Lab Assignment 4


## NAME: HARISH CHANDRA JYOSHI

## CLASS ID: 06

## TEAM ID: 04

## UMKC EMAIL ID: hjddh@mail.umkc.edu

## NAME: ATLURI VENKATA AKHILA KRISHNA

## CLASS ID: 01

## TEAM ID: 04

## UMKC EMAIL ID: vagq2@mail.umkc.edu


## Video link: https://youtu.be/mGymAaEUWu4

---

### Task 1: Implementing Spark Mlib classification algorithms on a data set.

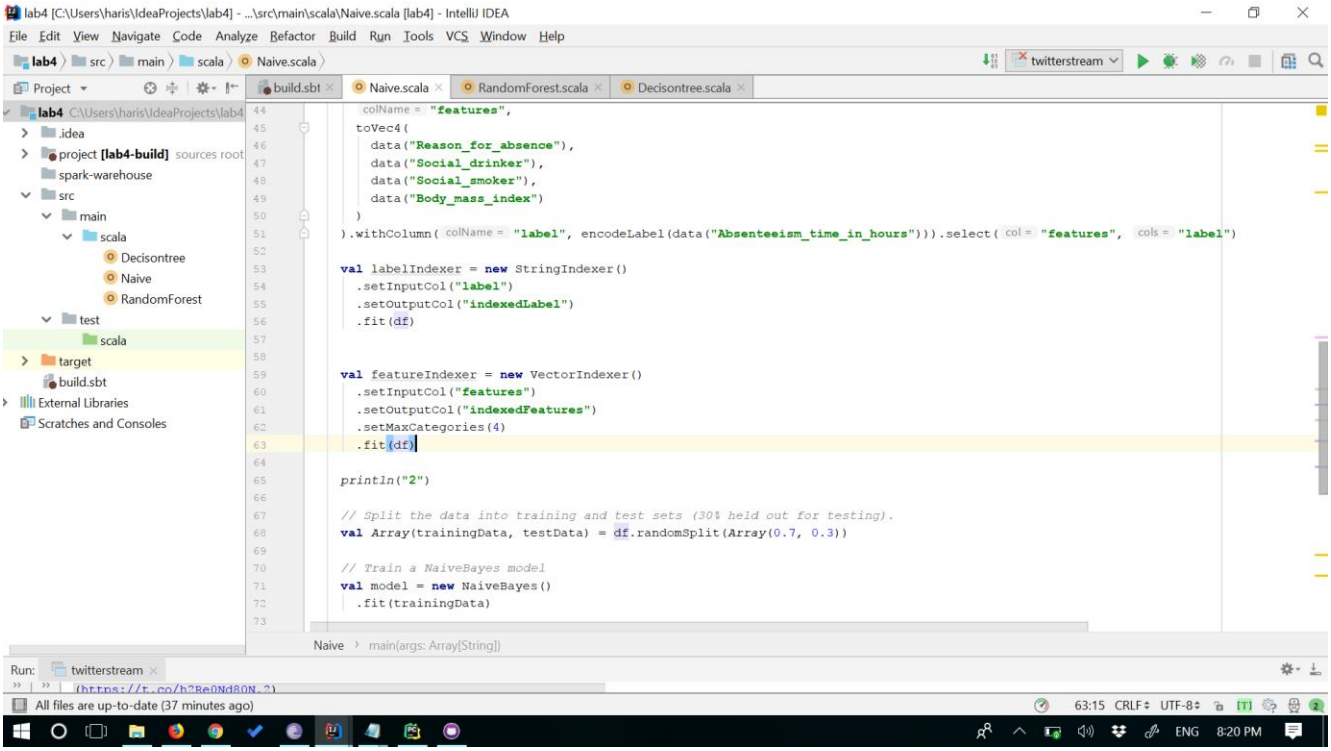
DataSet used: https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work

The input data is split into 70% for training and 30% for testing. Classification is performed based on the columns 'Reason for absence', 'social Drinker', 'Social smoker', 'Body mass Index', 'Absenteeism at work'.

# Lab Assignment 4

## 1.Naive Bayes:

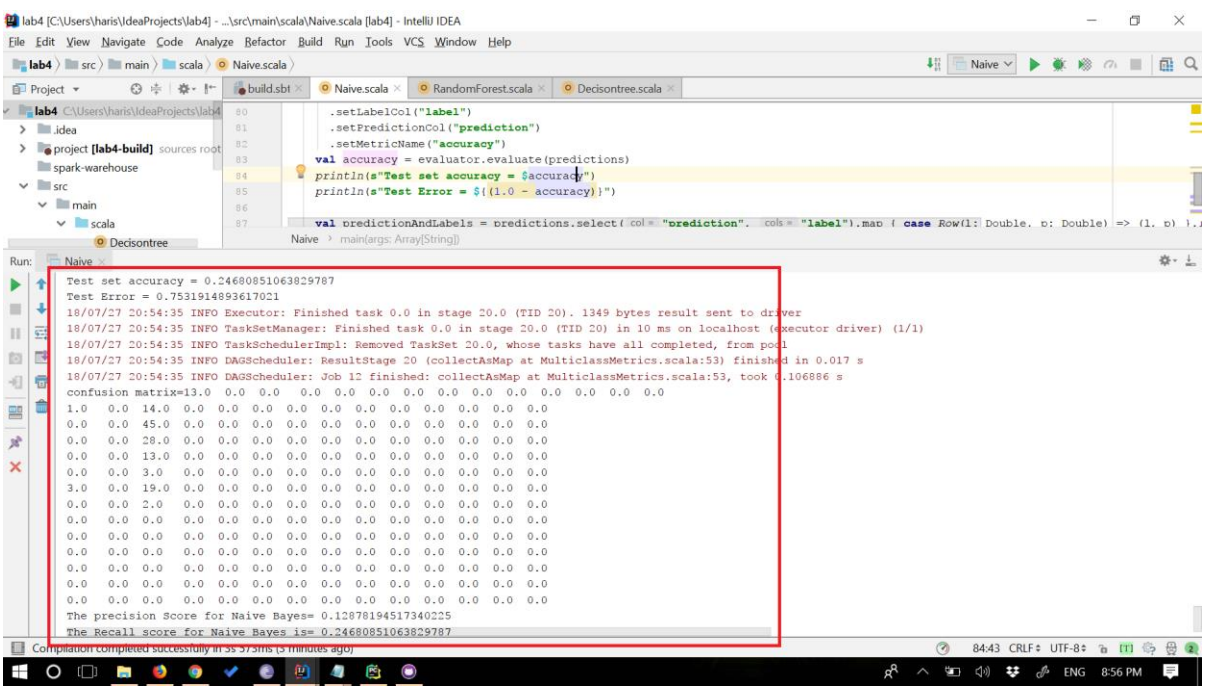
It's classification task based on Baye's theorem.



```
      colName = "features",
      toVec4(
        data("Reason_for_absence"),
        data("Social_drinker"),
        data("Social_smoker"),
        data("Body_mass_index")
      )
    ).withColumn( colName = "label", encodeLabel(data("Absenteeism_time_in_hours"))).select( col = "features",  cols = "label")

    val labelIndexer = new StringIndexer()
      .setInputCol("label")
      .setOutputCol("indexedLabel")
      .fit(df)


    val featureIndexer = new VectorIndexer()
      .setInputCol("features")
      .setOutputCol("indexedFeatures")
      .setMaxCategories(4)
      .fit(df)

    println("2")

    // Split the data into training and test sets (30% held out for testing).
    val Array(trainingData, testData) = df.randomSplit(Array(0.7, 0.3))

    // Train a NaiveBayes model
    val model = new NaiveBayes()
      .fit(trainingData)
```

All files are up-to-date (37 minutes ago)

## Output after Running the Naive Bayes algorithm.



```
      .setLabelCol("label")
      .setPredictionCol("prediction")
      .setMetricName("accuracy")
    val accuracy = evaluator.evaluate(predictions)
    println(s"Test set accuracy = $accuracy")
    println(s"Test Error = ${(1.0 - accuracy)}")

    val predictionAndLabels = predictions.select( col = "prediction",  cols = "label").map { case Row(l: Double, p: Double) => (l, p) }.
```

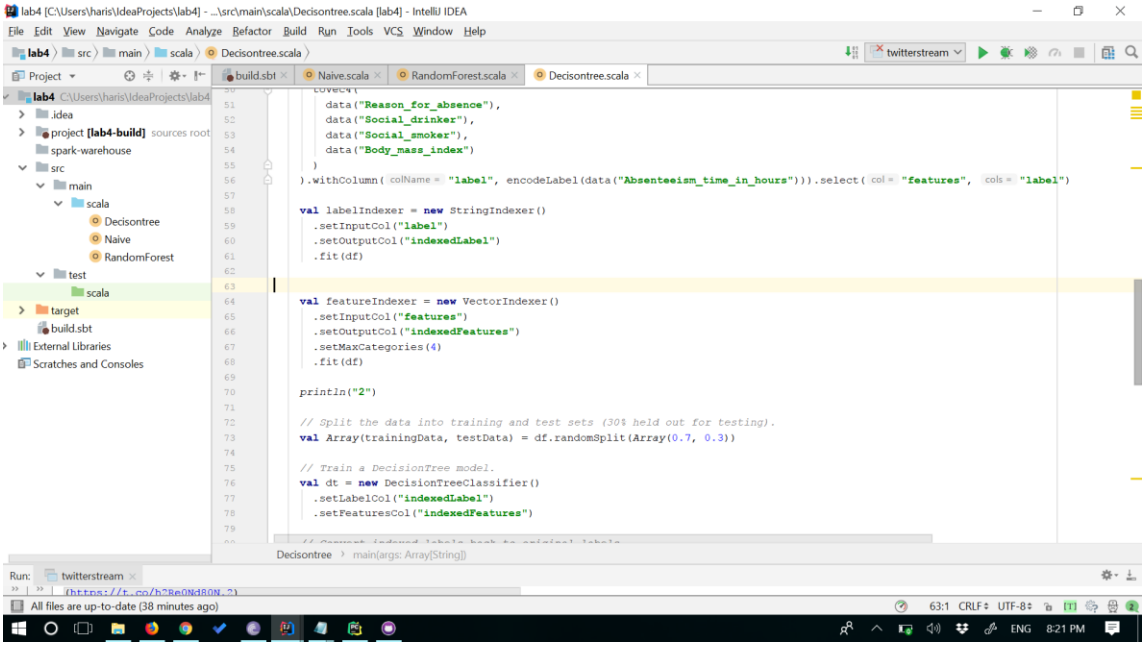
```
Test set accuracy = 0.24680851063829787
Test Error = 0.7531914893617021
18/07/27 20:54:35 INFO Executor: Finished task 0.0 in stage 20.0 (TID 20). 1349 bytes result sent to driver
18/07/27 20:54:35 INFO TaskSetManager: Finished task 0.0 in stage 20.0 (TID 20) in 10 ms on localhost (executor driver) (1/1)
18/07/27 20:54:35 INFO TaskSchedulerImpl: Removed TaskSet 20.0, whose tasks have all completed, from pool
18/07/27 20:54:35 INFO DAGScheduler: ResultStage 20 (collectAsMap at MulticlassMetrics.scala:53) finished in 0.017 s
18/07/27 20:54:35 INFO DAGScheduler: Job 12 finished: collectAsMap at MulticlassMetrics.scala:53, took 0.106886 s
confusion matrix=13.0  0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
1.0   0.0  14.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  45.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  28.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  13.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  3.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
3.0   0.0  19.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  2.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0   0.0  0.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
The precision Score for Naive Bayes= 0.12878194517340225
The Recall score for Naive Bayes is= 0.24680851063829787
```

Compilation completed successfully in 3s 573ms (3 minutes ago)

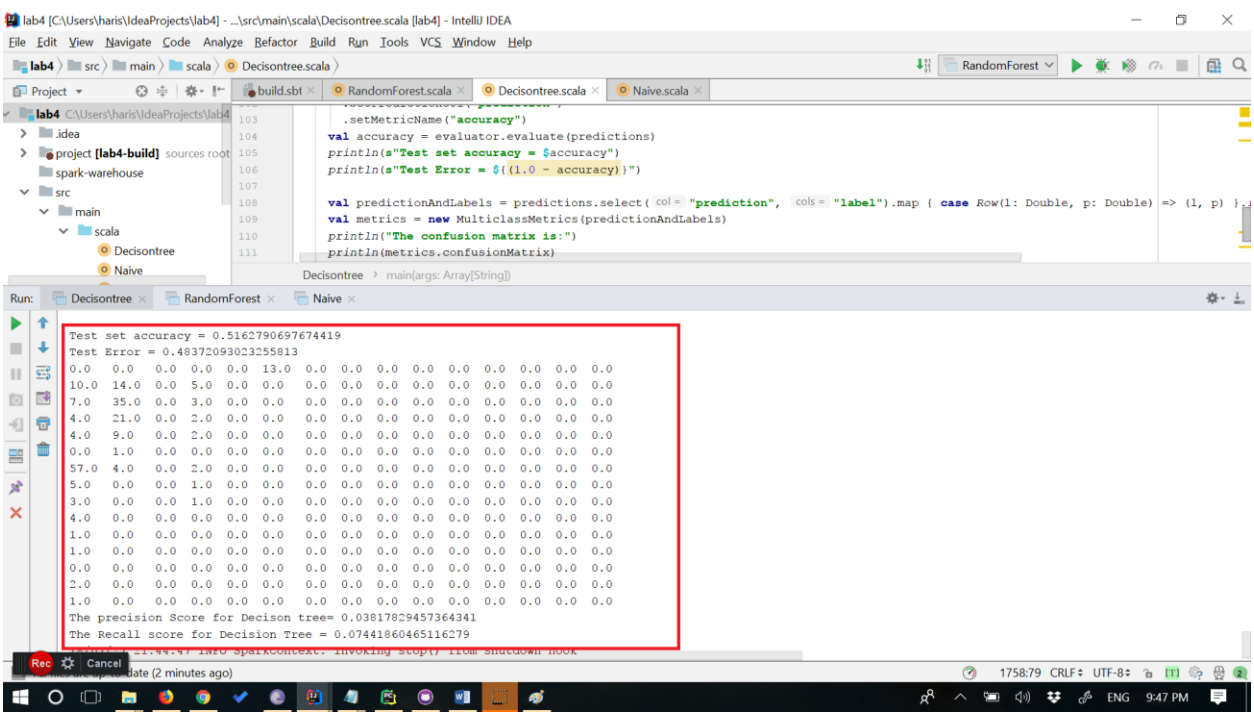
Lab Assignment 4

## 2. Decision Tree

The Input data is split into 70%-30%



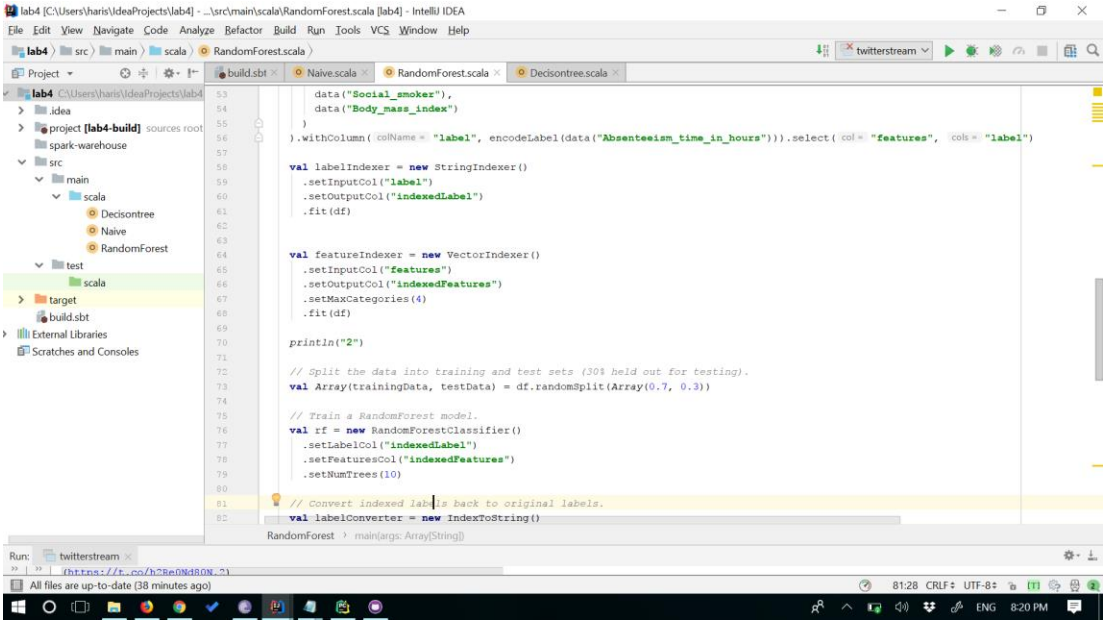

## output for Decision tree Algorithm



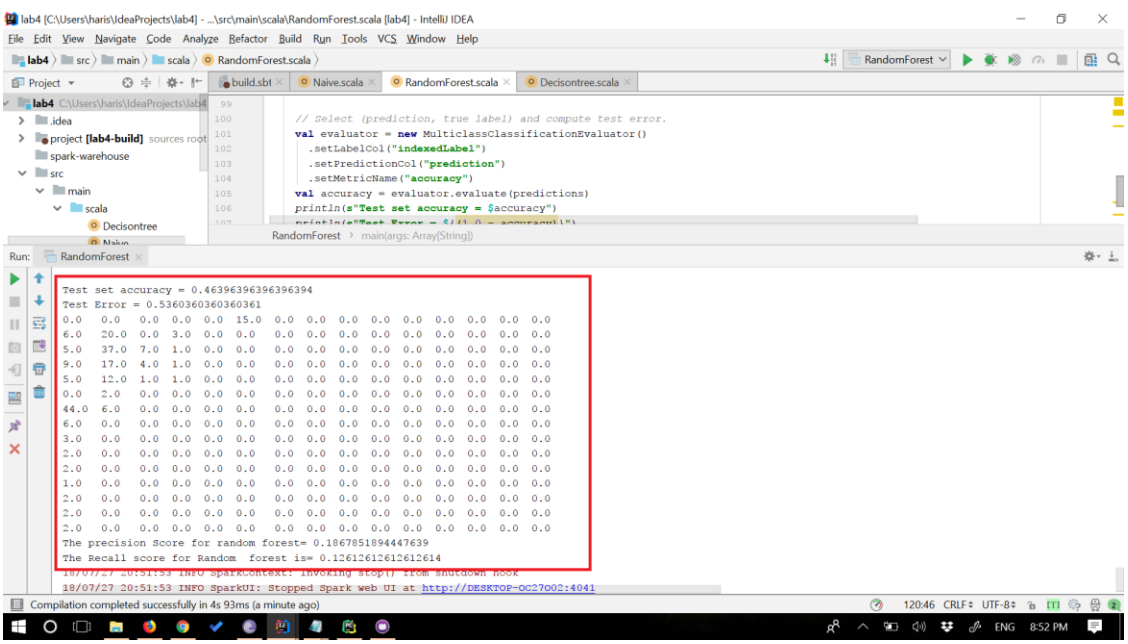
Lab Assignment 4

## 3. Random forest

The Input data is split into 70%-30%




## Output for random Forest algorithm



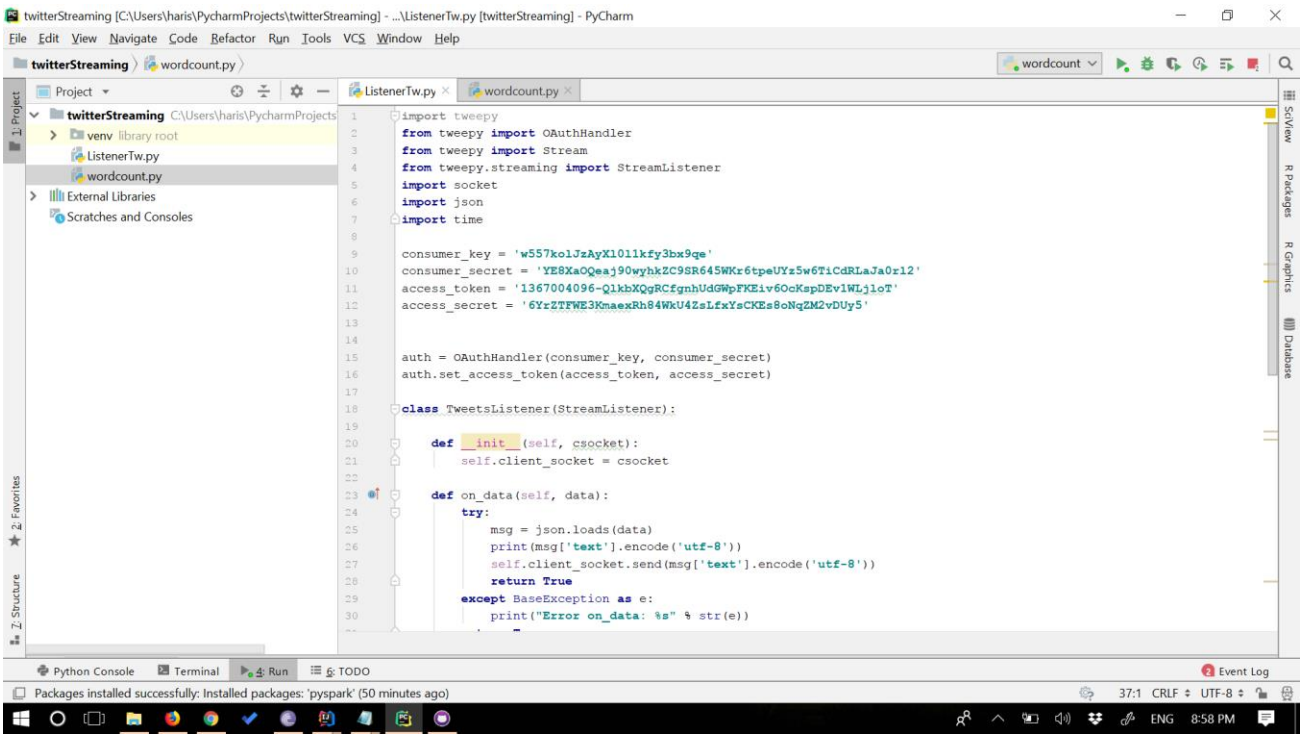
**Conclusion:**

According to the Results, Random Forest has the highest accuracy of 46% compared to decison tree and naive bayes. The confusion matrix and precision and recall values are shown for all the algorithms.

## Task 2: Spark Streaming

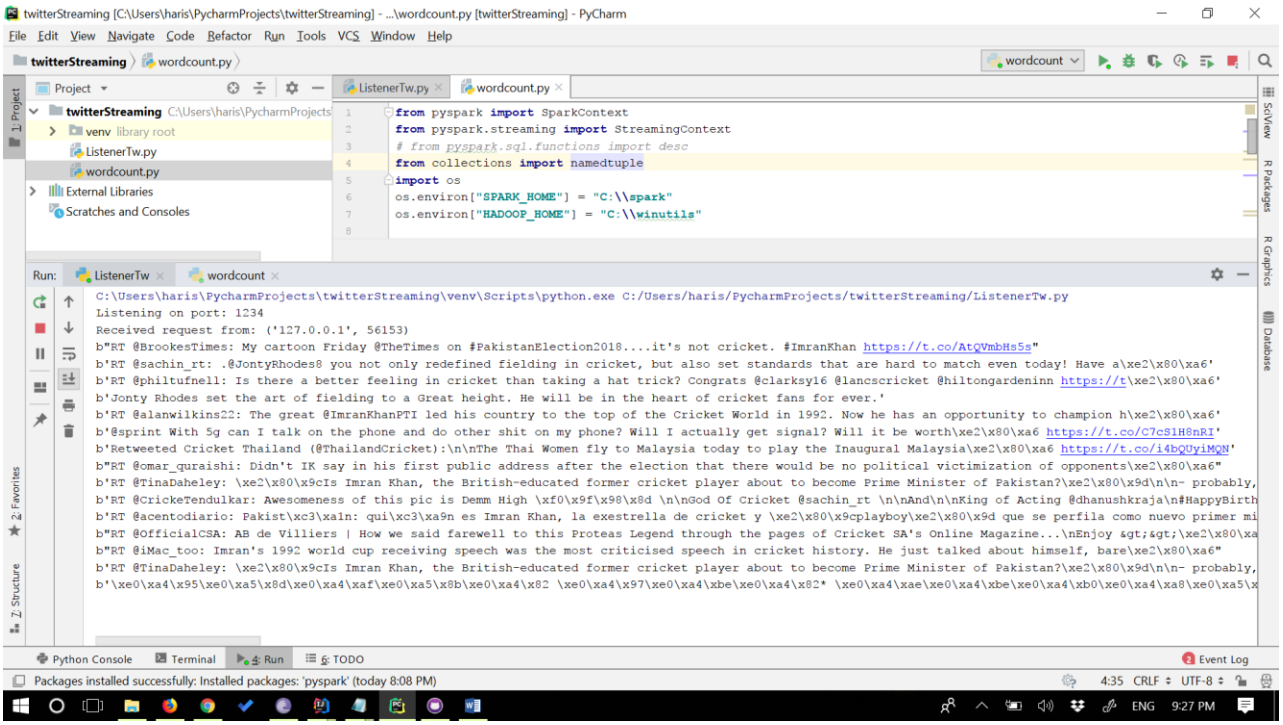
In this task, streaming is performed on the Twitter data which is fetched using access and consumer keys.

### 1.Tweets collection:

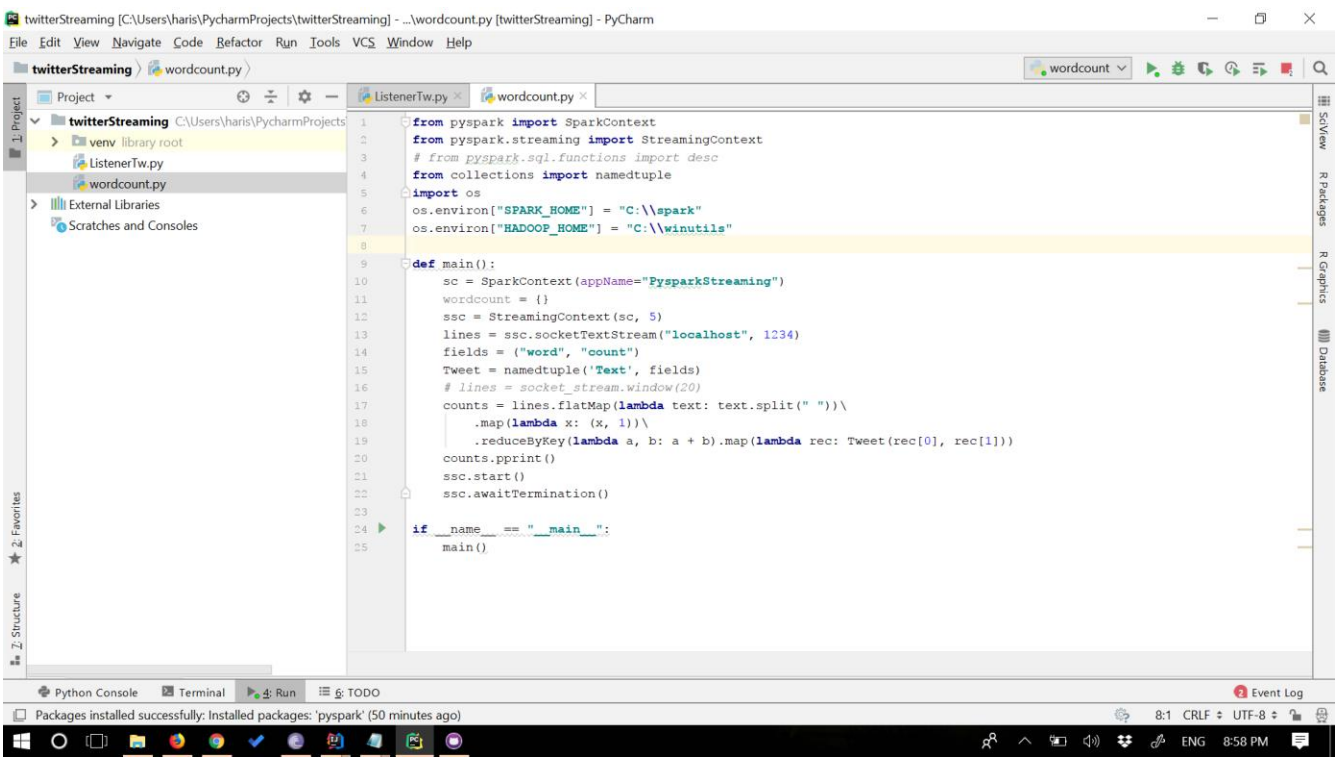


Lab Assignment 4

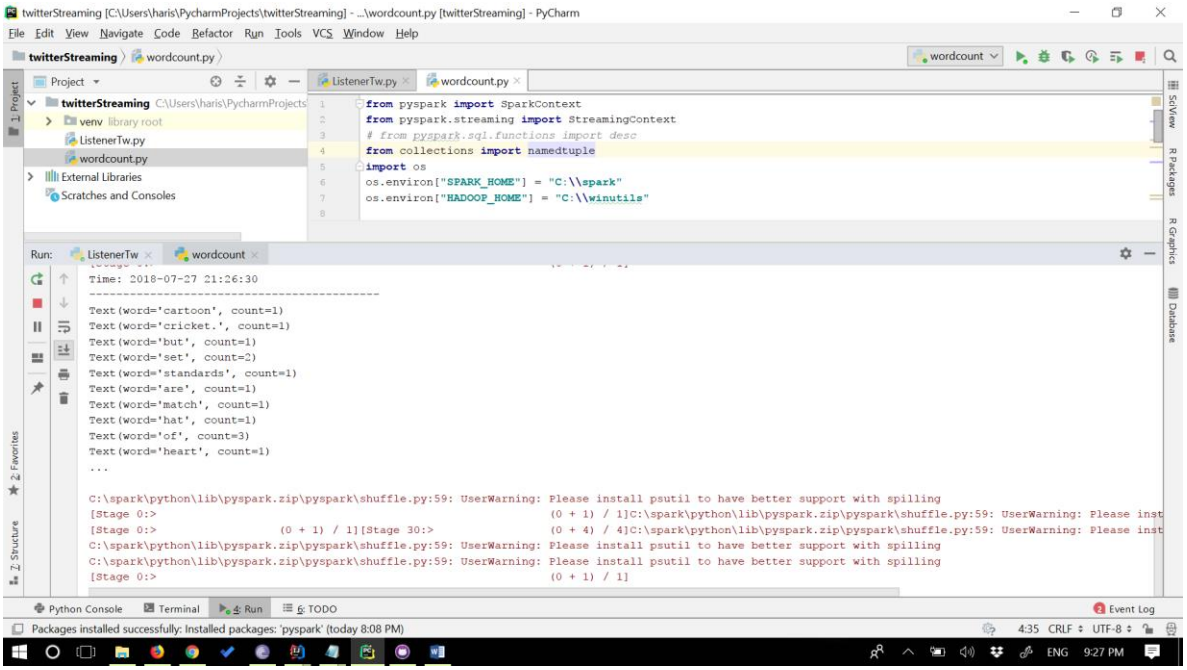
## Output for tweets collection:




## 2. Streaming and performing word count on twitter data:



**output for Wordcount of Twitter data:**




References:


1. https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work

2. https://spark.apache.org/docs/1.5.2/streaming-programming-guide.html

3. https://spark.apache.org/docs/2.1.0/mllib-naive-bayes.html

4. https://spark.apache.org/docs/2.2.0/mllib-decision-tree.html

5.https://github.com/dennyglee/databricks/blob/master/notebooks/Users/denny%40databricks.com/blog%20books/Scalable%20Decision%20Trees%20with%20MLlib.scala

6. https://www.linkedin.com/pulse/apache-spark-streaming-twitter-python-laurent-weichberger/