Saavn Project ReadMe

Summary

Overview

My project was to write and execute a MapReduce program to figure out the top 100 trending songs from Saavn’s stream data on a daily basis, ranging from 25th-31st December.

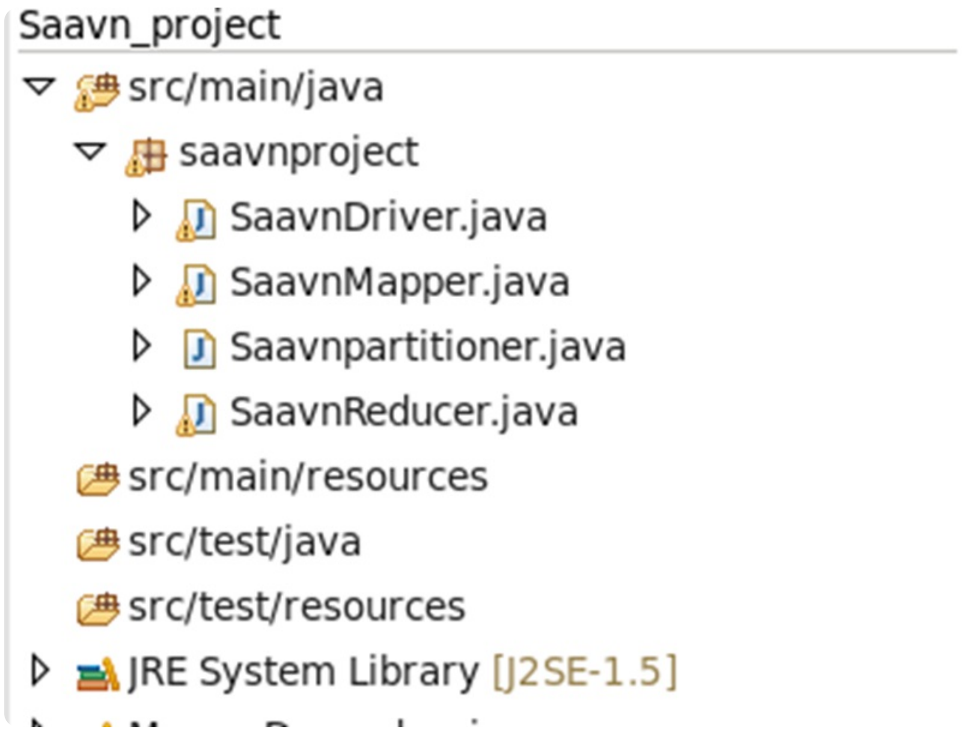
The record was in the format of tuple containing attributes (song ID, user ID, timestamp, hour, date)

Program snapshot and the command to run the code

Sharing the details of the mapreduce program

- Created a mapreduce program with a Driver class, Mapper class and a reducer class.

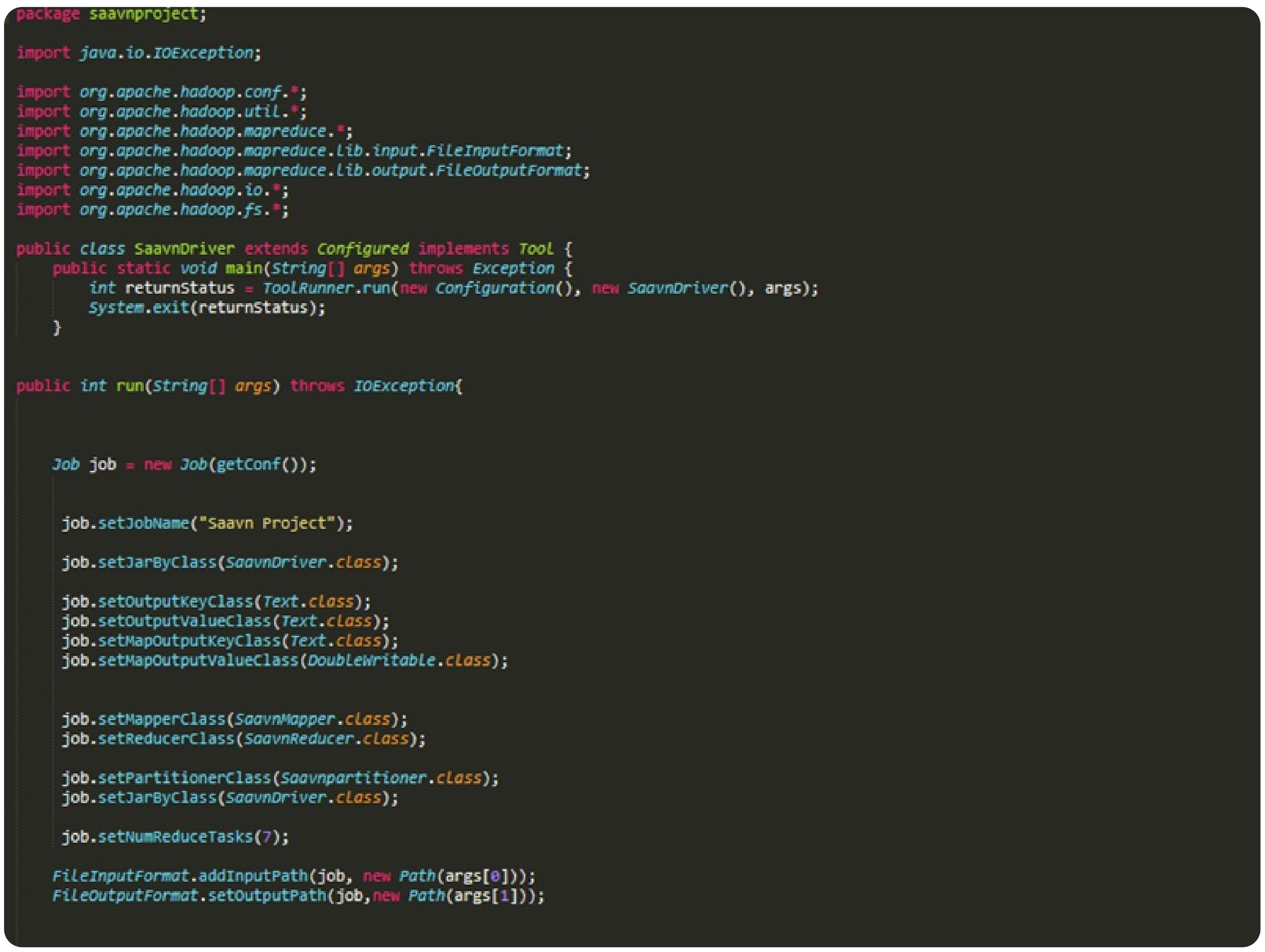
Command used to run the Jar file was
"hadoop jar saavnFinalProject.jar saavnproject.SaavnDriver s3a://mapreduce-project-bde/part-00000 s3a://Saavn-top-100-songs/output4 "



```
Saavn_project
▽ src/main/java
    ▽ saavnproject
        ▷ SaavnDriver.java
        ▷ SaavnMapper.java
        ▷ Saavnpartitioner.java
        ▷ SaavnReducer.java
  src/main/resources
  src/test/java
  src/test/resources
▷ JRE System Library [J2SE-1.5]
```

Driver class

Driver class extended the configuration where all the job configuration was set.



```java
package saavnproject;

import java.io.IOException;

import org.apache.hadoop.conf.*;
import org.apache.hadoop.util.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.io.*;
import org.apache.hadoop.fs.*;

public class SaavnDriver extends Configured implements Tool {
    public static void main(String[] args) throws Exception {
        int returnStatus = ToolRunner.run(new Configuration(), new SaavnDriver(), args);
        System.exit(returnStatus);
    }


public int run(String[] args) throws IOException{



    Job job = new Job(getConf());


     job.setJobName("Saavn Project");

     job.setJarByClass(SaavnDriver.class);

     job.setOutputKeyClass(Text.class);
     job.setOutputValueClass(Text.class);
     job.setMapOutputKeyClass(Text.class);
     job.setMapOutputValueClass(DoubleWritable.class);


     job.setMapperClass(SaavnMapper.class);
     job.setReducerClass(SoavnReducer.class);

     job.setPartitionerClass(Saavnpartitioner.class);
     job.setJarByClass(SaavnDriver.class);

     job.setNumReduceTasks(7);

    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job,new Path(args[1]));
```

Mapper class

In the mapper class, the file was split into an array of string from where songId, date and hours was extracted.

After getting the date field, the date was further split into days. Thereafter, used an 'if statement' to write the data in the context write with respect to the days.

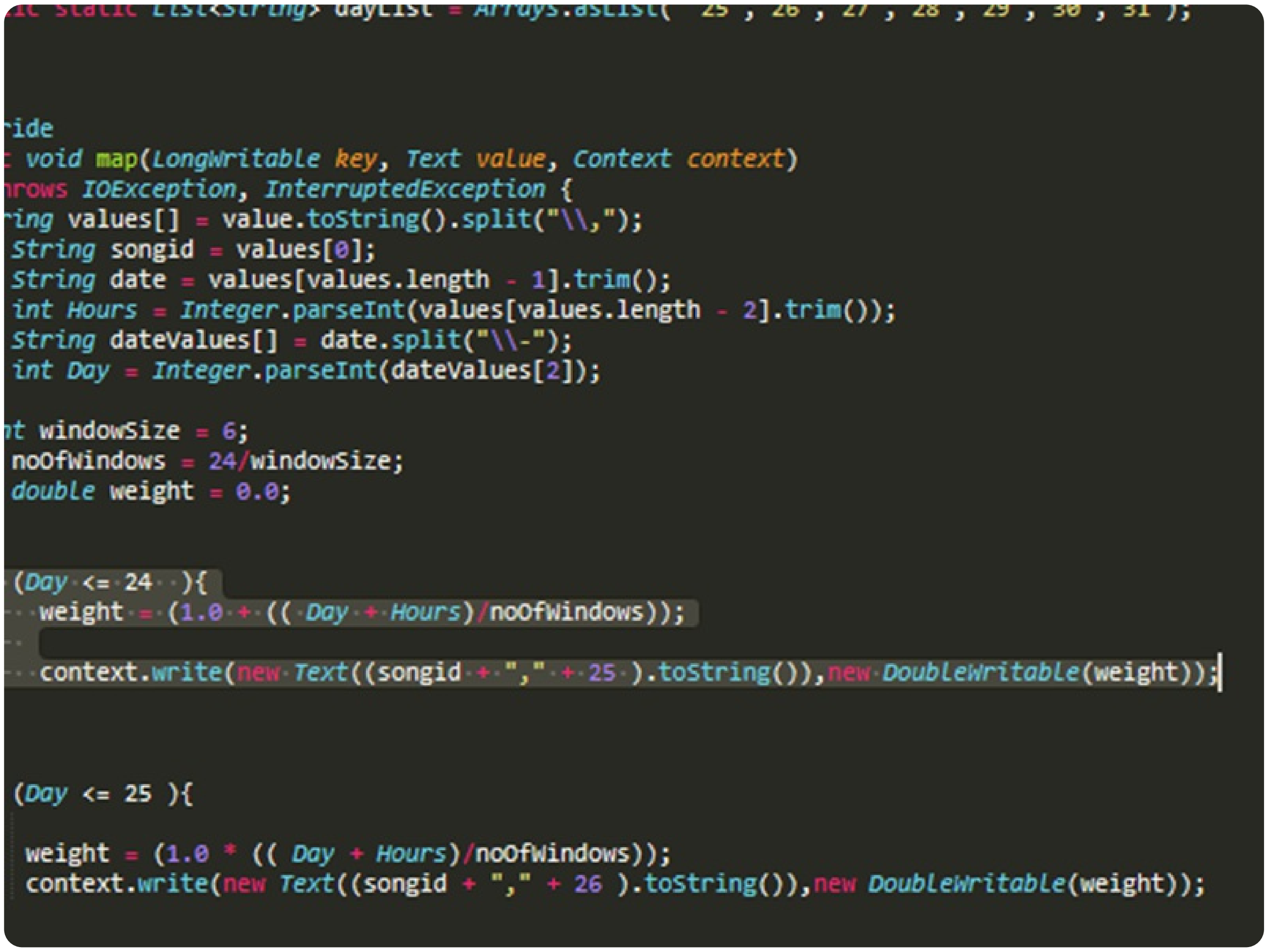
The formula worked to fetch he data of 25th Dec. Here is the formula: **if(days<=24){**
}
This helped me in getting the data of the songs from 25th Dec onward.

Then I followed the decaying window algorithm, to attach the weight of each songid as per the below logic.

To get he window per day, I took the window size of 6 hours, hence the total number of window came out to be 24/6 = 4 windows.

For the weight part, the weight increase on the basis of the day and hour, at which the song got played in the day + a constant value of 1.0.

While writing in context.write, I have attached a partitioner identifier with the songid.
Reference image attached for the explanation.

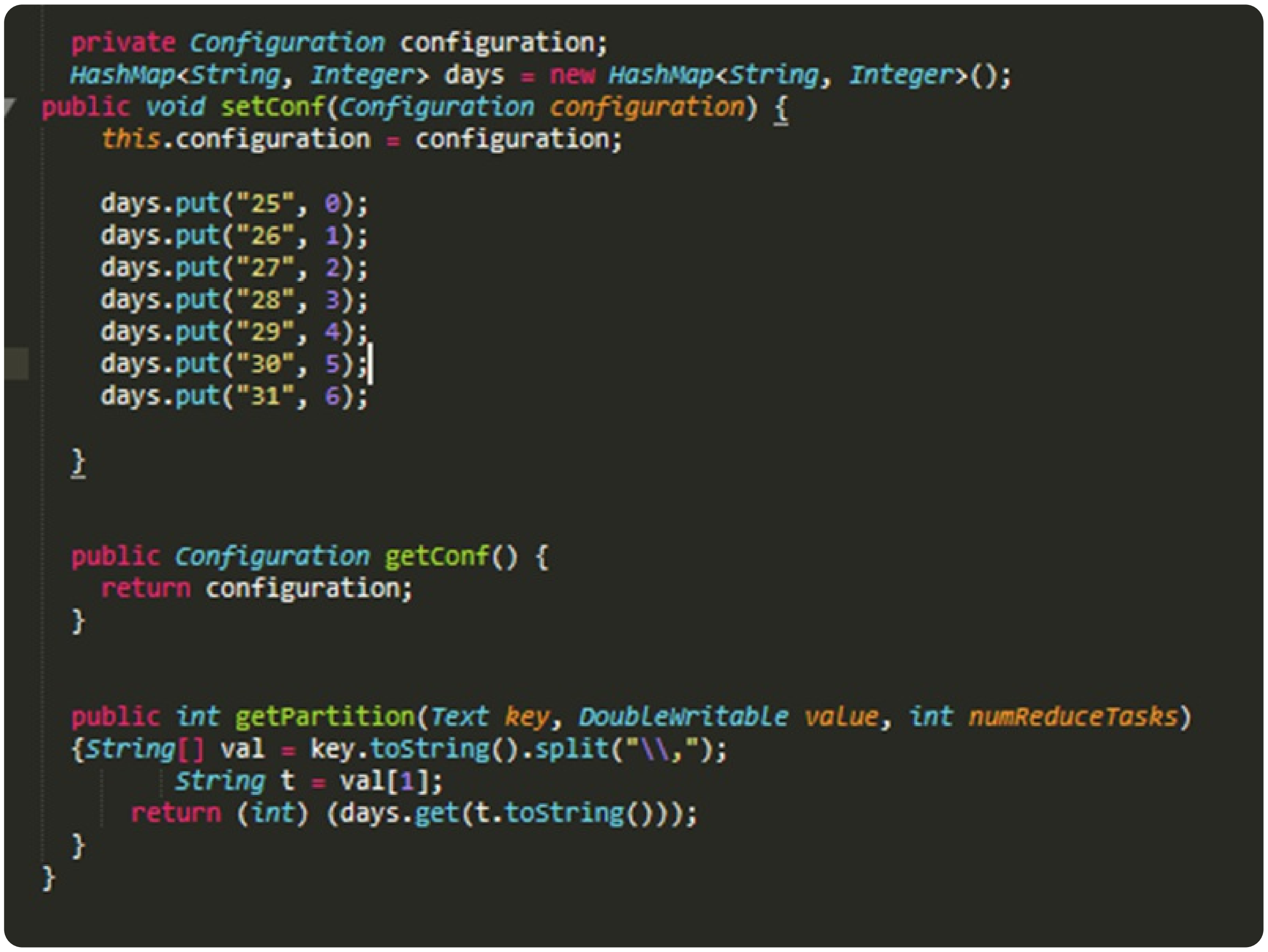


```
lic static List<String> dayList = Arrays.asList( "25","26","27","28","29","30","31" );



ride
: void map(LongWritable key, Text value, Context context)
hrows IOException, InterruptedException {
ring values[] = value.toString().split("\\,");
 String songid = values[0];
 String date = values[values.length - 1].trim();
 int Hours = Integer.parseInt(values[values.length - 2].trim());
 String dateValues[] = date.split("\\-");
 int Day = Integer.parseInt(dateValues[2]);

nt windowSize = 6;
 noOfWindows = 24/windowSize;
 double weight = 0.0;


(Day <= 24  ){
   weight = (1.0 + (( Day + Hours)/noOfWindows));

   context.write(new Text((songid + "," + 25 ).toString()),new DoubleWritable(weight));



(Day <= 25 ){

  weight = (1.0 * (( Day + Hours)/noOfWindows));
  context.write(new Text((songid + "," + 26 ).toString()),new DoubleWritable(weight));
```

Partitioner

In the Partitioner.class, I have created a hashmap with string and integer value.

String value contained the partitioning key E.g "25".

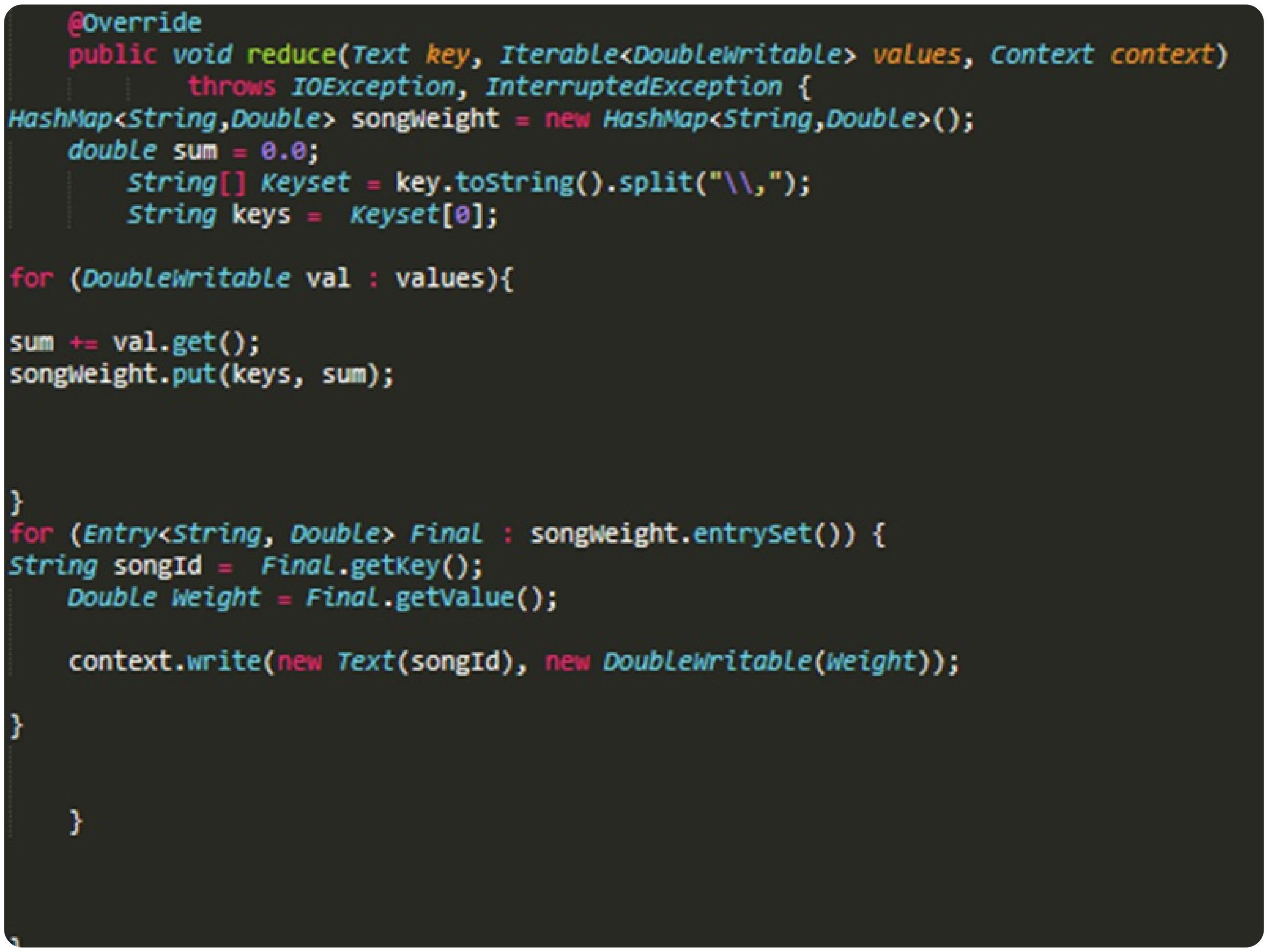


```
  private Configuration configuration;
  HashMap<String, Integer> days = new HashMap<String, Integer>();
public void setConf(Configuration configuration) {
    this.configuration = configuration;

    days.put("25", 0);
    days.put("26", 1);
    days.put("27", 2);
    days.put("28", 3);
    days.put("29", 4);
    days.put("30", 5);
    days.put("31", 6);

  }


  public Configuration getConf() {
    return configuration;
  }


  public int getPartition(Text key, DoubleWritable value, int numReduceTasks)
  {String[] val = key.toString().split("\\,");
          String t = val[1];
      return (int) (days.get(t.toString()));
  }
}
```

Reducer

In the reduce the key was separated from the identifier and the weight of the song was sum together with respect to the key.

and the value was put in a hashmap.

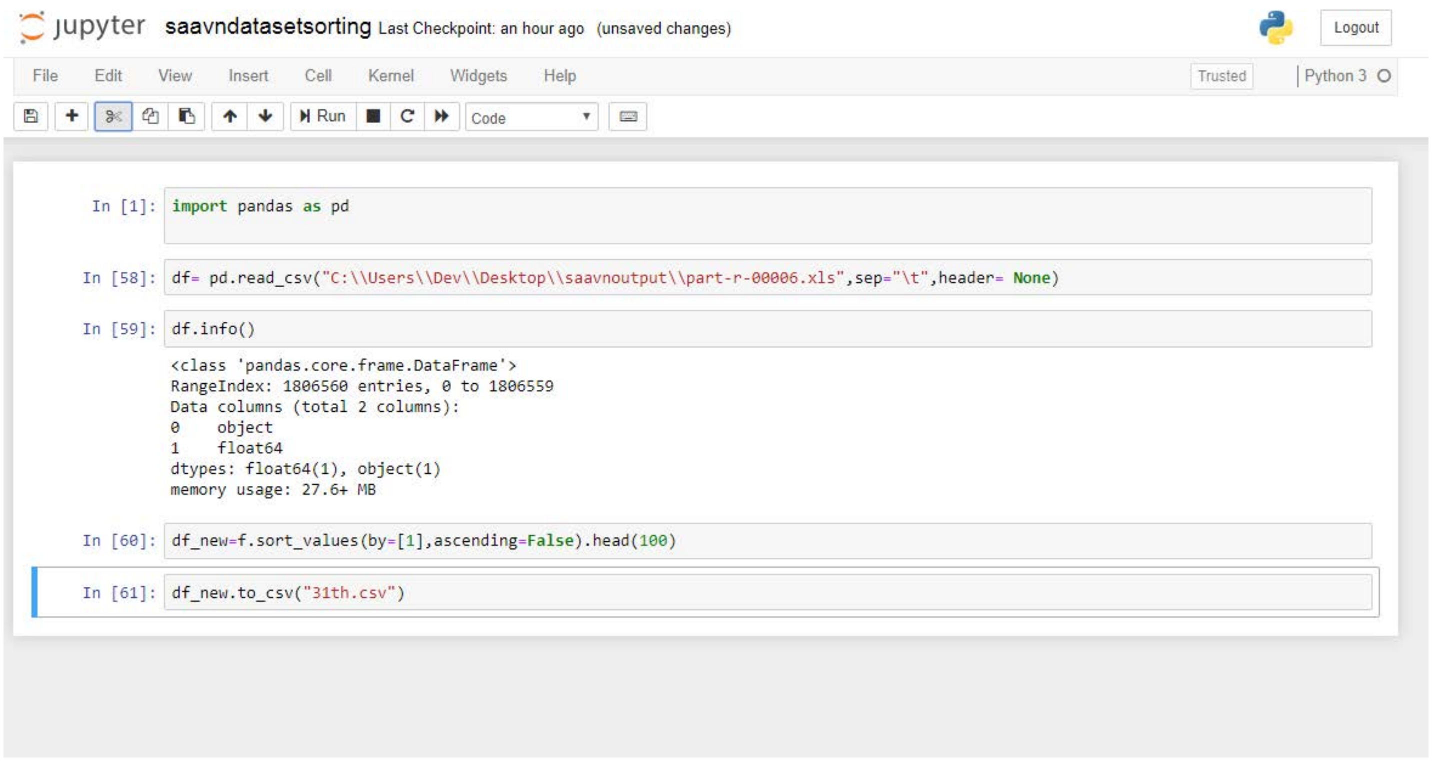


```
    @Override
    public void reduce(Text key, Iterable<DoubleWritable> values, Context context)
            throws IOException, InterruptedException {
HashMap<String,Double> songWeight = new HashMap<String,Double>();
    double sum = 0.0;
        String[] Keyset = key.toString().split("\\,");
        String keys =  Keyset[0];

for (DoubleWritable val : values){

sum += val.get();
songWeight.put(keys, sum);



}
for (Entry<String, Double> Final : songWeight.entrySet()) {
String songId =  Final.getKey();
    Double Weight = Final.getValue();

    context.write(new Text(songId), new DoubleWritable(Weight));

}


    }
```

Step followed to reach the 100 trending song from 25th to 31st output from mapreduce program output

After getting the output in form of 6 different files E.g part-r-00000, part-r-00001 etc.

Converted the files into xls format by adding .xls at the end of the filename and uploaded the data in to jupyter notebook where i used the python to sort the data and taking only top 100 records.


jupyter saavndatasetsorting Last Checkpoint: an hour ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [1]: import pandas as pd

In [58]: df= pd.read_csv("C:\\Users\\Dev\\Desktop\\saavnoutput\\part-r-00006.xls",sep="\t",header= None)

In [59]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1806560 entries, 0 to 1806559
Data columns (total 2 columns):
0    object
1    float64
dtypes: float64(1), object(1)
memory usage: 27.6+ MB

In [60]: df_new=f.sort_values(by=[1],ascending=False).head(100)

In [61]: df_new.to_csv("31th.csv")
```