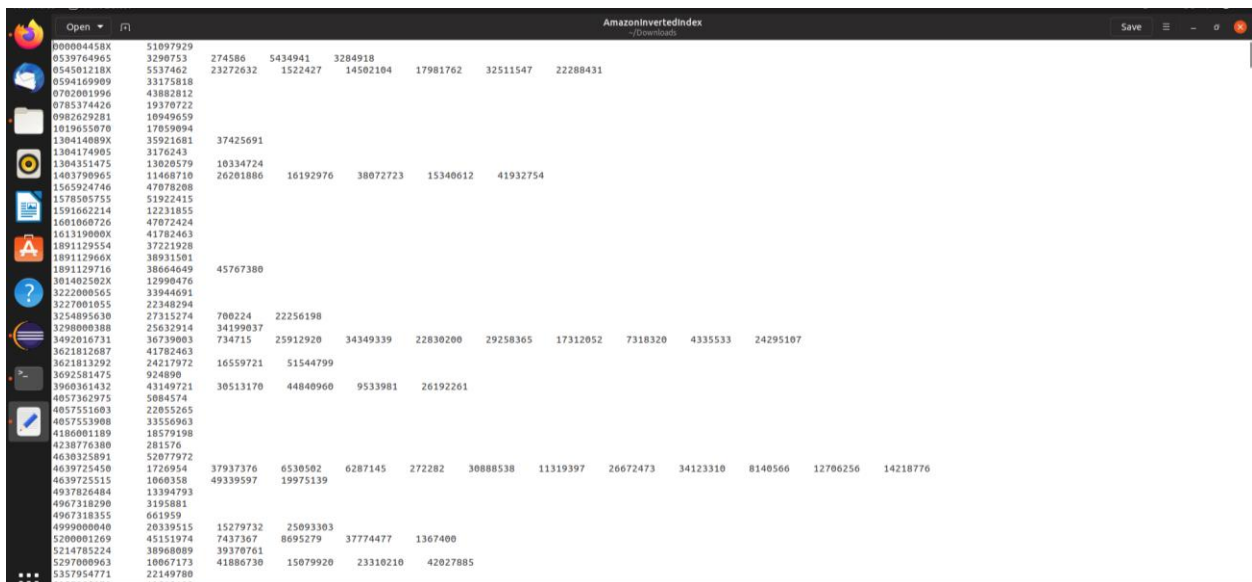# Amazon Inverted Index

(Numerical Summarization Pattern)

In this analysis, Inverted Indexing is used. It is very helpful as the output can show what Product are reviewed by which users. The key value will have multiple values.
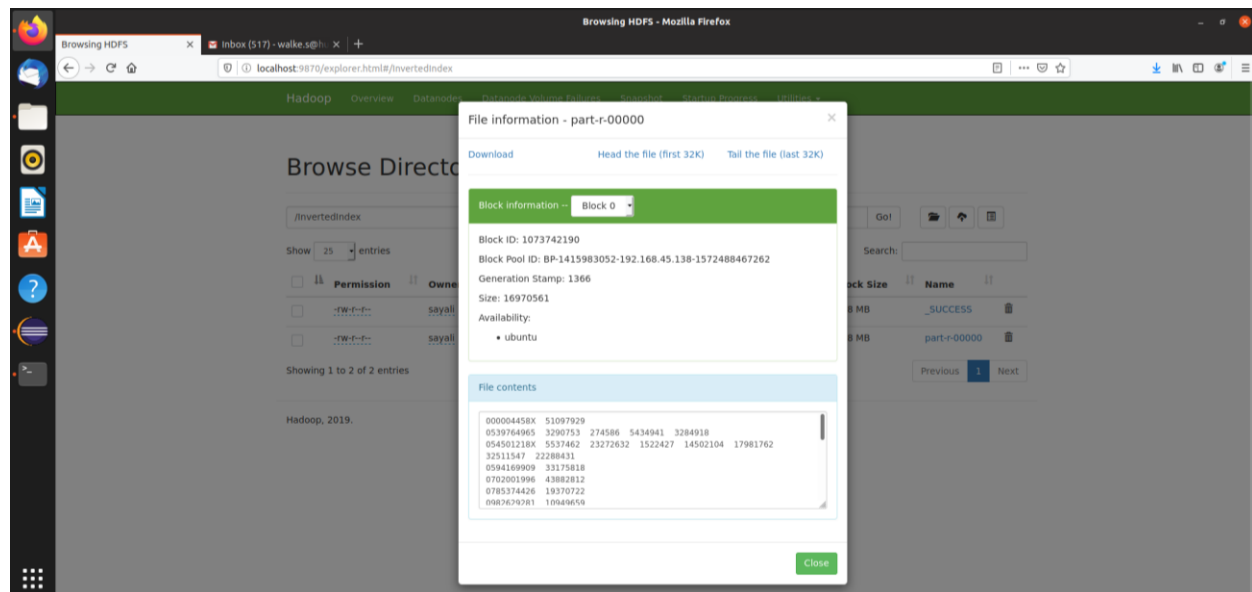
OUTPUT:

./hadoop jar /home/sayali/Desktop/AmazonInvertedIndex.jar
sayali.AmazonReviews2.AmazonInvertedIndex /AmazonReviews/AmazonReviews.tsv /InvertedIndex

Map Reduce Code

```java
import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

Mapper:

```java
public class AmazonInvertedIndexMapper extends Mapper<LongWritable, Text, Text, Text> {
        private Text product = new Text();
        private Text customerId = new Text();

        public void map(LongWritable key, Text values, Context context) throws IOException {

                if (key.get() == 0) {
                        return;
                }
                String[] tokens = values.toString().split("\\t");
                customerId.set(tokens[1]);
                product.set(tokens[3]);
                try {
                        context.write(product, customerId);
                } catch (InterruptedException e) {
                        e.printStackTrace();
                }
        }
}
```

Reducer:

```java
public class AmazonInvertedIndexReducer extends Reducer<Text, Text, Text, Text> {
        private Text result = new Text();

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

                StringBuilder sb = new StringBuilder();
                boolean first = true;

                for (Text id : values) {
                        if (first) {
```

```
                                        first = false;
                        } else {
                                sb.append("   ");


                        }
                        sb.append(id.toString());
                }
                result.set(sb.toString());
                context.write(key, result);
        }
}
```
Main:
```java
public class AmazonInvertedIndex {

    public static void main(String[] args)throws IOException, InterruptedException,
ClassNotFoundException {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf,"Inverted Index");
        job.setJarByClass(AmazonInvertedIndex.class);
        job.setMapperClass(AmazonInvertedIndexMapper.class);
        job.setReducerClass(AmazonInvertedIndexReducer.class);

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(Text.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```