# COMP24111

*Lab Exercise 3 by Anuj Vaishnav, ID: 9549869*

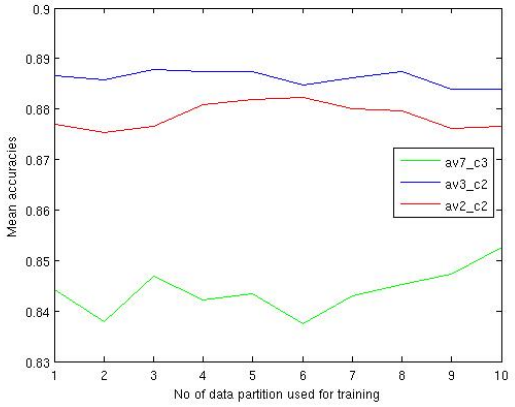
# 1 Part1:

Parmeter are stored in following nested Cell structure:
Level1: [feature probablities] [prior]
(Where each row has distinct class)
Level2: [cell of attribute cells]
(Where each row is a feature)
Level3: [attributes] [attribute probablities]

Probablities estimation process 1) Seperate examples based on classes 2) For each Class dataset and for each feature count the number of times an attribute occurs 3) Smoothing is performed to avoid 0 probablities with below equation P(X = a | C = c1) = (Nc + mp) / (N + M) Where Nc: number of training examples for X = a and C = c1, n: number of training examples for C = c1, p: prior estimate, m: weight to prior (number of "virtual" examples)

*Confusion Matrix for av7_ c3 with NaiveBayes:*

| True value | Predicted | Predicted | Predicted |
|---|---|---|---|
| | NotSpam | Spam | Unknown |
| NotSpam | 1024 | 0 | 59 |
| Spam | 8 | 611 | 41 |
| Uknown | 73 | 158 | 146 |

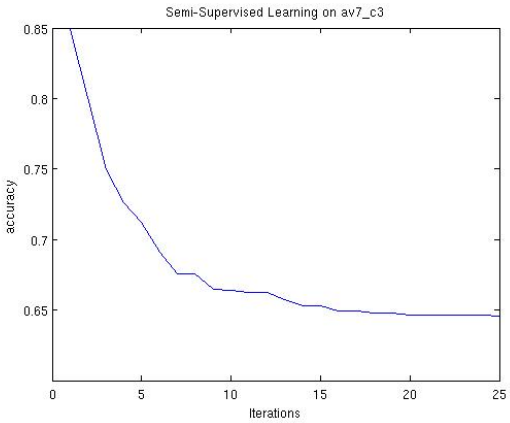



# 2 Part2: Semi-supervised Naive Bayes, reliefF filtering

## 2.1 Semi-supervised

A semi-supervised algorithm is constructed by using combination of labeled and unlabeled data by running the supervised learning algorithm in a loop. Algorithm used is describe below:
Let D = Labeled dataset + Unlabeled dataset.
*Train model on Labeled dataset*
*Until(Convergence)*
*Predict Class P(C|X) for all examples in D*
*Re-train model based on predicted Labels*

This training algorithm is an instance of the more general expectation- maximization algorithm (EM): the prediction step inside the loop is the E-step of EM, while the re-training of naive Bayes is the M-step. [Quoted from Wikipedia]




## 2.2 ReliefF filter

*Description of the Method*: "ReliefF-MI is based on the principles of the ReliefF algorithm [3]. This method works by randomly sampling instances from the training data. For each sampled instance R, its k nearest neighbors from the same class (called nearest hit) and the opposite class of each sampled instance (called nearest miss) are found. Multi-class datasets are handled by finding the nearest neighbors from each class that are different from the class of the current sampled instance, and weighting their contributions by the prior probability of each class estimated from the training data. The weight updating of attribute A (W[A]) is computed as the average of all the examples of magnitude of the difference between the distance to the k nearest hits and the distance to the k nearest misses, projecting on the attribute A. Each weight reflects its ability to distinguish among class labels, thus a high weight indicates that there is differentiation to this attribute among instances from different classes and it holds the same value for instances in the same class. Features are ranked by weight and those that exceed a user-specified threshold are selected to form the final subset." [Quoted from *Feature Selection is the ReliefF for Multiple Instance Learning.* By Amelia Zafra et.al.]
Relief: $W_i = W_i - (x_i - neahHit_i)^2 + (x_i - nearMiss_i)^2$