

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:

This assignment can be approached by developing two different applications. One is an android application which will run in mobile device in a network and another application will run in IntelliJ application in desktop which will run in the same network.

The android application will start as a socket client where it will create a port and assign to the application and will wait for incoming connection. Then if we run IntelliJ application with the IP address same as mobile, then the application will connect to mobile and will send the twitter word count data output to mobile in form of streaming until the application was terminated.

Report for Lab Assignment 5 & 6

IntelliJ Code TwitterStreaming.scala Explanation:

```
//Setting the API keys of twitter to variables
val Consumer_Key="XJPlyOuAo7tC8YBxE1GDiTq8n"
val Consumer_Secret="Zeyv8qYR9u5wmOqvBStqLV8HMA1Qvh3R9d3vDj02gfpTfAzrgs"
val Access_Token="143736380-2VpESDsrrWMcOipe7zLIMBr0ncHwnDbMxjckpU7nT"
val Access_Token_Secret="jEdqt83lwfuD01rPFpUyGmS4OexheFFsru7CZEH9iqnss"

//Assigning the API keys
System.setProperty("twitter4j.oauth.consumerKey", Consumer_Key)
System.setProperty("twitter4j.oauth.consumerSecret", Consumer_Secret)
System.setProperty("twitter4j.oauth.accessToken", Access_Token)
System.setProperty("twitter4j.oauth.accessTokenSecret", Access_Token_Secret)

//Creating Spark Configuration
val sparkConf = new SparkConf().setAppName("STweetsApp").setMaster("local[*]")
//Creating streaming context
val ssc = new StreamingContext(sparkConf, Seconds(2))
val stream = TwitterUtils.createStream(ssc, None, filters)
stream.print()
//Getting Hashtags stream data based on criteria like "#"
val hashTags = stream.flatMap(status => status.getText.split("
").filter(_.startsWith("#")))
//Getting top hashtags for 30 seconds
val topCounts30 = hashTags.map((_, 1)).reduceByKeyAndWindow(_ + _, Seconds(30))
    .map{case (topic, count) => (count, topic)}
    .transform(_.sortByKey(false))
//Searching for each RDD and sending the values to SocketClient.scala
topCounts30.foreachRDD(rdd => {
    val topList = rdd.take(10)
    println("\nPopular topics in last 30 seconds (%s total):".format(rdd.count()))
    topList.foreach{case (count, tag) => println("%s (%s tweets)".format(tag, count))}
    topList.foreach{case (count, tag) => SocketClient.sendCommandToRobot("\n( " + tag + " ,
"+ count+" )") }
})
//Starting streaming context
ssc.start()
//Stop the streaming context either by terminating or if the timeout happens for 1000000
seconds
ssc.awaitTerminationOrTimeout(100000)
```

Report for Lab Assignment 5 & 6

IntelliJ Code SocketClient.scala Explanation:

```
def sendCommandToRobot(string: String)
{
    try {

        //Assigning the IP value of the mobile to address variable
        lazy val address: Array[Byte] = Array(192.toByte, 168.toByte, 1.toByte, 120.toByte)
        val ia = InetAddress.getByAddress(address)

        //Create port address
        val socket = new Socket(ia, 1234)

        //Creating out variable and print the stream to
        val out = new PrintStream(socket.getOutputStream)

        //Printing the string and sending to Mobile Client
        out.print(string)
        out.flush()
        out.close()
        socket.close()
    }
}
```

Report for Lab Assignment 5 & 6

Screenshots for Question 1:



Report for Lab Assignment 5 & 6

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

Screenshots:

Code:

```
import org.apache.spark.{SparkContext, SparkConf}
import org.apache.spark.mllib.classification.NaiveBayes
import org.apache.spark.mllib.feature.TfidfTransformer
import org.apache.spark.mllib.linalg.Vector
import org.apache.spark.mllib.linalg.distributed.RowMatrix

val conf = new SparkConf().setAppName("SparkMachineLearning-Text-1")
val sc = new SparkContext(conf)

def train(tweets: List[String]): NaiveBayes = {
    // Create a TfidfTransformer
    val tfidf = new TfidfTransformer()

    // Create a NaiveBayes classifier
    val model = NaiveBayes.train(tweets.map(_.split(" ")).map(_.map(_.toLowerCase)).map(_.mkString(" "))), tfidf)

    model
}

def test(tweets: List[String], model: NaiveBayes): List[String] = {
    // Create a TfidfTransformer
    val tfidf = new TfidfTransformer()

    // Create a NaiveBayes classifier
    val predictions = tweets.map(_.split(" ")).map(_.map(_.toLowerCase)).map(_.mkString(" ")).map(model.predict(_))

    predictions
}

val lines = sc.wholeTextFiles("data/testing/1/*")
val data = lines.map(line => {
    val test = createLabeledDocumentTest(line._2, labelToNumeric, stopWords)
    println(test.body)
    test
})

val X_test = tfidf.transformTest(sc, data)

val predictionAndLabel = model.predict(X_test)
println("PREDICTION")
predictionAndLabel.foreach(x => {
    labelToNumeric.foreach { y => if (y._2 == x) {
        println(y._1)
    }
})

ssc.start()

ssc.awaitTerminationOrTimeout(300)
```

Report for Lab Assignment 5 & 6

Prediction:

```
FeatureExtractionText1 - [D:\#UMKC\Big Data Analytics & Applications\Tutorial\SparkMachineLearning-Text-1] - [featureextractiontext1] - ...src\main\scala\edu\umkc\fv\FeatureVector1.scala - IntelliJ IDEA 15.0.4
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

SparkMachineLearning-Text-1 | src | main | scala | edu | umkc | fv | FeatureVector1.scala | FeatureVector1
Project | SparkMachineLearning-Text-1 | featureextractiontext1 | data | testing | 1
FeatureVector1.scala | NLPUtils.scala | build.sbt | FeatureVector1
val rawText => createLabeledDocument(rawText, labelToNumeric, stopWords))
val X_train = tfidfTransformer(training)
X_train.foreach(vv => println(vv))

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

Run | FeatureVector1
16/03/02 21:44:02 INFO DAGScheduler: Submitting 2 missing tasks from ResultStage 5 (MapPartitionsRDD[17] at mapPartitions at NaiveBayes.scala:90)
16/03/02 21:44:02 INFO TaskSchedulerImpl: Adding task set 5.0 with 2 tasks
16/03/02 21:44:02 INFO TaskSetManager: Starting task 0.0 in stage 5.0 (TID 10, localhost, PROCESS_LOCAL, 3052 bytes)
16/03/02 21:44:02 INFO TaskSetManager: Starting task 1.0 in stage 5.0 (TID 11, localhost, PROCESS_LOCAL, 2387 bytes)
16/03/02 21:44:02 INFO Executor: Running task 0.0 in stage 5.0 (TID 10)
16/03/02 21:44:02 INFO Executor: Running task 1.0 in stage 5.0 (TID 11)
16/03/02 21:44:02 INFO BlockManager: Found block rdd_12_1 locally
16/03/02 21:44:02 INFO BlockManager: Found block rdd_12_0 locally
16/03/02 21:44:02 INFO BlockManager: Found block rdd_12_1 locally
16/03/02 21:44:02 INFO BlockManager: Found block rdd_12_0 locally
16/03/02 21:44:02 INFO BlockManager: Found block rdd_12_0 locally
VOTE BERNIE
VOTE BERNIE
VOTE BERNIE
TRUMP
VOTE BERNIE
VOTE BERNIE
VOTE BERNIE
16/03/02 21:44:02 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
16/03/02 21:44:02 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
16/03/02 21:44:02 INFO Executor: Finished task 1.0 in stage 5.0 (TID 11). 2044 bytes result sent to driver
16/03/02 21:44:02 INFO Executor: Finished task 0.0 in stage 5.0 (TID 10). 2044 bytes result sent to driver
16/03/02 21:44:02 INFO TaskSetManager: Finished task 1.0 in stage 5.0 (TID 11) in 19 ms on localhost (1/2)
16/03/02 21:44:02 INFO TaskSetManager: Finished task 0.0 in stage 5.0 (TID 10) in 20 ms on localhost (2/2)
16/03/02 21:44:02 INFO DAGScheduler: ResultStage 5 (foreach at FeatureVector1.scala:72) finished in 0.022 s
16/03/02 21:44:02 INFO TaskSchedulerImpl: Removed TaskSet 5.0, whose tasks have all completed, from pool
16/03/02 21:44:03 INFO DAGScheduler: Task 5 finished: foreach at FeatureVector1.scala:72, task 0.02216 s

Java Enterprise | Run | Debug | TODO | Terminal | Messages | Event Log
All files are up-to-date (10 minutes ago) | 77 chars, 7 lines | 338:1 | CRLF+ | UTF-8+ | [T] | [D]
```