Gabriel Rosca - 10159887

COMP24111 - Machine Learning and Optimisation
Exercise 3 Lab report

Naïve Bayes Classifier for Spam Filtering

1)What are those “parameters” that need learning in discrete and continuous naïve Bayes?

The parameters that need learning in discrete and continuous **Naïve Bayes** are:

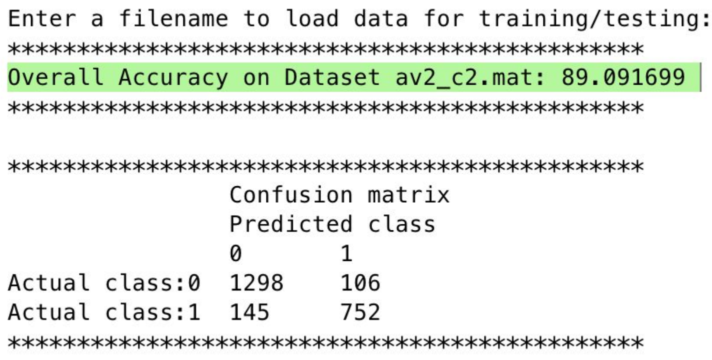
- the probability model where we store the probabilities for each unique value for each feature possible. I created a matrix of structures(number of features x number of classes which contains a vector to store the probability for each unique value that current feature can hold for the current class).

-a vector structure(size of number features) which acts as a frequency structure for each feature. I created a matrix to store the frequency for every unique value of the current feature, a matrix to store the frequency for each unique value of the current feature for each class(number of classes x number of unique values of the current feature)

Using those two structures we can now test our test data and calculate the accuracy of our Naive Bayes classifier.

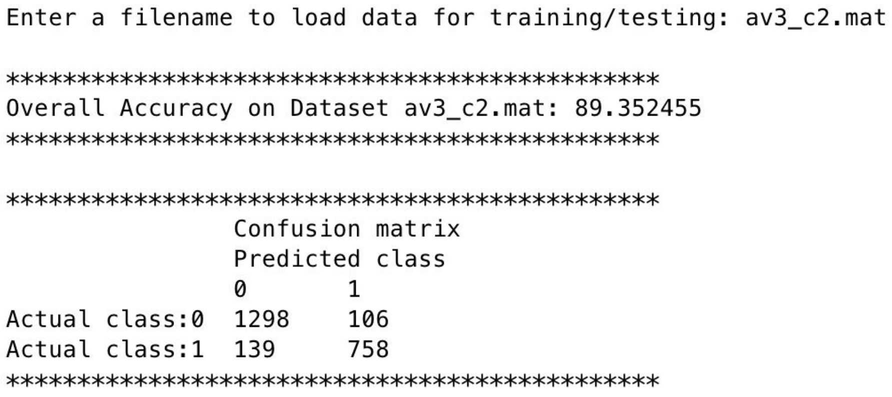
3) What are test results on all the given data sets described in Parts 1 & 2?

For 2 classes and 2 unique values for each feature we get the following results:



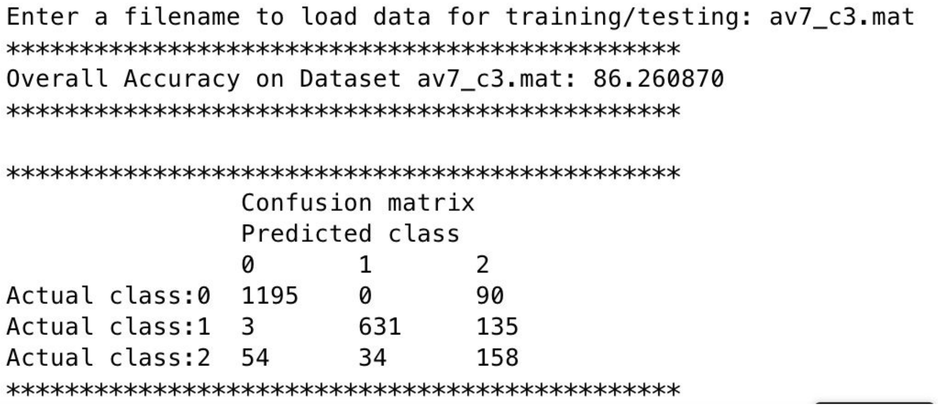
```
Enter a filename to load data for training/testing: av2_c2.mat
*********************************************
Overall Accuracy on Dataset av2_c2.mat: 89.091699
*********************************************

*********************************************
                Confusion matrix
                Predicted class
                0       1
Actual class:0  1298    106
Actual class:1  145     752
*********************************************
```

For 2 classes and 3 unique values for each feature we get the following results:

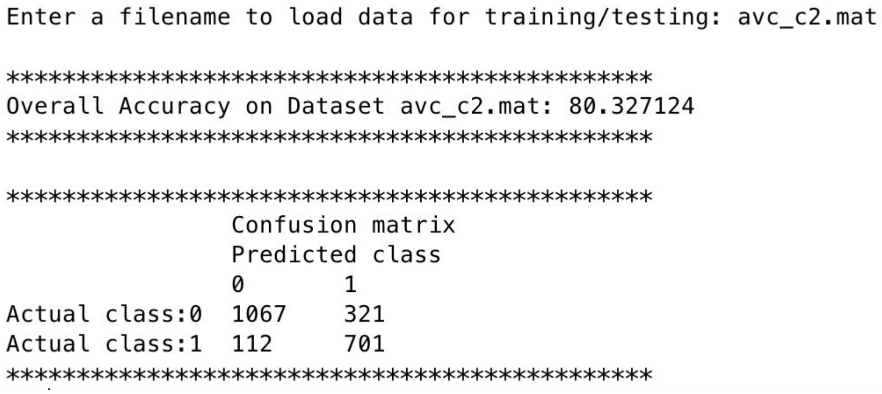


```
Enter a filename to load data for training/testing: av3_c2.mat

*********************************************
Overall Accuracy on Dataset av3_c2.mat: 89.352455
*********************************************

*********************************************
                Confusion matrix
                Predicted class
                0       1
Actual class:0  1298    106
Actual class:1  139     758
*********************************************
```

For 3 classes and 7 unique values for each feature we get the following results:



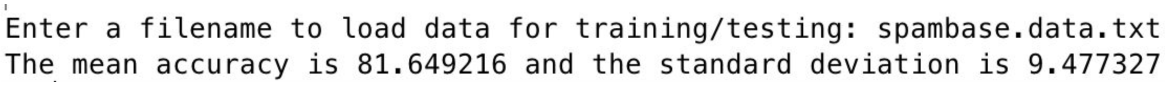
```
Enter a filename to load data for training/testing: av7_c3.mat
*********************************************
Overall Accuracy on Dataset av7_c3.mat: 86.260870
*********************************************

*********************************************
                Confusion matrix
                Predicted class
                0       1       2
Actual class:0  1195    0       90
Actual class:1  3       631     135
Actual class:2  54      34      158
*********************************************
```



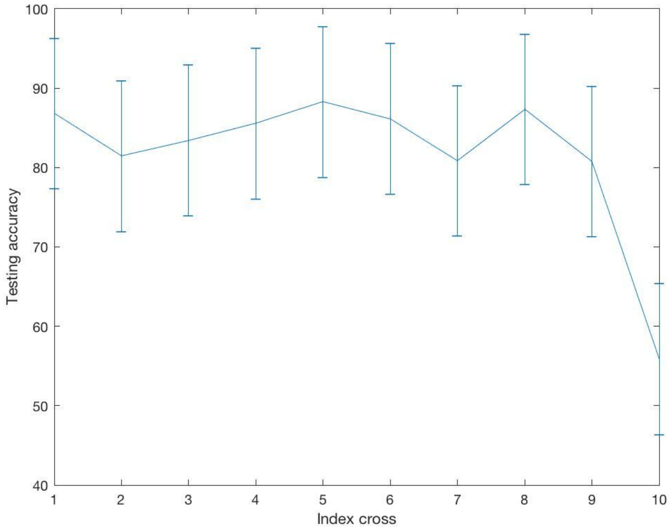
```
Enter a filename to load data for training/testing: avc_c2.mat

*********************************************
Overall Accuracy on Dataset avc_c2.mat: 80.327124
*********************************************

*********************************************
                Confusion matrix
                Predicted class
                0       1
Actual class:0  1067    321
Actual class:1  112     701
*********************************************
```

For 2 classes and continuous values for each feature:

For spambase input data:



```
Enter a filename to load data for training/testing: spambase.data.txt
The mean accuracy is 81.649216 and the standard deviation is 9.477327
```



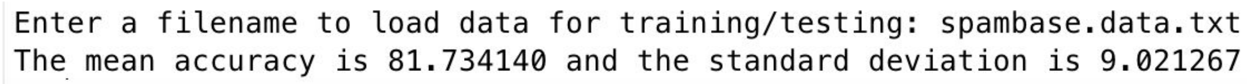
4) What are the motivation and setting(s) in your cross-validation experiments?

We implement cross-validation in order to check the how our program will behave/generalize to an independent to an independent data-set. We split out data set into 10 equal parts and we iterate through all the parts and choose the current part as a triaining data and the 9 others as test data. We observe that the data is partitioned as follows: 1813 samples from class 1 and the rest 2787 from class 0. In order to have an balanced data set for training I picked 181 samples from class 1 and 278 samples from class 0(459 samples for training). Using this method we get an average accuracy of 81.64%.

5) Based on your observation and analysis on experimental results achieved in Parts 1 & 2, can you grasp any non-trivial implication? If any, *in your report*, you must *explicitly* describe your experimental evidence or theoretical justification that leads to such an implication.

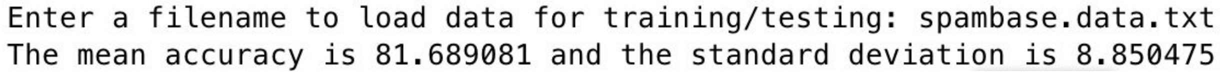
Some probabilities for a certain value of a feature we get a zero probability. In order to avoid a total probability of zero we implemented a zero conditional probability using only one weight to prior(number of virtual examples). Based on my experiments the higher the number of virtual examples the higher the accuracy will get until it reaches a high point then it decreases by a bit but the standard deviation decreases too.

For m = 2 we get the following results:



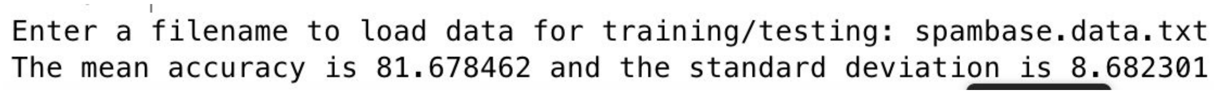
```
Enter a filename to load data for training/testing: spambase.data.txt
The mean accuracy is 81.734140 and the standard deviation is 9.021267
```

For m = 3 we get the following results:



```
Enter a filename to load data for training/testing: spambase.data.txt
The mean accuracy is 81.689081 and the standard deviation is 8.850475
```

For m = 4 we get the following results:



```
Enter a filename to load data for training/testing: spambase.data.txt
The mean accuracy is 81.678462 and the standard deviation is 8.682301
```