**Lab 2 big data**

**Student: Zeyad Mohamed ID:21ztem**

1. **Create inputdata folder**

Graphical user interface, text, application, email

Description automatically generated

1. **Upload stock.csv file to inputdata folder which we created in step 1**

**Graphical user interface, application

Description automatically generated**

1. **Create inputdata and outputdata folder inside hdfs and gave all users all permissions.**

**Text

Description automatically generated**

1. **Create copy the content of stock.csv from inputdata folder in home folder into inputdata folder in hdfs.**

****

1. **Create jarfiles folder to upload all assets folder inside it.**

**Graphical user interface, text, application

Description automatically generated**

1. **Upload jar and java files.**

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **Create MaxCloseDriver class**

**package** Mprd;

**import** java.io.IOException;

**import** org.apache.hadoop.conf.Configuration;

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

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

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

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

**import** org.apache.hadoop.mapreduce.Job;

**import** org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

**import** org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

**public** **class** MaxCloseDriver {

**public** **static** **void** main(String[] args) **throws** IOException, ClassNotFoundException, InterruptedException {

Configuration conf = **new** Configuration();

Job job = **new** Job(conf);

job.setJarByClass(MaxCloseDriver.**class**);

job.setMapperClass(MaxCloseMapper.**class**);

job.setNumReduceTasks(1);

job.setReducerClass(MaxCloseReducer.**class**);

//common

job.setOutputKeyClass(Text.**class**); //word//Text

job.setOutputValueClass(FloatWritable.**class**); //3//IntWritable

FileInputFormat.*addInputPath*(job, **new** Path(args[0]));

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

FileSystem fs = FileSystem.*get*(conf);

fs.~~delete~~(**new** Path(args[1]));

job.waitForCompletion(**true**);

}

}

**Graphical user interface, text, application

Description automatically generated**

1. **Create MaxCloseMapper class**

**package** Mprd;

**import** java.io.IOException;

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

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

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

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

**public** **class** MaxCloseMapper **extends** Mapper<LongWritable, Text, Text, FloatWritable>{

@Override

**protected** **void** map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, FloatWritable>.Context context)

**throws** IOException, InterruptedException {

String inputLine = value.toString();

String[] wordArray = inputLine.split(",");

String sticker\_value = wordArray[0];

String close\_value = wordArray[5];

context.write(**new** Text(sticker\_value), **new** FloatWritable (Float.*parseFloat*(close\_value)));

}

}

**Graphical user interface, text, application, email

Description automatically generated**

1. **Create MaxCloseReducer class**

**package** Mprd;

**import** java.io.IOException;

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

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

**import** org.apache.hadoop.mapreduce.Reducer.

**import** java.util.Iterator.

**public** **class** MaxCloseReducer **extends** Reducer<Text, FloatWritable, Text, FloatWritable>{

@Override

**protected** **void** reduces(Text key, Iterable<FloatWritable> values,

Reducer<Text, FloatWritable, Text, FloatWritable>.Context context) **throws** IOException, InterruptedException {

**float** max\_close = 0;

Iterator<FloatWritable> iterator = values.iterator();

**while**(iterator.hasNext())

{

**float** value = iterator.next().get();

**if**(value>max\_close)

{

max\_close = value;

}

}

context.write(**new** Text(key), **new** FloatWritable(max\_close));

}

}

Graphical user interface, text, application, email

Description automatically generated

1. **Export all created classed with two jar files as one jar file called lab\_2.jar**
2. **Execute lab\_2.jar file and save the output file in the outputdata folder in hdfs**

**Text, letter

Description automatically generated**

1. **Read the output**

**Text

Description automatically generated**

File output:

AAPL 182.01

ADBE 688.37

AMD 161.91

IMPP 7.5

QCOM 189.28