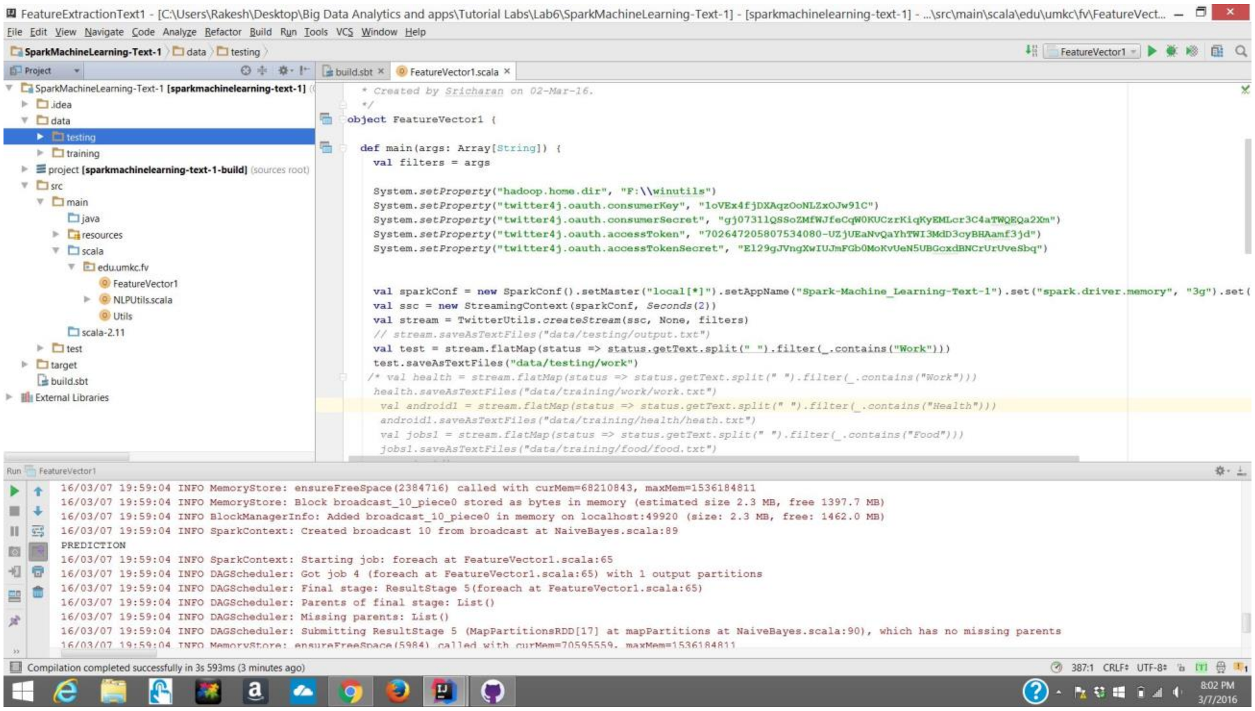
Spark Machine Learning

In this part of assignment large data sets of training and testing data must be collected from Twitter. The content of tweets must be collected in a streaming way.

Here I am collecting data on food, health and work as filters. Divided the data into training and testing data.

Below are the screenshots:

1. PREDICTION is happening.



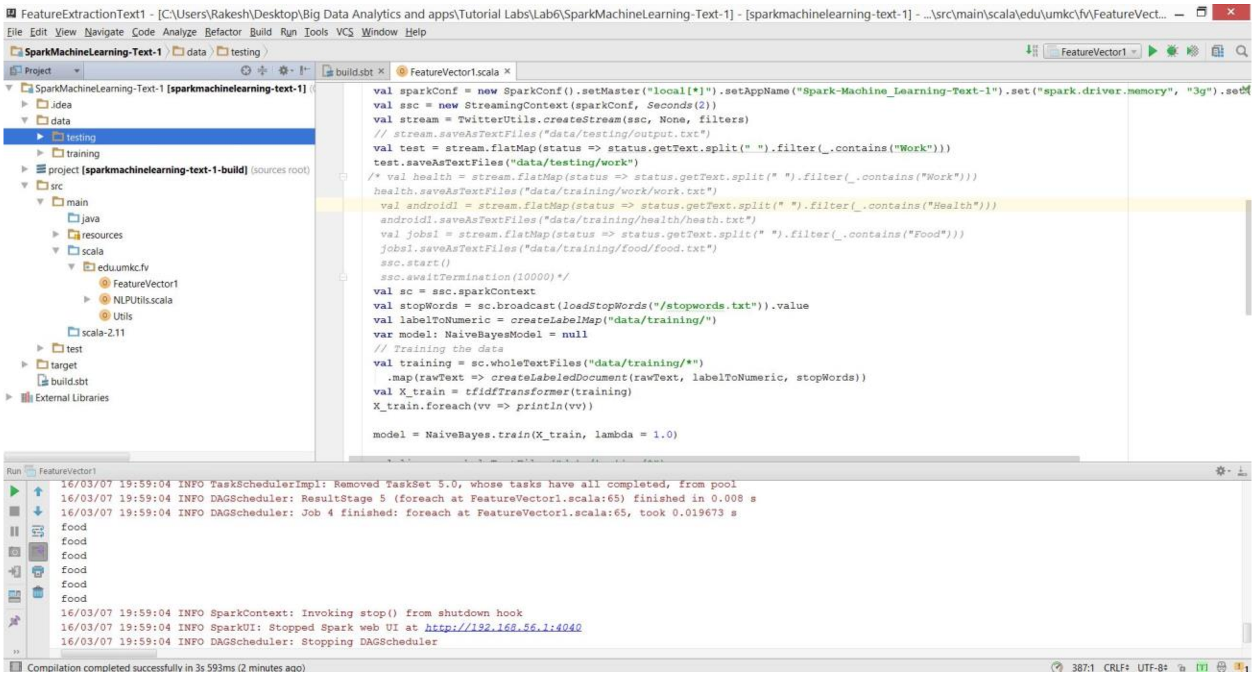
FeatureExtractionText1 - [C:\Users\Rakesh\Desktop\Big Data Analytics and apps\Tutorial Labs\Lab6\SparkMachineLearning-Text-1] - [sparkmachinelearning-text-1] - ...\src\main\scala\edu\umkc\fv\FeatureVect...
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkMachineLearning-Text-1 data testing
FeatureVector1
Project
build.sbt
FeatureVector1.scala
SparkMachineLearning-Text-1 [sparkmachinelearning-text-1]
.idea
data
testing
training
project [sparkmachinelearning-text-1-build] (sources root)
src
main
java
resources
scala
edu.umkc.fv
FeatureVector1
NLPUtils.scala
Utils
scala-2.11
test
target
build.sbt
External Libraries
Run FeatureVector1
387:1 CRLF UTF-8
8:02 PM
3/7/2016

```
 * Created by Sricharan on 02-Mar-16.
 */
object FeatureVector1 {

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

    System.setProperty("hadoop.home.dir", "F:\\winutils")
    System.setProperty("twitter4j.oauth.consumerKey", "1oVEx4fjDXAqzOoNLZxOJw91C")
    System.setProperty("twitter4j.oauth.consumerSecret", "gj07311QSSoZMfWJfeCqW0KUCzrKiqKyEMLcr3C4aTWQEQa2Xm")
    System.setProperty("twitter4j.oauth.accessToken", "702647205807534080-UZjUEaNvQaYhTWI3MdD3cyBHAamf3jd")
    System.setProperty("twitter4j.oauth.accessTokenSecret", "E129gJVngXwIUJmFGb0MoKvUeN5UBGcxdBNCrUrUveSbq")


    val sparkConf = new SparkConf().setMaster("local[*]").setAppName("Spark-Machine_Learning-Text-1").set("spark.driver.memory", "3g").set(
    val ssc = new StreamingContext(sparkConf, Seconds(2))
    val stream = TwitterUtils.createStream(ssc, None, filters)
    // stream.saveAsTextFiles("data/testing/output.txt")
    val test = stream.flatMap(status => status.getText.split(" ").filter(_.contains("Work")))
    test.saveAsTextFiles("data/testing/work")
   /* val health = stream.flatMap(status => status.getText.split(" ").filter(_.contains("Work")))
    health.saveAsTextFiles("data/training/work/work.txt")
     val android1 = stream.flatMap(status => status.getText.split(" ").filter(_.contains("Health")))
     android1.saveAsTextFiles("data/training/health/heath.txt")
     val jobs1 = stream.flatMap(status => status.getText.split(" ").filter(_.contains("Food")))
     jobs1.saveAsTextFiles("data/training/food/food.txt")
```

```
16/03/07 19:59:04 INFO MemoryStore: ensureFreeSpace(2384716) called with curMem=68210843, maxMem=1536184811
16/03/07 19:59:04 INFO MemoryStore: Block broadcast_10_piece0 stored as bytes in memory (estimated size 2.3 MB, free 1397.7 MB)
16/03/07 19:59:04 INFO BlockManagerInfo: Added broadcast_10_piece0 in memory on localhost:49920 (size: 2.3 MB, free: 1462.0 MB)
16/03/07 19:59:04 INFO SparkContext: Created broadcast 10 from broadcast at NaiveBayes.scala:89
PREDICTION
16/03/07 19:59:04 INFO SparkContext: Starting job: foreach at FeatureVector1.scala:65
16/03/07 19:59:04 INFO DAGScheduler: Got job 4 (foreach at FeatureVector1.scala:65) with 1 output partitions
16/03/07 19:59:04 INFO DAGScheduler: Final stage: ResultStage 5(foreach at FeatureVector1.scala:65)
16/03/07 19:59:04 INFO DAGScheduler: Parents of final stage: List()
16/03/07 19:59:04 INFO DAGScheduler: Missing parents: List()
16/03/07 19:59:04 INFO DAGScheduler: Submitting ResultStage 5 (MapPartitionsRDD[17] at mapPartitions at NaiveBayes.scala:90), which has no missing parents
16/03/07 19:59:04 INFO MemoryStore: ensureFreeSpace(5984) called with curMem=70595559, maxMem=1536184811
```

Compilation completed successfully in 3s 593ms (3 minutes ago)

2. Based on the testing data it is predicting as food.



FeatureExtractionText1 - [C:\Users\Rakesh\Desktop\Big Data Analytics and apps\Tutorial Labs\Lab6\SparkMachineLearning-Text-1] - [sparkmachinelearning-text-1] - ...\src\main\scala\edu\umkc\fv\FeatureVect...
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkMachineLearning-Text-1 data testing
FeatureVector1
Project
build.sbt
FeatureVector1.scala
SparkMachineLearning-Text-1 [sparkmachinelearning-text-1]
.idea
data
testing
training
project [sparkmachinelearning-text-1-build] (sources root)
src
main
java
resources
scala
edu.umkc.fv
FeatureVector1
NLPUtils.scala
Utils
scala-2.11
test
target
build.sbt
External Libraries
Run FeatureVector1
387:1 CRLF UTF-8

```
val sparkConf = new SparkConf().setMaster("local[*]").setAppName("Spark-Machine_Learning-Text-1").set("spark.driver.memory", "3g").set
val ssc = new StreamingContext(sparkConf, Seconds(2))
val stream = TwitterUtils.createStream(ssc, None, filters)
// stream.saveAsTextFiles("data/testing/output.txt")
val test = stream.flatMap(status => status.getText.split(" ").filter(_.contains("Work")))
test.saveAsTextFiles("data/testing/work")
/* val health = stream.flatMap(status => status.getText.split(" ").filter(_.contains("Work")))
health.saveAsTextFiles("data/training/work/work.txt")
 val android1 = stream.flatMap(status => status.getText.split(" ").filter(_.contains("Health")))
 android1.saveAsTextFiles("data/training/health/heath.txt")
 val jobs1 = stream.flatMap(status => status.getText.split(" ").filter(_.contains("Food")))
 jobs1.saveAsTextFiles("data/training/food/food.txt")
 ssc.start()
 ssc.awaitTermination(10000)*/
val sc = ssc.sparkContext
val stopWords = sc.broadcast(loadStopWords("/stopwords.txt")).value
val labelToNumeric = createLabelMap("data/training/")
var model: NaiveBayesModel = null
// 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)
```

```
16/03/07 19:59:04 INFO TaskSchedulerImpl: Removed TaskSet 5.0, whose tasks have all completed, from pool
16/03/07 19:59:04 INFO DAGScheduler: ResultStage 5 (foreach at FeatureVector1.scala:65) finished in 0.008 s
16/03/07 19:59:04 INFO DAGScheduler: Job 4 finished: foreach at FeatureVector1.scala:65, took 0.019673 s
food
food
food
food
food
food
16/03/07 19:59:04 INFO SparkContext: Invoking stop() from shutdown hook
16/03/07 19:59:04 INFO SparkUI: Stopped Spark web UI at http://192.168.56.1:4040
16/03/07 19:59:04 INFO DAGScheduler: Stopping DAGScheduler
```

Compilation completed successfully in 3s 593ms (2 minutes ago)