

**NAME: HARISH CHANDRA JYOSHI**

**CLASS ID: 06**

**TEAM ID: 04**

**UMKC EMAIL ID: [hjddh@mail.umkc.edu](mailto:hjddh@mail.umkc.edu)**

**NAME: ATLURI VENKATA AKHILA KRISHNA**

**CLASS ID: 01**

**TEAM ID: 04**

**UMKC EMAIL ID: [vagq2@mail.umkc.edu](mailto:vagq2@mail.umkc.edu)**

**Video link: <https://youtu.be/mGymAaEUWu4>**

---

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

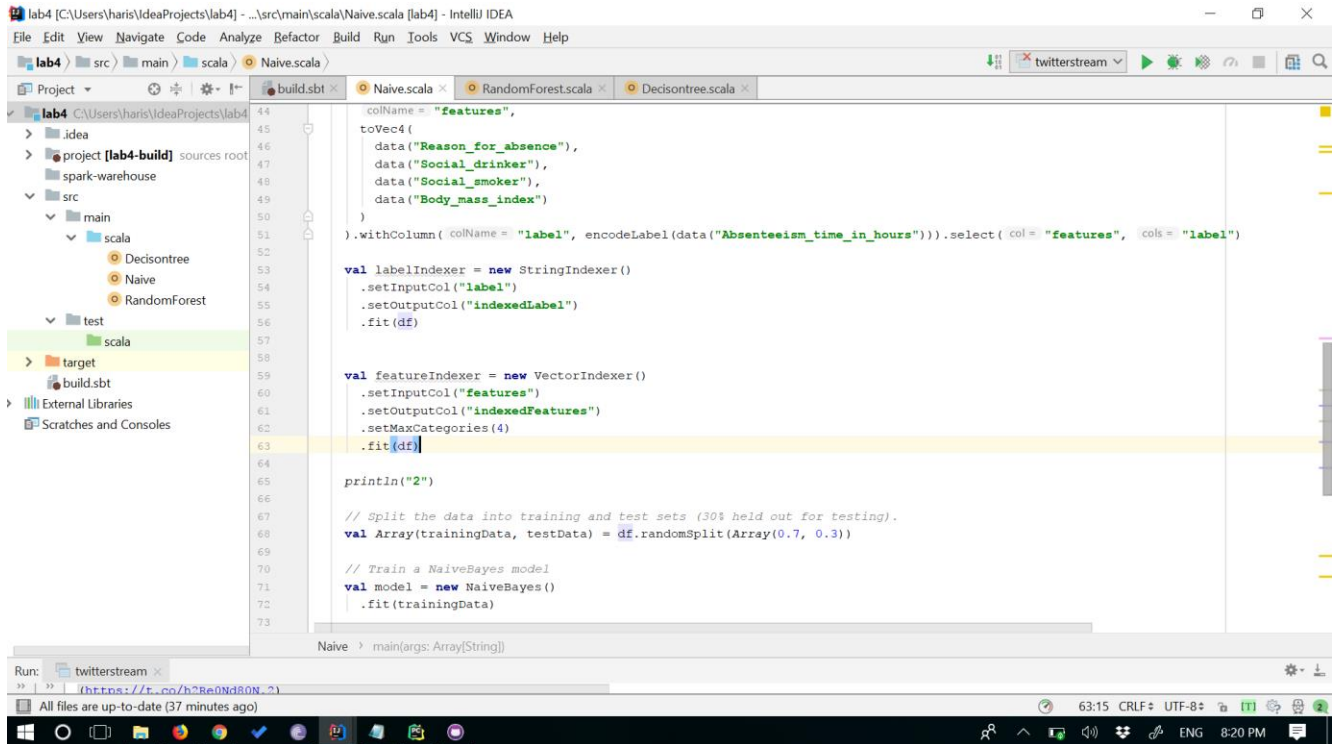
DataSet used: <https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work>

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

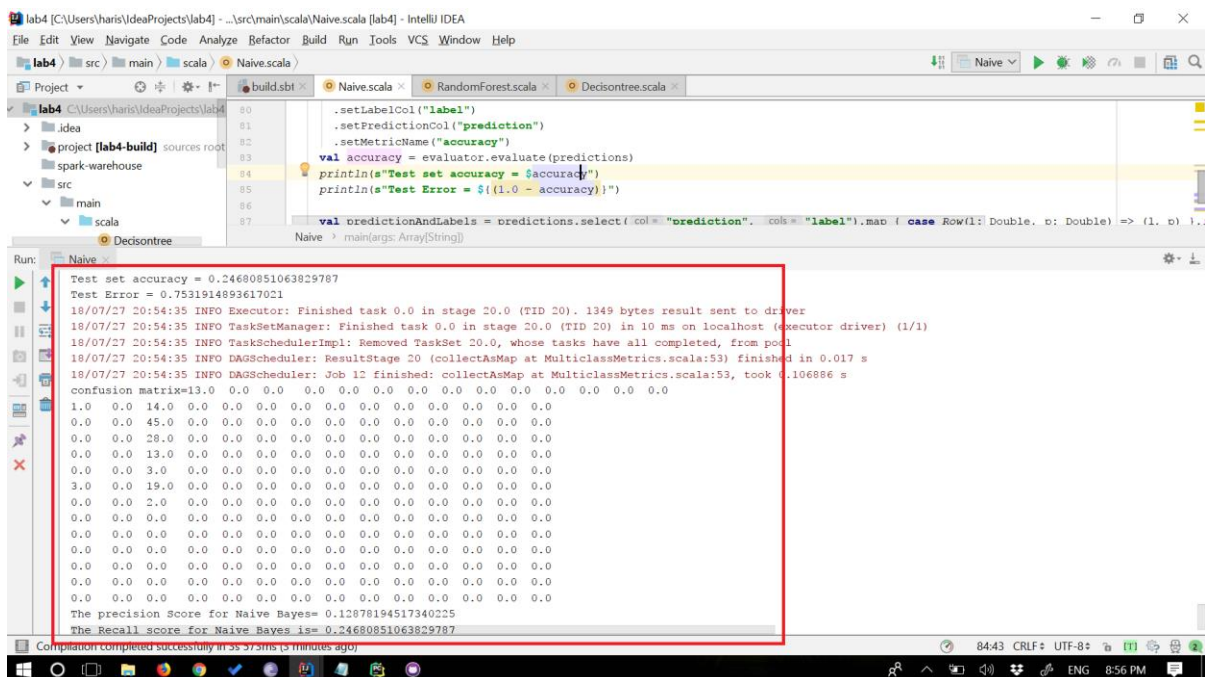
## 1. Naive Bayes:

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



```
44 colName = "features",
45 toVec4(
46   data("Reason_for_absence"),
47   data("Social_drinker"),
48   data("Social_smoker"),
49   data("Body_mass_index")
50 )
51 ).withColumn( colName = "label", encodeLabel(data("Absenteeism_time_in_hours")))
52 .select( col = "features", cols = "label")
53
54 val labelIndexer = new StringIndexer()
55   .setInputCol("label")
56   .setOutputCol("indexedLabel")
57   .fit(df)
58
59 val featureIndexer = new VectorIndexer()
60   .setInputCol("features")
61   .setOutputCol("indexedFeatures")
62   .setMaxCategories(4)
63   .fit(df)
64
65 println("2")
66
67 // Split the data into training and test sets (30% held out for testing).
68 val Array(trainingData, testData) = df.randomSplit(Array(0.7, 0.3))
69
70 // Train a NaiveBayes model
71 val model = new NaiveBayes()
72   .fit(trainingData)
73
74 Naive > main(args: Array[String])
```

## Output after Running the Naive Bayes algorithm.



```
80 .setLabelCol("label")
81 .setPredictionCol("prediction")
82 .setMetricName("accuracy")
83 val accuracy = evaluator.evaluate(predictions)
84 println(s"Test set accuracy = $accuracy")
85 println(s"Test Error = ${(1.0 - accuracy)}")
86
87 val predictionAndLabels = predictions.select( col = "prediction", cols = "label").map { case Row(p: Double, l: Double) => (l, p) }

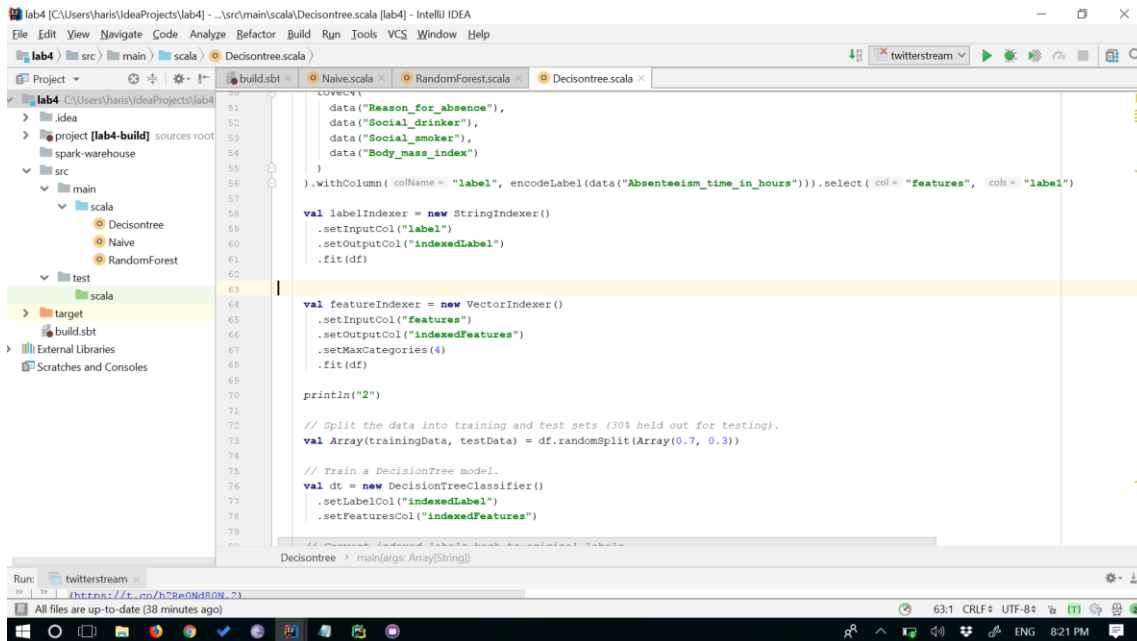
Naive > main(args: Array[String])

Run: Naive
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 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 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 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 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 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 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 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 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
```

# Lab Assignment 4

## 2. Decision Tree

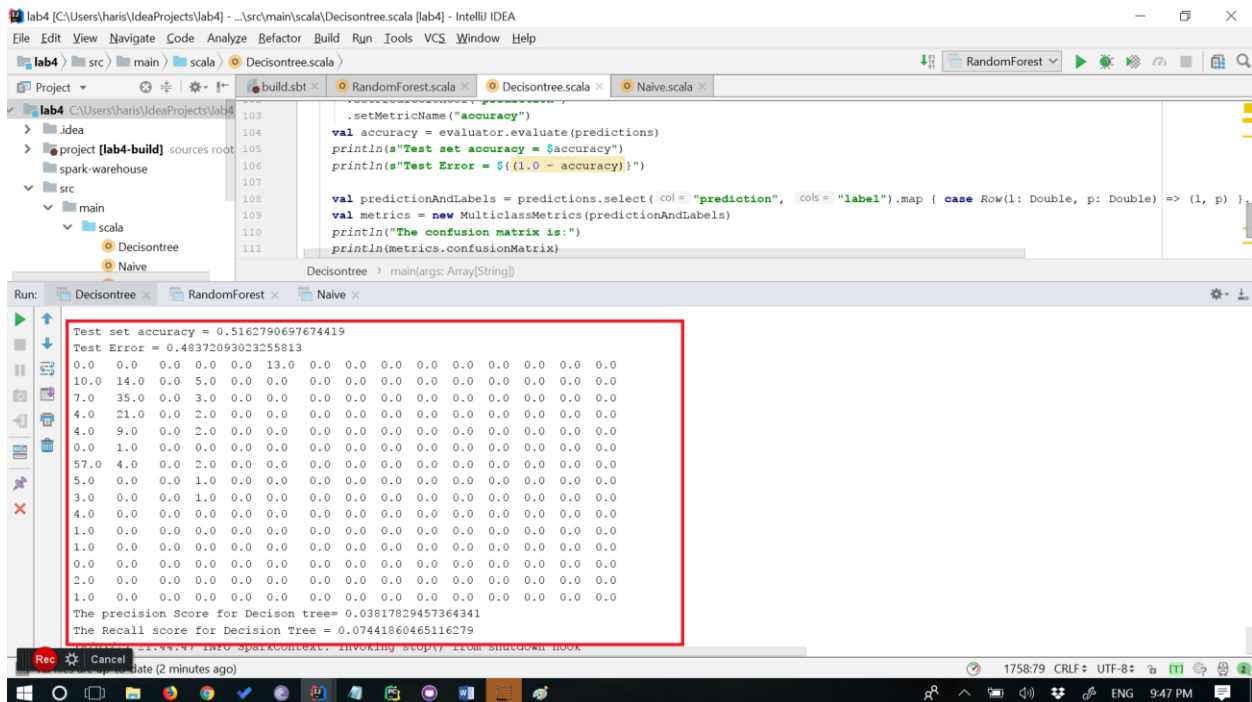
The Input data is split into 70%-30%



```
114 [C:\Users\haris\IdeaProjects\lab4] - ...src\main\scala\Decisontree.scala [lab4] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
lab4 [C:\Users\haris\IdeaProjects\lab4] src main scala Decisontree.scala
Project lab4 [C:\Users\haris\IdeaProjects\lab4]
  idea
  project [lab4-build] sources root
  spark-warehouse
  src
    main
      scala
        Decisontree
        Naive
        RandomForest
    test
      scala
      target
  build.sbt
  External Libraries
  Scratches and Consoles

51 data("Reason_for_absence"),
52 data("Social_drinker"),
53 data("Social_smoker"),
54 data("Body_mass_index")
55 )
56 ).withColumn(colName = "label", encodeLabel(data("Absenteeism_time_in_hours"))).select(col = "features", cols = "label")
57
58 val labelIndexer = new StringIndexer()
59   .setInputCol("label")
60   .setOutputCol("indexedLabel")
61   .fit(df)
62
63
64 val featureIndexer = new VectorIndexer()
65   .setInputCol("features")
66   .setOutputCol("indexedFeatures")
67   .setMaxCategories(4)
68   .fit(df)
69
70 println("2")
71
72 // Split the data into training and test sets (30% held out for testing).
73 val Array(trainingData, testData) = df.randomSplit(Array(0.7, 0.3))
74
75 // Train a DecisionTree model.
76 val dt = new DecisionTreeClassifier()
77   .setLabelCol("indexedLabel")
78   .setFeaturesCol("indexedFeatures")
79
80 // Compute trained model's predictions on test data
81 val predictionAndLabels = testData.select(col = "prediction", cols = "label").map { case Row(l: Double, p: Double) => (l, p) }
82
83 val metrics = new MulticlassMetrics(predictionAndLabels)
84 println("The confusion matrix is:")
85 println(metrics.confusionMatrix)
86
87 Decisontree > main(args: Array[String])
```

## output for Decision tree Algorithm



```
114 [C:\Users\haris\IdeaProjects\lab4] - ...src\main\scala\Decisontree.scala [lab4] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
lab4 [C:\Users\haris\IdeaProjects\lab4] src main scala Decisontree.scala
Project lab4 [C:\Users\haris\IdeaProjects\lab4]
  idea
  project [lab4-build] sources root
  spark-warehouse
  src
    main
      scala
        Decisontree
        Naive
        RandomForest
    test
      scala
      target
  build.sbt
  External Libraries
  Scratches and Consoles

103   .setMetricName("accuracy")
104   val accuracy = evaluator.evaluate(predictions)
105   println(s"Test set accuracy = $accuracy")
106   println(s"Test Error = ${(1.0 - accuracy)}")
107
108   val predictionAndLabels = predictions.select(col = "prediction", cols = "label").map { case Row(l: Double, p: Double) => (l, p) }
109   val metrics = new MulticlassMetrics(predictionAndLabels)
110   println("The confusion matrix is:")
111   println(metrics.confusionMatrix)
112
113   Decisontree > main(args: Array[String])

Run: Decisontree x RandomForest x Naive x
Test set accuracy = 0.5162790697674419
Test Error = 0.48372093023255813
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
10.0 14.0 0.0 5.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
7.0 35.0 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
4.0 21.0 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
4.0 9.0 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 1.0 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
57.0 4.0 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
5.0 0.0 0.0 1.0 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 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
4.0 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
1.0 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
1.0 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
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
1.0 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 Decision tree= 0.03817829457364341
The Recall score for Decision Tree = 0.07441860465116279
21:44:47 INFO sparkcontext: Invoking stop() from shutdown hook
date (2 minutes ago)
1758:79 CRLF UTF-8 ENG 9:47 PM
```

## Lab Assignment 4

### 3. Random forest

The Input data is split into 70%-30%

The screenshot displays the IntelliJ IDEA IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The project 'lab4' is open, with the file 'RandomForest.scala' selected. The code in the editor is as follows:

```
53 data("Social_smoker"),
54 data("Body_mass_index")
55 )
56 ).withColumn(colName = "label", encodeLabel(data("Absenteeism_time_in_hours"))).select(col = "features", cols = "label")
57
58 val labelIndexer = new StringIndexer()
59   .setInputCol("label")
60   .setOutputCol("indexedLabel")
61   .fit(df)
62
63
64 val featureIndexer = new VectorIndexer()
65   .setInputCol("features")
66   .setOutputCol("indexedFeatures")
67   .setMaxCategories(4)
68   .fit(df)
69
70 println("2")
71
72 // Split the data into training and test sets (30% held out for testing).
73 val Array(trainingData, testData) = df.randomSplit(Array(0.7, 0.3))
74
75 // Train a RandomForest model.
76 val rf = new RandomForestClassifier()
77   .setLabelCol("indexedLabel")
78   .setFeaturesCol("indexedFeatures")
79   .setNumTrees(10)
80
81 // Convert indexed labels back to original labels.
82 val labelConverter = new IndexToString()
83   .setInputCol("indexedLabel")
84   .setOutputCol("label")
85   .fit(trainingData)
```

The project structure on the left shows the following hierarchy:

- lab4 (C:\Users\haris\IdeaProjects\lab4)
  - .idea
  - project [lab4-build] sources root
    - spark-warehouse
    - src
      - main
        - scala
          - DecisionTree
          - Naive
          - RandomForest
        - test
          - scala
        - target
        - build.sbt
      - External Libraries
      - Scratches and Consoles

The bottom status bar shows the current file is 'twitterstream.scala' and all files are up-to-date (38 minutes ago). The system tray at the bottom indicates the time is 8:28 PM and the language is set to ENG.

## Output for random Forest algorithm

The screenshot shows the IntelliJ IDEA IDE with a Scala project named 'lab4'. The project structure includes 'src', 'main', 'scala', and 'RandomForest.scala'. The 'Run' console displays the output of a multiclass classification evaluation using a RandomForest model. The output is highlighted with a red box.

```
Test set accuracy = 0.46396396396396394
Test Error = 0.5360360360360361
0.0 0.0 0.0 0.0 0.0 15.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
6.0 20.0 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 0.0
5.0 37.0 7.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
9.0 17.0 4.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
5.0 12.0 1.0 1.0 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
44.0 6.0 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
6.0 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
3.0 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
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
1.0 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
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
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
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
The precision Score for random forest= 0.1867851894447639
The Recall score for Random forest is= 0.12612612612614
```

The console also shows the following messages:

```
18/07/27 20:51:53 INFO SparkContext: Invoking stop() from shutdown hook
18/07/27 20:51:53 INFO SparkUI: Stopped Spark web UI at http://DESKTOP-OC27002:4041
```

The bottom status bar indicates: "Compilation completed successfully in 4s 93ms (a minute ago)".

## Lab Assignment 4

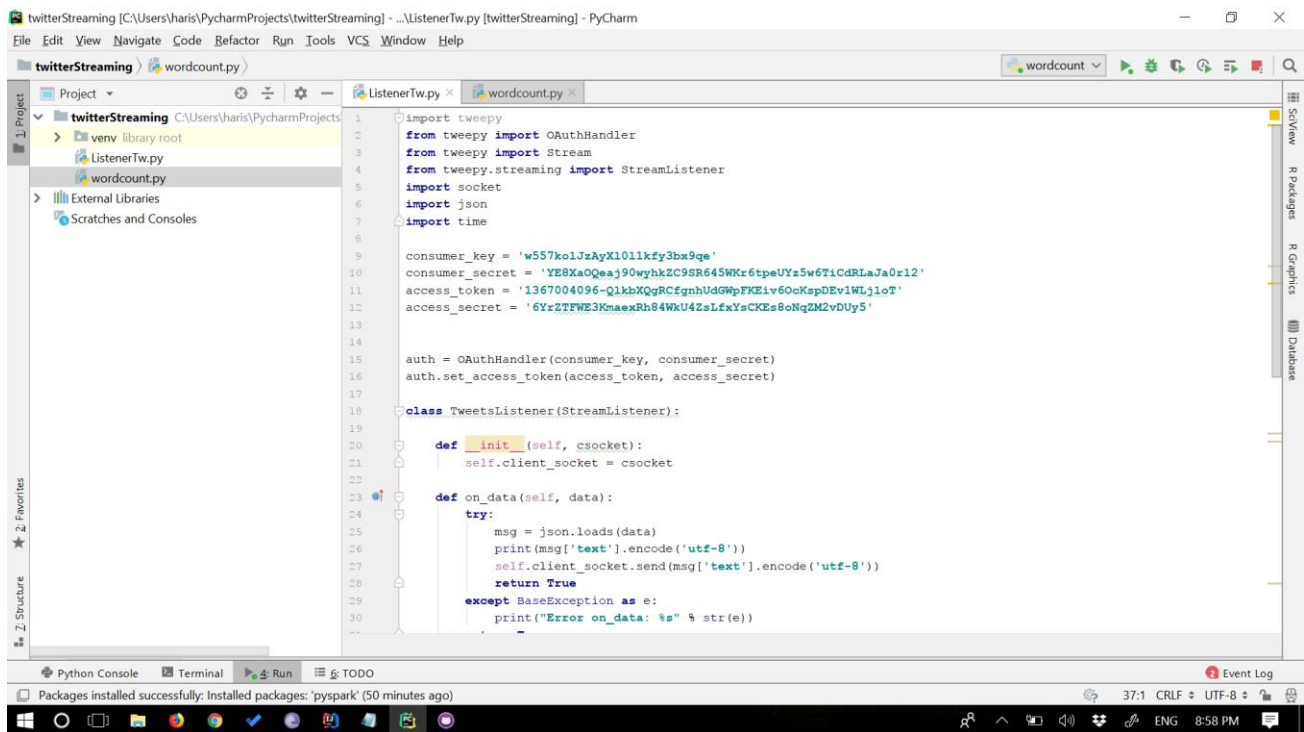
### Conclusion:

According to the Results, Random Forest has the highest accuracy of 46% compared to decision 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:

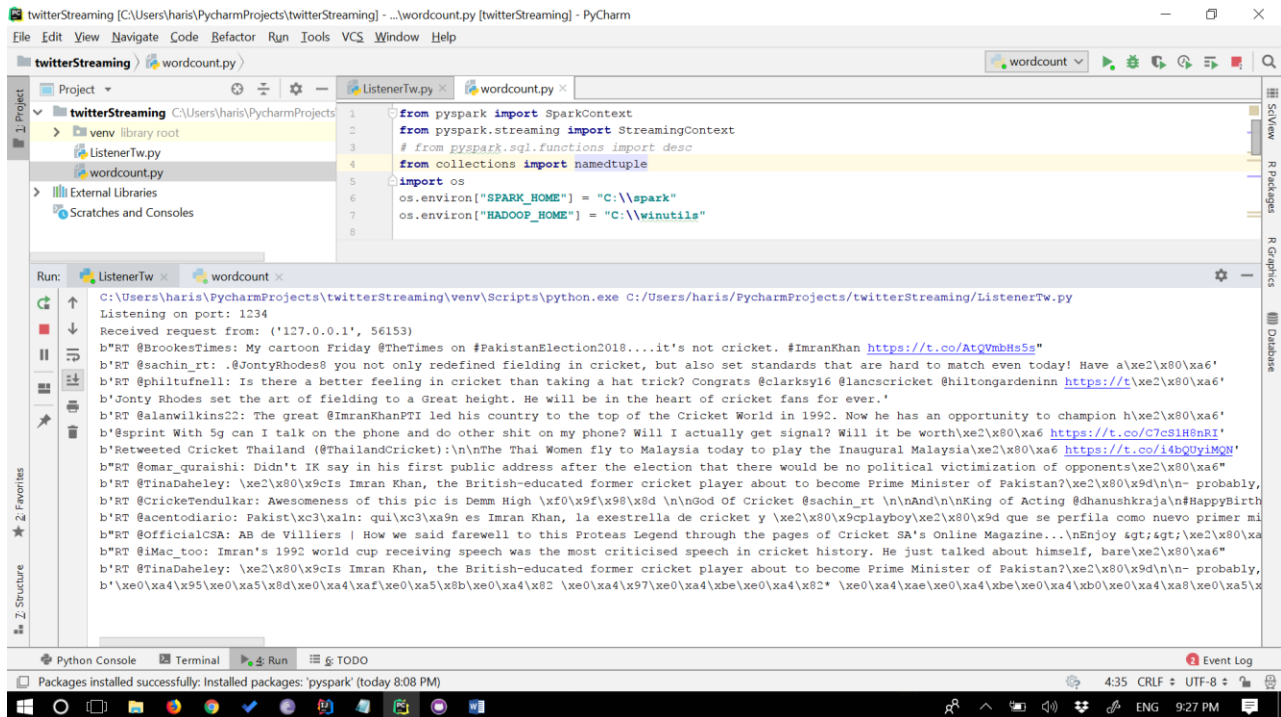


The screenshot displays the PyCharm IDE interface. The main editor window shows a Python script named `ListenerTw.py` within a project called `twitterStreaming`. The script imports `tweepy`, `socket`, `json`, and `time`. It defines consumer and access keys/secrets. An `OAuthHandler` is created and configured. A `TweetsListener` class inherits from `StreamListener` and implements `__init__` and `on_data` methods. The `on_data` method processes incoming JSON data, prints the text content, and sends it to a client socket. The script also includes a `wordcount.py` file in the project structure.

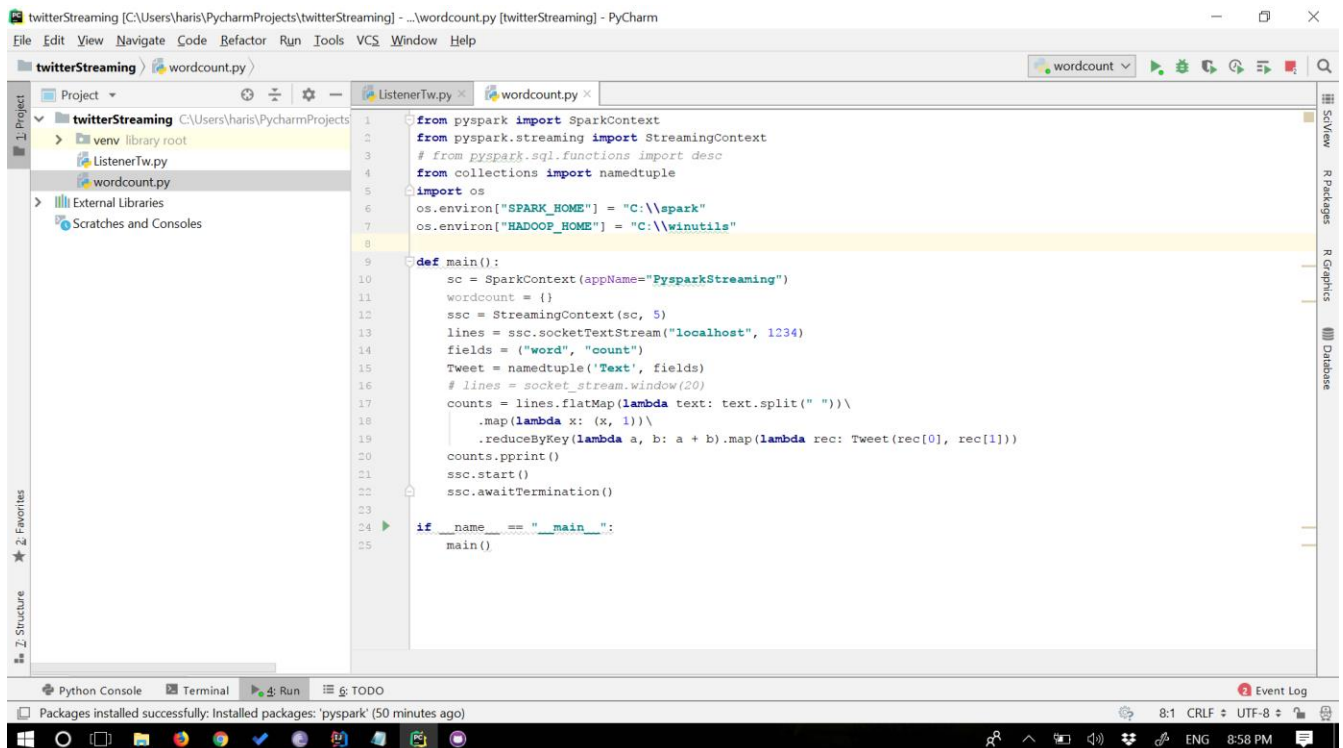
```
1 import tweepy
2 from tweepy import OAuthHandler
3 from tweepy import Stream
4 from tweepy.streaming import StreamListener
5 import socket
6 import json
7 import time
8
9 consumer_key = 'w557kolJzAyXl01lkfy3bx9qe'
10 consumer_secret = 'YEB8XaQGeaj90vyhkZC9SR645WKr6tpeUYz5w6TiCdRLaJa0r12'
11 access_token = '1367004096-Q1kbXQgRCfgnhUdGWpFKEiv6OcKspDEv1WLj1oT'
12 access_secret = '6Yr2TFWE3KmaexRh84WkU4ZsLfxYsCKEs8oNgZM2vDUy5'
13
14
15 auth = OAuthHandler(consumer_key, consumer_secret)
16 auth.set_access_token(access_token, access_secret)
17
18 class TweetsListener(StreamListener):
19
20     def __init__(self, csocket):
21         self.client_socket = csocket
22
23     def on_data(self, data):
24         try:
25             msg = json.loads(data)
26             print(msg['text'].encode('utf-8'))
27             self.client_socket.send(msg['text'].encode('utf-8'))
28             return True
29         except BaseException as e:
30             print("Error on_data: %s" % str(e))
```

## Lab Assignment 4

## Output for tweets collection:



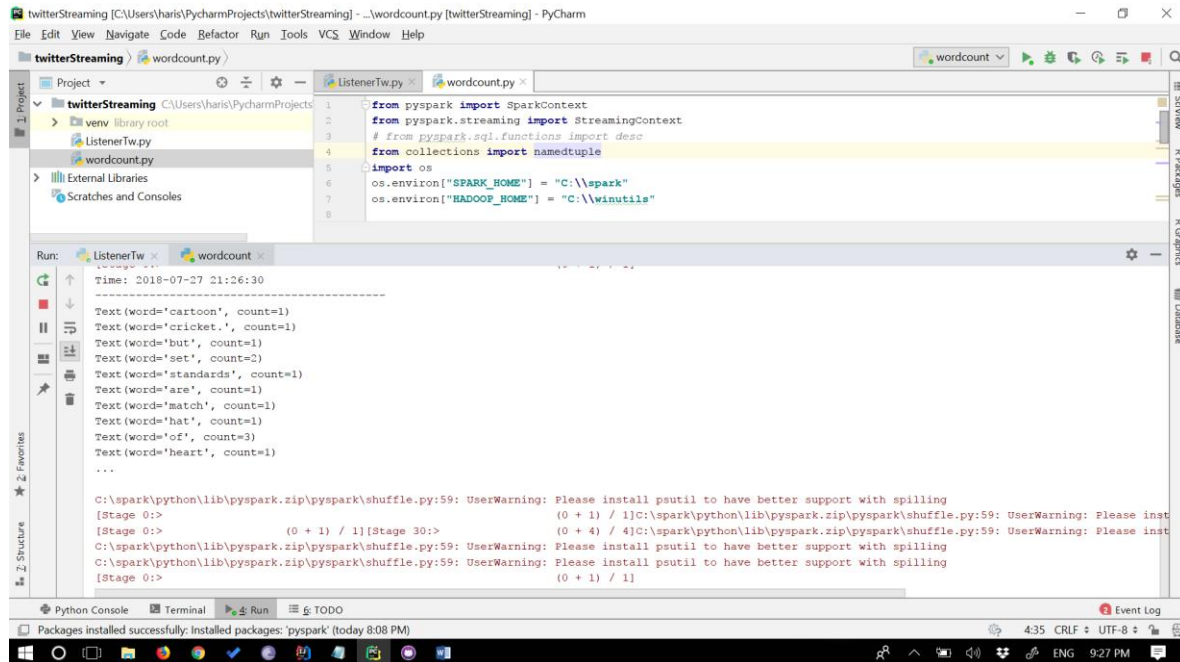
## 2. Streaming and performing word count on twitter data:





## Lab Assignment 4

### output for Wordcount of Twitter data:



The screenshot shows a PyCharm IDE window titled 'twitterStreaming'. The 'wordcount.py' file is open, displaying the following code:

```
1 from pyspark import SparkContext
2 from pyspark.streaming import StreamingContext
3 # from pyspark.sql.functions import desc
4 from collections import namedtuple
5 import os
6 os.environ["SPARK_HOME"] = "C:\\spark"
7 os.environ["HADOOP_HOME"] = "C:\\winutils"
8
```

The 'Run' console at the bottom shows the output of the script, which is a word count for various words:

```
Time: 2018-07-27 21:26:30
-----
Text(word='cartoon', count=1)
Text(word='cricket.', count=1)
Text(word='but', count=1)
Text(word='set', count=2)
Text(word='standards', count=1)
Text(word='are', count=1)
Text(word='match', count=1)
Text(word='hat', count=1)
Text(word='of', count=3)
Text(word='heart', count=1)
...
C:\spark\python\lib\pyspark.zip\pyspark\shuffle.py:59: UserWarning: Please install psutil to have better support with spilling
[Stage 0:]> (0 + 1) / 1 [Stage 30:]> (0 + 1) / 1 C:\spark\python\lib\pyspark.zip\pyspark\shuffle.py:59: UserWarning: Please inst
(0 + 4) / 4 C:\spark\python\lib\pyspark.zip\pyspark\shuffle.py:59: UserWarning: Please inst
C:\spark\python\lib\pyspark.zip\pyspark\shuffle.py:59: UserWarning: Please install psutil to have better support with spilling
[Stage 0:]> (0 + 1) / 1 [Stage 30:]> (0 + 1) / 1 C:\spark\python\lib\pyspark.zip\pyspark\shuffle.py:59: UserWarning: Please inst
(0 + 4) / 4 C:\spark\python\lib\pyspark.zip\pyspark\shuffle.py:59: UserWarning: Please inst
```

### 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/>