CS 5644: Assignment #2
Date Assigned: Sep 27, 2017
Date Due: Oct 11, 2017, 11:59 ET

1. (60 points) Machine learning has now permeated multiple disciplines, even politics. The current landscape in the US is rife with data scientists and other quantitative experts making predictions about ongoing and upcoming elections. Consider the Congressional Voting Records dataset from the UCI machine learning repository (https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records).

The dataset contains two files: one with a ".names" suffix and one with a ".data" suffix. The actual data is in the ".data" suffix and ".names" describes the metadata (i.e., describes what the different columns mean). Note that each row of the ".data" file contains one instance and includes both features and the class label (please take care to note the order). The machine learning problem here is to take the votes of US congressmen/congresswomen as input and predict whether they are a Republican or a Democrat. In particular, our goal is to solve this problem using both decision trees and a naïve Bayes classifier.

First, spend some time understanding the structure of the dataset, how the instances are organized, how the features/class are organized, and so on. You need to "massage" this data into the form that scikit-learn requires before you can apply either a decision tree or a naïve Bayes classifier. So spend some time understanding and planning how you will do this massaging. You can do this in Python or in Excel or any way you choose. Note that this step is a natural part of the machine learning and knowledge discovery process. Data is rarely given in the form that machine learning can be directly applied, so that considerable effort goes into cleaning, manipulating, and massaging it. Do not apply scikit-learn before ensuring that it is in the form required.

Just like the PlayTennis dataset, the features are binary-valued but note that some features have missing values for some rows (instances). You need to decide how you will handle them. There are three possibilities here: i) discard instances that have missing feature values, ii) treat "missing" as if it is a value (and thus a binary feature becomes a ternary, or three-valued, feature), iii) impute missing values (i.e., for each feature, replace missing values with the most common value for that feature), so that they are no longer missing or unknown. If you read the ".notes" file, it explains why some values are missing and what they mean.

- Implement a decision tree and Naïve Bayes classifier for classification, with each of the above three ways of dealing with missing values. So you are experimenting with 6 scenarios.

- Perform 5-fold cross validation and report precision, recall, and F1-scores for each of the 6 scenarios.

2. (20+20=40 points) For what type of dataset would you choose decision trees as a classifier over Naive Bayes? Vice versa?

(continued)

What to submit:

Exactly one zipped file containing:

- a PDF document summarizing answers to questions 1 and 2. Do not submit pages and pages of code. Instead distill your lessons and experiences succinctly.
- Either hyperlinks to or actual attachments of your data files and your iPython notebook(s).