COMP24111 – EX3

Matt Walton 10137735

Part 1 – Implementation for discrete input attribute values

For this part of the exercise I successfully implemented a generic naïve Bayes classifier. Naïve Bayes is a classification algorithm which uses Bayes theorem along with assumption that all input features are class conditionally independent.

When training the algorithm I go through each value of each feature and record the probability of a certain value of that feature occurring given the assumption that it is of a certain class (P(X|C)). This is the parameters that need learning in discrete and continuous naïve Bayes. This is stored in a 3D matrix so that each feature has a probability for every possible value given every possible class.

In my implementation I also return another vector labelProbs which contains the probabilities of each class occurring (P(C)) which is used in the testing.

The final class is determined using the MAP classification rule where the class is assigned which has the greatest possibility.

Test Results

Av2_c2 – 2 discrete values for each feature with a binary classification

Accuracy: 88.87%

Confusion matrix:

	Actually Ham	**Actually Spam**
Predic Ham	1292	144
Predic Spam	112	753

Av3_c2 – 3 discrete values for each feature with a binary classification

Accuracy: 89.22%

Confusion matrix:

	Actually Ham	**Actually Spam**
Predic Ham	1294	138
Predic Spam	110	759

Av7_c3 – 7 discrete values for each feature with a 3 classes

Accuracy: 86.21%

Confusion matrix:

	Actually Ham	**Actually Spam**	**Actually N/A**
Predic Ham	1190	0	50
Predic Spam	2	634	37
Predic N/A	93	135	159

By adding the undetermined class we have managed to almost eliminate the false positives between spam and ham emails.

Part 2 – Continuous

I attempted to implement the continuous function Bayes classifier however ran into some issues and subsequently ran out of time.