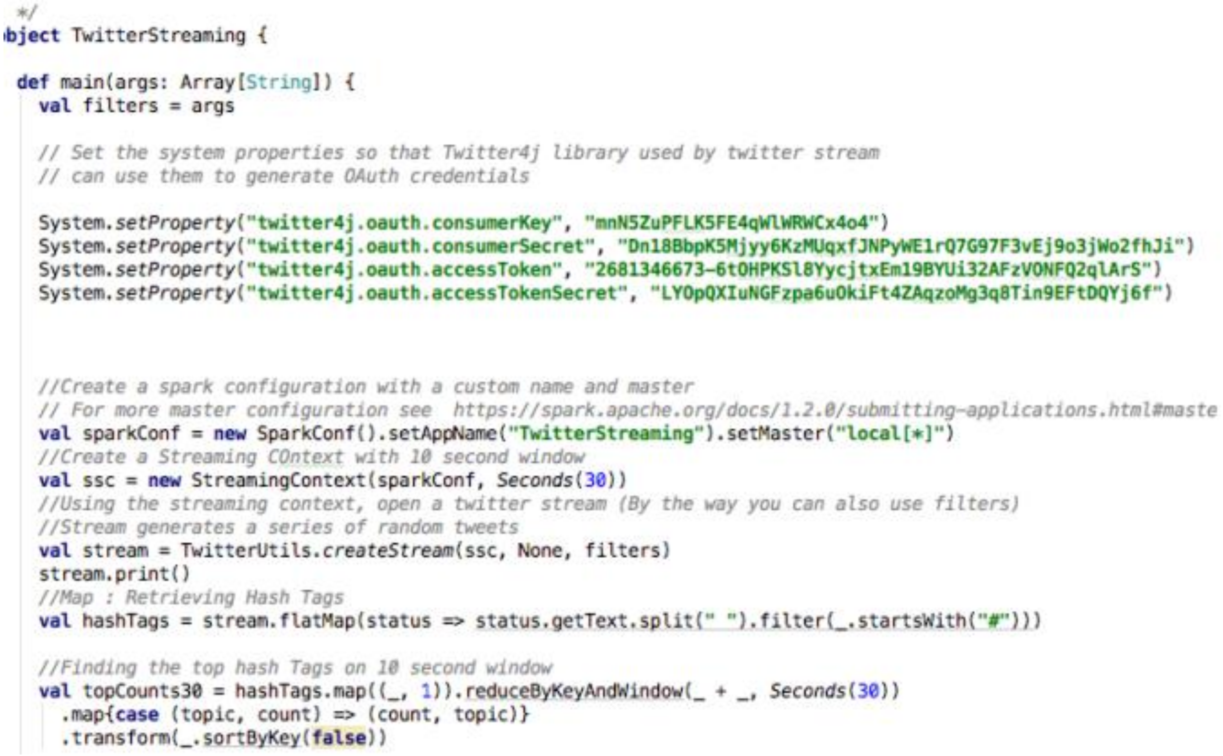
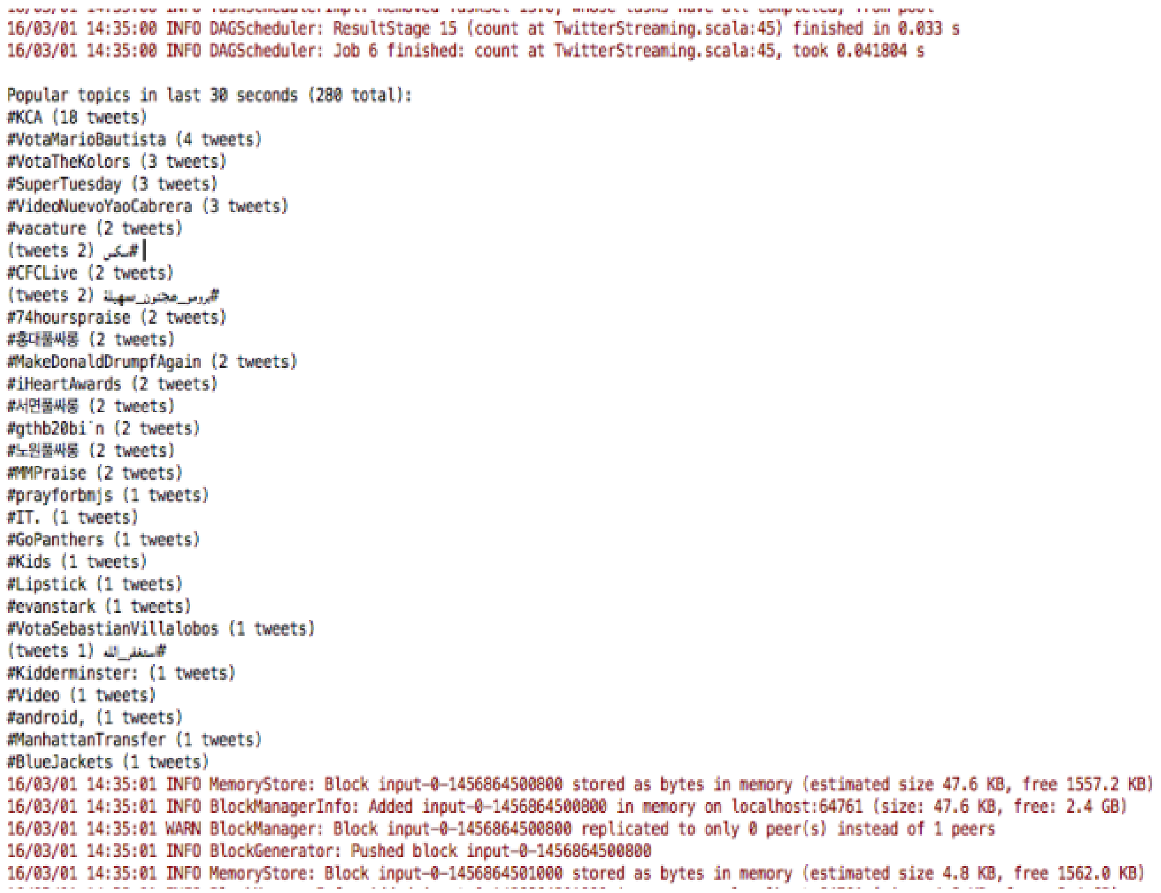
Report for Lab Assignment 5_6

Question 1 : Implement a smart application with big data analytics related to your project showing the collaboration between Spark and Smart Apps. Implement Twitter Streaming and perform word count on it and publish the results and showcase it in your Smart Phone/WatchApplication

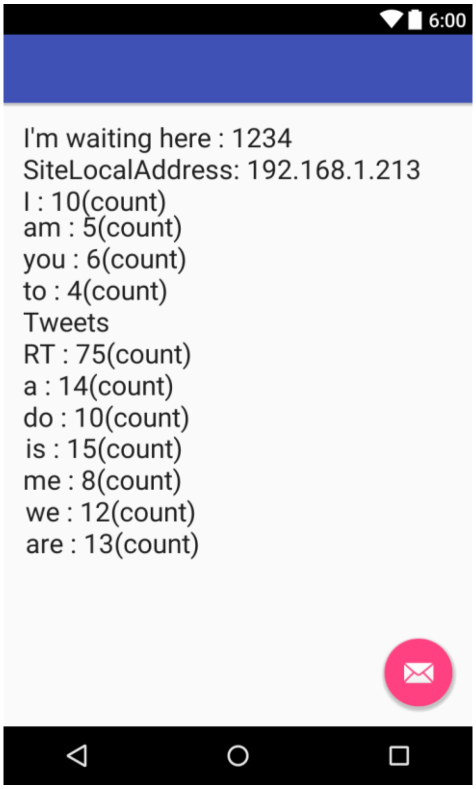
Description:



```
 */
bject TwitterStreaming {

 def main(args: Array[String]) {
   val filters = args

   // Set the system properties so that Twitter4j library used by twitter stream
   // can use them to generate OAuth credentials

   System.setProperty("twitter4j.oauth.consumerKey", "mnN5ZuPFLK5FE4qWlWRWCx4o4")
   System.setProperty("twitter4j.oauth.consumerSecret", "Dn18BbpK5Mjyy6KzMUqxfJNPyWE1rQ7G97F3vEj9o3jWo2fhJi")
   System.setProperty("twitter4j.oauth.accessToken", "2681346673-6tOHPKSl8YycjtxEm19BYUi32AFzVONFQ2qlArS")
   System.setProperty("twitter4j.oauth.accessTokenSecret", "LYOpQXIuNGFzpa6uOkiFt4ZAqzoMg3q8Tin9EFtDQYj6f")



   //Create a spark configuration with a custom name and master
   // For more master configuration see  https://spark.apache.org/docs/1.2.0/submitting-applications.html#maste
   val sparkConf = new SparkConf().setAppName("TwitterStreaming").setMaster("local[*]")
   //Create a Streaming COntext with 10 second window
   val ssc = new StreamingContext(sparkConf, Seconds(30))
   //Using the streaming context, open a twitter stream (By the way you can also use filters)
   //Stream generates a series of random tweets
   val stream = TwitterUtils.createStream(ssc, None, filters)
   stream.print()
   //Map : Retrieving Hash Tags
   val hashTags = stream.flatMap(status => status.getText.split(" ").filter(_.startsWith("#")))

   //Finding the top hash Tags on 10 second window
   val topCounts30 = hashTags.map((_, 1)).reduceByKeyAndWindow(_ + _, Seconds(30))
     .map{case (topic, count) => (count, topic)}
     .transform(_.sortByKey(false))
```



```
16/03/01 14:35:00 INFO TaskSchedulerImpl: Removed TaskSet 15.0, whose tasks have all completed, from pool
16/03/01 14:35:00 INFO DAGScheduler: ResultStage 15 (count at TwitterStreaming.scala:45) finished in 0.033 s
16/03/01 14:35:00 INFO DAGScheduler: Job 6 finished: count at TwitterStreaming.scala:45, took 0.041804 s

Popular topics in last 30 seconds (280 total):
#KCA (18 tweets)
#VotaMarioBautista (4 tweets)
#VotaTheKolors (3 tweets)
#SuperTuesday (3 tweets)
#VideoNuevoYaoCabrera (3 tweets)
#vacature (2 tweets)
(tweets 2) #سكس
#CFCLive (2 tweets)
(tweets 2) #بروس_محترف_سهيلة
#74hourspraise (2 tweets)
#홍대풀싸롱 (2 tweets)
#MakeDonaldDrumpfAgain (2 tweets)
#iHeartAwards (2 tweets)
#서면풀싸롱 (2 tweets)
#gthb20bi˙n (2 tweets)
#노원풀싸롱 (2 tweets)
#MMPraise (2 tweets)
#prayforbmjs (1 tweets)
#IT. (1 tweets)
#GoPanthers (1 tweets)
#Kids (1 tweets)
#Lipstick (1 tweets)
#evanstark (1 tweets)
#VotaSebastianVillalobos (1 tweets)
(tweets 1) #سنفر_الله
#Kidderminster: (1 tweets)
#Video (1 tweets)
#android, (1 tweets)
#ManhattanTransfer (1 tweets)
#BlueJackets (1 tweets)
16/03/01 14:35:01 INFO MemoryStore: Block input-0-1456864500800 stored as bytes in memory (estimated size 47.6 KB, free 1557.2 KB)
16/03/01 14:35:01 INFO BlockManagerInfo: Added input-0-1456864500800 in memory on localhost:64761 (size: 47.6 KB, free: 2.4 GB)
16/03/01 14:35:01 WARN BlockManager: Block input-0-1456864500800 replicated to only 0 peer(s) instead of 1 peers
16/03/01 14:35:01 INFO BlockGenerator: Pushed block input-0-1456864500800
16/03/01 14:35:01 INFO MemoryStore: Block input-0-1456864501000 stored as bytes in memory (estimated size 4.8 KB, free 1562.0 KB)
```



I'm waiting here : 1234
SiteLocalAddress: 192.168.1.213
I : 10(count)
am : 5(count)
you : 6(count)
to : 4(count)
Tweets
RT : 75(count)
a : 14(count)
do : 10(count)
is : 15(count)
me : 8(count)
we : 12(count)
are : 13(count)

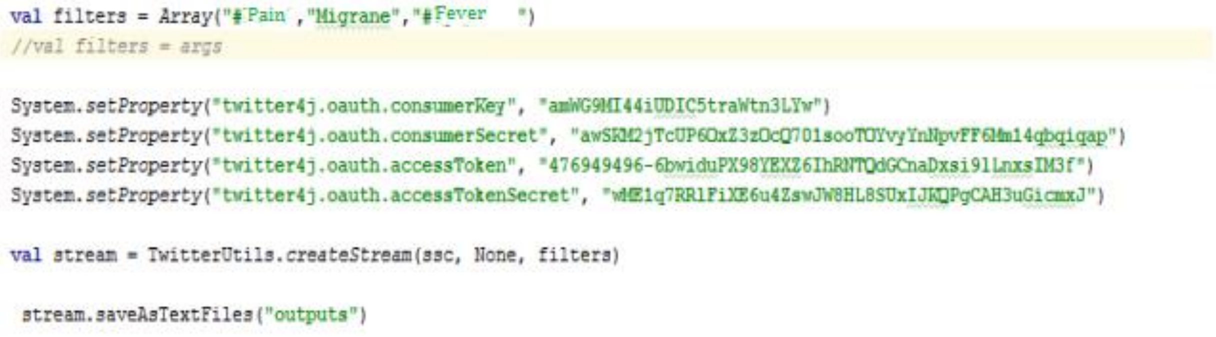
Question 2: Perform a machine learning algorithm with the Twitter Streaming data to categorize each Tweet

1) Training datasets: Collect different categories of Tweets related to your project.(Categories can be based on HashTags /Subjects etc.)
2) Test data: the upcoming twitter stream.

Description:

Use the Training data to generate different category of data such as Fever, Pain, Migrane etc.

After running the program, text files of the three categories are generated.



```
val filters = Array("#Pain ,"Migrane","#Fever   ")
//val filters = args

System.setProperty("twitter4j.oauth.consumerKey", "amWG9MI44iUDIC5traWtn3LYw")
System.setProperty("twitter4j.oauth.consumerSecret", "awSKM2jTcUP6OxZ3zOcQ701sooTOYvyYnNpvFF6Mm14qbqiqap")
System.setProperty("twitter4j.oauth.accessToken", "476949496-6bwiduPX98YEXZ6IhRNTQdGCnaDxsi91LnxsIM3f")
System.setProperty("twitter4j.oauth.accessTokenSecret", "wME1q7RR1FiXE6u4ZswJW8HL8SUxIJKQPgCAH3uGicmxJ")

val stream = TwitterUtils.createStream(ssc, None, filters)

 stream.saveAsTextFiles("outputs")
```