CS 391L: Machine Learning **Spring 2021**

Homework 1 - Programming

Lecture: Prof. Adam Klivans

Keywords: decision trees

1. Read the online documentation on decision trees and random forests in scikit-learn to find out how to use decision trees and random forests. Notice that training a classifier is done using the `fit` method, and that for decision trees this is done using a more sophisticated evolution (known as CART) of the ID3 algorithm covered in class.

   (a) Use the breast cancer data set from Homework 0 to create a training set. Recall that the label is 0 if the patient's data indicates a malignant cancer and 1 otherwise. Compute the base rate of malignant cancer occurrence over the entire data set.

   (b) The goal is to build a decision tree that, based on the other features in the set, predicts whether or not a patient has malignant cancer. So this is a classification problem. Using `tree.DecisionTreeClassifier` and other functions in the scikit-learn library, one can build a decision tree and calculate both its training accuracy when fitted to the entire data set as well as its accuracy using 10-fold cross validation (which gives a better idea of true accuracy).

   Vary the depth of your decision tree (use `max_depth` = 1,2,. . . ,10) and plot both training accuracy and cross-validated accuracy (as a function of the depth, on the x-axis). Plot both curves on the same plot and use a legend to label them.

   (c) Now try the random forest classifier of the scikit-learn library and use the best depth you get from (b) as `max_depth`. Vary the number of trees in the forest via the parameter `n_estimators` and plot its 10-fold cross-validated accuracy (use `n_estimators` = $1, 2, \ldots, 20$). Do you see an improvement using random forests versus using a single tree? (Note: use the `n_estimators`=1 result as the result for a single tree.)

   (d) Using the method for building a decision tree you used in part (b), build a tree but randomly hold out a .2, .4, .6, and .8 fraction of the data set (so you will need to build 4 different trees for each depth value). For each fraction held out, plot a curve of the test accuracy (the accuracy on the held-out set) against depth. You should have four curves. Plot them all on the same plot and use a legend to label them.