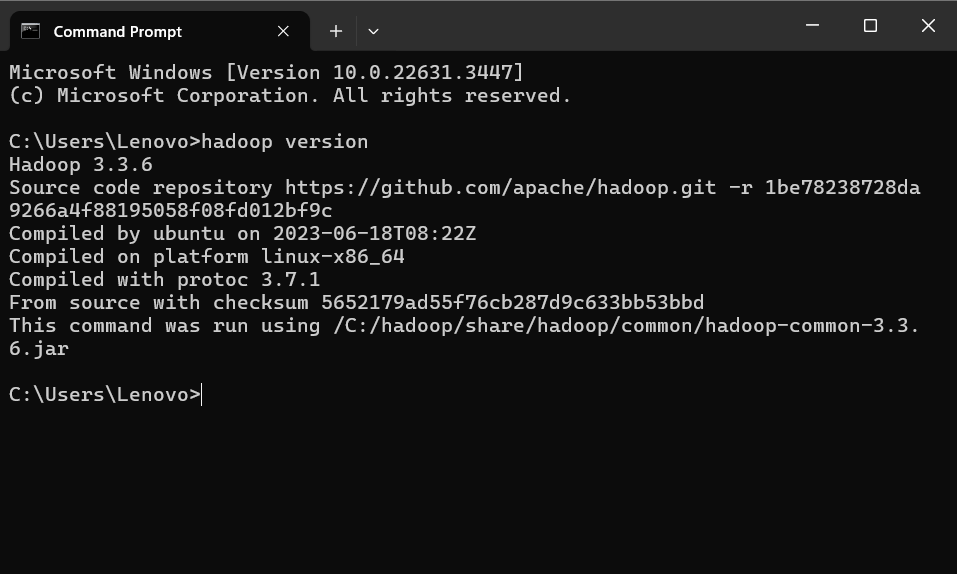
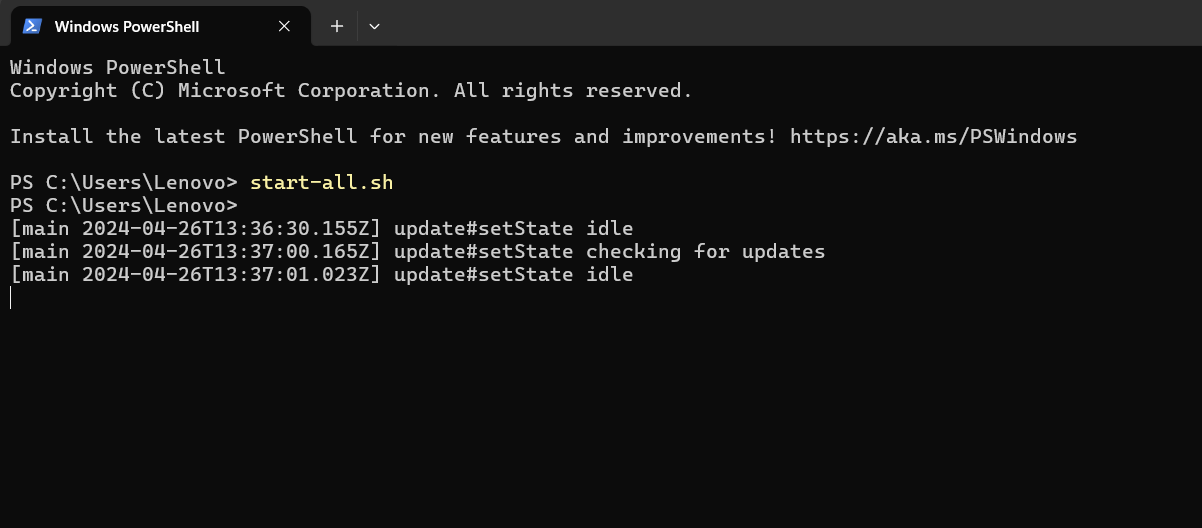
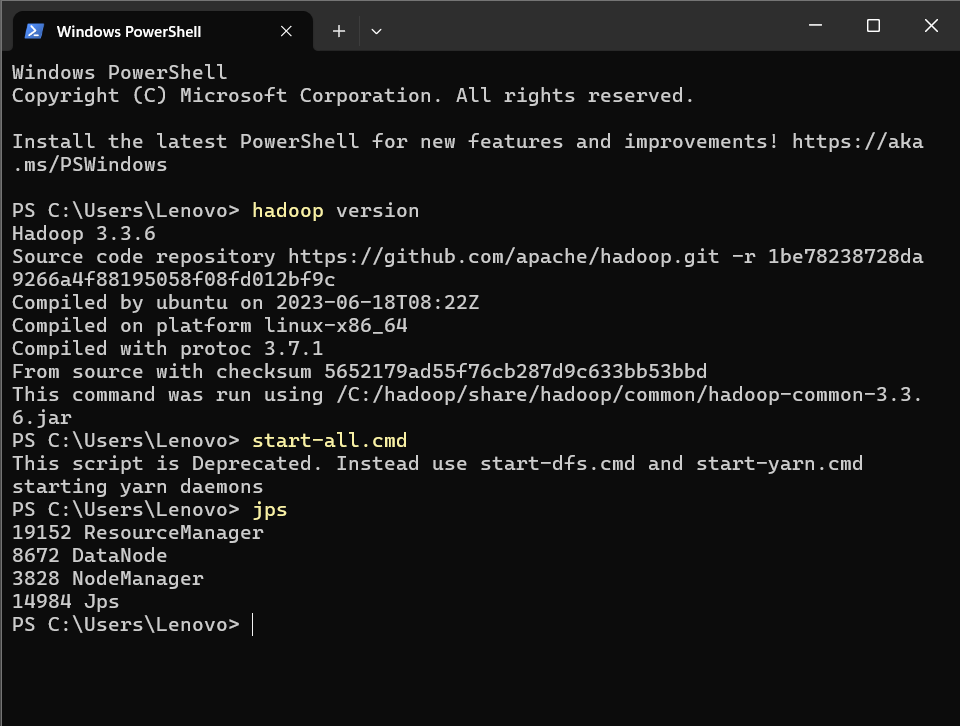
**POLI VARDHINI REDDY**

**BUSINESS INTELLIGENCE USING HADOOP EXAMPLE**

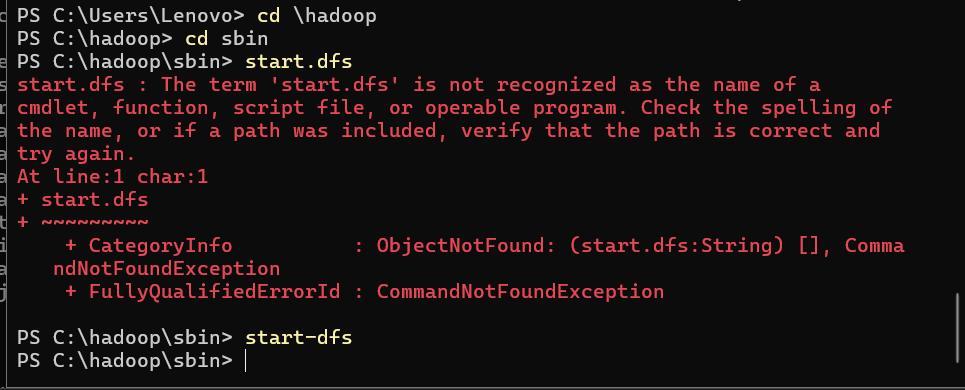
Check Hadoop is installed or not

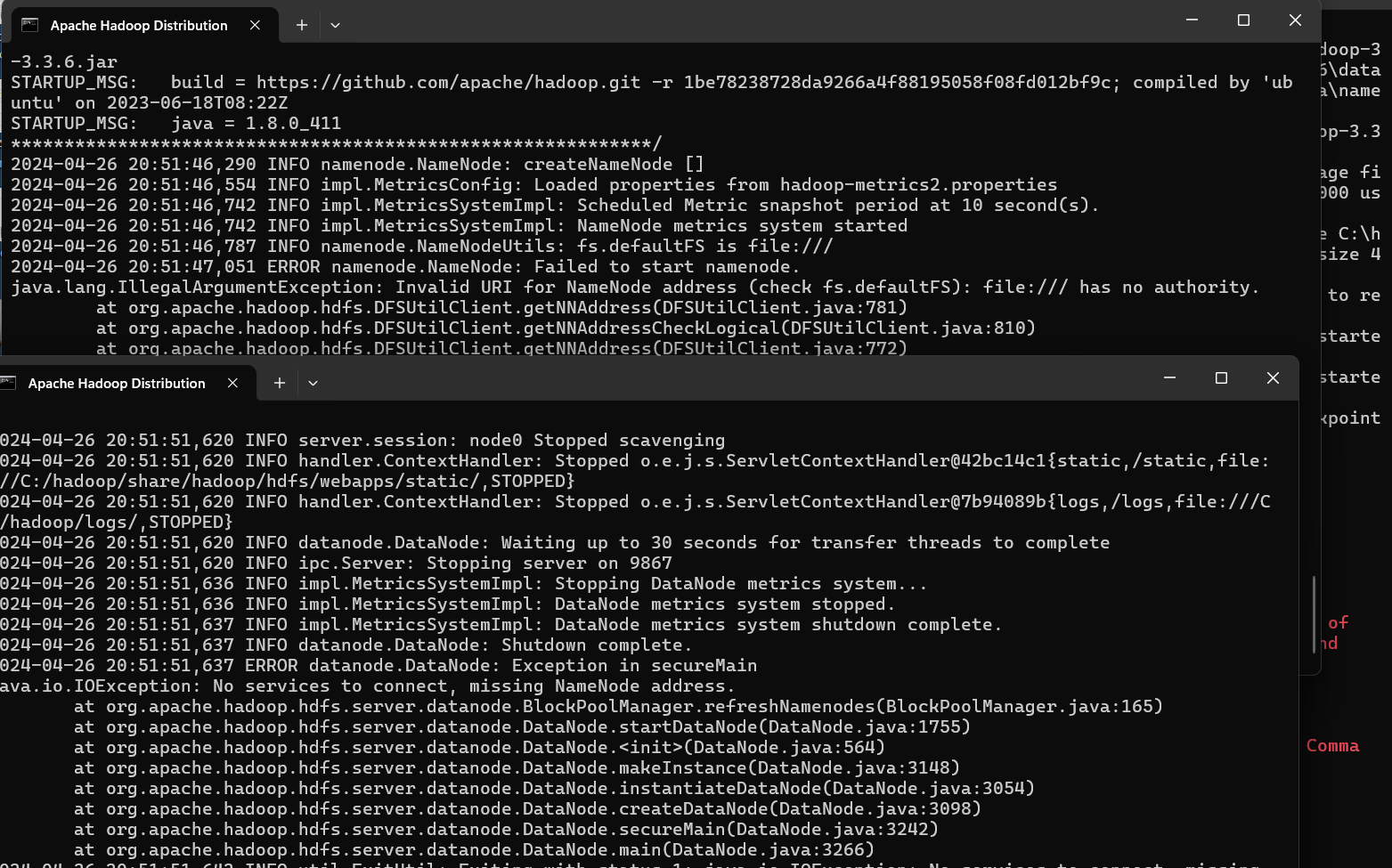


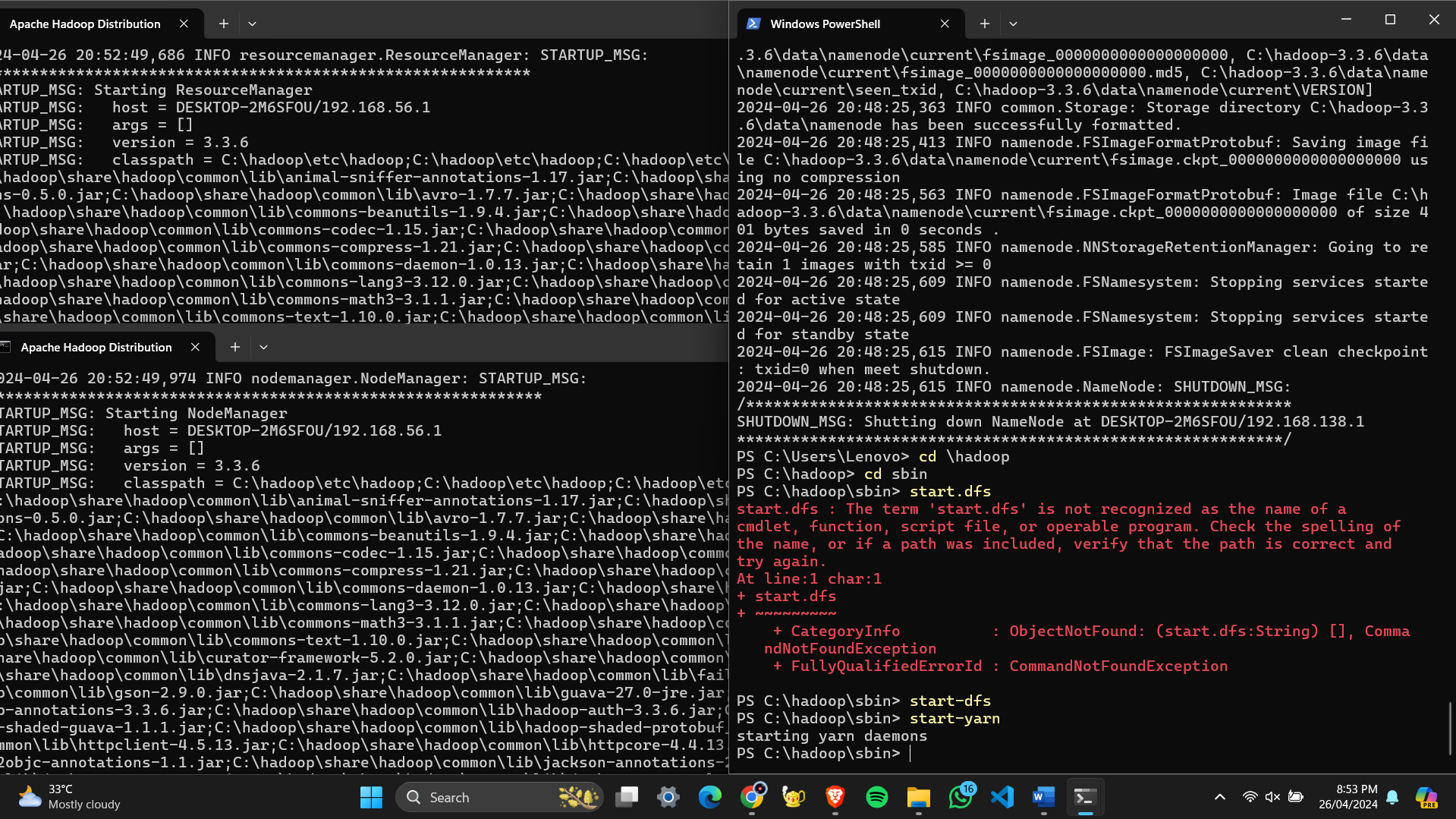


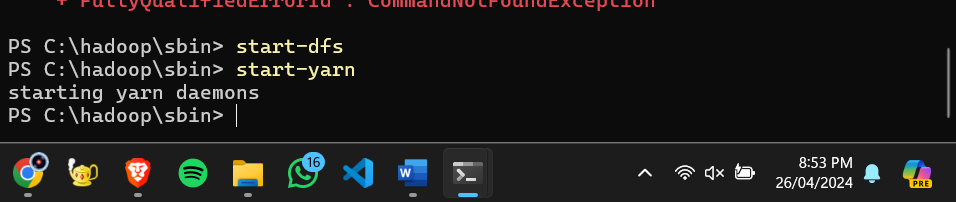


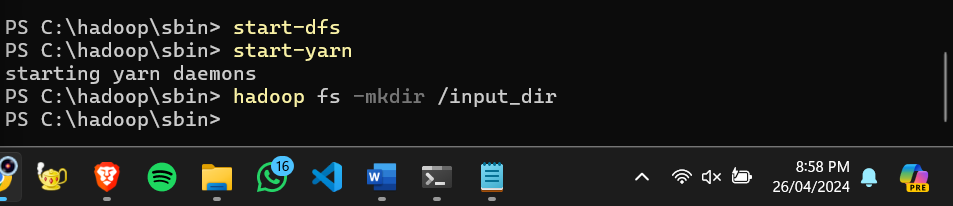


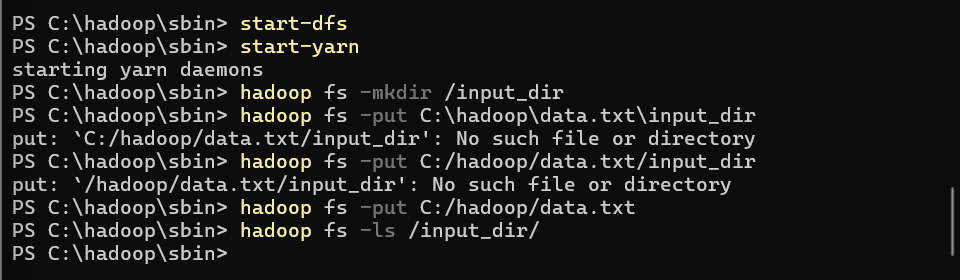


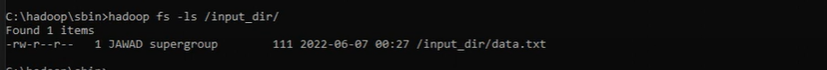


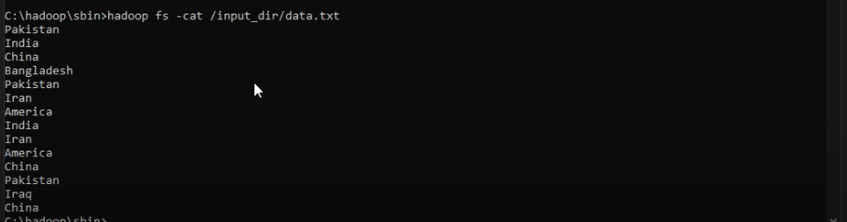


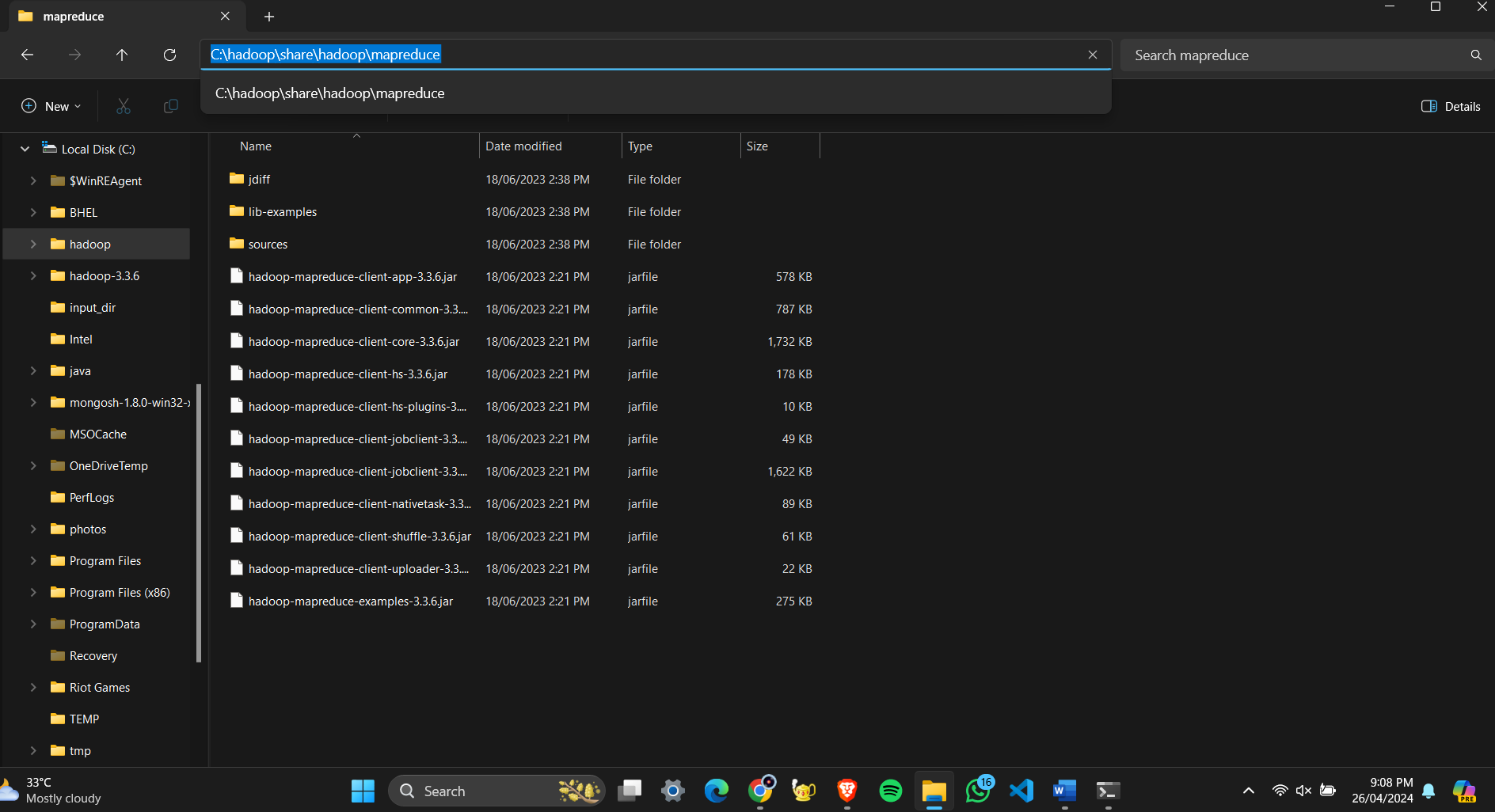


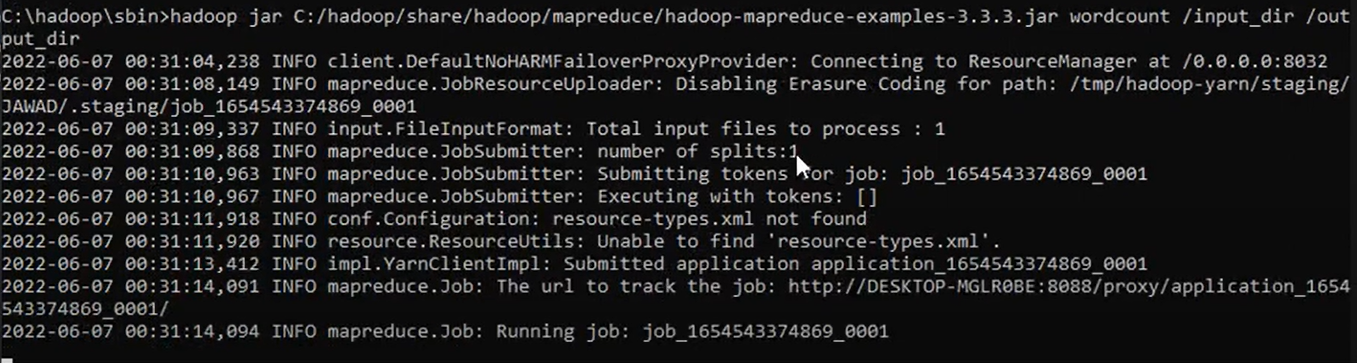




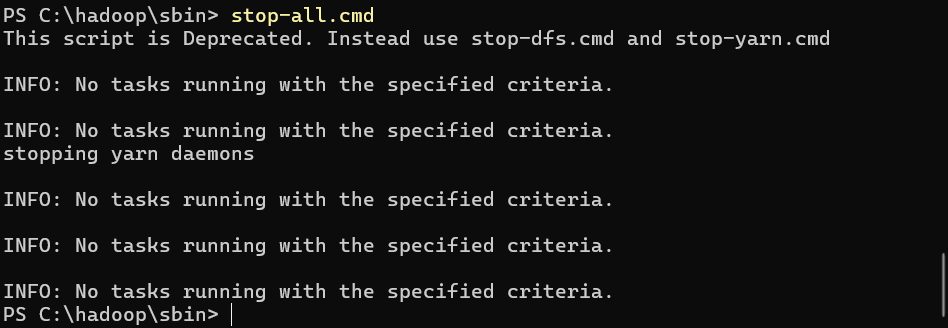












**BUSINESS INTELLIGENCE USING HADOOP EXAMPLE**

**Business intelligence (BI) using Hadoop involves processing and analyzing large volumes of data to extract meaningful insights and make informed business decisions. Hadoop provides a scalable and distributed framework for storing, processing, and analyzing big data. In Java, you can develop BI applications using Hadoop's MapReduce paradigm or higher-level frameworks like Apache Hive or Apache Spark.**

**CODE**

MAPPER CLASS:

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

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

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

**import java.io.IOException;**

**public class SalesMapper extends Mapper<LongWritable, Text, Text, DoubleWritable> {**

**@Override**

**public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {**

**// Assuming each line in the input represents a sales transaction in the format: product\_name,sales\_amount**

**String[] parts = value.toString().split(",");**

**if (parts.length == 2) {**

**String product = parts[0];**

**double salesAmount = Double.parseDouble(parts[1]);**

**context.write(new Text(product), new DoubleWritable(salesAmount));**

**}**

**}**

**}**

**REDUCER CLASS**

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

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

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

**import java.io.IOException;**

**public class SalesReducer extends Reducer<Text, DoubleWritable, Text, DoubleWritable> {**

**@Override**

**public void reduce(Text key, Iterable<DoubleWritable> values, Context context) throws IOException, InterruptedException {**

**double totalSales = 0;**

**for (DoubleWritable value : values) {**

**totalSales += value.get();**

**}**

**context.write(key, new DoubleWritable(totalSales));**

**}**

**}**

**MAIN CLASS**

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.DoubleWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class SalesAnalysis {

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

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "sales analysis");

job.setJarByClass(SalesAnalysis.class);

job.setMapperClass(SalesMapper.class);

job.setReducerClass(SalesReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(DoubleWritable.class);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

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

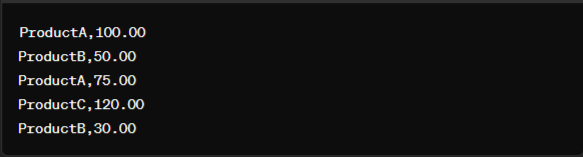
TextOutputFormat.setOutputPath(job, new Path(args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

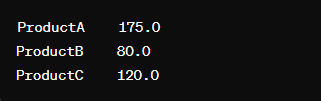
}

}

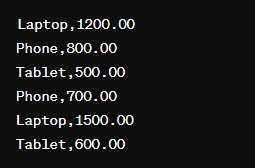
**INPUT:**

****

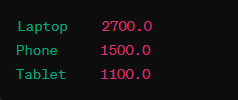
**OUTPUT:**

****

**INPUT:**

****

**OUTPUT:**

****