Data Mining - Assignment 4

1) brief introduction of the classification methods in your classification framework

a) **Basic classification : Naive Bayes**

Implementation

- The training data set is read and parsed one line at a time. The count of the attributes against a particular value is stored in the form of a hash-map for each class.
- The total occurrence of each class in the entire training set is also computed.
- The count of value for each attribute (for a specific class) is calculated as *total class count – occurrence of every other value for that attribute*.
- Subsequently, the test file is read one line at a time, a classification is made for each tuple using the Bayes' theorem. Based on the values of the predicted class and the actual class, the measures true positive, true negative, false positive and false negative are updated.
- The objective is to derive the maximum posteriori. P (class/tuple) is computed for each class and the class which offers the maximum probability is chosen as the class predicted by the classifier. The product of probability of occurrence of each attribute is computed using the computations made on the training data set (for each class). This is then multiplied by the probability of occurrence of that class. Obviously, the assumption is that the attributes are independent of each other.

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

- The accuracy of both the training dataset and test dataset are tested against this model.
- The issue of zero probability is avoided by using laplacian correction. By default 1 is added to the count of any attribute and to compute the probability, the denominator is taken as occurrence count of the class in the training data set + no of unique values the attribute can take.

b) **Ensemble classification – AdaBoost**

The Naive Bayes classifier designed above is used to generate K classifiers using the following algorithm

- Initially all tuples are given the same weight (tuple count = D).
- At each round, D tuples are selected with replacement. The probability of selecting a tuple is proportional to its weight.
- The Naive Bayes model is used on the selected tuples.
- The error associated with a particular model is the weighted sum of all the misclassified tuples.
- If the error of the classifier exceeds 0.5, the entire process is carried out again.
- Once the K classifiers have been decided upon, for each test tuple, the prediction of each classifier is taken into account and the following weight equal to $\log((1 - \text{err}(\text{classifier}_i))/\text{err}(\text{classifier}_i))$ is assigned to the prediction.

- The class that has a higher cumulative weight assigned to it is chosen as the prediction.
- As before, the true positive, true negative, false positive and false negative values are updated suitably in each step and the measures are computed.

*2) all model evaluation measures you calculated above (8 metrics * 2 methods * 4 datasets * 2 (training and test))*

Adult

Naive Bayes

Training

```
Accuracy            : 79.75077881619937%
Error Rate          : 20.24922118380062%
Sensitivity         : 81.77215189873418%
Specificity         : 79.0909090909091%
Precision           : 56.076388888888886%
F-1 Score           : 0.6652935118434603
F Score(beta = 0.5) : 0.5983697665802149
F Score (beta = 2)  : 0.7490723562152134
```

Test

```
Accuracy            : 79.34810699056726%
Error Rate          : 20.651893009432744%
Sensitivity         : 79.70722535589579%
Specificity         : 79.23436835389197%
Precision           : 54.8673384487381%
F-1 Score           : 0.6499479822592126
F Score(beta = 0.5) : 0.5851441416571361
F Score (beta = 2)  : 0.7308933277505479
```

AdaBoost

Training

```
Accuracy            : 80.1246105919003%
Error Rate          : 19.87538940809969%
Sensitivity         : 81.26582278481013%
Specificity         : 79.75206611570248%
Precision           : 56.71378091872792%
F-1 Score           : 0.6680541103017691
F Score(beta = 0.5) : 0.6036103798420459
F Score (beta = 2)  : 0.7479030754892825
```

Test

```
Accuracy            : 79.56454322263859%
Error Rate          : 20.435456777361416%
Sensitivity         : 79.26403438087564%
Specificity         : 79.65971926839643%
Precision           : 55.24148259078997%
F-1 Score           : 0.6510755653612796
F Score(beta = 0.5) : 0.5880594635526682
F Score (beta = 2)  : 0.7292181476722347
```

Breast Cancer

Naive Bayes

Training

```
Accuracy            : 73.88888888888889%
Error Rate          : 26.111111111111114%
Sensitivity         : 48.214285714285715%
Specificity         : 85.48387096774194%
Precision           : 60.0%
F-1 Score           : 0.5346534653465347
F Score(beta = 0.5) : 0.5720338983050848
F Score (beta = 2)  : 0.5018587360594795
```

Test

```
Accuracy            : 74.52830188679245%
Error Rate          : 25.471698113207548%
Sensitivity         : 44.827586206896555%
Specificity         : 85.71428571428571%
Precision           : 54.166666666666664%
F-1 Score           : 0.49056603773584906
F Score(beta = 0.5) : 0.52
F Score (beta = 2)  : 0.46428571428571425
```

AdaBoost

Training

```
Accuracy            : 72.77777777777777%
Error Rate          : 27.22222222222222%
Sensitivity         : 44.642857142857146%
Specificity         : 85.48387096774194%
Precision           : 58.139534883720934%
F-1 Score           : 0.5050505050505051
F Score(beta = 0.5) : 0.5482456140350879
F Score (beta = 2)  : 0.46816479400749067
```

Test

```
Accuracy            : 75.47169811320755%
Error Rate          : 24.528301886792452%
Sensitivity         : 44.827586206896555%
Specificity         : 87.01298701298701%
Precision           : 56.52173913043478%
F-1 Score           : 0.4999999999999999
F Score(beta = 0.5) : 0.5371900826446281
F Score (beta = 2)  : 0.4676258992805755
```

led

Naive Bayes

Training

```
Accuracy            : 84.13991375179684%
Error Rate          : 15.860086248203162%
```

```
Sensitivity         : 62.539184952978054%
Specificity         : 93.65079365079364%
Precision           : 81.26272912423626%
F-1 Score           : 0.7068201948627104
F Score(beta = 0.5) : 0.766717909300538
F Score (beta = 2)  : 0.6556030233322379
```

Test

```
Accuracy            : 83.68606701940035%
Error Rate          : 16.313932980599645%
Sensitivity         : 58.97435897435898%
Specificity         : 94.76372924648787%
Precision           : 83.46774193548387%
F-1 Score           : 0.6911519198664441
F Score(beta = 0.5) : 0.7706626954579302
F Score (beta = 2)  : 0.6265133171912833
```

AdaBoost

Training

```
Accuracy            : 84.18782942022041%
Error Rate          : 15.812170579779588%
Sensitivity         : 63.479623824451416%
Specificity         : 93.30572808833678%
Precision           : 80.67729083665338%
F-1 Score           : 0.7105263157894736
F Score(beta = 0.5) : 0.7653061224489794
F Score (beta = 2)  : 0.6630648330058939
```

Test

```
Accuracy            : 84.03880070546738%
Error Rate          : 15.961199294532626%
Sensitivity         : 60.3988603988604%
Specificity         : 94.6360153256705%
Precision           : 83.46456692913385%
F-1 Score           : 0.7008264462809917
F Score(beta = 0.5) : 0.7754206291148501
F Score (beta = 2)  : 0.6393244873341376
```

Poker

Naive Bayes

Training

```
Accuracy            : 72.07293666026871%
Error Rate          : 27.927063339731284%
Sensitivity         : 99.19786096256684%
Specificity         : 3.061224489795918%
Precision           : 72.24926971762414%
F-1 Score           : 0.836056338028169
F Score(beta = 0.5) : 0.764003294892916
F Score (beta = 2)  : 0.9231152027867628
```

Test

```
Accuracy            : 66.3716814159292%
Error Rate          : 33.6283185840708%
Sensitivity         : 97.60348583877996%
Specificity         : 0.91324200913242%
Precision           : 67.36842105263158%
F-1 Score           : 0.7971530249110319
F Score(beta = 0.5) : 0.7181789034947097
F Score (beta = 2)  : 0.8956417433026788
```

AdaBoost

Training

```
Accuracy            : 71.97696737044146%
Error Rate          : 28.023032629558543%
Sensitivity         : 99.19786096256684%
Specificity         : 2.7210884353741496%
Precision           : 72.17898832684824%
F-1 Score           : 0.8355855855855856
F Score(beta = 0.5) : 0.7633744855967078
F Score (beta = 2)  : 0.9228855721393036
```

Test

```
Accuracy            : 66.66666666666666%
Error Rate          : 33.33333333333333%
Sensitivity         : 98.25708061002179%
Specificity         : 0.45662100456621%
Precision           : 67.41405082212258%
F-1 Score           : 0.7996453900709221
F Score(beta = 0.5) : 0.7192982456140351
F Score (beta = 2)  : 0.9001996007984032
```

3) does your framework perform equally good on training and test datasets? Why or why not?

Both Naive Bayes and AdaBoost (which in this case has been implemented on Naive Bayes) work equally well on training and test data sets.

Naive Bayes is **simple** model that **does not over-fit** the data and given a similar distribution should therefore return similar accuracies. It treats the attributes independent of one another, basically, it assumes there is no relation between the co-occurrence of attributes in a particular class. As mentioned earlier in the report, the only information needed to draw inferences from the classifier are attribute count by class and the class count. Irrespective of whether a tuple belongs to the training or the test data, to predict its class, the probabilities of the occurrence of individual words are considered for that class. Intuitively, it therefore makes sense that given enough training tuples, the classifier should not discriminate between test and training data.

4) parameters you chose during implementation and why you chose these parameters;

The number of classifiers (K) in the AdaBoost implementation of Naive Bayes has been chosen as 30. The idea was to tune the parameter so as to achieve an optimal result. K was initially set to 2 and increased iteratively to determine if increasing the classifiers resulted in any significant improvement in accuracy. Using values of K= 10 or 20 provided results very similar to those provided by K = 30. Increasing the number of learners to over 30 was also tried, however, there was no marked difference. K was set at 30, a reasonably high number of classifiers to successfully conclude that increasing the classifiers did not have the desired effect.

5) and also your conclusion on whether the ensemble method improves the performance of the basic classification method you chose, why or why not;

AdaBoost on Naive Bayes does give marginally improved results in comparison to the base classifier, however, there is **no significant improvement** as would be expected from a boosting model.

Naive Bayes as a model has a high bias but less variance – variance as the word indicates is the variability in prediction and bias is the difference between the correct value and the prediction. Ensemble methods tend to work well on models that have a high variance and a reasonably low bias – allowing one to converge to solutions that are more optimal than those predicted by the base classifier. As mentioned earlier, Naive Bayes is a simple model that generalizes the training data by using the concept of attribute independence given class and is therefore less likely to produce any significant improvement.

6) verify your guess on whether your basic classification method is ensemble compatible.

The figure below depicts a chart representing the concept of bias and variance (picked from the website – http://scott.fortmann-roe.com/docs/BiasVariance.html). Naive Bayes is depicted by the High Bias/Low Variance image. In a scenario like this, it is expected that boosting cannot significantly improve the results. The method represented by Low Bias/High Variance on the other hand can however gain significantly by using boosting techniques, since they are are spread out around the bulls eye and can therefore converge to a more optimum result. Also, the article "A study of AdaBoost with Naive Bayesian Classifiers : Weakness and Improvement" by Kai Ming Ting and Zijian Zheng explains this very same behavior.

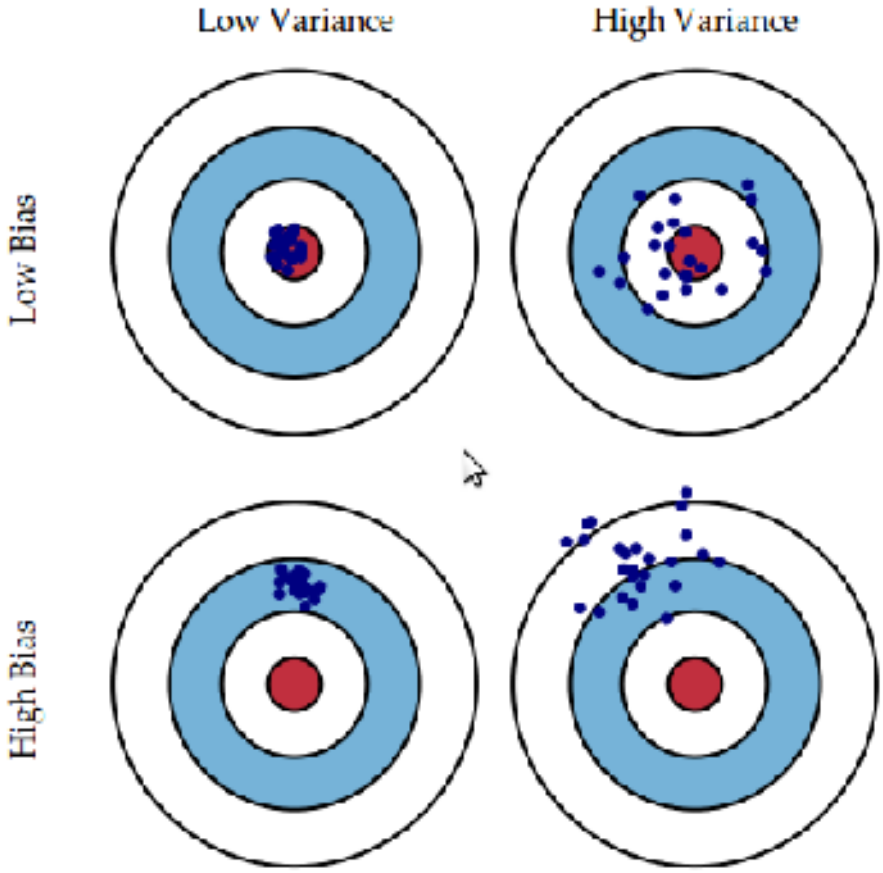

Fig. 1 Graphical illustration of bias and variance.