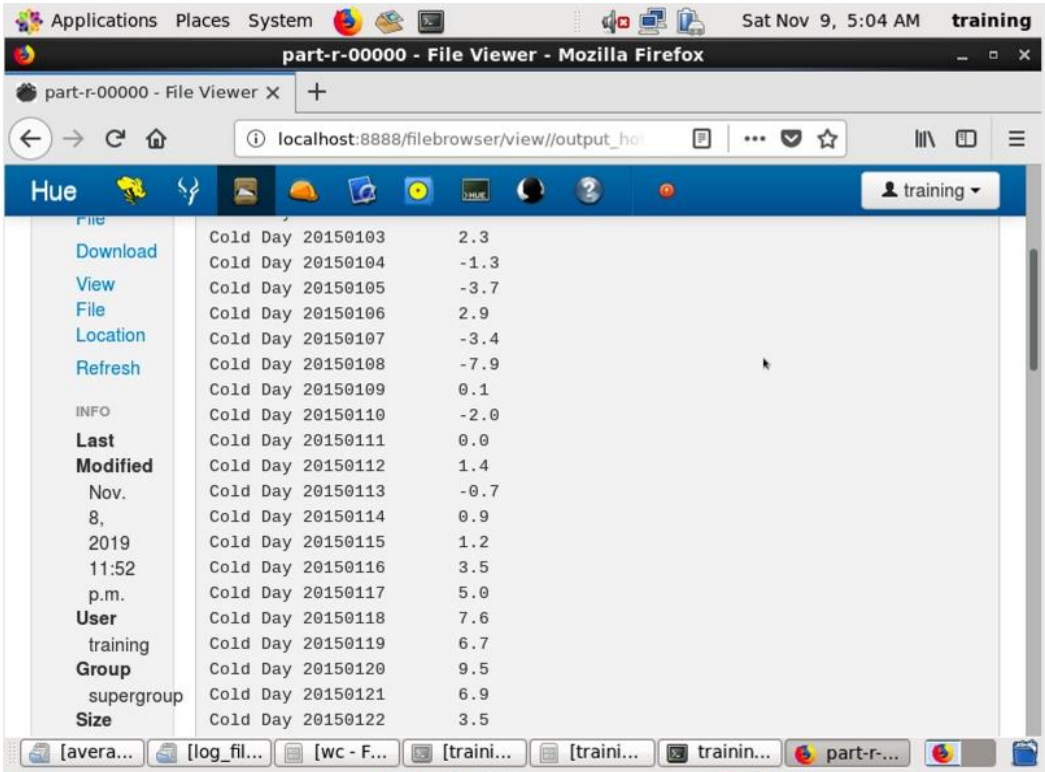


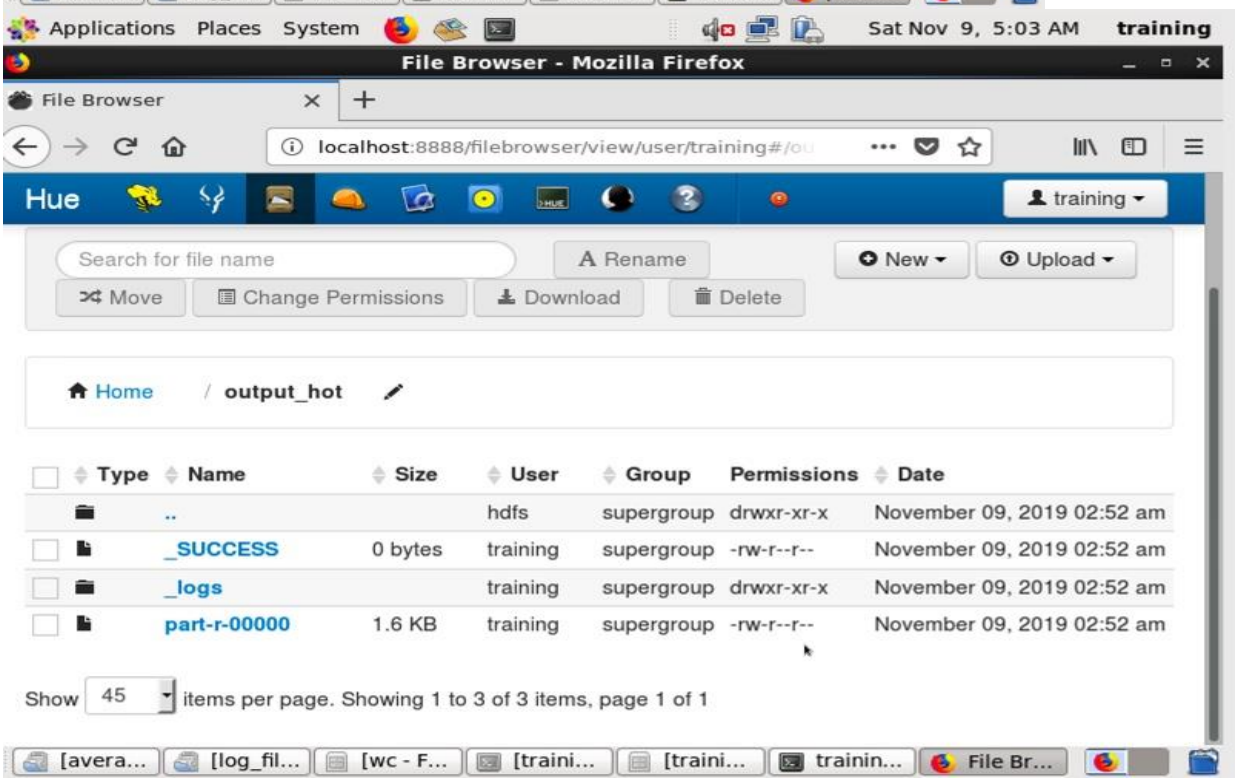
Run the Jar file for output.

Hadoop jar temperature.jar /Noaa_weather_data.txt /output_hotandcold



Check the Output directory in the HDFS.





Results analysis:

Depending on the 2015 sample dataset only two days above 35.0 recorded.