COMP-SCI 5540 Principles of Big Data Management

University of Missouri-Kansas City

Department of Computer Science and Electrical Engineering

PROJECT 3 REPORT

Team Number:14

Done by:

- Nageswara Rao Nandigam(nrnxh9)
- Revanth Chakilam(rcww4)
- Syed Moin(slrp3)
- Devender Sarda(dspc8)

Tasks:

Main Requirements:

- Using a collection of tweets, implement three analytical tasks. A single task could consist of multiple analytical queries.
  - One task must be implemented via RDD transformations and actions only. Must NOT be a simple word count (e.g. most used language).
  - The other two tasks must be implemented via Spark SQL and DFs.
  - One of your analytical queries must use the input file trends.txt.

Extra Requirement:

- Implement a graphical user interface that enable the user to dynamically execute your analytical tasks and provide a visual representation of the results.
- Flow: the user selects an analytical task, the task is executed in the back-end, the results are returned and displayed to the user in a visual representation (e.g. pie chart).

COMP-SCI 5540 Principles of Big Data Management

University of Missouri-Kansas City

Department of Computer Science and Electrical Engineering

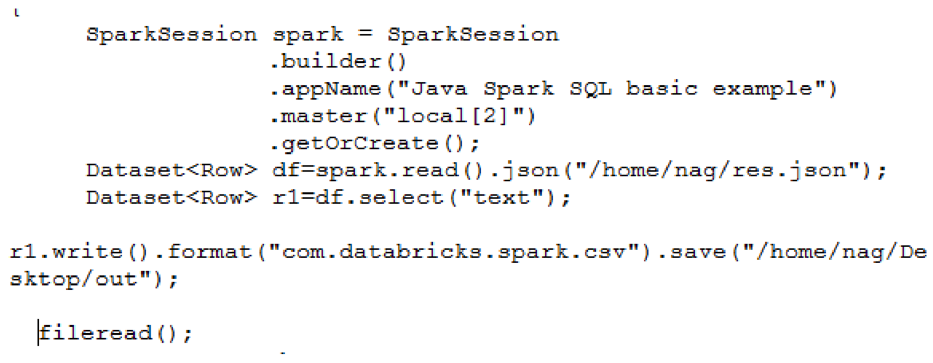
First Task: -

We find out movie review of The Fate of the Furious based on sentiment score of

tweets text that are tweeted on this movie.

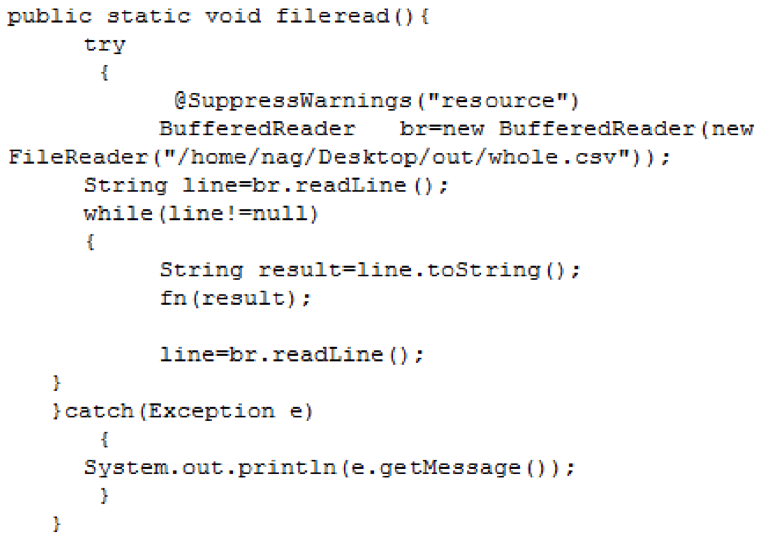
Solution: -

We extract the JSON tweet text data with SPARK Sql and dataset and passing to public API to get sentiment score of this text.

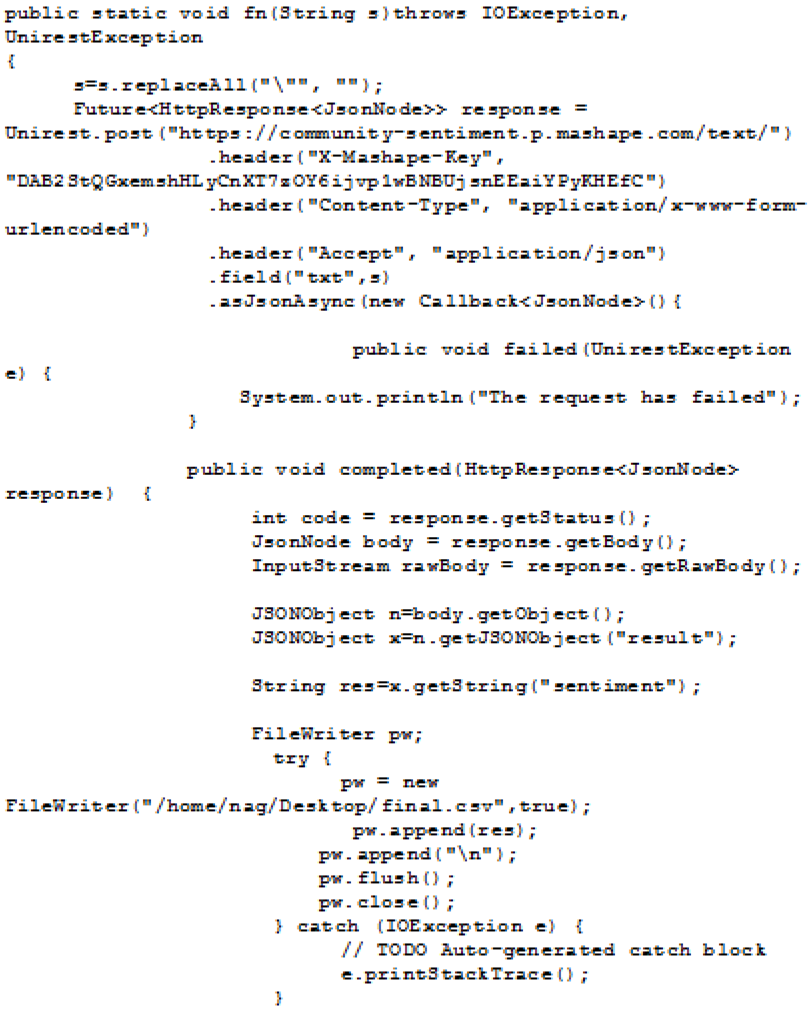


```
      SparkSession spark = SparkSession
                      .builder()
                      .appName("Java Spark SQL basic example")
                      .master("local[2]")
                      .getOrCreate();
      Dataset<Row> df=spark.read().json("/home/nag/res.json");
      Dataset<Row> r1=df.select("text");

r1.write().format("com.databricks.spark.csv").save("/home/nag/De
sktop/out");

   fileread();
```

Pass this text file to Unirest API for doing sentiment analysis of each text and storing in temp file



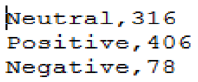
```
public static void fileread(){
      try
       {
             @SuppressWarnings("resource")
            BufferedReader   br=new BufferedReader(new
FileReader("/home/nag/Desktop/out/whole.csv"));
      String line=br.readLine();
      while(line!=null)
      {
            String result=line.toString();
            fn(result);

            line=br.readLine();
    }
    }catch(Exception e)
       {
      System.out.println(e.getMessage());
       }
    }
```



```
public static void fn(String s)throws IOException,
UnirestException
{
      s=s.replaceAll("\"", "");
      Future<HttpResponse<JsonNode>> response =
Unirest.post("https://community-sentiment.p.mashape.com/text/")
                .header("X-Mashape-Key",
"DAB2StQGxemshHLyCnXT7zOY6ijvp1wBNBUjsnEEaiYPyKHEfC")
                .header("Content-Type", "application/x-www-form-
urlencoded")
                .header("Accept", "application/json")
                .field("txt",s)
                .asJsonAsync(new Callback<JsonNode>(){

                            public void failed(UnirestException
e) {
                   System.out.println("The request has failed");
               }

              public void completed(HttpResponse<JsonNode>
response)  {
                    int code = response.getStatus();
                    JsonNode body = response.getBody();
                    InputStream rawBody = response.getRawBody();

                    JSONObject n=body.getObject();
                    JSONObject x=n.getJSONObject("result");

                    String res=x.getString("sentiment");

                    FileWriter pw;
                      try {
                           pw = new
FileWriter("/home/nag/Desktop/final.csv",true);
                            pw.append(res);
                         pw.append("\n");
                         pw.flush();
                         pw.close();
                      } catch (IOException e) {
                           // TODO Auto-generated catch block
                           e.printStackTrace();
                      }
```

Final out put storing in CSV file

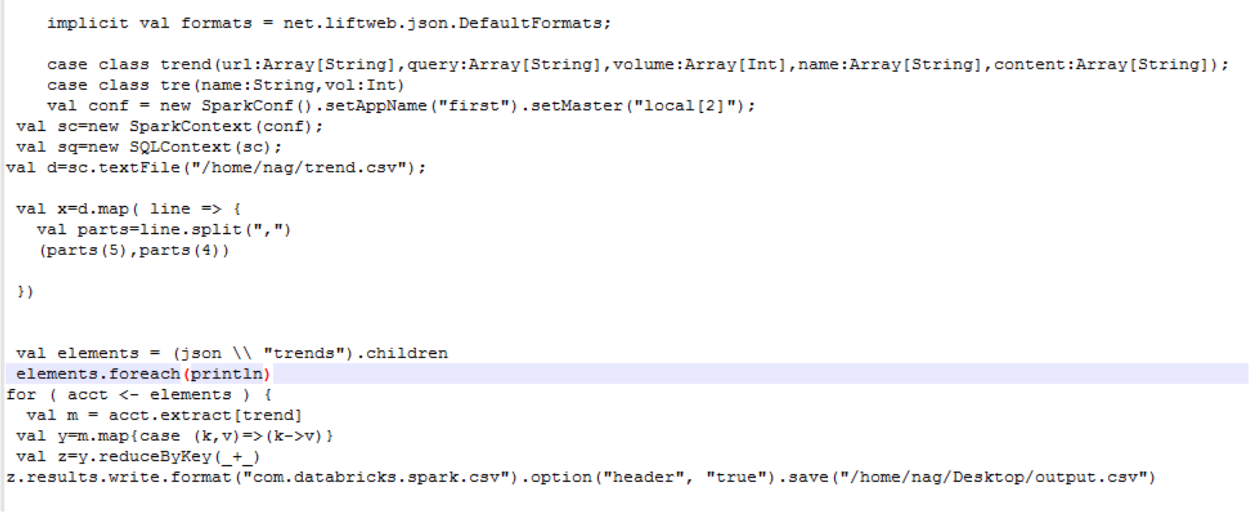
OUTPUT:



```
Neutral,316
Positive,406
Negative,78
```

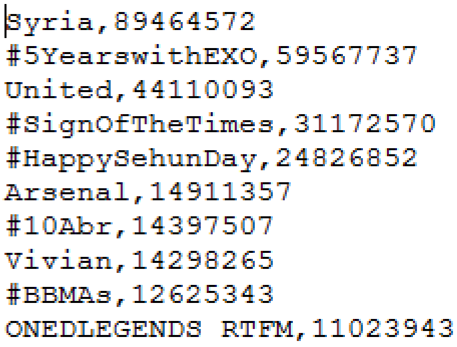
Task 2:

We find out top 10 trending hashtags from trends.txt file using SPARK RDD's



```
    implicit val formats = net.liftweb.json.DefaultFormats;

    case class trend(url:Array[String],query:Array[String],volume:Array[Int],name:Array[String],content:Array[String]);
    case class tre(name:String,vol:Int)
    val conf = new SparkConf().setAppName("first").setMaster("local[2]");
 val sc=new SparkContext(conf);
 val sq=new SQLContext(sc);
val d=sc.textFile("/home/nag/trend.csv");

 val x=d.map( line => {
   val parts=line.split(",")
   (parts(5),parts(4))

 })


 val elements = (json \\ "trends").children
 elements.foreach(println)
for ( acct <- elements ) {
  val m = acct.extract[trend]
 val y=m.map{case (k,v)=>(k->v)}
 val z=y.reduceByKey(_+_)
z.results.write.format("com.databricks.spark.csv").option("header", "true").save("/home/nag/Desktop/output.csv")
```

Output:



```
Syria,89464572
#5YearswithEXO,59567737
United,44110093
#SignOfTheTimes,31172570
#HappySehunDay,24826852
Arsenal,14911357
#10Abr,14397507
Vivian,14298265
#BBMAs,12625343
ONEDLEGENDS RTFM,11023943
```

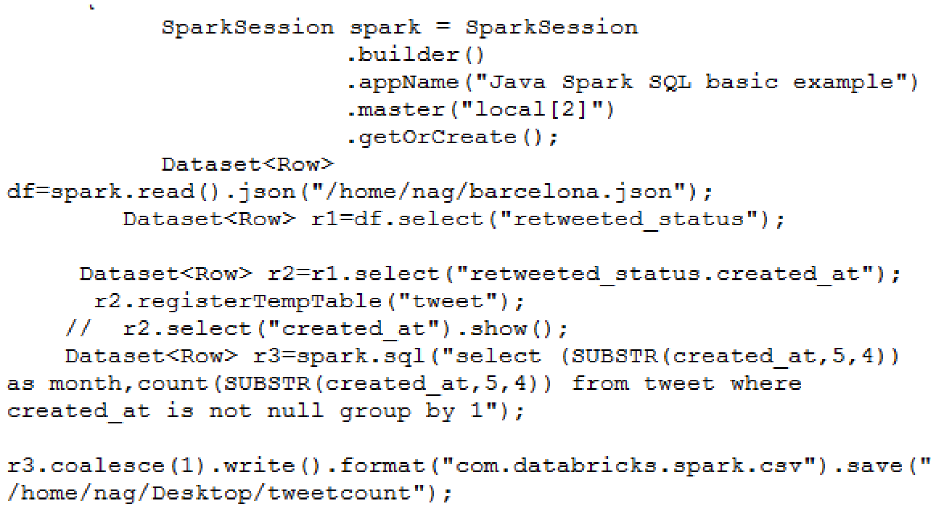
COMP-SCI 5540 Principles of Big Data Management

University of Missouri-Kansas City

Department of Computer Science and Electrical Engineering

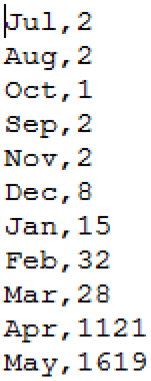
Task 3:

We find out the no of tweets that are tweeted on FC Barcelona team for each month using SPARK SQL and dataset



```
          SparkSession spark = SparkSession
                      .builder()
                      .appName("Java Spark SQL basic example")
                      .master("local[2]")
                      .getOrCreate();
          Dataset<Row>
df=spark.read().json("/home/nag/barcelona.json");
        Dataset<Row> r1=df.select("retweeted_status");

     Dataset<Row> r2=r1.select("retweeted_status.created_at");
      r2.registerTempTable("tweet");
    //  r2.select("created_at").show();
    Dataset<Row> r3=spark.sql("select (SUBSTR(created_at,5,4))
as month,count(SUBSTR(created_at,5,4)) from tweet where
created_at is not null group by 1");

r3.coalesce(1).write().format("com.databricks.spark.csv").save("
/home/nag/Desktop/tweetcount");
```

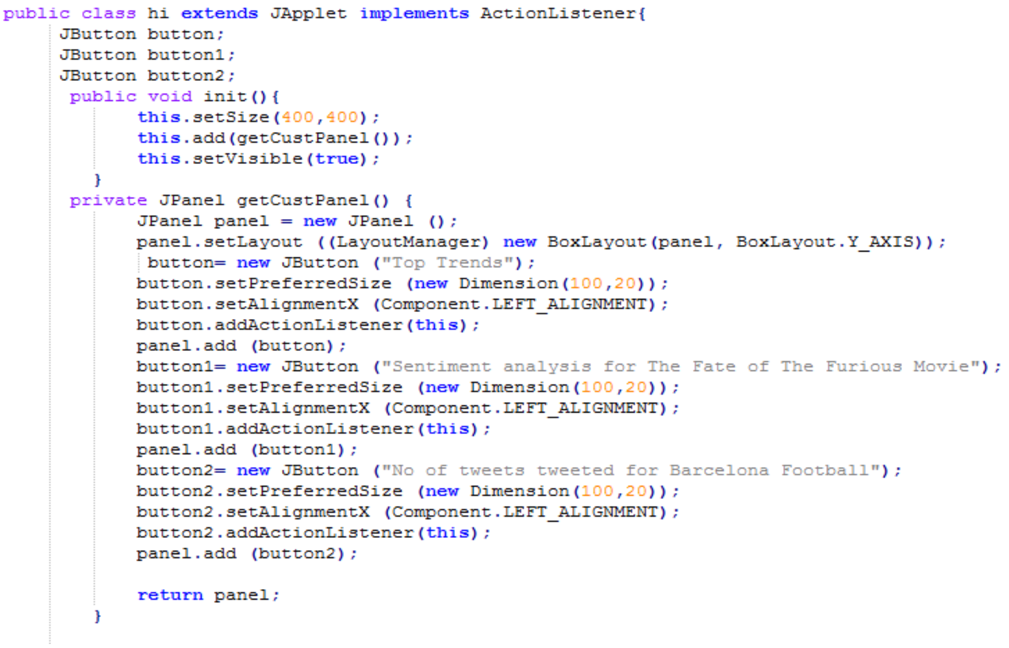
OUTPUT:



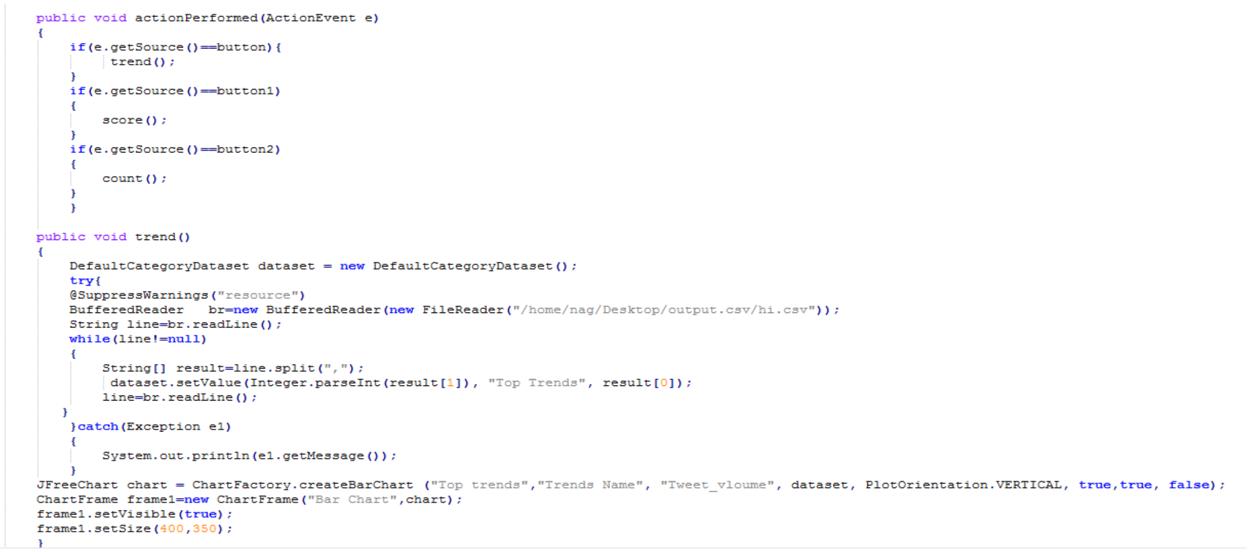
```
Jul,2
Aug,2
Oct,1
Sep,2
Nov,2
Dec,8
Jan,15
Feb,32
Mar,28
Apr,1121
May,1619
```

Extra Requirement:

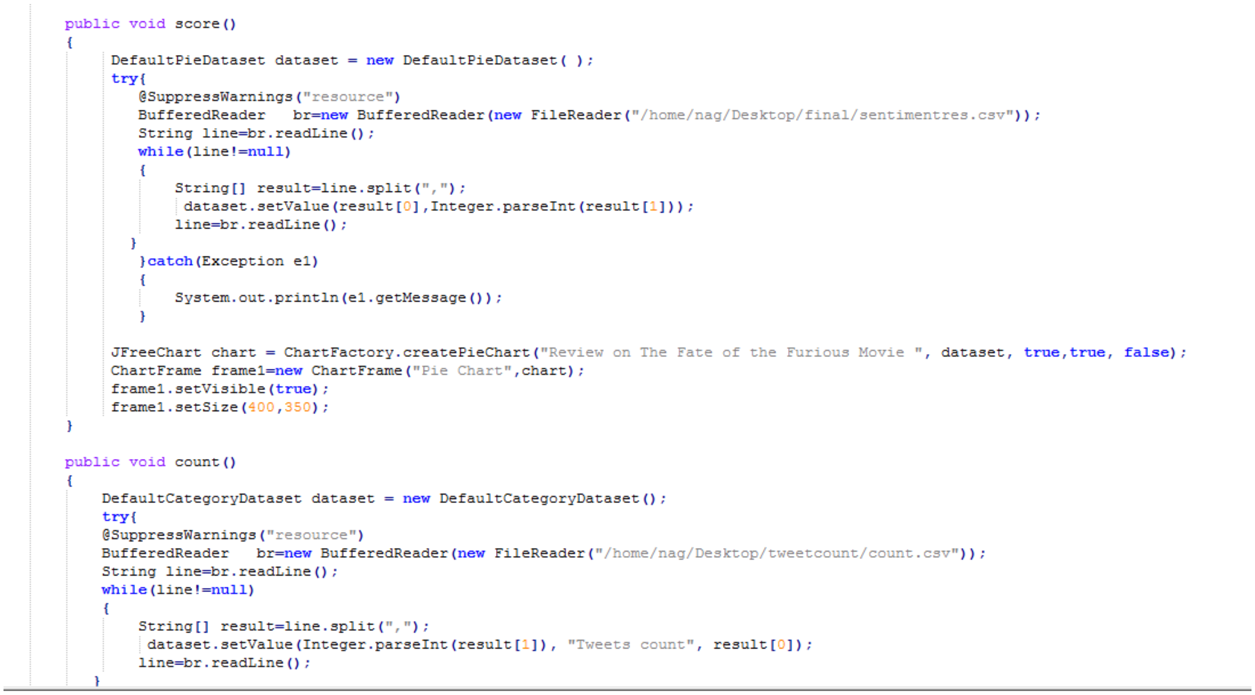
Implemented the GUI for the all tasks with the help of JAVA Applet functionality



```
public class hi extends JApplet implements ActionListener{
    JButton button;
    JButton button1;
    JButton button2;
     public void init(){
           this.setSize(400,400);
           this.add(getCustPanel());
           this.setVisible(true);
     }
     private JPanel getCustPanel() {
           JPanel panel = new JPanel ();
           panel.setLayout ((LayoutManager) new BoxLayout(panel, BoxLayout.Y_AXIS));
            button= new JButton ("Top Trends");
           button.setPreferredSize (new Dimension(100,20));
           button.setAlignmentX (Component.LEFT_ALIGNMENT);
           button.addActionListener(this);
           panel.add (button);
           button1= new JButton ("Sentiment analysis for The Fate of The Furious Movie");
           button1.setPreferredSize (new Dimension(100,20));
           button1.setAlignmentX (Component.LEFT_ALIGNMENT);
           button1.addActionListener(this);
           panel.add (button1);
           button2= new JButton ("No of tweets tweeted for Barcelona Football");
           button2.setPreferredSize (new Dimension(100,20));
           button2.setAlignmentX (Component.LEFT_ALIGNMENT);
           button2.addActionListener(this);
           panel.add (button2);

           return panel;
     }
```

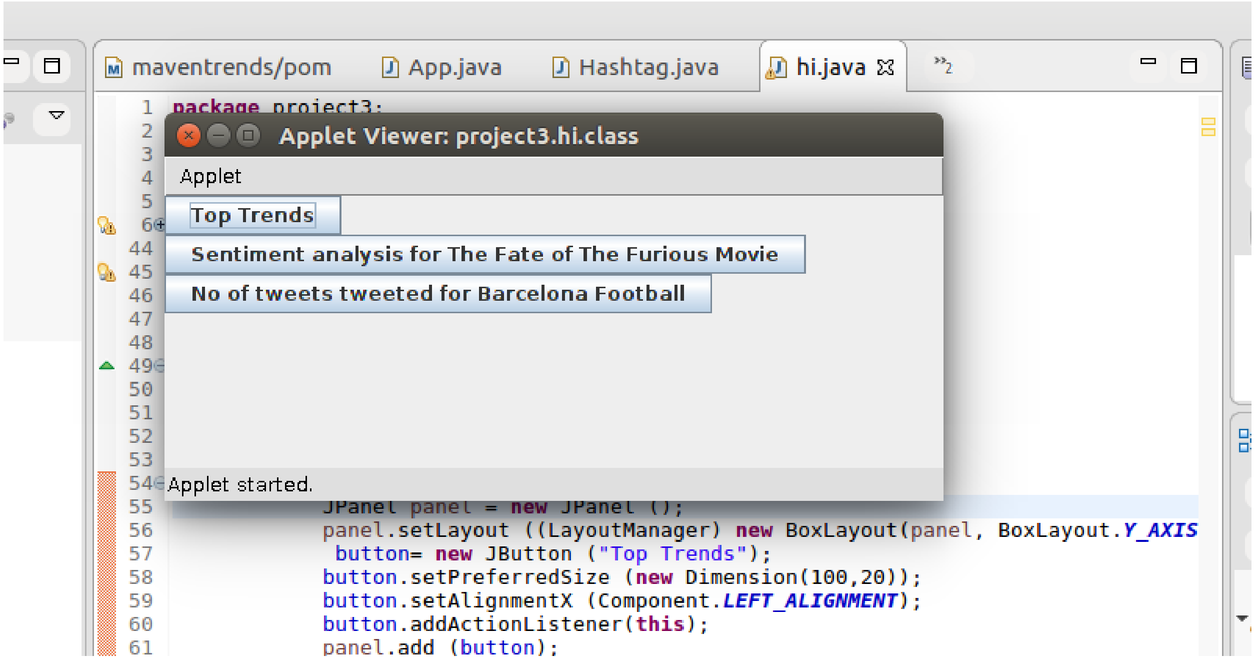


```
public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==button){
         trend();
    }
    if(e.getSource()==button1)
    {
        score();
    }
    if(e.getSource()==button2)
    {
        count();
    }
    }

public void trend()
{
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();
    try{
    @SuppressWarnings("resource")
    BufferedReader   br=new BufferedReader(new FileReader("/home/nag/Desktop/output.csv/hi.csv"));
    String line=br.readLine();
    while(line!=null)
    {
        String[] result=line.split(",");
         dataset.setValue(Integer.parseInt(result[1]), "Top Trends", result[0]);
        line=br.readLine();
   }
    }catch(Exception e1)
    {
        System.out.println(e1.getMessage());
    }
JFreeChart chart = ChartFactory.createBarChart ("Top trends","Trends Name", "Tweet_vloume", dataset, PlotOrientation.VERTICAL, true,true, false);
ChartFrame frame1=new ChartFrame("Bar Chart",chart);
frame1.setVisible(true);
frame1.setSize(400,350);
}
```



```
public void score()
{
     DefaultPieDataset dataset = new DefaultPieDataset( );
     try{
        @SuppressWarnings("resource")
        BufferedReader   br=new BufferedReader(new FileReader("/home/nag/Desktop/final/sentimentres.csv"));
        String line=br.readLine();
        while(line!=null)
        {
            String[] result=line.split(",");
             dataset.setValue(result[0],Integer.parseInt(result[1]));
            line=br.readLine();
       }
        }catch(Exception e1)
        {
            System.out.println(e1.getMessage());
        }

     JFreeChart chart = ChartFactory.createPieChart("Review on The Fate of the Furious Movie ", dataset, true,true, false);
     ChartFrame frame1=new ChartFrame("Pie Chart",chart);
     frame1.setVisible(true);
     frame1.setSize(400,350);
}

public void count()
{
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();
    try{
    @SuppressWarnings("resource")
    BufferedReader   br=new BufferedReader(new FileReader("/home/nag/Desktop/tweetcount/count.csv"));
    String line=br.readLine();
    while(line!=null)
    {
        String[] result=line.split(",");
         dataset.setValue(Integer.parseInt(result[1]), "Tweets count", result[0]);
        line=br.readLine();
   }
```

Output GUI: -

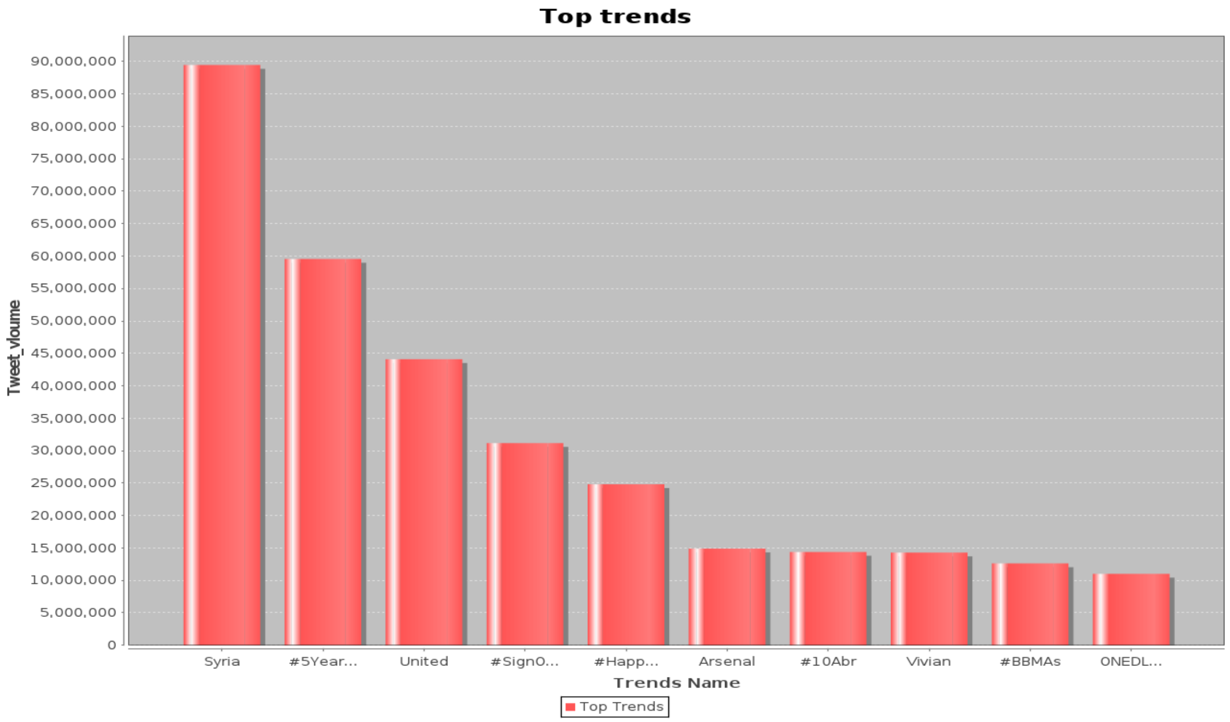

maventrends/pom
App.java
Hashtag.java
hi.java
»2
Applet Viewer: project3.hi.class
Applet
Top Trends
Sentiment analysis for The Fate of The Furious Movie
No of tweets tweeted for Barcelona Football

Applet started.

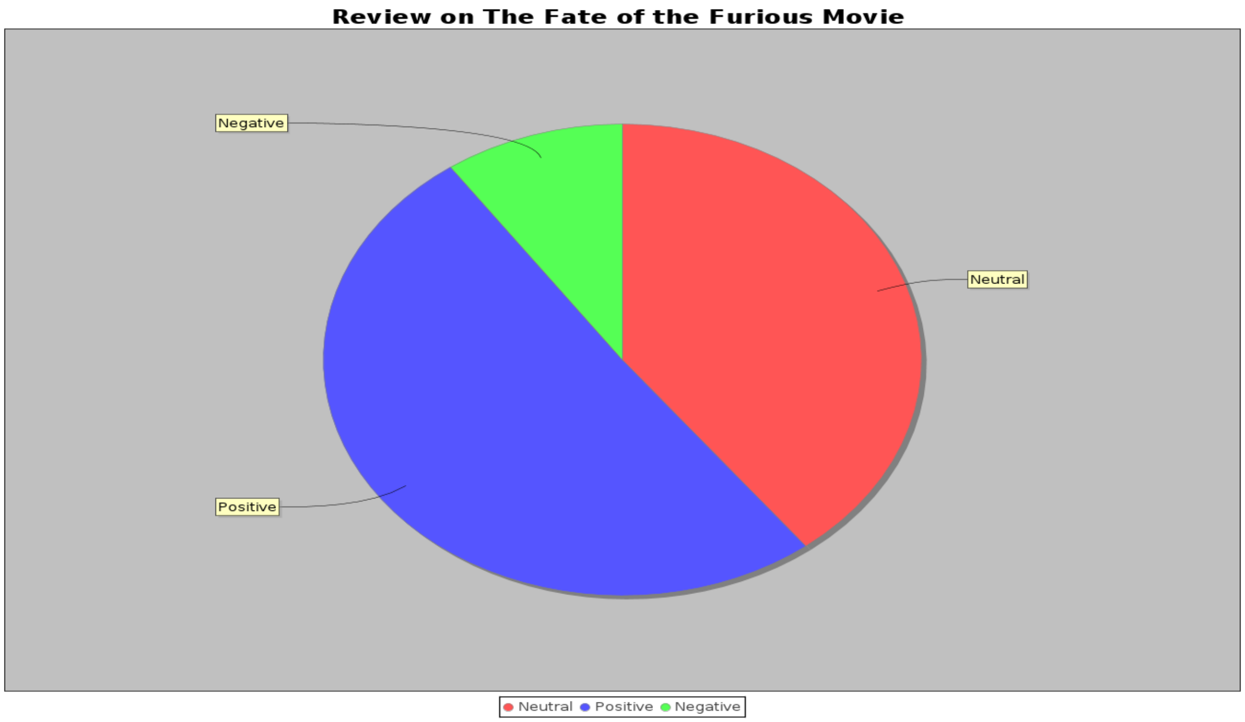
```
package project3;
```

```
            JPanel panel = new JPanel ();
            panel.setLayout ((LayoutManager) new BoxLayout(panel, BoxLayout.Y_AXIS
             button= new JButton ("Top Trends");
            button.setPreferredSize (new Dimension(100,20));
            button.setAlignmentX (Component.LEFT_ALIGNMENT);
            button.addActionListener(this);
            panel.add (button);
```

Top trends :



Review on Fate of the Furious movie

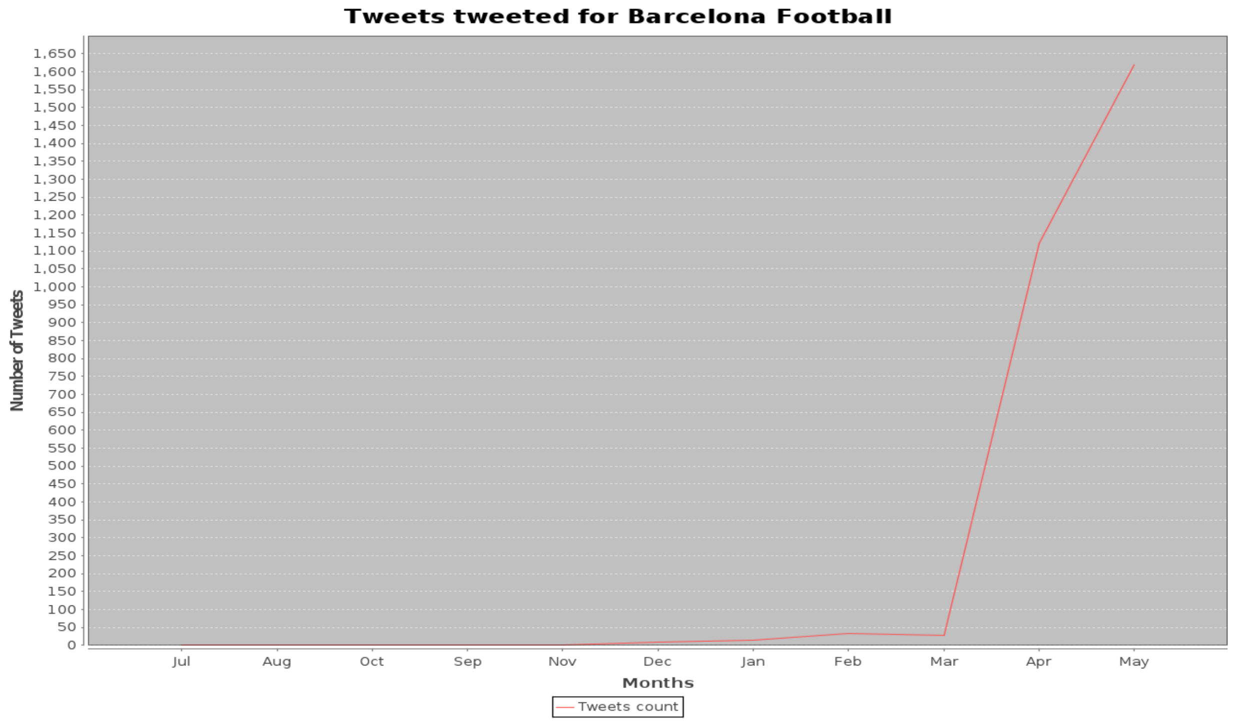


COMP-SCI 5540 Principles of Big Data Management

University of Missouri-Kansas City

Department of Computer Science and Electrical Engineering

Tweets tweeted for FC Barcelona Team:


Tweets tweeted for Barcelona Football
Number of Tweets
Jul
Aug
Oct
Sep
Nov
Dec
Jan
Feb
Mar
Apr
May
Months
Tweets count