REPORT LAB ASSIGNMENT – 5 & 6

Sandeep – 5

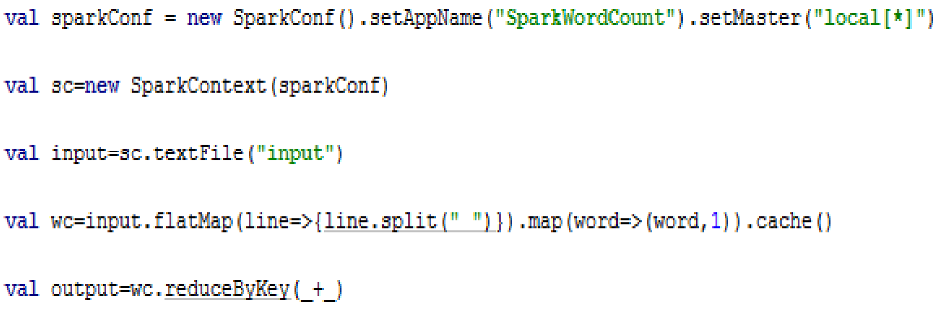
1. Question:

Spark and Smartphone/Watch Application

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/Watch Application.

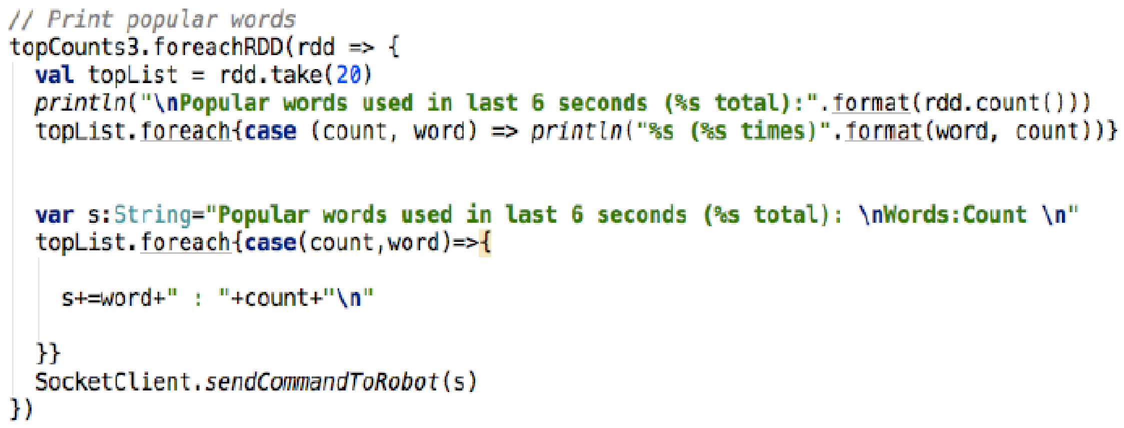
Description:

Considered the input which consists of different symptoms with its related diseases, used the map reduce code compute the word count.
Collected the top tags and have send the data to phone using socket connection.



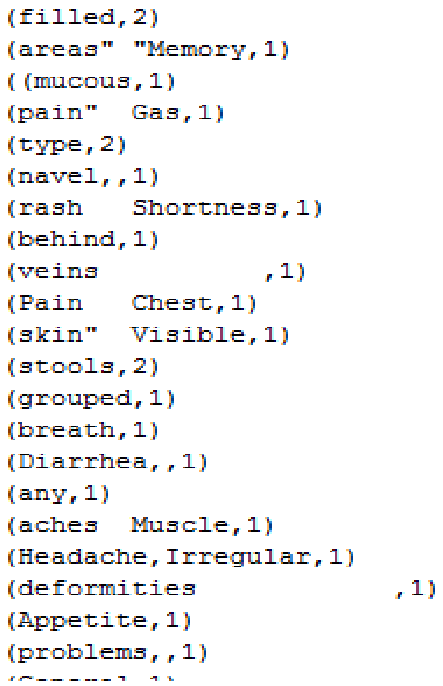
```
val sparkConf = new SparkConf().setAppName("SparkWordCount").setMaster("local[*]")

val sc=new SparkContext(sparkConf)

val input=sc.textFile("input")

val wc=input.flatMap(line=>{line.split(" ")}).map(word=>(word,1)).cache()

val output=wc.reduceByKey(_+_)
```

………

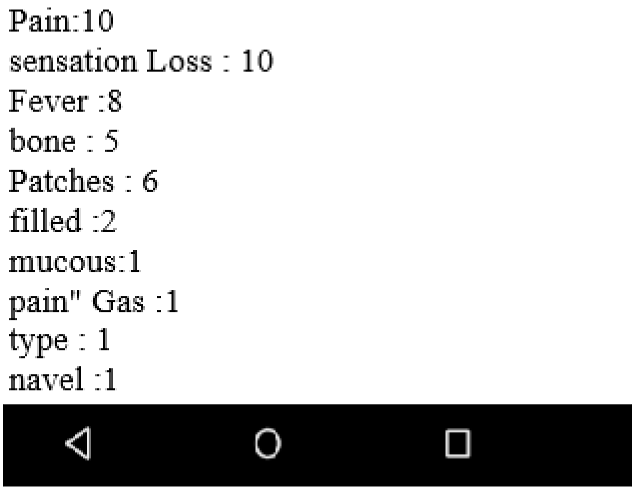


```
// Print popular words
topCounts3.foreachRDD(rdd => {
  val topList = rdd.take(20)
  println("\nPopular words used in last 6 seconds (%s total):".format(rdd.count()))
  topList.foreach{case (count, word) => println("%s (%s times)".format(word, count))}


  var s:String="Popular words used in last 6 seconds (%s total): \nWords:Count \n"
  topList.foreach{case(count,word)=>{

    s+=word+" : "+count+"\n"

  }}
  SocketClient.sendCommandToRobot(s)
})
```

………



```
(filled,2)
(areas" "Memory,1)
((mucous,1)
(pain"  Gas,1)
(type,2)
(navel,,1)
(rash   Shortness,1)
(behind,1)
(veins           ,1)
(Pain   Chest,1)
(skin"  Visible,1)
(stools,2)
(grouped,1)
(breath,1)
(Diarrhea,,1)
(any,1)
(aches  Muscle,1)
(Headache,Irregular,1)
(deformities               ,1)
(Appetite,1)
(problems,,1)
(General,1)
```



```
Pain:10
sensation Loss : 10
Fever :8
bone : 5
Patches : 6
filled :2
mucous:1
pain" Gas :1
type : 1
navel :1
```

2. Question

Spark ML Lib Application

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.)

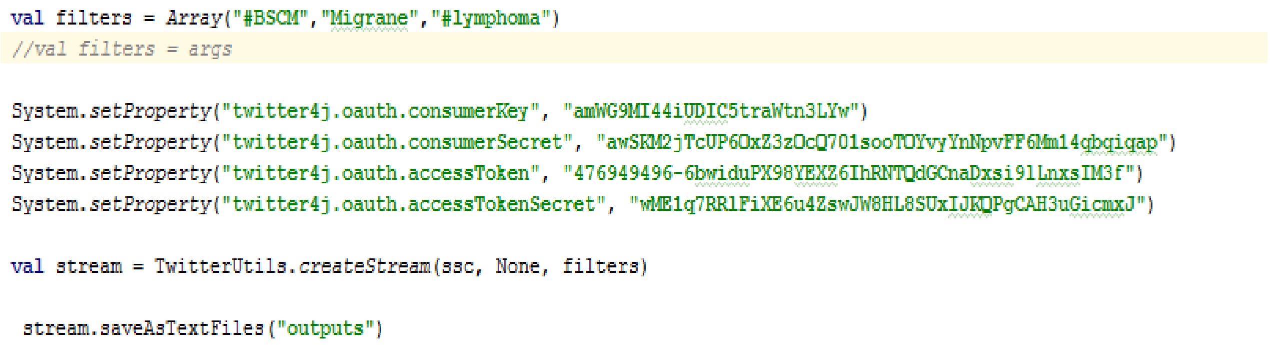
2) Test data: the upcoming twitter stream.

Description

Get the Training data

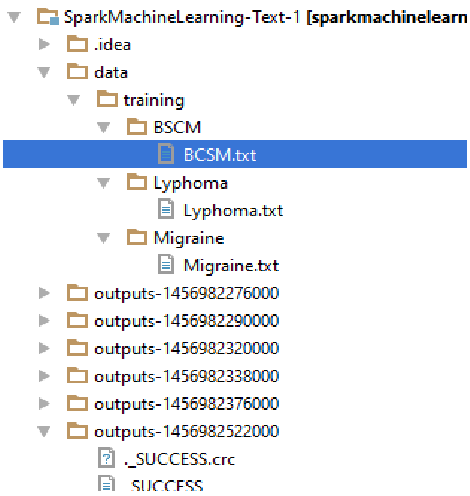
Collect the training data from the twitter using spark twitter4J library support and categorize the data in to different categories such as BSCM (Cancer), Lymphoma, Migraine.

To collect data from Twitter.



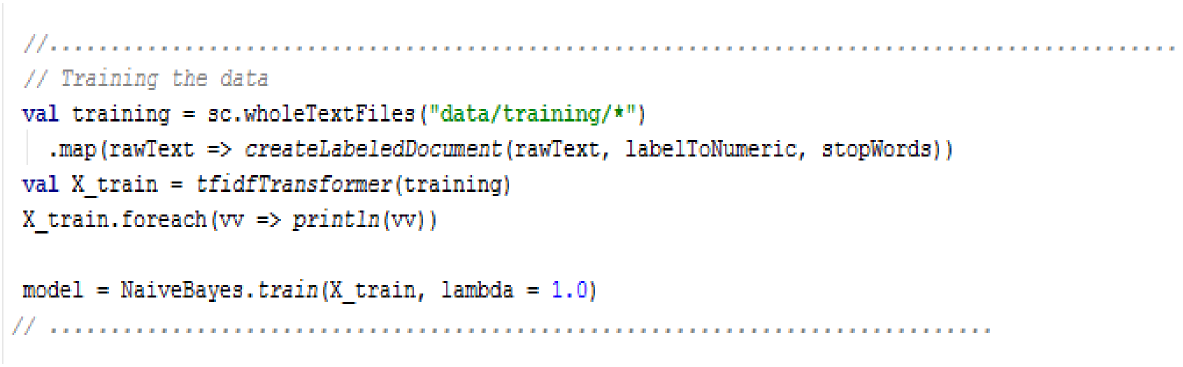
```
val filters = Array("#BSCM","Migrane","#lymphoma")
//val filters = args

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")
```

Organised the training data in to differnet folders for the input.



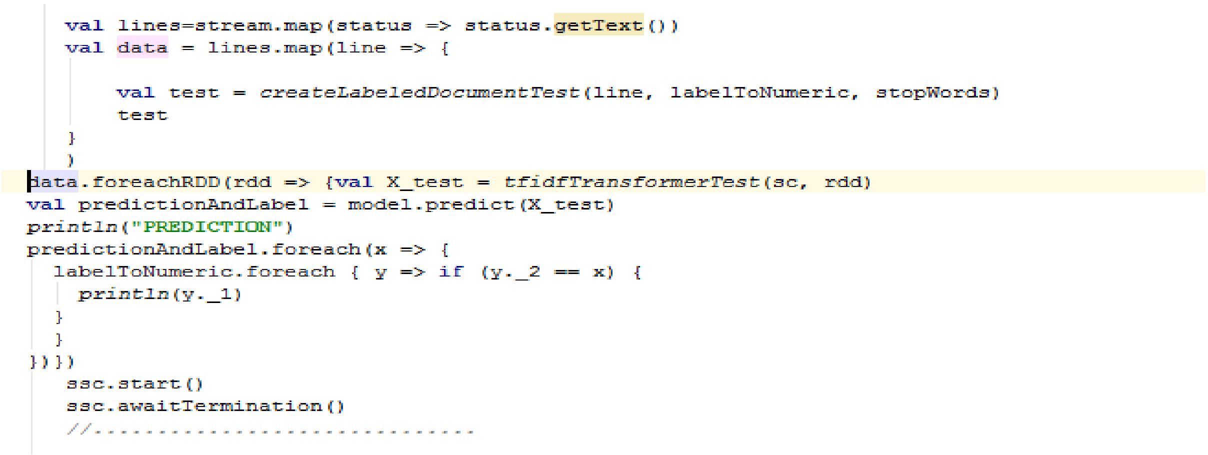
```
▼ SparkMachineLearning-Text-1 [sparkmachinelearn
  ▶ .idea
  ▼ data
    ▼ training
      ▼ BSCM
          BCSM.txt
      ▼ Lyphoma
          Lyphoma.txt
      ▼ Migraine
          Migraine.txt
  ▶ outputs-1456982276000
  ▶ outputs-1456982290000
  ▶ outputs-1456982320000
  ▶ outputs-1456982338000
  ▶ outputs-1456982376000
  ▼ outputs-1456982522000
      ._SUCCESS.crc
      _SUCCESS
```

Machine Learning analysis though Naïve Bayes model.



```
//........................................................................
// Training the data
val training = sc.wholeTextFiles("data/training/*")
  .map(rawText => createLabeledDocument(rawText, labelToNumeric, stopWords))
val X_train = tfidfTransformer(training)
X_train.foreach(vv => println(vv))

model = NaiveBayes.train(X_train, lambda = 1.0)
// ..........................................................................
```

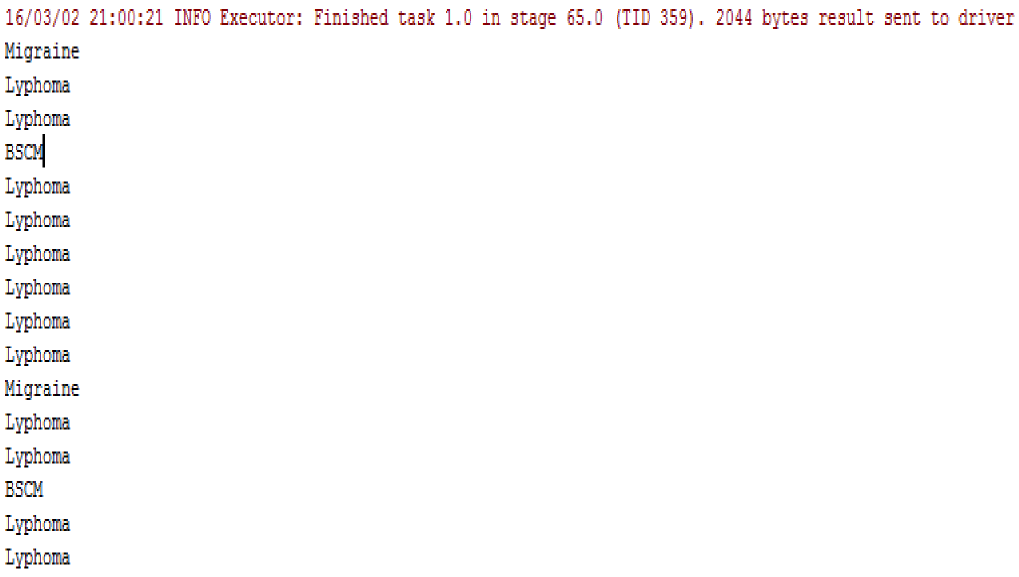
………



```
    val lines=stream.map(status => status.getText())
    val data = lines.map(line => {

        val test = createLabeledDocumentTest(line, labelToNumeric, stopWords)
        test
    }
    )
data.foreachRDD(rdd => {val X_test = tfidfTransformerTest(sc, rdd)
val predictionAndLabel = model.predict(X_test)
println("PREDICTION")
predictionAndLabel.foreach(x => {
  labelToNumeric.foreach { y => if (y._2 == x) {
    println(y._1)
  }
  }
}))
    ssc.start()
    ssc.awaitTermination()
    //------------------------------
```

We will extract the text from the stream and would create a labeled document using standard NLP library functions.

We will iterate through each RDD and would predict the category of the tweets received.



```
16/03/02 21:00:21 INFO Executor: Finished task 1.0 in stage 65.0 (TID 359). 2044 bytes result sent to driver
Migraine
Lyphoma
Lyphoma
BSCM
Lyphoma
Lyphoma
Lyphoma
Lyphoma
Lyphoma
Lyphoma
Migraine
Lyphoma
Lyphoma
BSCM
Lyphoma
Lyphoma
```