Titanic dataset

**Objective:**

Find the average age of males and females who died in the Titanic tragedy.

**Mapper:**

**public** **class** TaskMapper **extends** Mapper<LongWritable, Text, Text, IntWritable> {

Text gender=**new** Text();

IntWritable age=**new** IntWritable();

**public** **void** map(LongWritable key, Text value, Context context)

**throws** IOException, InterruptedException {

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

**if**(lineArray[1].equals("1") && isNotEmpty(lineArray[5]) && lineArray[5].matches("\\d+"))

{

**if**(lineArray[4].equalsIgnoreCase("male"))

{

gender.set("male");

age.set(Integer.*parseInt*(lineArray[5]));

context.write(gender, age);

}

**else**

{

gender.set("female");

age.set(Integer.*parseInt*(lineArray[5]));

context.write(gender, age);

}

}

}

**public** **boolean** isNotEmpty(String s)

{

**return** s!=**null** && !s.equals("") && s.length()!=0;

}

}

**Reducer:**

**public** **class** TaskReducer **extends** Reducer<Text, IntWritable, Text, IntWritable>

{

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

{

**int** totalAge=0;

**int** count=0;

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

count++;

totalAge=totalAge+value.get();

}

**int** avgAge=totalAge/count;

context.write(key, **new** IntWritable(avgAge));

}

}

**Job:**

job.setJarByClass(Task1.**class**);

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

job.setMapOutputValueClass(IntWritable.**class**);

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

job.setOutputValueClass(IntWritable.**class**);

job.setMapperClass(TaskMapper.**class**);

job.setReducerClass(TaskReducer.**class**);

job.setInputFormatClass(TextInputFormat.**class**);

job.setOutputFormatClass(TextOutputFormat.**class**);

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

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

**Run command:**

hadoop jar S5\_A3.jar mapreduce.demo.task1.Task1 /niki/TitanicData.txt /niki/A3\_2

Key will be gender and the value will be the age. All the invalid age values are excluded like null and alphanumeric values if any.

Reducer has logic to take average of the male and females age.

**Output:**

