## Analysis 05 – Implement partitioning on the basis on year the Movies and TV shows added in Netflix dataset

|  |  |
| --- | --- |
| **public** **class** YearMapper **extends** Mapper<LongWritable, Text, IntWritable, Text> {  **private** IntWritable outKey = **new** IntWritable();  **public** **void** map(LongWritable key, Text value, Context context) **throws** IOException, InterruptedException {  String input = value.toString();  // Split data on comma and ignore comma within double quotes  String[] movieData = input.split(",(?=(?:[^\"]\*\"[^\"]\*\")\*[^\"]\*$)", -1);  **try** {  String dateAdded = movieData[6].replace("\"", "");    **if** (dateAdded.equals("date\_added"))  **return**;  String year = dateAdded.split(",")[1];  **if**(!year.isEmpty()) {  **int** yearAdded = Integer.*parseInt*(year.replace(" ", ""));  outKey.set(yearAdded);  }  } **catch** (Exception ex) {  System.***out***.println(ex);  }  context.write(outKey, value);  }  } | |
| **public** **class** YearReducer **extends** Reducer<IntWritable, Text, Text, NullWritable> {  **protected** **void** reduce(IntWritable key, Iterable<Text> values, Context context)  **throws** IOException, InterruptedException {    **for** (Text val : values) {  context.write(val, NullWritable.*get*());  }    }  } | |
| **public** **class** YearPartition **extends** Partitioner<IntWritable, Text> {  @Override  **public** **int** getPartition(IntWritable key, Text value, **int** numPartition) {  **return** (key.hashCode()) % numPartition;  }  } | |
| **public** **class** Driver {  **public** **static** **void** main(String[] args) **throws** IOException, ClassNotFoundException, InterruptedException {  Configuration conf = **new** Configuration();  Job job = Job.*getInstance*(conf);  job.setJarByClass(Driver.**class**);  FileSystem fs = FileSystem.*get*(conf);  **if** (fs.exists(**new** Path(args[1]))) {  fs.delete(**new** Path(args[1]), **true**);  }  job.setNumReduceTasks(20);  job.setMapperClass(YearMapper.**class**);  job.setReducerClass(YearReducer.**class**);  job.setPartitionerClass(YearPartition.**class**);  TextInputFormat.*addInputPath*(job, **new** Path(args[0]));  TextOutputFormat.*setOutputPath*(job, **new** Path(args[1]));  job.setMapOutputKeyClass(IntWritable.**class**);  job.setMapOutputValueClass(Text.**class**);  job.setOutputKeyClass(Text.**class**);  job.setOutputValueClass(NullWritable.**class**);  System.*exit*(job.waitForCompletion(**true**) ? 0 : 1);  }  } | |
|  |
|  |