Due date: October 28nd, by 11:59pm.

Task 1 (5 pt)

Consider the following training set:

Example	Attribute 1	Attribute 2	Attribute 3	Attribute 4	Class
$\mathbf{x_1}$	1	1	1	1	+
$\mathbf{x_2}$	1	1	0	1	+
$\mathbf{x_3}$	0	1	1	0	+
$\mathbf{x_4}$	1	0	0	1	+
$\mathbf{x_5}$	1	0	0	0	-
$\mathbf{x_6}$	1	0	1	0	-
$\mathbf{x_7}$	0	1	0	0	-
$\mathbf{x_8}$	0	0	1	0	-

(i) For each attribute, each value, each class, calculate the conditional probability P(Attribute=value|class) using the generalized Laplace estimate:

$$P(A_i = v_j \mid c_k) = \frac{n_{ijk} + 1}{n_k + s_i}$$

where n_k is the number of samples in class c_k, n_{ijk} is the number of samples in class c_k where Attribute $A_i = v_j$, and s_i is the number of possible values for A_i .

(ii) Show how a naïve Bayes classifier trained on this dataset would classify the following example

$\mathbf{x_*}$	1	1	0	0	?

(show your work)

Task 2 (5 pt)

In this task, you will experiment with classification of dog versus cat images using support vector machine. The images come from the competition at https://www.kaggle.com/c/dogs-vs-cats.

Each instance in DogsVsCats.train is a 64-color histogram of the corresponding image. The values of the features are normalized fraction of pixels in the image of a given color bin. Cats are the negative class and dogs are the positive class.

Download svm_light from http://svmlight.joachims.org/. Follow the directions (under "Source Code and Binaries" and "Installation") to download and install this package on your computer. Read the "How to Use" section.

(i) Compare the accuracy of the linear kernel versus the polynomial kernel (with degree 5) using 10-fold cross validation, which is:

Divide the training set into 10 disjoint subsets of approximately equal size. Each subset should have roughly the same number of positive and negative examples. You might use available software/libraries/packages or your own programs for partitioning the data. For each iteration, train the SVM using 9 subsets and test its *accuracy* on the last subset, called **validation set**. Report the average accuracy on 10 measures for each kernel.

(ii) For each kernel in (i), train the SVM on the entire training set and get the **training accuracy** and classify the testing set to get the **testing accuracy**. Report your results.

In your report, give the training, validation, and test accuracies of each kernel. Which gives a better prediction of the test accuracy, the training accuracy or the validation accuracy?

Task 3 – Boosting (10 pt)

Implement the AdaBoost algorithm described in class to boost decision stump, i.e., depth-1 decision tree. Use *weighted* entropy and information gain to choose splitting attribute. Replace count of samples by sum of sample weights. For example,

$$p_i = \frac{\left|\{x \in S, x.class = c_i\}\right|}{|S|} \quad \text{can be adapted to} \quad p_i = \frac{\sum_{x_k \in S \& x_k.class = c_i} d(k)}{\sum_{x_k \in S} d(k)}.$$

Your program should take arguments in the following order:

- T, the number of boosting iterations
- file1, the training data file
- file2, the testing data file

The output of your code should include the testing accuracy of the ensemble classifier and the strength of each classifier, one number per line. Essentially, the code should print on stdout (in java, you can call System.out.print):

testing accuracy
α_1
α_2
…

α_T

(i) Set T=10. Run the ensemble classifier *H* on the test data. Report the accuracy of *H*.
(ii) Set T=20. Run the ensemble classifier *H* on the test data. Report the accuracy of *H*.

In addition, include a paragraph commenting on your results. Provide/Refer to intermediate results, such as error and strength of the decision stump at each iteration, to support your conclusion if needed. Submit separately your source code Adaboost.java/.cpp./.py and compiling instructions if needed.

Note: source code that does not compile or produce specified output will not be graded. (If you are not able to generate output as instructed, seek help early.)

The training and testing data is taken from the Mushroom dataset (https://archive.ics.uci.edu/ml/datasets/Mushroom) on UIC Machine Learning repository (with one column removed to eliminate missing data). The first column indicates class label, e=edible and p=poisonous. And other columns contain attribute values.