I have worked with the following people for understanding the assignment better:
Tarun Gulati
Nasheed

Note: To implement the SVM in Fourth Classifier, I have used the PyML library. I have included the files from this library in the codefiles.zip.

Question 1:

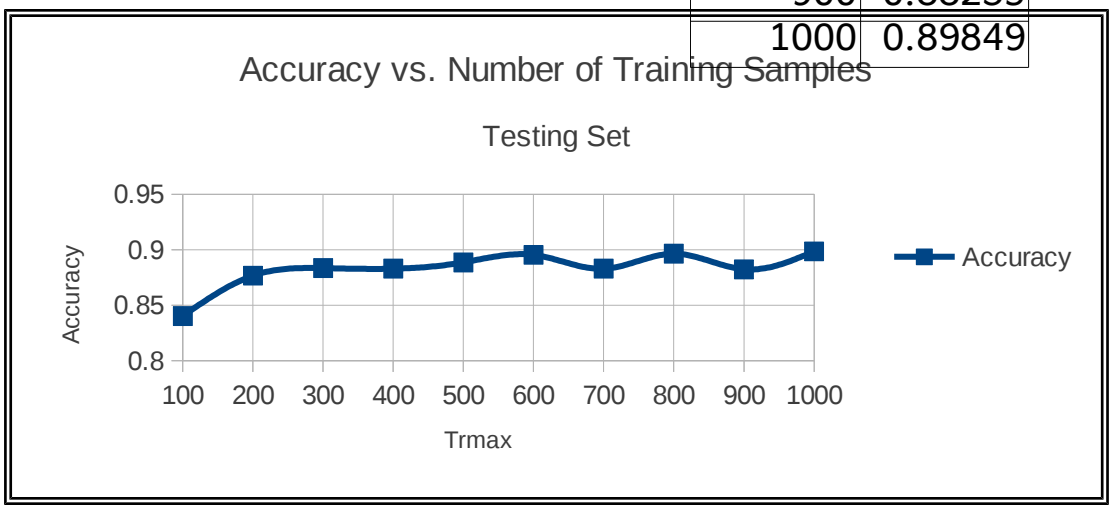
Part A:

Implemented the Naive Bayes Classifier using both Minimum Likelihood Estimation and Maximum A Posteriori Estimation (MLE has been commented for the calculation of the probability of the labels as uniform prior was assumed. The code works when a non uniform prior is assumed as well). Currently alpha and beta values of 2 each are used. The code is tested for other beta values as well.

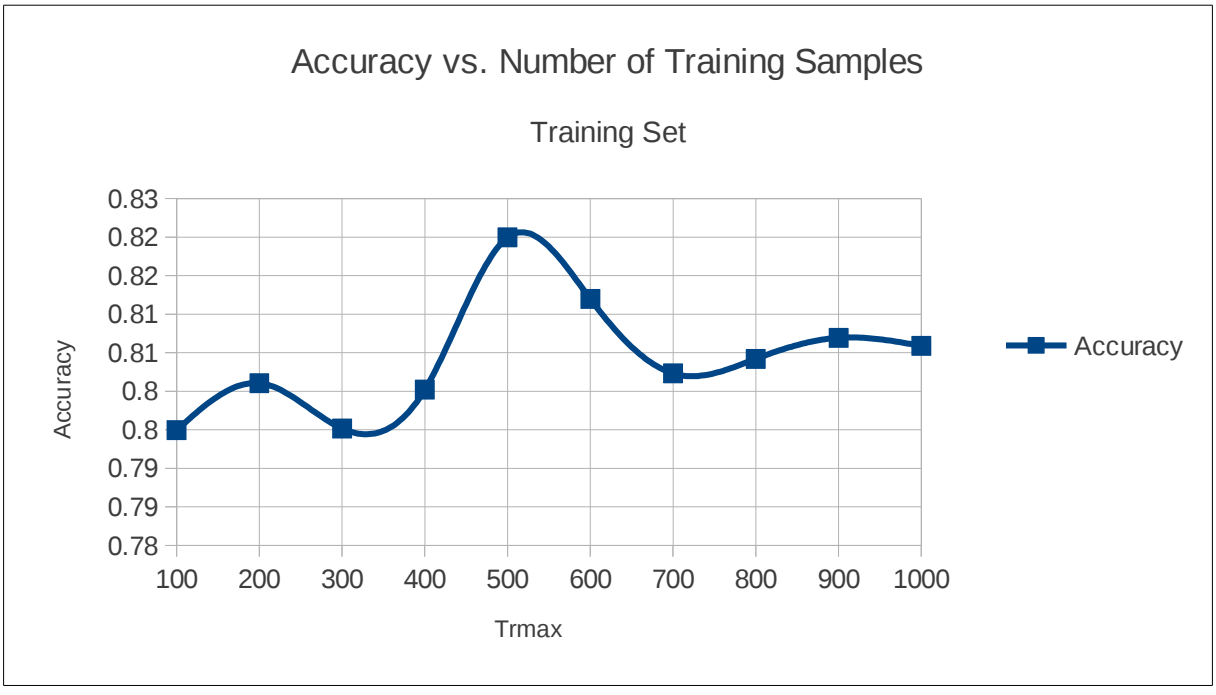
Part B:

Positive topics: Earn
Negative Topics: acq, crude, gold.
Following are the values obtained:

Trmax	Accuracy
100	0.84019
200	0.87663
300	0.88339
400	0.88287
500	0.8886
600	0.89537
700	0.88287
800	0.89641
900	0.88235
1000	0.89849



Trmax	Accuracy
100	0.79496
200	0.80105
300	0.79517
400	0.80021
500	0.81996
600	0.81198
700	0.80231
800	0.8042
900	0.80693
1000	0.80588



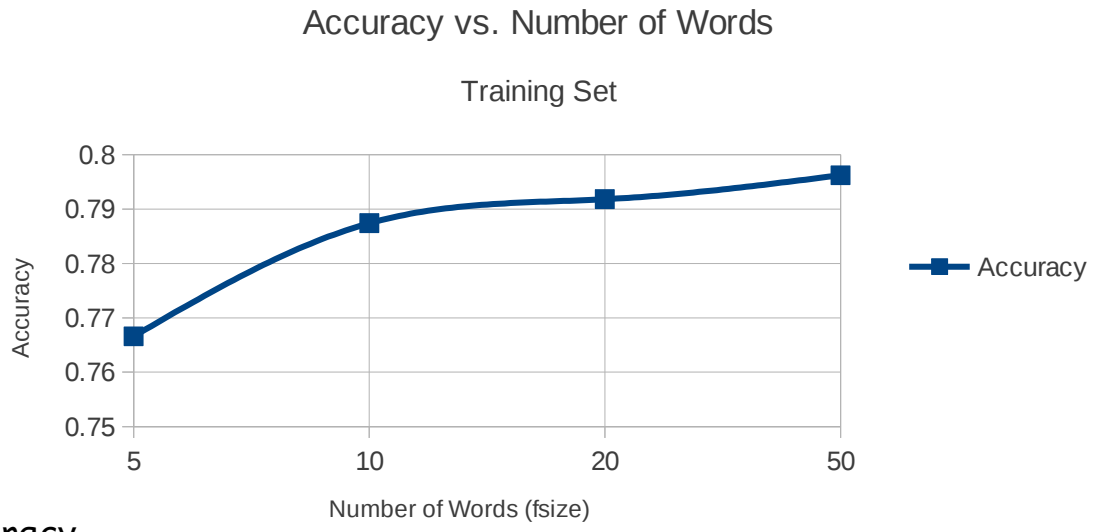
Observation: As the number of training samples used increases, the accuracy does improve. But this improvement is not consistent and it looks to be decreasing after a

certain training set size. This is because the randomness in choosing the files for the examples increases as the size of the training set increases. Also, another possible explanation for this is overfitting.

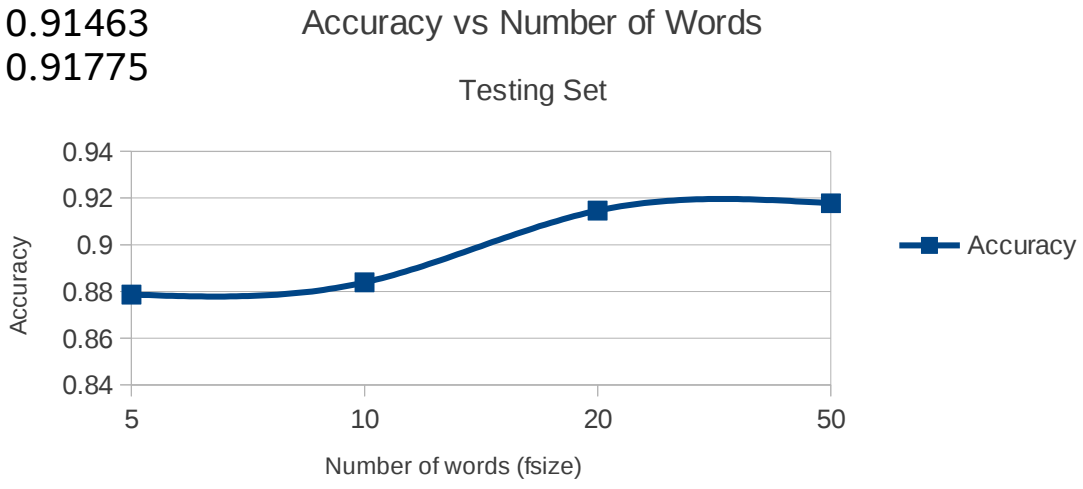
Part C:

Following were the values obtained:

Fsize	Accuracy
5	0.7666
10	0.7874
20	0.79181
50	0.79622



Fsize	Accuracy
5	0.87871
10	0.88392
20	0.91463
50	0.91775



Observation: As with the training set, as the number of feature words being used increases, the accuracy again increases. This happens as we are giving more information to our classifier during training which helps it learn better. Overfitting is one cause of a drop in the accuracy at some points.

Question 2

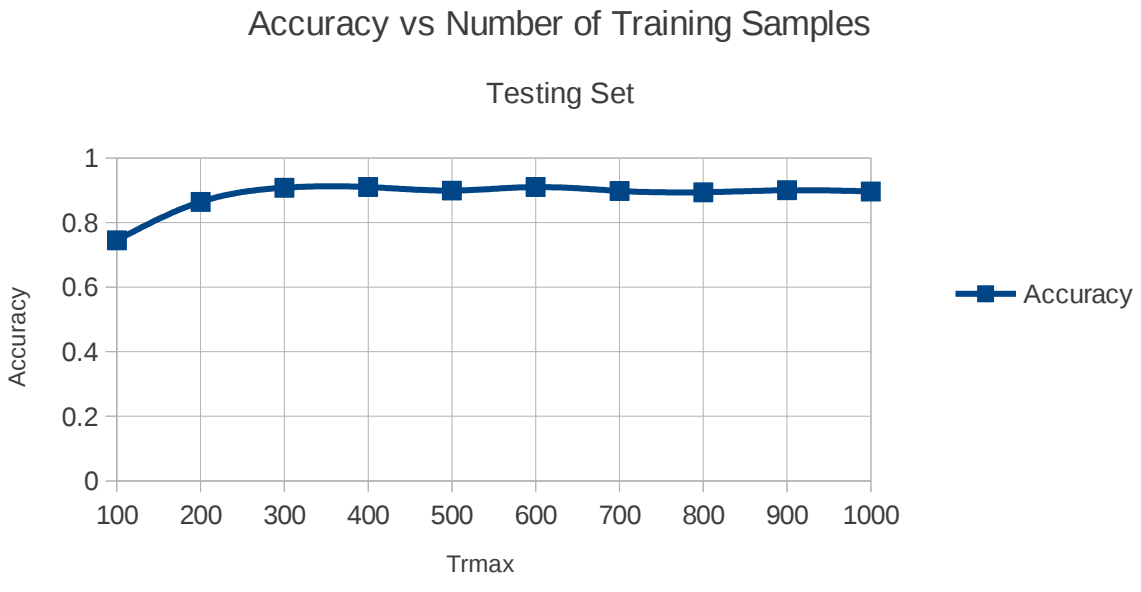
Part A:

The Decision tree classifier was implemented in the code. To pick the best attribute to classify on, I have used the feature which get the maximum value of Information Gain, as described in R&N.

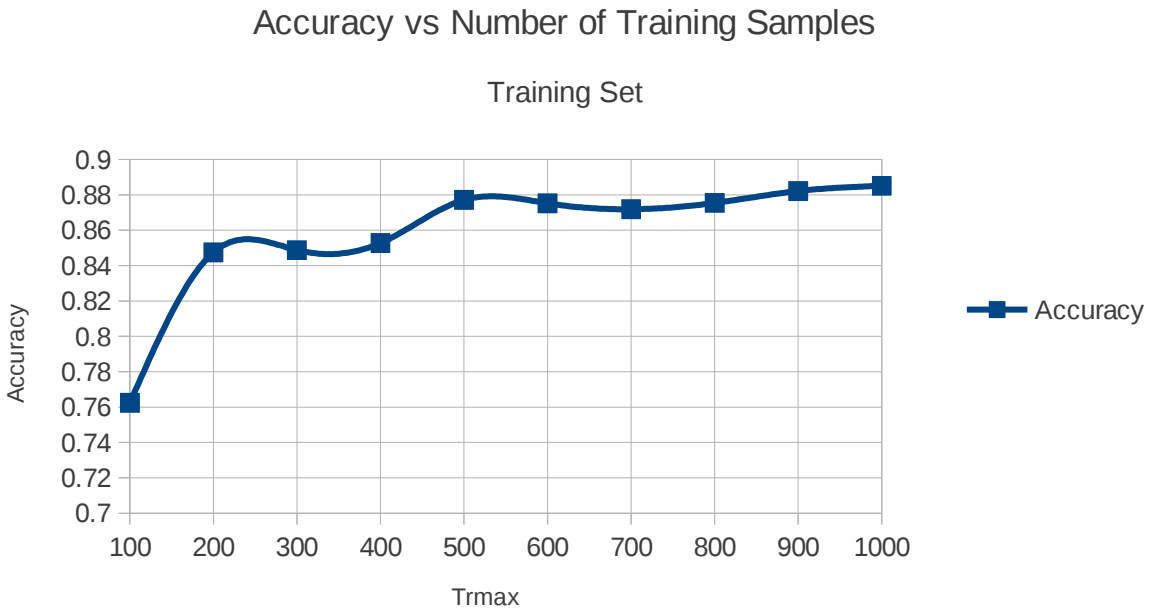
Part B:

Following were the values obtained:

Trmax	Accuracy
100	0.74493
200	0.86413
300	0.90786
400	0.90994
500	0.89901
600	0.90994
700	0.89797
800	0.89381
900	0.90005
1000	0.89641



Trmax	Accuracy
100	0.7624
200	0.84748
300	0.84874
400	0.85273
500	0.8771
600	0.87521
700	0.87185
800	0.87542
900	0.88214
1000	0.88508

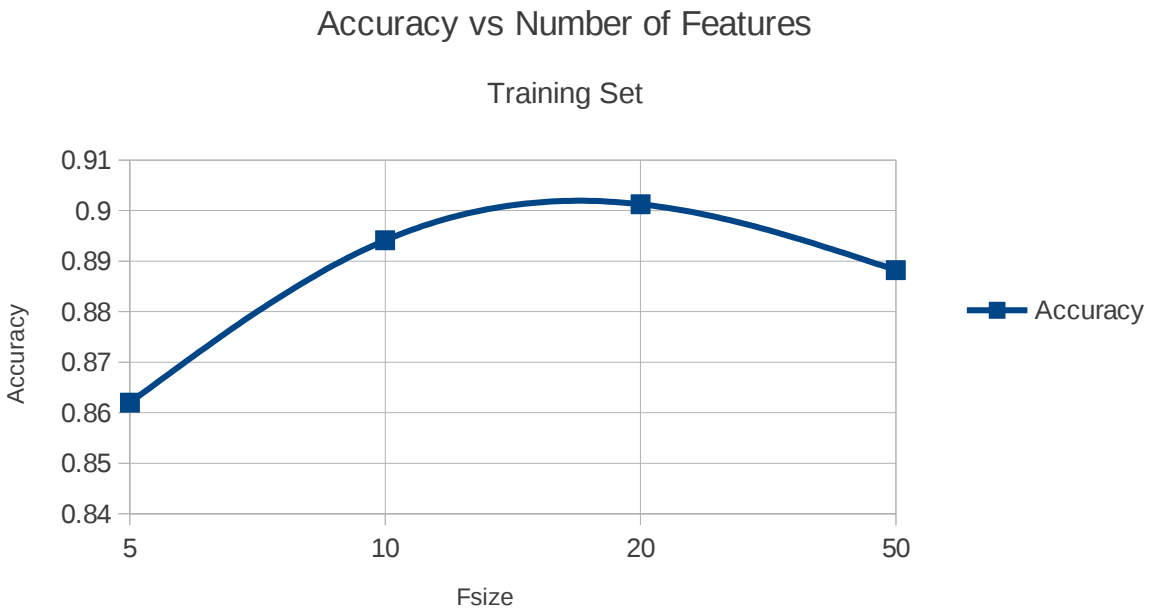


Observation: As the number of training set samples increases, we see a general trend that the accuracy increases.

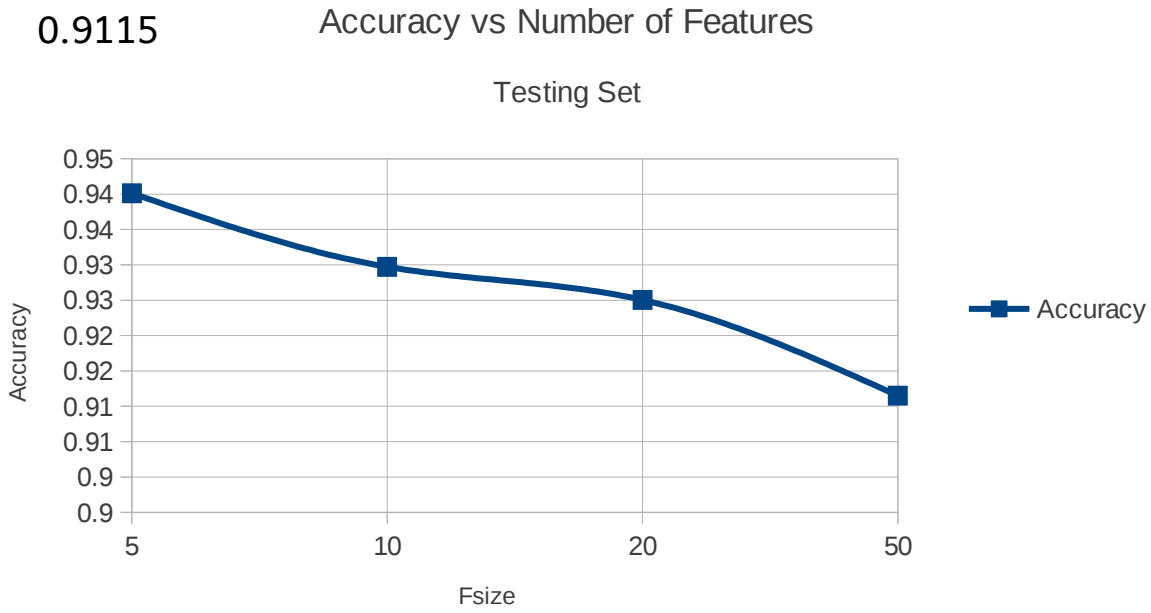
Part C:

Following were the values obtained:

Fsize	Accuracy
5	0.86198
10	0.89412
20	0.90126
50	0.88824



Fsize	Accuracy
5	0.94014
10	0.92972
20	0.92504
50	0.9115

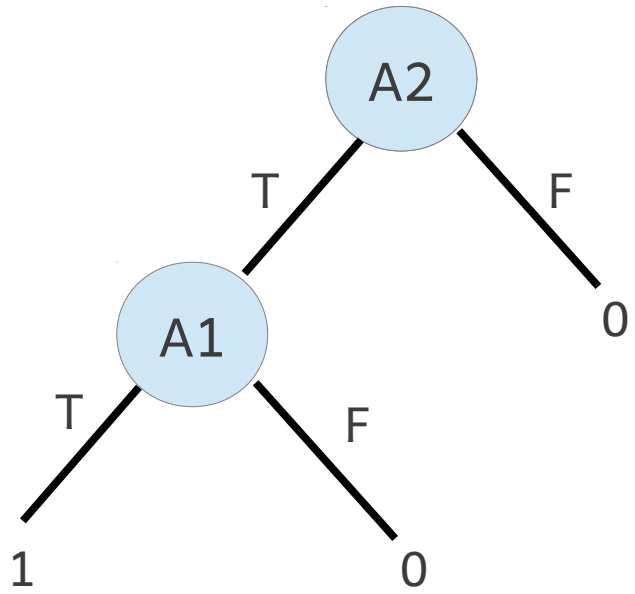


Observation: As the number of features used to train the classifier increases, we see that the accuracy of the classifier increases as well.

Part D:

When the decision tree learning algorithm is applied to the training set D, we will get a tree (say X). Now, as we increase the size of the training set samples, the tree returned will be better informed, and at some stage, be consistent with the training set D.
Now for a given training set D, there is a possibility that there could be more than one tree that are consistent with it. Hence, increasing the size of the sample set for training does not guarantee that the tree returned (which will consistent with D at some stage) will be the same as the original tree T. It could may well be possible that T != X but X is consistent with D. (X may be a minimzed tree).

Part E:
The following was the tree that was obtained:



To get this tree, the following was done:

1. Out of the possible attributes, we pick the one that gives us the minimum error.
2. For each of its values, we construct branches. (In this case one for True and one for False)
3. We repeat from 1 until we reach the leaf nodes.

Question 3:

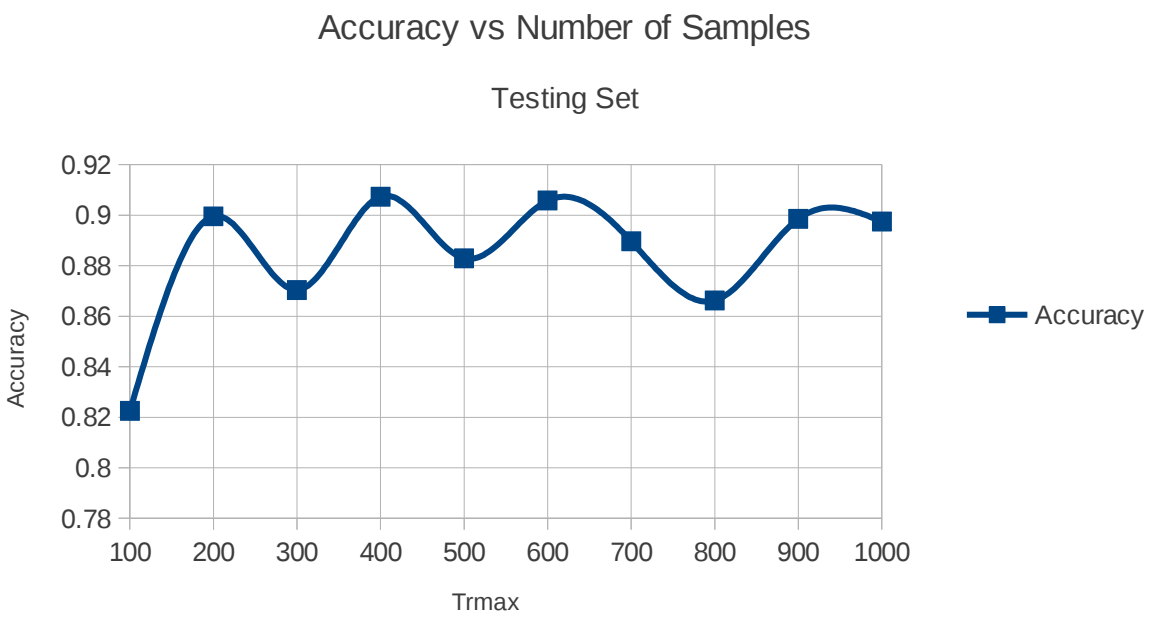
Part A:

I have implemented two new classifiers. The first is in ThirdClassfier() and can be used using the -c d option. This is a Dtree with the modification that all the stop words are removed from the training and the testing examples. I did this as for identification of a topic, stop words do not contribute much to the information on the topic. This not only reduces the size of the tree, but also reduces the time required during training and testing.

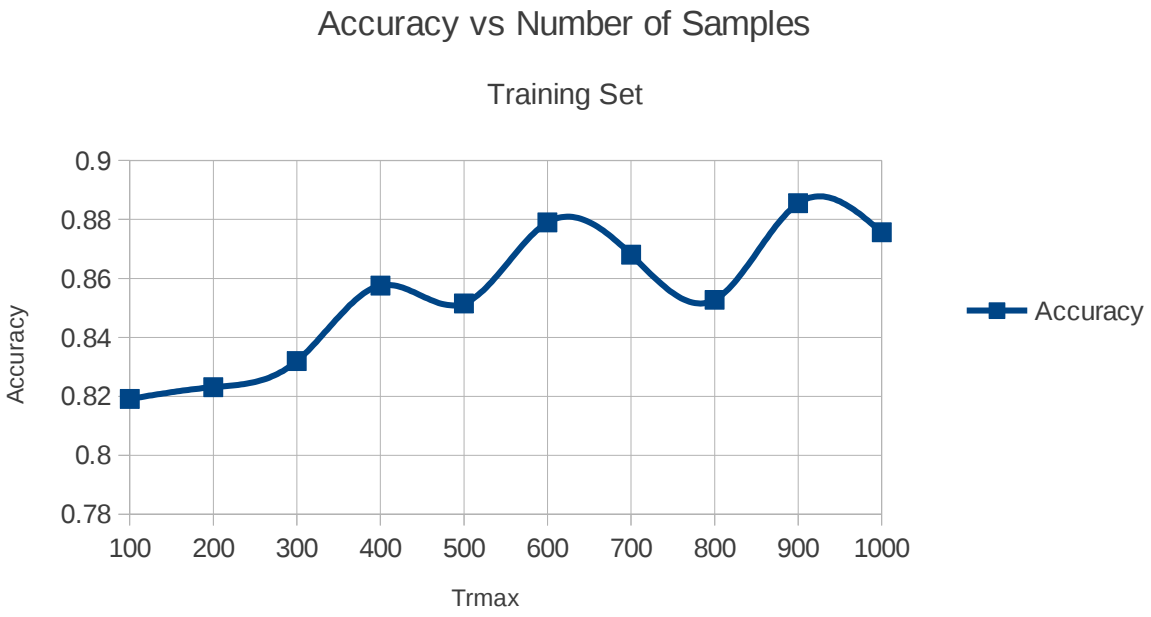
The second new classifier is in FourthClassifier() and can be used using the -c f option. This is a SVM. To implement this I have used the PyML library. I have had to change the batch_test() function in classify.py to handle this. For testing, I am first appending all the test cases to a file along with their correct labels (this is the correct format to pass data to an SVM according to PyML). Hence, the testing only happens after all the feature sets are written to the file. Also for this classifier, I am not using the given calculation of accuracy as PyML automatically calculates the accuracy.

Part B:

Following are the values obtained for the Third Classifier:

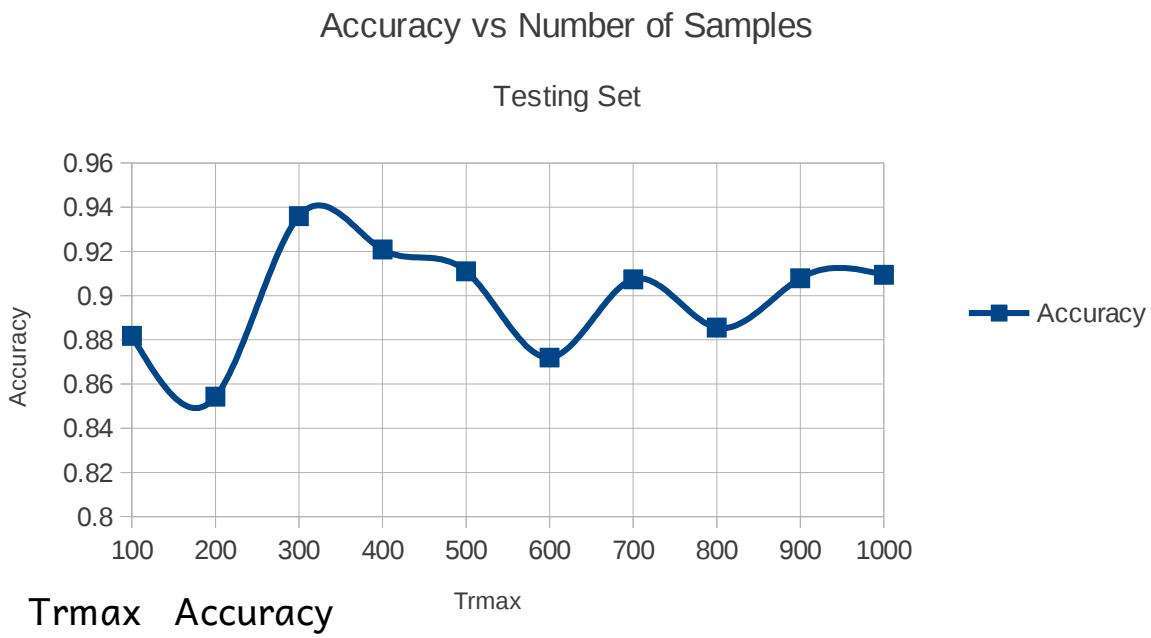


Trmax	Accuracy
100	0.81912
200	0.82311
300	0.83193
400	0.85756
500	0.85147
600	0.87899
700	0.86807
800	0.85273
900	0.8855
1000	0.87563

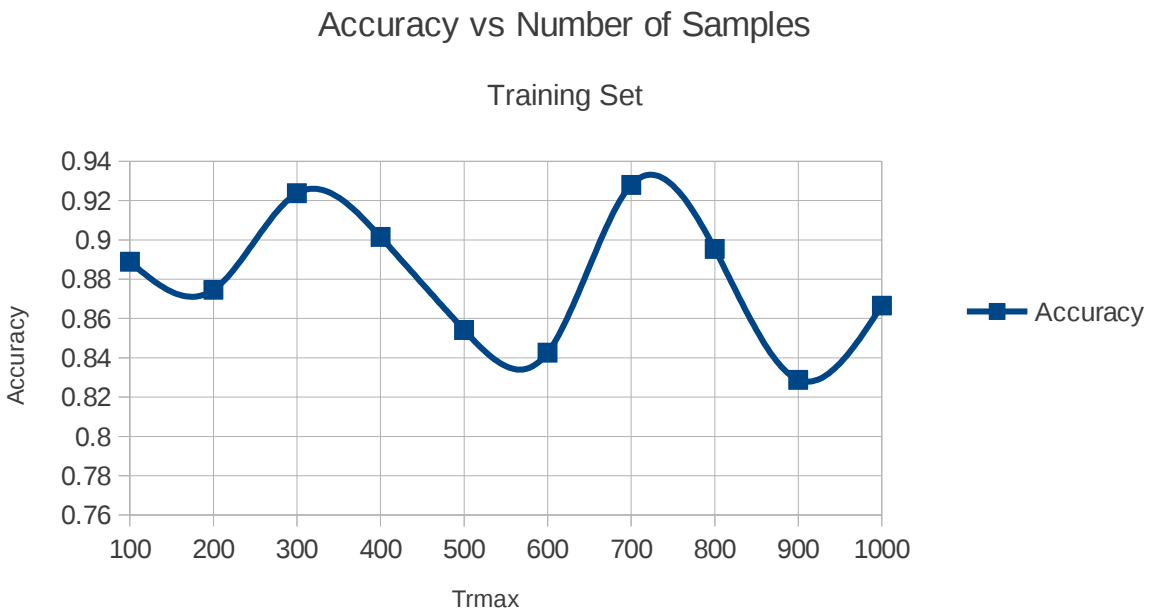


Following are the values obtained for the Fourth Classifier:

Trmax	Accuracy
100	0.88183
200	0.85424
300	0.93597
400	0.92088
500	0.91098
600	0.87194
700	0.90734
800	0.88548
900	0.90786
1000	0.90942



Trmax	Accuracy
100	0.88887
200	0.87458
300	0.92374
400	0.90147
500	0.8542
600	0.84265
700	0.92794
800	0.89538
900	0.82878
1000	0.8666

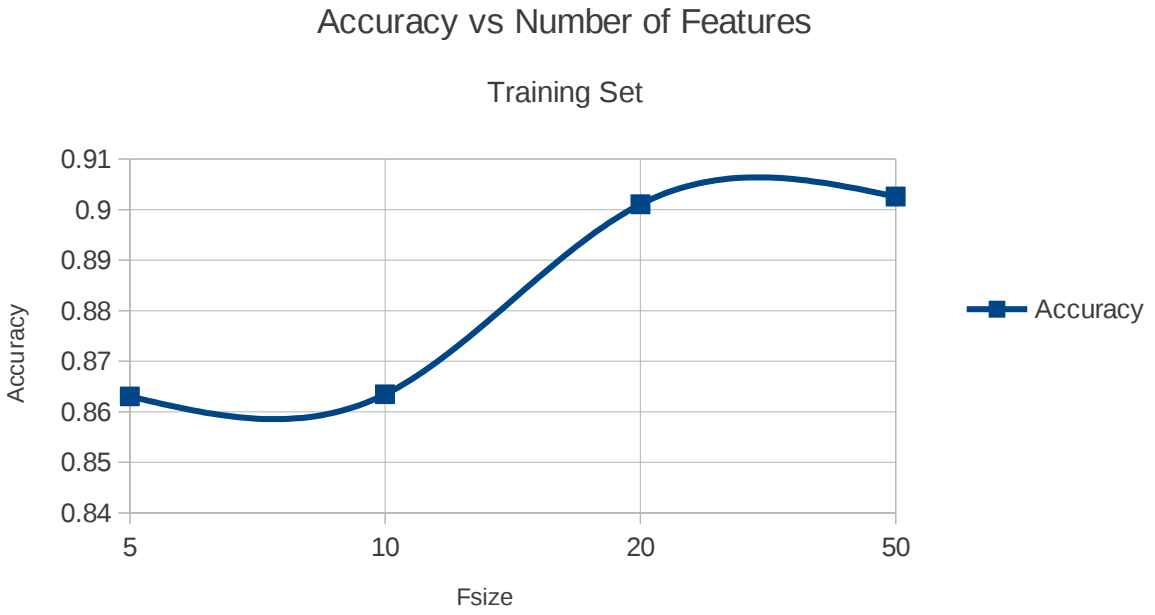


Observation: As for the previous, the accuracy increases (not always) as the number of training samples increases.

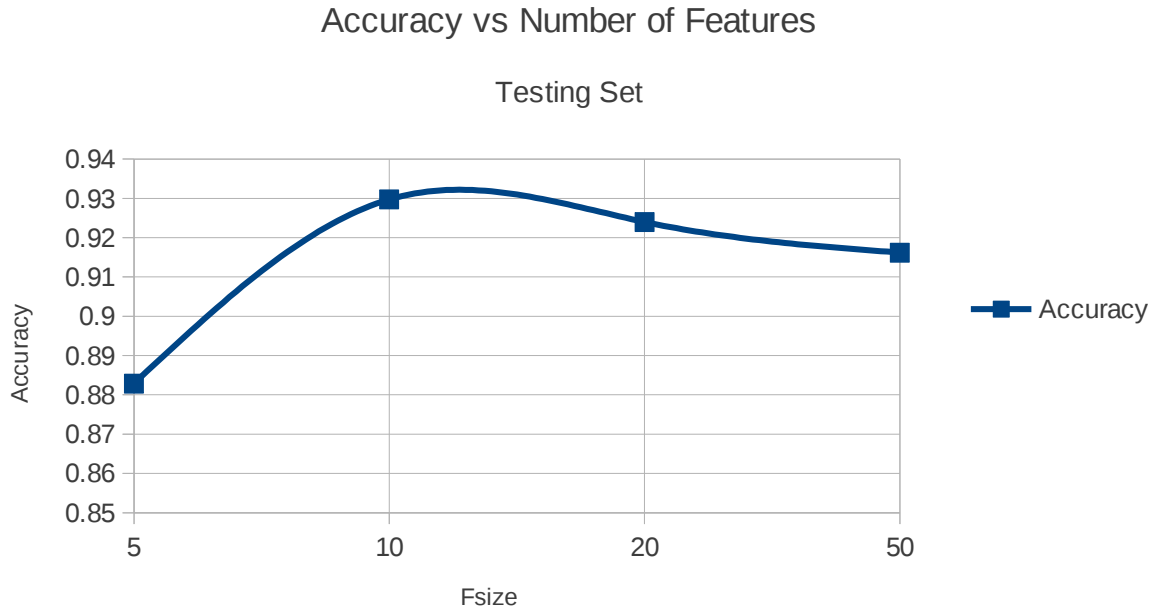
Part C:

Following were the values obtained for fsize-accuracy for Fourth Classifier (SVM):

Fsize	Accuracy
5	0.86303
10	0.86345
20	0.90105
50	0.90258

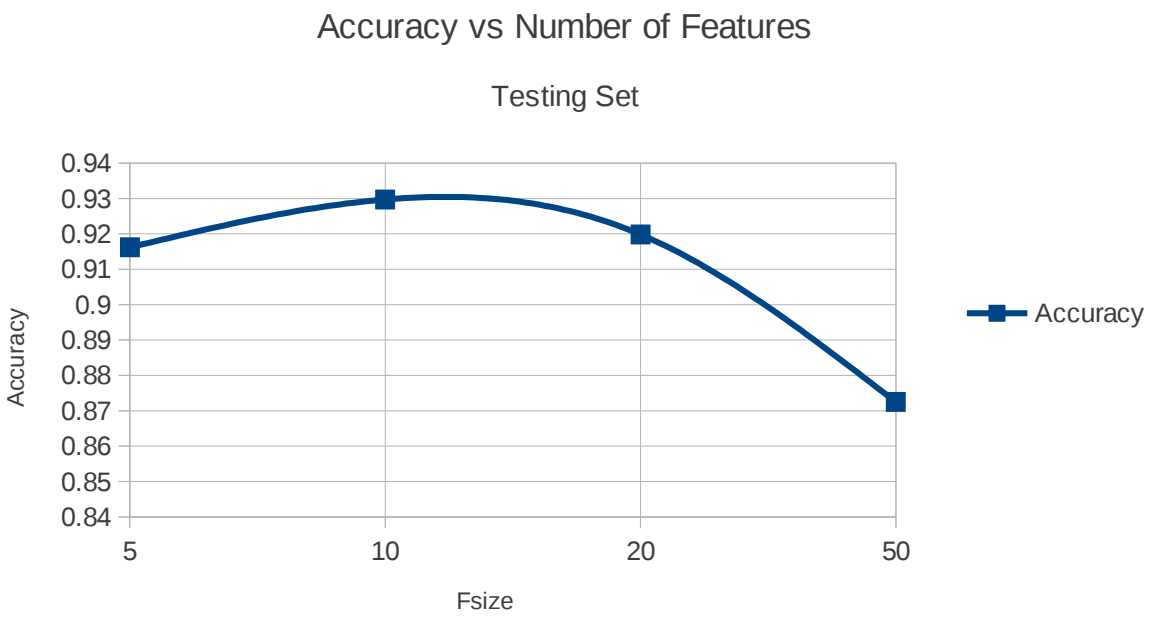


Fsize	Accuracy
5	0.88287
10	0.92972
20	0.924
50	0.91619

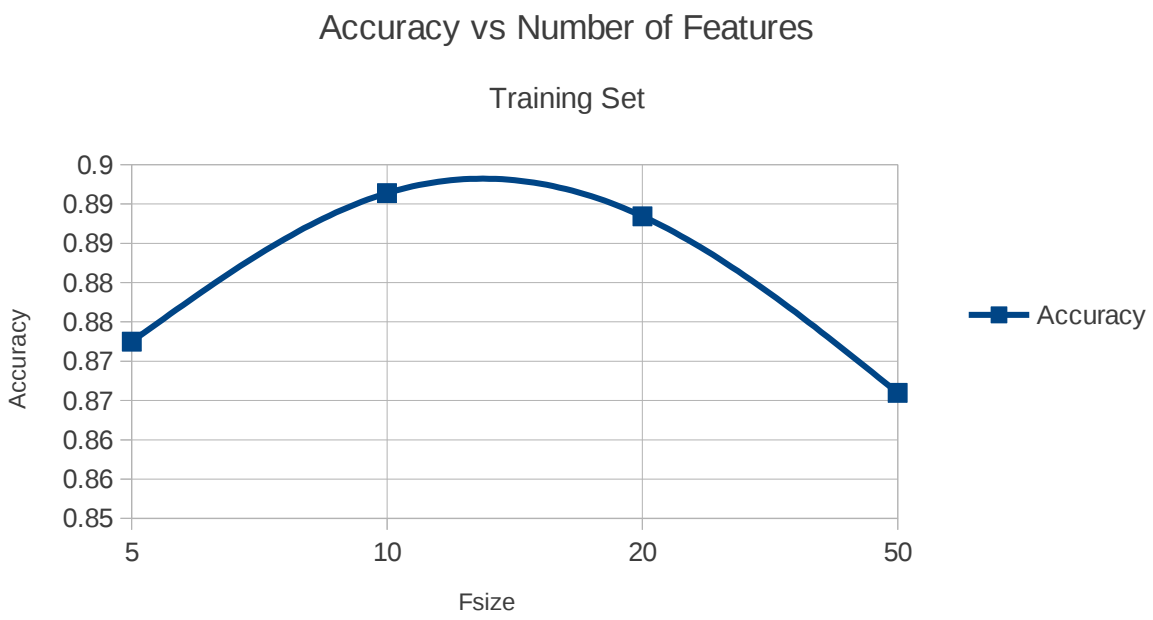


Following were the values obtained for the Third Classifier:

Fsize	Accuracy
5	0.91619
10	0.92972
20	0.91983
50	0.87246



Fsize	Accuracy
5	0.87248
10	0.89139
20	0.88845
50	0.86597



Observation: Again, as the fsize increases, the accuracy of the classifier increases upto a point.