CS 429/529: Machine Learning
Martin Cenek
Fall 2022 Due: NLT 23:59 11/20/2022
Homework 6:

In this homework you will implement a probabilistic machine learning algorithm that will implement an e-mail spam classifier. So, implement a Naive Bayesian Classifier to generalize the training corpus of instances to predict a spam/non-spam label of a new, never seen, email feature vector.

The datafile for you to use contains SpamAssassin (SA) feature vectors (lines) for e-mail messages that were classified as a spam or non-spam – the vectors were returned from the SA's pre-processor. In addition to each binary feature vector, each line is marked as a span or non-spam.

Data description:
Column 1:int has a feature vector instance number
Column 2: {-1,1} indicates if a message is a spam -1:non spam and 1:spam followed
Column 3 the feature vector: Each feature vector is 344 characters long, where each character represents a value of a unique word or an e-mail feature that would potentially identify e-mail as spam or non-spam. The value of each character is a boolean value that identifies weather or not the feature occurred in an e-mail (value 1 for yes) and value 0 if the word does not appear in the document.

The dataset has 15497 e-mail messages with both spam and non-spam messages. Feel free to split up the dataset into a training and validation subsets that your algorithm will use for training and validation phase.

1. Build the Naive Bayesian Classifier. Assert the conditional independence among all 334 reported features of the instance vector.
2. Assess the classification accuracy of your Naive Bayesian Classifier by restarting the (partitioning-training-assessment) process 20 times. Each partitioning will have different number of training instances to build your classifier starting with only 100 training e-mails for the first partitioning and the subsequent partition size increasing by 100. The last partitioning will be 80/20 (12400/3100). Make sure your partitions are equally weighted for the spam/non-spam labels. For each constructed classifier, record the precision and recall (with associated false classification counts) and plot these models on a ROC plot.

Graduate Students:
No extra requirement. Enjoy your freedom instead.

What to submit:
On the moodle, submit an archive file (.zip or similar) with your (1) code base, (2) readme file with the instructions on the architecture, the compilation instructions and the usage (how and what the interface needs as its inputs) and (3) one page write-up of your findings. Please be as complete and thorough as you can be, use plots and tables summarize your findings.