# Big Data Hadoop Training

Session 4 Assignment 1 Solution:

Q) **Write a Map Reduce program to filter out the invalid records. The output of this program will act as input for subsequent tasks. Map only job will fit for this context**.

A) **Java Program**: Takes HDFS Path for **television.txt** as input and filters-out the invalid records and stores the result in output path as a **part-file.**

**Note:** Input file kept into HDFS in **/tmp/access\_folder/ folder**

Output path provided: **/tmp/access\_folder/television**

**import** java.io.IOException;

**import** org.apache.hadoop.fs.Path;

**import** org.apache.hadoop.io.LongWritable;

**import** org.apache.hadoop.io.NullWritable;

**import** org.apache.hadoop.io.Text;

**import** org.apache.hadoop.mapred.FileInputFormat;

**import** org.apache.hadoop.mapred.FileOutputFormat;

**import** org.apache.hadoop.mapred.JobClient;

**import** org.apache.hadoop.mapred.JobConf;

**import** org.apache.hadoop.mapred.MapReduceBase;

**import** org.apache.hadoop.mapred.Mapper;

**import** org.apache.hadoop.mapred.OutputCollector;

**import** org.apache.hadoop.mapred.Reporter;

**import** org.apache.hadoop.mapred.TextInputFormat;

**import** org.apache.hadoop.mapred.TextOutputFormat;

**public** **class** filter\_invalid\_records{

**public** **static** **class** Map **extends** MapReduceBase **implements** Mapper<LongWritable, Text, NullWritable, Text> {

**public** **void** map(LongWritable key, Text value, OutputCollector<NullWritable, Text> output, Reporter reporter) **throws** IOException {

**boolean** invalid\_record = **false**;

String line = value.toString();

String[] elements = line.split("\\|"); // Delimiter "|" to separate words

**for** (**int** i=0; i<elements.length; i++){

**if**(elements[i].equals("NA")) {

invalid\_record = **true**;

**break**; // No need of further checking for "NA" in other fields

}

}

**if**(!invalid\_record) { // If it is not an invalid record then only write it into output

output.collect(NullWritable.*get*(), **new** Text(line));;

}

}

}

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

JobConf conf = **new** JobConf(filter\_invalid\_records.**class**);

conf.setJobName("filter");

conf.setOutputKeyClass(NullWritable.**class**);

conf.setOutputValueClass(Text.**class**);

conf.setMapperClass(Map.**class**);

// Setting Number of Mappers to one

conf.setNumMapTasks(1);

// Setting Number of reducers to zero

conf.setNumReduceTasks(0); // **Map-only job will fit for this task**

conf.setInputFormat(TextInputFormat.**class**);

conf.setOutputFormat(TextOutputFormat.**class**);

FileInputFormat.*setInputPaths*(conf, **new** Path(args[0]));

FileOutputFormat.*setOutputPath*(conf, **new** Path(args[1]));

JobClient.*runJob*(conf);

}

}

**Input format: “**hdfs://<hostname>:<port\_no>/<HDFS\_Input\_File\_Path>**” “**hdfs://<hostname>:<port\_no>/<HDFS\_Output\_File\_Path>**”**

**Running in Eclipse:**

* Input given in Run Configurations Argument Tab as : "hdfs://pavan:8020/tmp/access\_folder/television.txt" "hdfs://pavan:8020/tmp/access\_folder/television"

**Note: Here Hostname: pavan and Port no:8020 is required since files are in HDFS**

* Become a HDFS user and filtered Output can be seen with the help of :

**hadoop fs –cat /tmp/access\_folder/television/part-00000** command

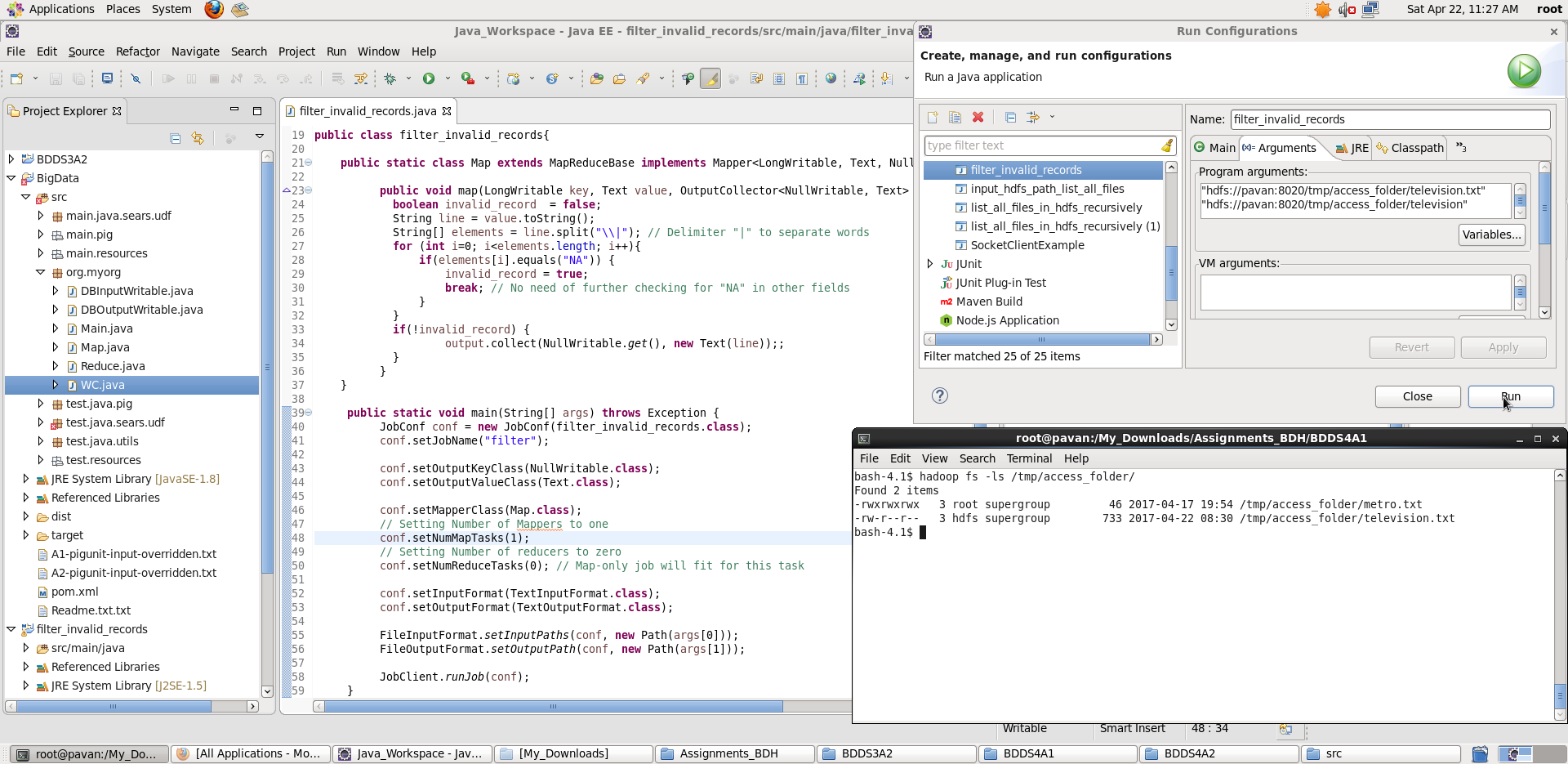
**Running in Command Line:**

* Make the folder structure BDDS4A1/src/main/java/filter\_invalid\_records.java
* Then in BDDS4A1 folder, open terminal
* Create the Jar file target/BDDS4A1.jar using Maven as **root user** # mvn clean install
* Now , become HDFS user # su hdfs
* Run the Hadoop Job : “hadoop jar <jar\_file\_path> <class\_name> <input\_hdfs\_file\_path> <output\_hdfs\_file\_path>”
* In our case : **hadoop jar target/BDDS4A1.jar filter\_invalid\_records /tmp/access\_folder/television.txt /tmp/access\_folder/television**
* **Note: Here both input and output present in HDFS, so no Hostname and Port No specification required.**
* Filtered Output can be seen with the help of :

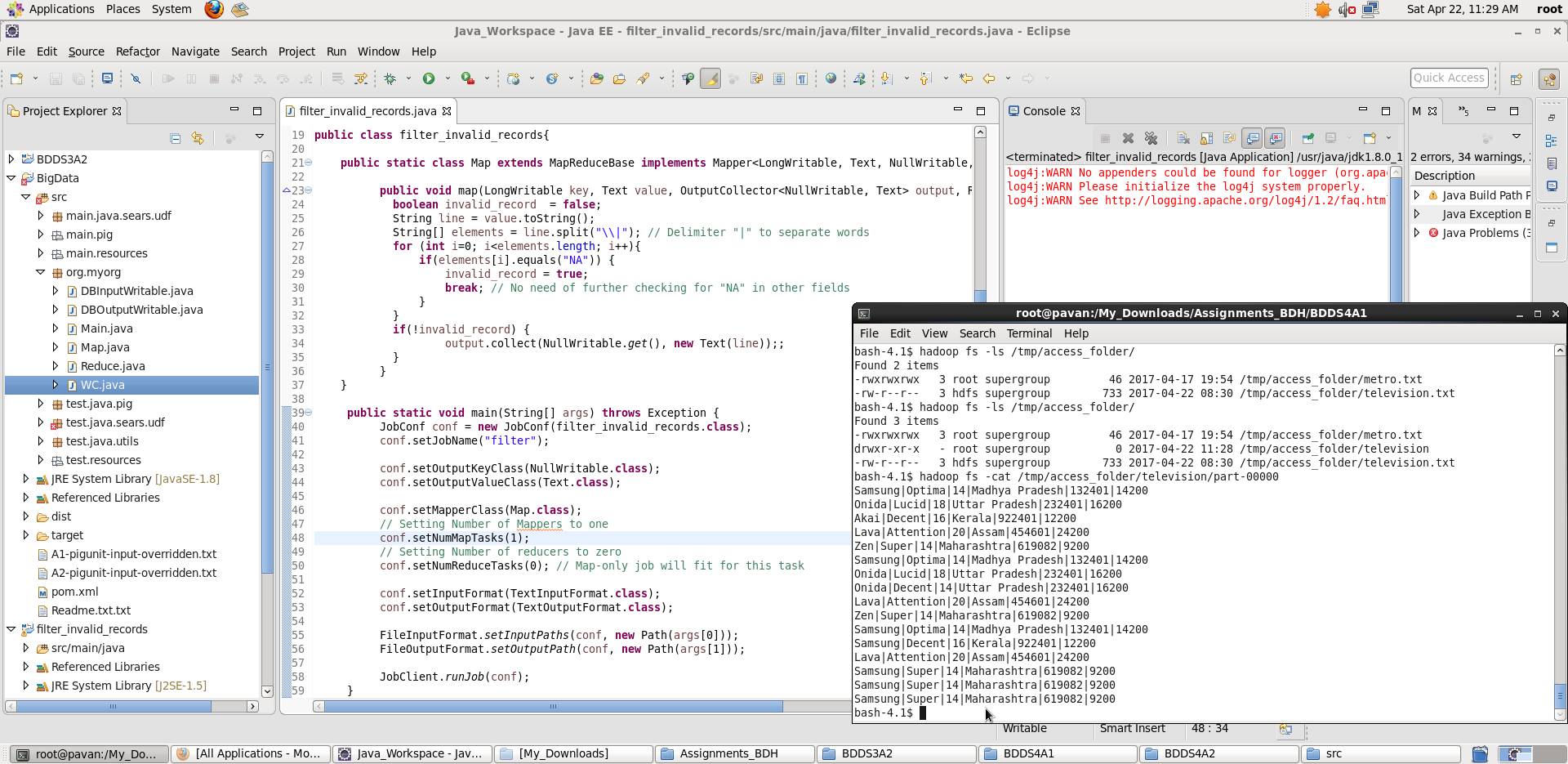
**hadoop fs –cat /tmp/access\_folder/television/part-00000** command

Screenshots

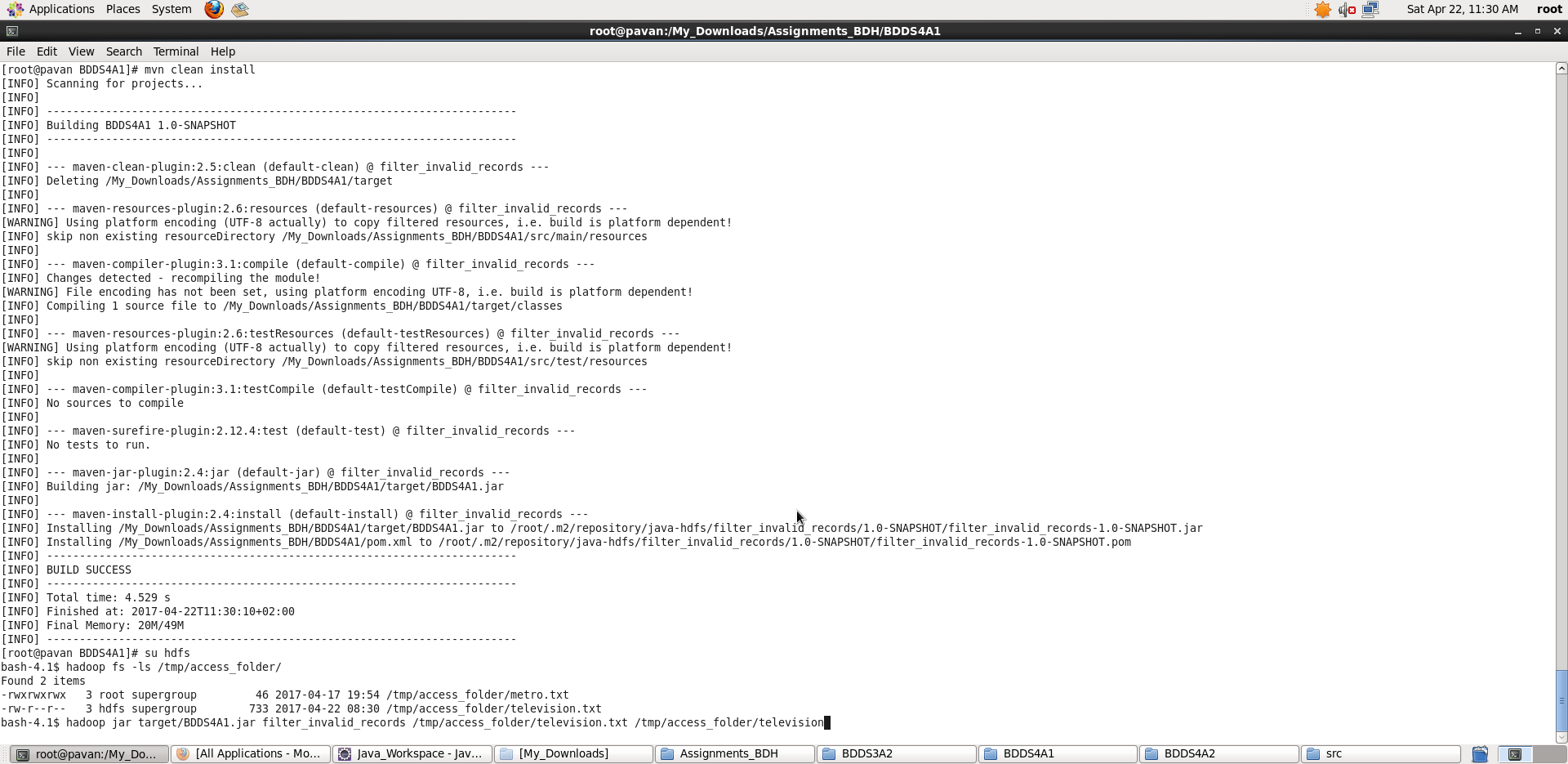
Eclipse Java code and Input provided in Run Configurations:



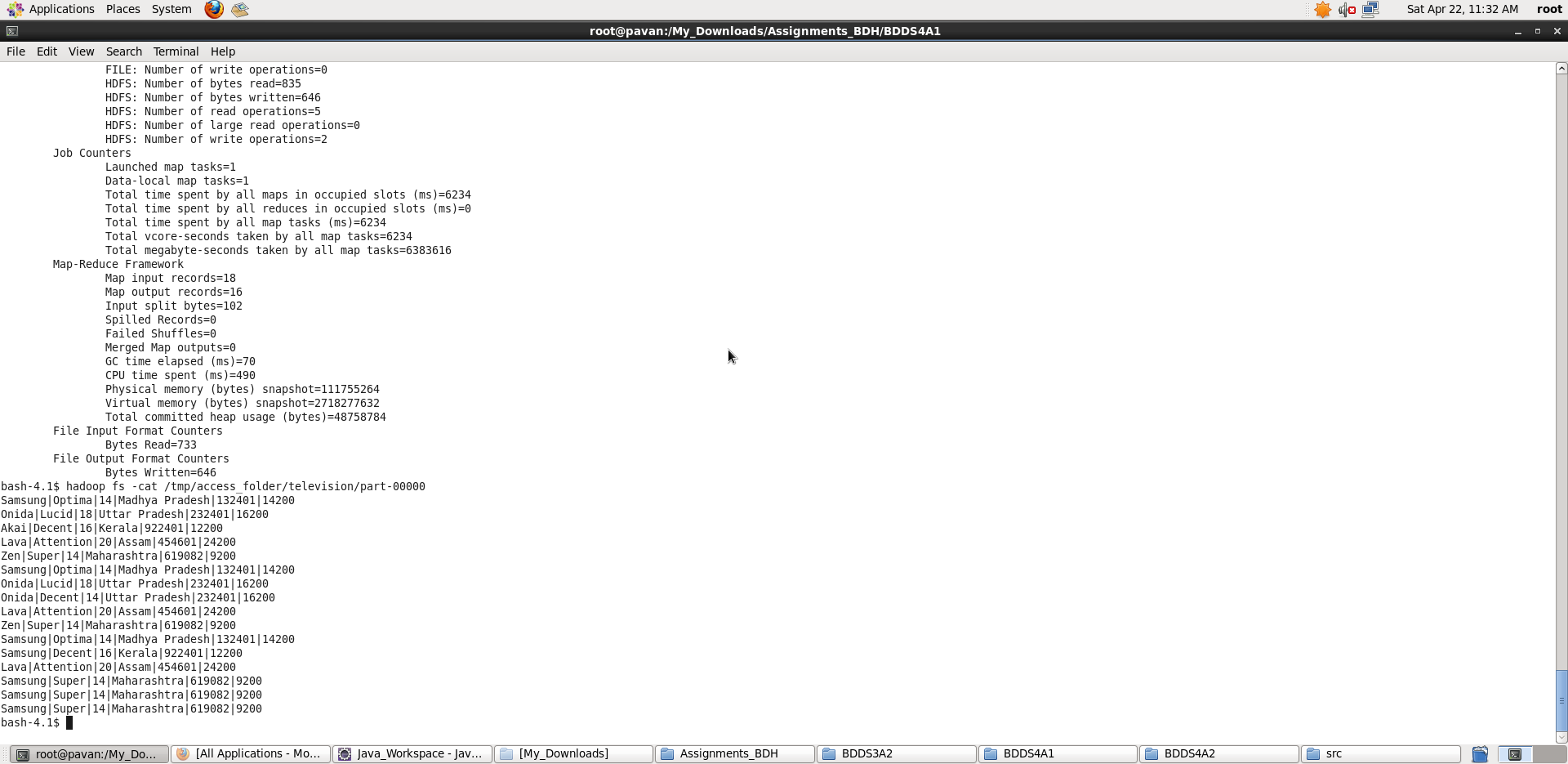
Once we **Run**, we see a **part-file generated.** Displaying **filtered output** using hadoop fs –cat command:



**Submitting Job as HDFS user:**

****

Once we **Run the Hadoop Job**, we see a **part-file generated.** Displaying **filtered output** using hadoop fs –cat command:

****

**Thus, we achieved filtering out the invalid records from the given data\_set.**