**Mapreduce\_ChurnAnalysis**

This document has the following contents.

1. Original dataset used for this assignment.
2. Renamed dataset used for this assignment.
3. Java classes code
   1. ChurnMapper
   2. ChurnReducer
   3. ChurnAnalysis
4. Log4j-1.2.17 jar file
5. Output file in csv format
6. Tableau file for results evaluation
7. **Original Dataset**

Kaggle link: <https://www.kaggle.com/blastchar/telco-customer-churn>



1. **Renamed Dataset**



1. **Java Classes**

**3.1 ChurnMapper.java**

|  |
| --- |
| **package** org.myorg;  **import** **java.io.IOException**;  **import** **org.apache.hadoop.io.IntWritable**;  **import** **org.apache.hadoop.io.LongWritable**;  **import** **org.apache.hadoop.io.Text**;  **import** **org.apache.hadoop.mapreduce.Mapper**;  **public** **class** **ChurnMapper** **extends** Mapper<LongWritable, Text, Text, IntWritable> {    **private** **final** **static** IntWritable one = **new** IntWritable(**1**);  **private** Text KEY = **new** Text();  **@Override**  **public** **void** **map**(LongWritable key, Text value, Context context)  **throws** IOException, InterruptedException {  String line = value.toString();  **if** (!line.startsWith("customerID")){  String[] fields = line.split(",");  String result = fields[**15**].concat(",").concat(fields[**4**]).concat(",").concat(fields[**20**]);  KEY.set(result);  context.write(KEY,one);  }  }  } |

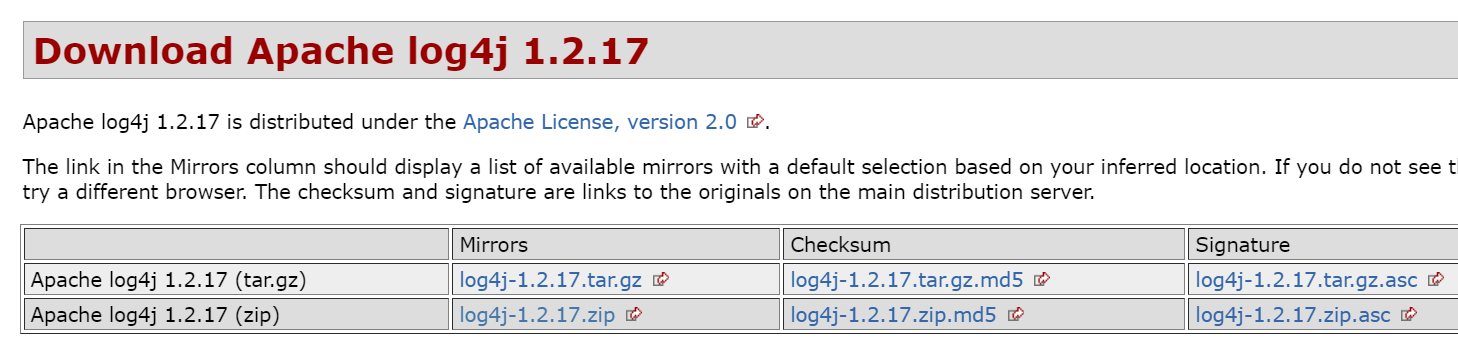
**3.2 ChurnReducer.java**

|  |
| --- |
| **package** org.myorg;  **import** **java.io.IOException**;  **import** **org.apache.hadoop.io.IntWritable**;  **import** **org.apache.hadoop.io.Text**;  **import** **org.apache.hadoop.mapreduce.Reducer**;  **public** **class** **ChurnReducer** **extends** Reducer<Text, IntWritable, Text, IntWritable> {  **@Override**  **public** **void** **reduce**(Text key, Iterable<IntWritable> values, Context context)  **throws** IOException, InterruptedException {    **int** sum = **0**;    **for** (IntWritable value : values){  sum += value.get();  }    context.write(key, **new** IntWritable(sum));  }  } |

**3.3 ChurnAnalysis.java**

|  |
| --- |
| **package** org.myorg;  **import** **org.apache.hadoop.conf.Configuration**;  **import** **org.apache.hadoop.fs.Path**;  **import** **org.apache.hadoop.io.IntWritable**;  **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**;  **import** **org.apache.hadoop.fs.FileSystem**;  **public** **class** **ChurnAnalysis** {  **public** **static** **void** **main**(String[] args) **throws** Exception {  Configuration conf = **new** Configuration();  conf.set("mapreduce.output.textoutputformat.separator", ",");    **if** (args.length != **2**) {  System.out.printf("Usage: ChurnAnalysis <input dir> <output dir>\n");  System.exit(-**1**);  }    Job job;  job = Job.getInstance(conf,"Telecommunication Churn Analysis");  job.setJarByClass(ChurnAnalysis.class);    FileInputFormat.addInputPath(job, **new** Path(args[**0**]));  FileOutputFormat.setOutputPath(job, **new** Path(args[**1**]));    job.setMapperClass(ChurnMapper.class);  job.setReducerClass(ChurnReducer.class);  job.setCombinerClass(ChurnReducer.class);    job.setOutputKeyClass(Text.class);  job.setOutputValueClass(IntWritable.class);    //Delete Output If Exist  FileSystem hdfs = FileSystem.get(conf);  Path outputDir = **new** Path(args[**1**]);  **if** (hdfs.exists(outputDir)){  hdfs.delete(outputDir, **true**);  }    **boolean** success = job.waitForCompletion(**true**);  System.exit(success ? **0** : **1**);  }  } |

1. **Steps to download the log4j jar**
2. Login to URL : <https://logging.apache.org/log4j/1.2/download.html>
3. Click on the Apache log4j 1.2.17 (zip) file.



1. It navigates to the following location.

<https://www.apache.org/dyn/closer.cgi/logging/log4j/1.2.17/log4j-1.2.17.zip>

1. Download the below:

Text

Description automatically generated



1. **Output file in csv format**

****

1. **Tableau file**

****