**Challenge 2: Feature Extraction Using Different Classifying Mechanisms**

**Summary:**

The basic notion of this project is to classify data using certain supervised techniques. Here we used three classification mechanisms in order to evaluate the accuracy for our data. We also designed certain ROC curves to depict classifiers true positive and true negative values.

**Architecture:**

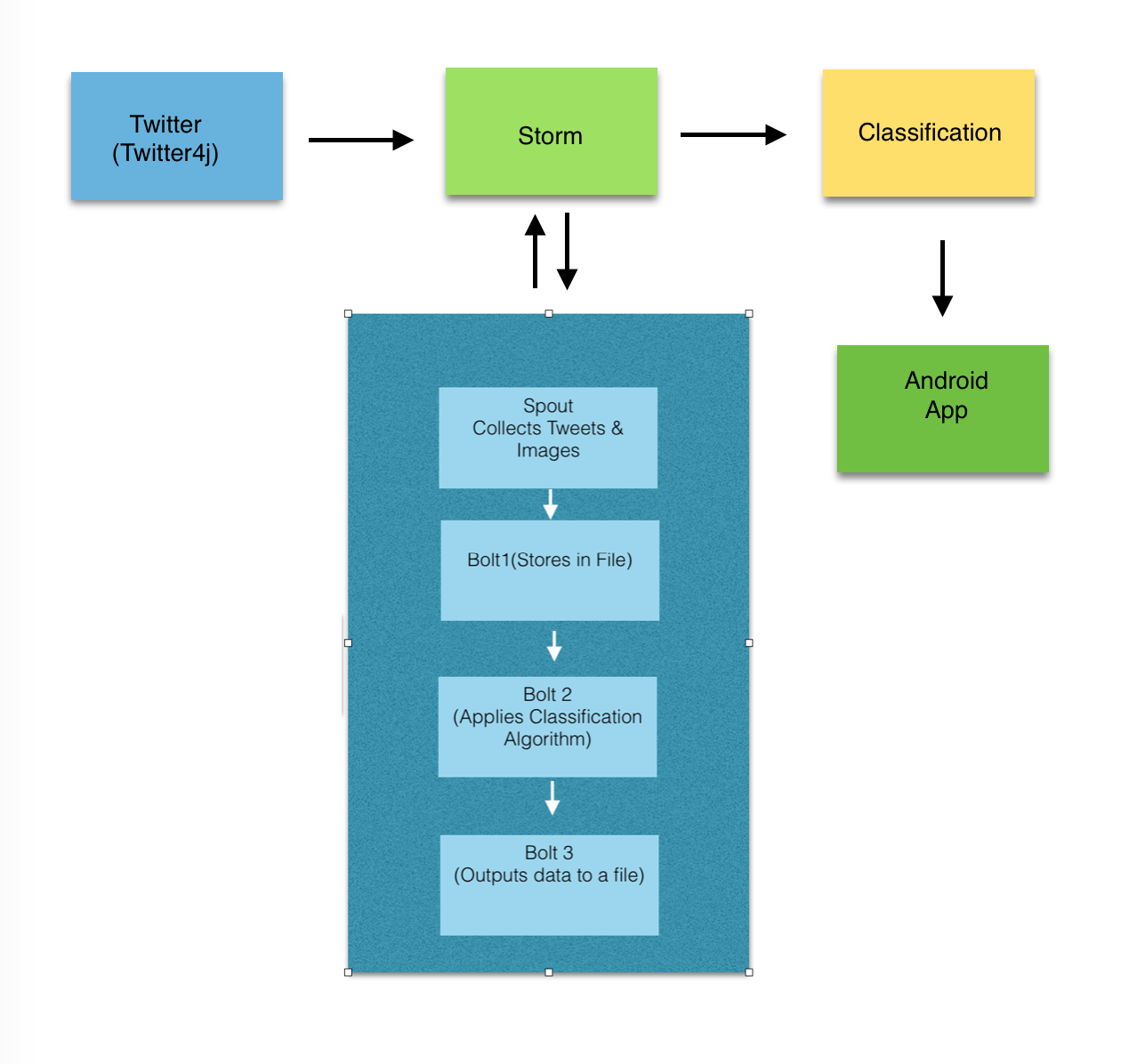


Figure Architecture

On a higher level we collect the tweets from the storm and classify those tweets based on the Naïve Bayes algorithm and then we shall store the resultant confusion matrix in a text file and store it on the server. Now using “SSH” techniques we shall extract these file in the android mobile application.

Now let us go in depth regarding the STORM execution. In the spout we shall collect the tweets and store them in a tuple. Now this tuple will be send to a bolt. Now in the first bolt we shall store the data in the tuple in a file. Then in the second we take this file as a test file and apply the classification model on to this test file and prepare a confusion matrix. Now this confusion matrix will be stored in a file using the third bolt. This is how we make use of the real time analytics in classifying the tweets. In bolt 2 we provide the trained data which we have collected and classify the test data. Due to some issues we used more of a model which we already prepared using a console application to test the data.

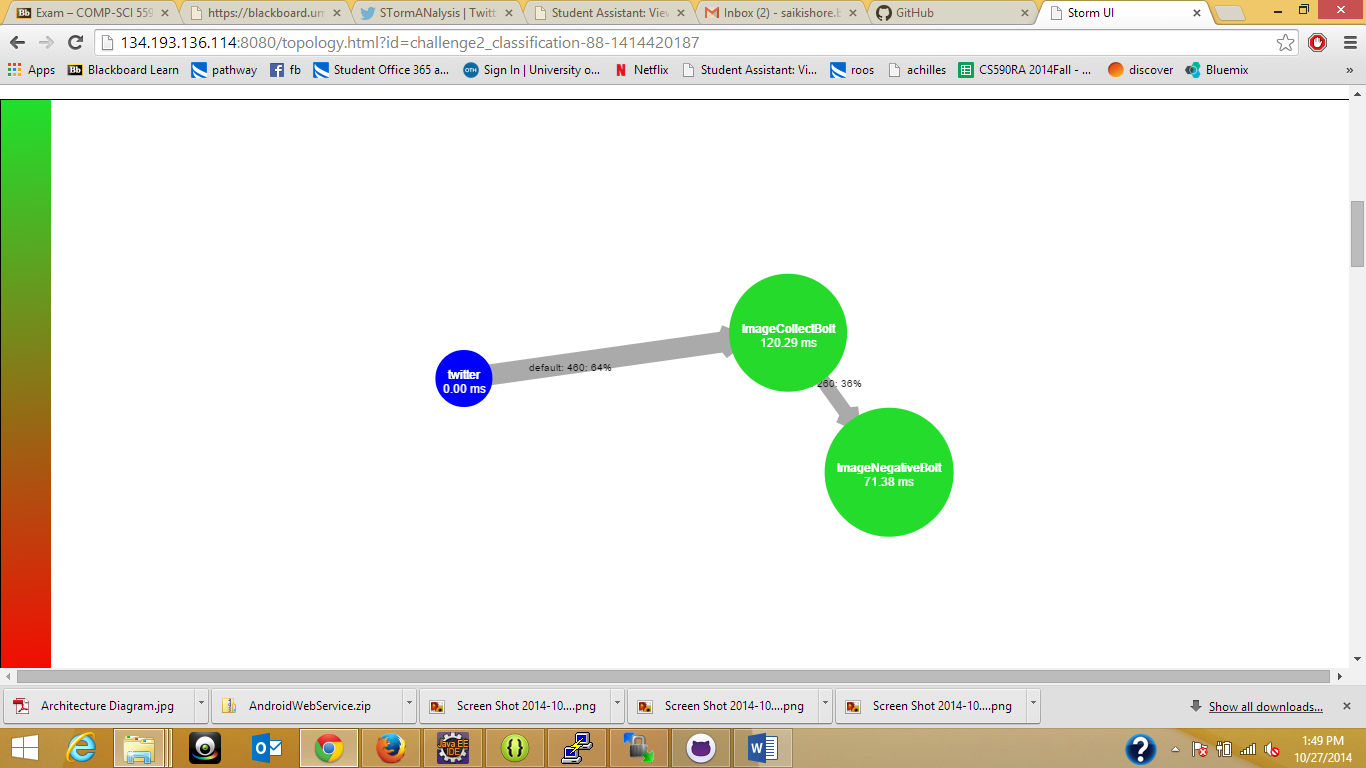


Figure Storm Topology

**Mobile Interface:**

We have developed a Native mobile application for Android users. The mobile interface has a button which on click displays the confusion matrix for a model. The following are the snapshots of the control flow:

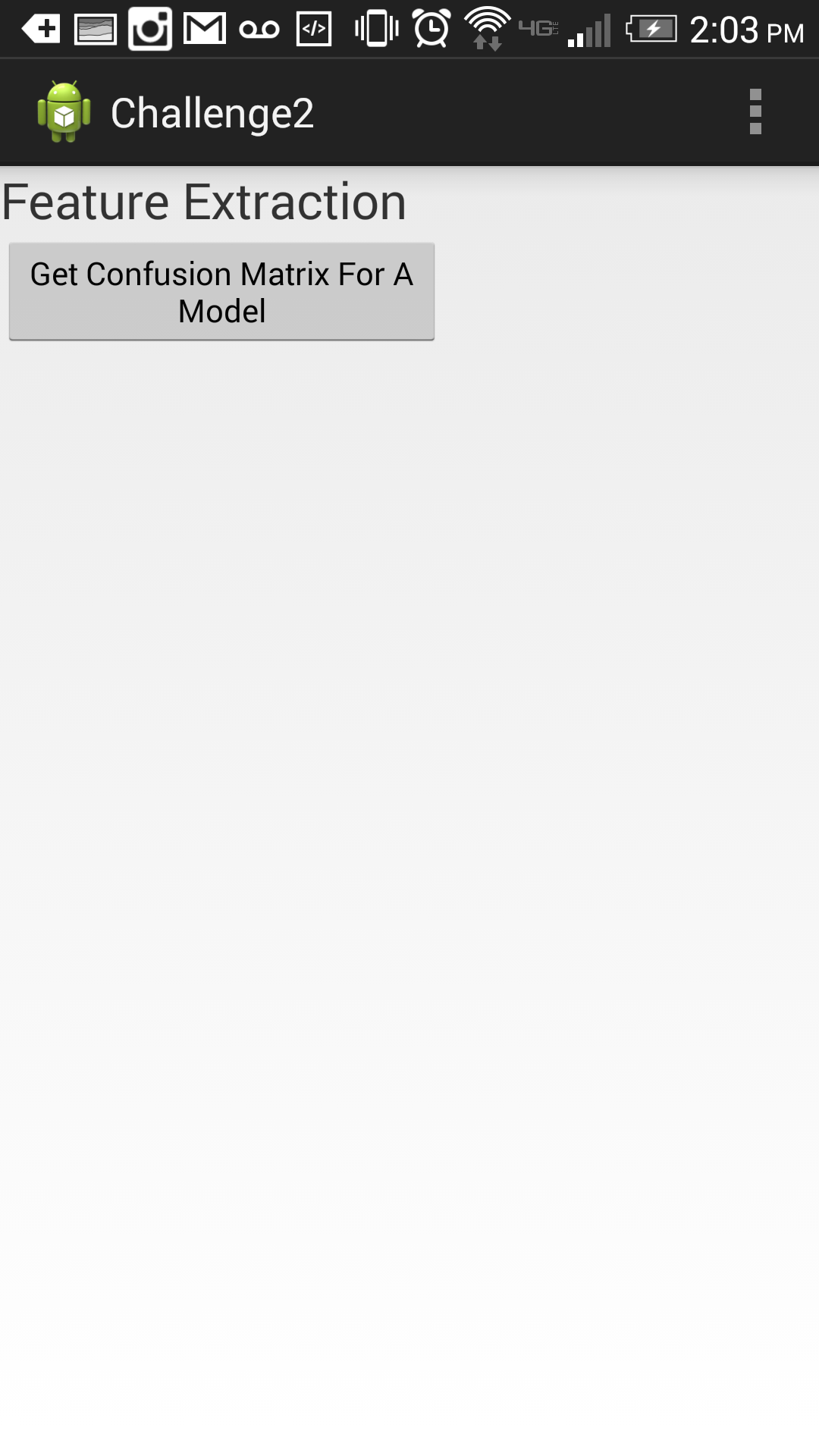


Figure Initial WebPage

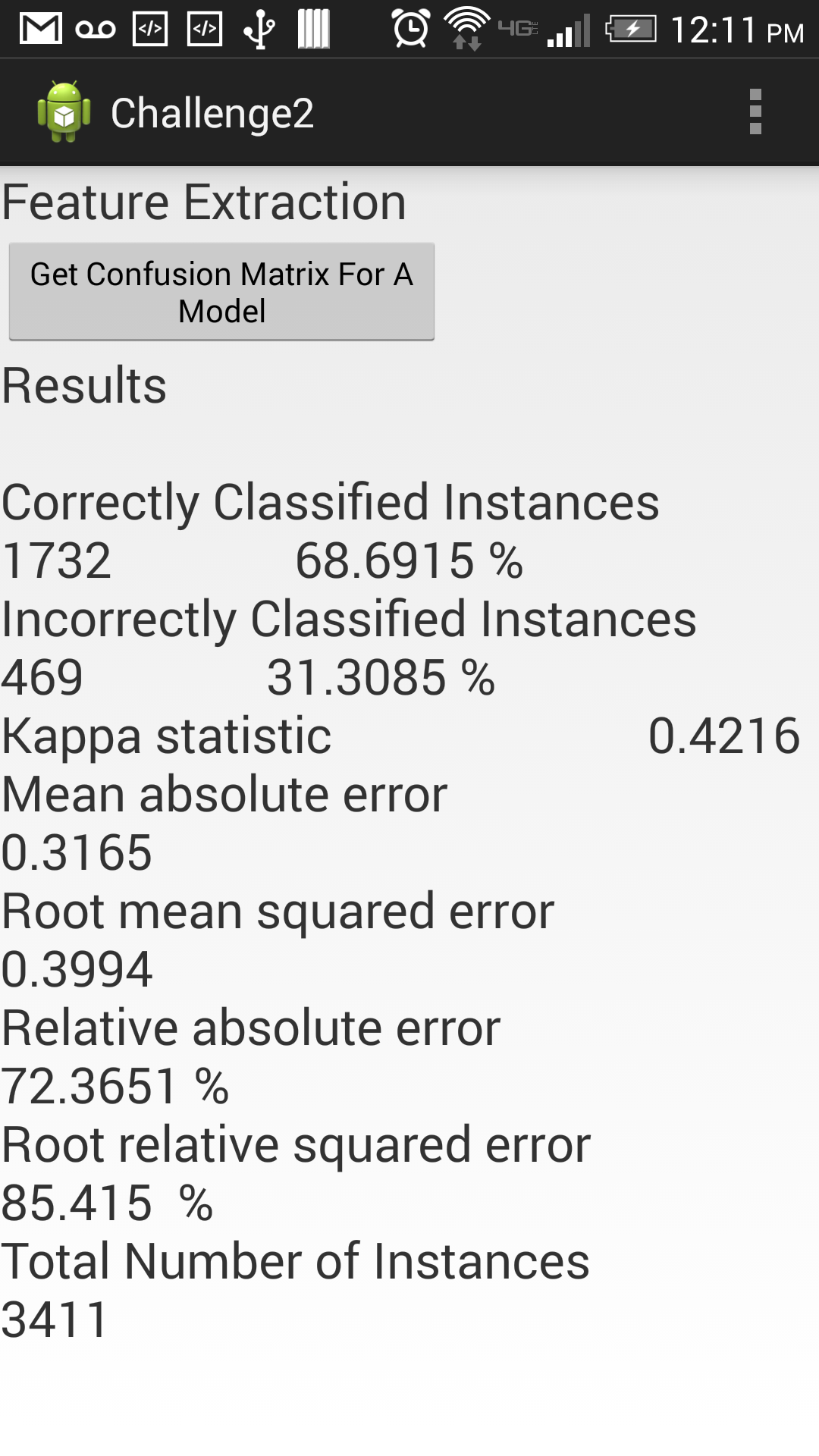


Figure Confusion matrix

**Data Collection:**

We have collected both text and image data from the twitter. However we did not perform any analytics on the images and it is merely used for data collection. Now we need to classify the data. For this we planned to use MashAPE APIs so as to train the data as per the tweet message rather featuring the tweets with vague information. However we classified the data on our own without usage of APIs. We wrote our own program to classify the tweets into five categories. The five categories are sad, happy, angry, excited and worried. Now we have sent this data to different algorithms mainly Decision Tree , Naïve Bayes and SVM. When we classified the data using decision tree with some portion of our data say 10% then we got the following accuracy with certain parameters:

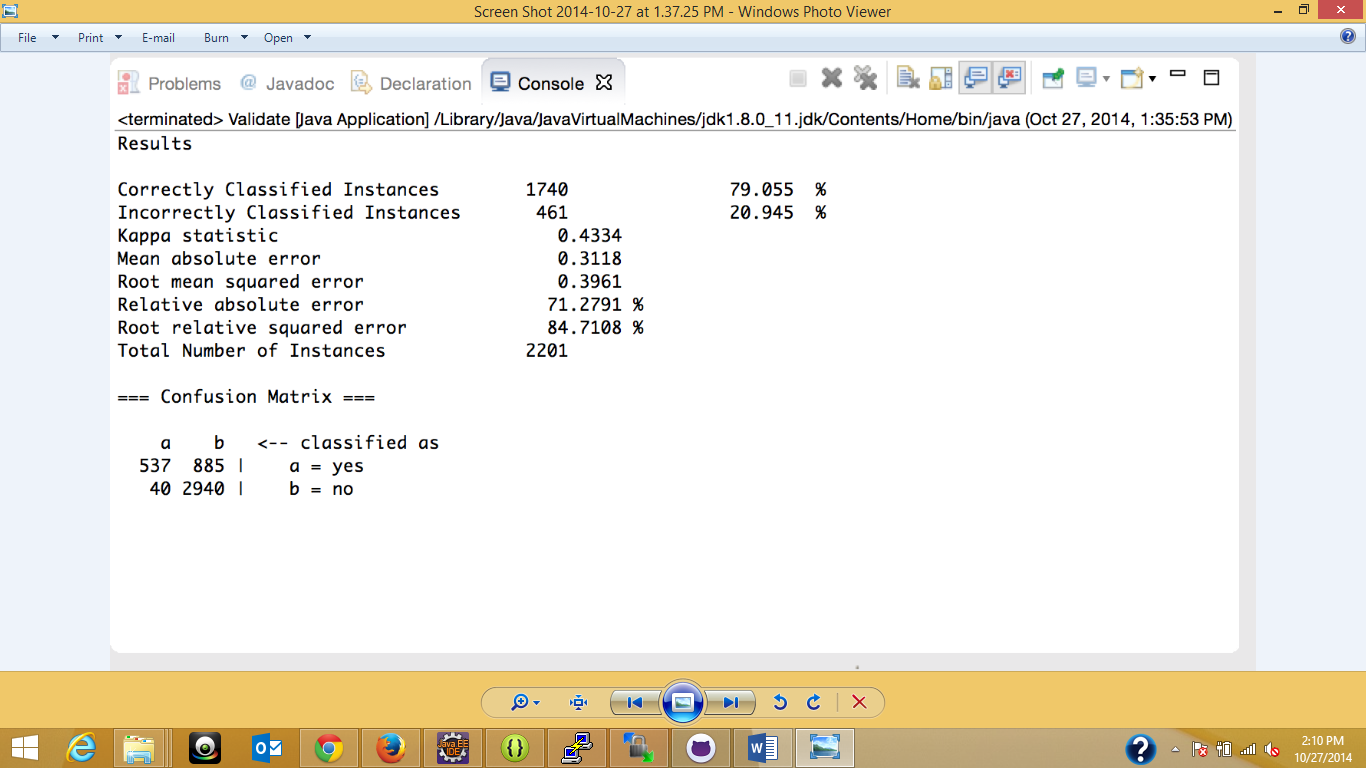


Figure naïve Bayes

Now we have applied similar techniques for other two algorithms with similar content of data.

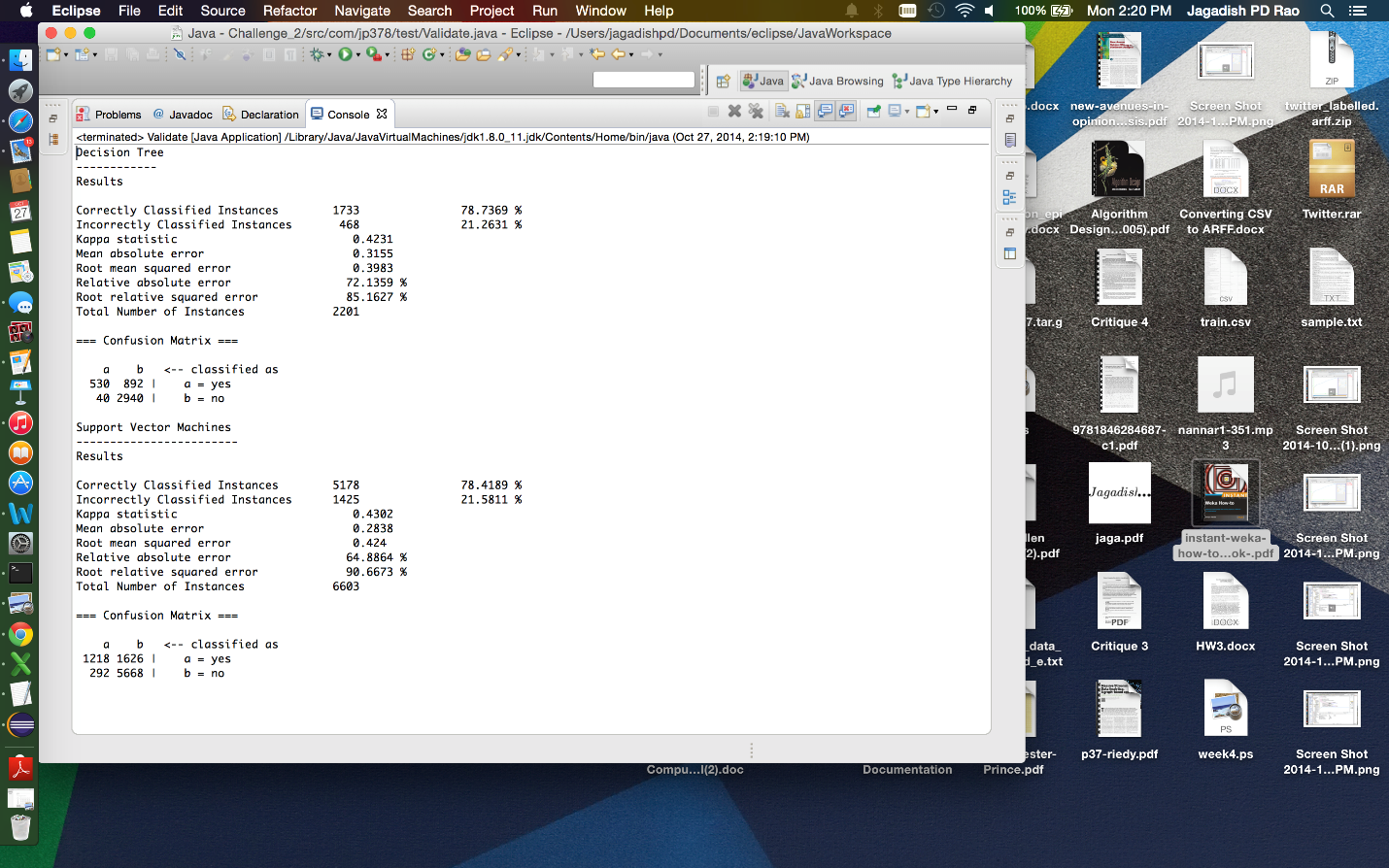


Figure SVM and Decision tree

Once we have applied the algorithms we have calculated the ROC curves and the ROC curves for three algorithms are shown below:

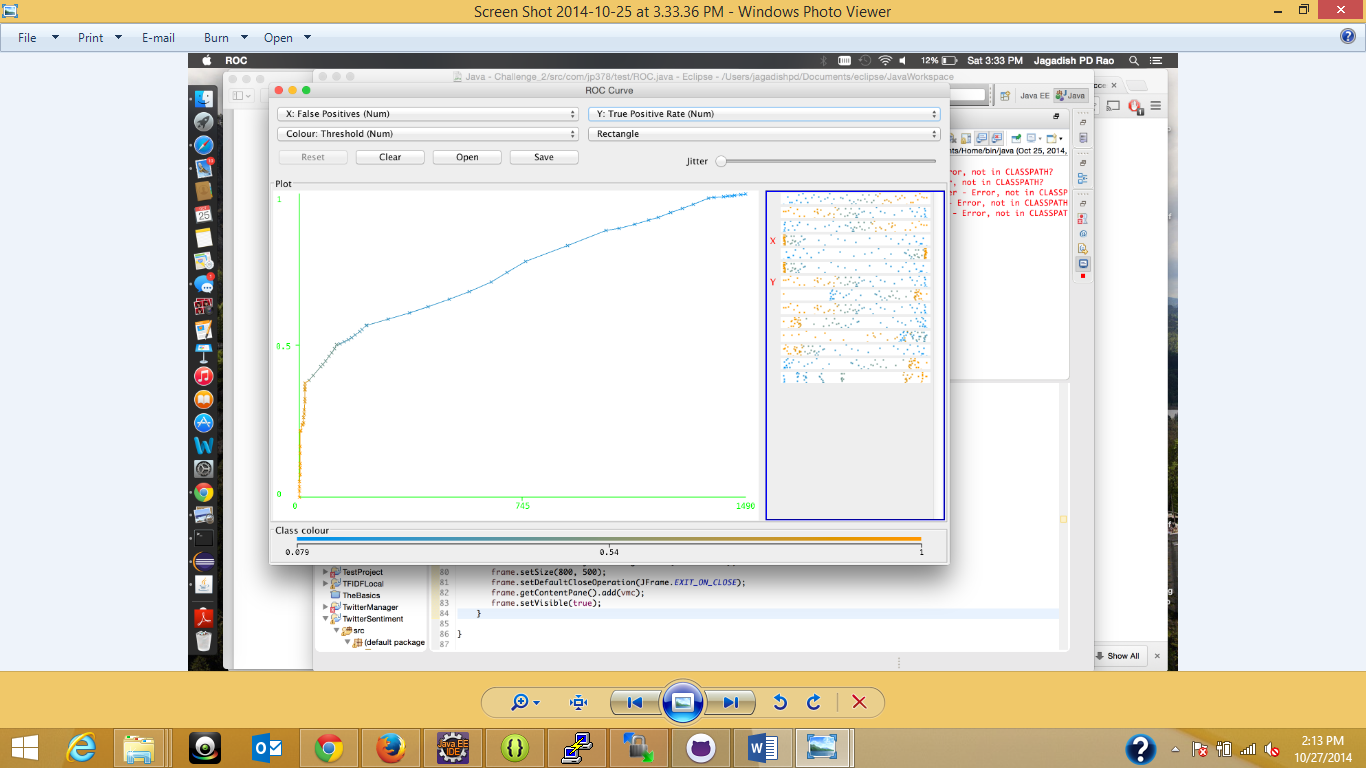


Figure ROC curve for Naive Bayes

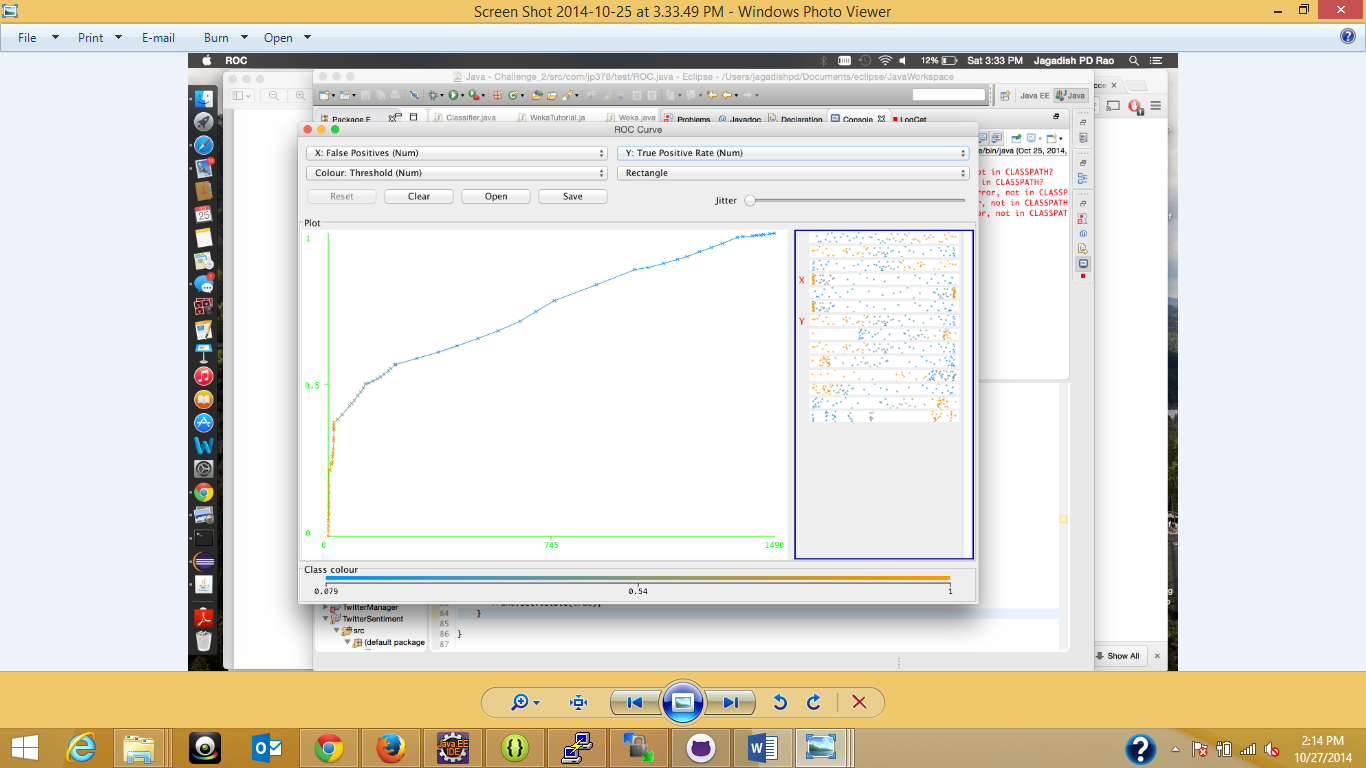


Figure ROC curve for SVM

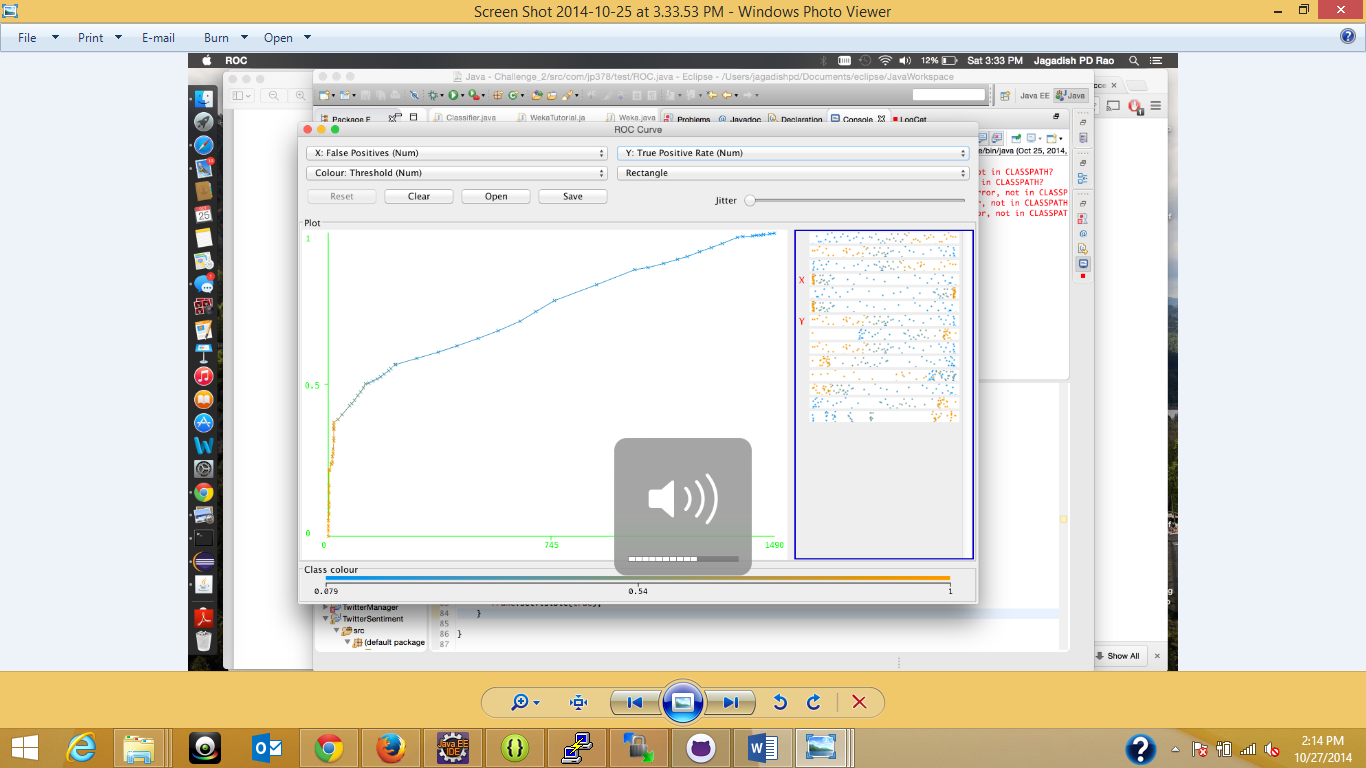


Figure ROC for decision tree

But since our data is huge and we cannot make parameters which are needed for the Decision tree we have fixed our approach to Naïve Bayes as it also has significant accuracy may not be desired but still admirable.

Now one of the bolts will initially filter the data as required to be converted for arff.

All the data and result files have been uploaded in the github repository:

* <https://github.com/SaiKishoreBandaru/Challenge2_RA>

**Results:**

The result file has been uploaded in the github in the above mentioned link.