Lab - 29.1.2018

1. Custom Partitioning

An organization maintains the salary details that contains years of experience,salary, gender and country name. Partition the file based on the salary. Create 3 partitions with salary <30000, salary<50000 and salary details other than the above two conditions. Write the country name and the salary details in the files.

Note

1. Create a map class. Write the country name and the salary.
2. Create the partitioner class as below:

("nr" is user defined variable name, which receives from number of reducers to create the partitions)

```
public static class dpart extends Partitioner<Text,IntWritable>
  {
     public int getPartition(Text key,IntWritable value,int nr)
     {
       if(value.get()<30000)
            return 0;
       if(value.get() < 50000)
            return 1;
        else
            return 2;
        } }
```

3. Create the reducer and write the contents

```
	context.write(key, value);
```

4. In the driver include the following with the other details:

```
  job.setPartitionerClass(dpart.class);
  job.setNumReduceTasks(3);
```

2. Number of mappers-split function

An organization maintains the salary details that contains years of experience,salary, gender and country name. Write the country name and the salary details in hdfs.

A) User defined number of mappers

```
conf.set("mapred.min.split.size","10000");
```

B) If the whole file have to run in one mapper: include the following class.

```
public class splitfalse extends TextInputFormat {
		protected boolean isSplitable(JobContext
context, Path file) {
        return false;
        }
        }
```

Include the below in driver(main)

```
job.setInputFormatClass(splitfalse.class);
```