1.JAVA PROGRAM

import java.io.IOException;  
  
  
import java.util.\*;  
import org.apache.hadoop.conf.Configuration;  
  
import org.apache.hadoop.fs.Path;  
  
  
import org.apache.hadoop.io.\*;  
  
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.input.TextInputFormat;  
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;  
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  
  
public class WordCount {  
  
    public static void main(String [] args) throws Exception  
  
    {  
  
    Configuration c=new Configuration();  
  
      
  
    Path input=new Path(args[0]);  
  
    Path output=new Path(args[1]);  
  
    Job j=Job.getInstance(c, "WordCount");  
  
    j.setJarByClass(WordCount.class);  
  
    j.setMapperClass(MapForWordCount.class);  
  
    j.setReducerClass(ReduceForWordCount.class);  
  
    j.setOutputKeyClass(Text.class);  
  
    j.setOutputValueClass(IntWritable.class);  
  
    FileInputFormat.addInputPath(j, input);  
  
    FileOutputFormat.setOutputPath(j, output);  
  
    System.exit(j.waitForCompletion(true)?0:1);  
  
    }  
  
    public static class MapForWordCount extends Mapper<LongWritable, Text, Text, IntWritable>{  
  
    public void map(LongWritable key, Text value, Context con) throws IOException, InterruptedException  
  
    {  
  
    String line = value.toString();  
    String[] word=line.split(",");  
    String Genre=word[word.length-1];  
    if(Genre.charAt(0)!='(')  
    {  
          
          
          
      
    String []genre=Genre.split("|");  
  
      
      
    for(int i=0;i<genre.length;i++){  
      
      
          Text outputKey = new Text(genre[i]);  
           
           
           
          IntWritable outputValue = new IntWritable(1);  
  
      con.write(outputKey, outputValue);  
    }  
      
    }}  
      
  
    }  
  
      
  
    public static class ReduceForWordCount extends Reducer<Text, IntWritable, Text, IntWritable>  
  
    {  
  
    public void reduce(Text word, Iterable<IntWritable> values, Context con) throws IOException, InterruptedException  
  
    {  
  
    int maxval=0;  
      
  
       for(IntWritable value : values)  
  
       {  
             
             
               maxval+=value.get();  
            
           
    }  
        
        
               con.write(word, new IntWritable(maxval));  
              
    }  
  
}  
      
  
}

2.Data set I chose is the movies dataset

I extracted the number of movies under each genre

3.

Action 3520

Adventure 2329

Animation 1027

Children 1139

Comedy 8374

Crime 2939

Documentary 2471

Drama 13344

Fantasy 1412

Film-Noir 330

Horror 2611

IMAX 196

Musical 1036

Mystery 1514

Romance 4127

Sci-Fi 1743

Thriller 4178

War 1194

Western 676