Machine Learning **S2018**

Problem Set 1

Handed Out: Jan 22^{nd}, 2018 *Due: Feb* 5^{th}, 2018

In this assignment you will implement a decision tree classifier that will be used to classify four synthetic datasets and one real dataset. You will submit a writeup in Word or PDF that summarizes your results and all code as a zip file. Submit the writeup (with attached source code) to the Canvas submission locker before 11:59pm on the due date.

Classify Synethetic Data (30 points)

Write a method to estimate a decision tree of maximum depth 3 and apply it to the sythetic training datasets. Go ahead and train and test on the same dataset (this is bad form in general, but that is ok for this assignment).

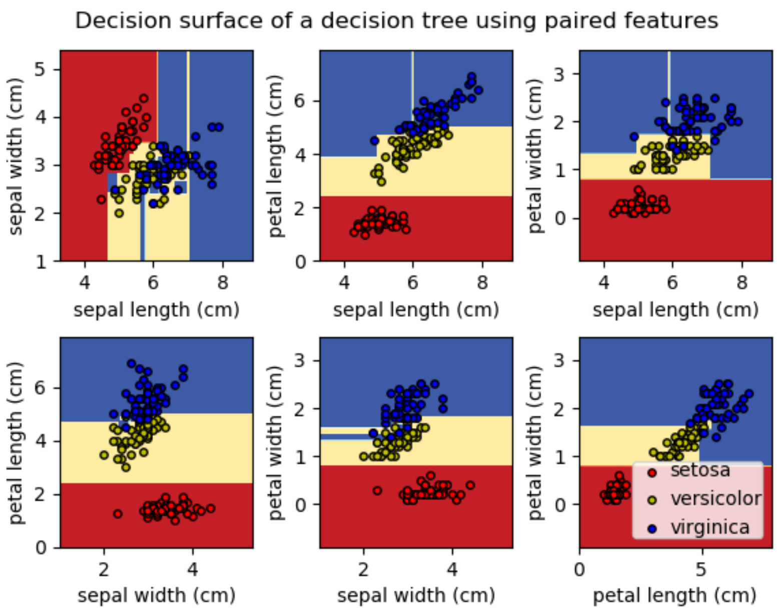
Details:

- You may not use a high-level function for decision tree fitting.
- Along each dimension, consider a finite set of possible splits. One possible way to do this is to separate data into a finite number of equidistant bins based on the maximal and minimal value along a given dimension. Another possible way to do this is to separate data into a finite number of bins such that each bin contains roughly the same number of examples. These are just two possible ways to do things. If you have a favorite way of discretizing data, I encourage you to use it!
- As mentioned at the start of class, all code should be written in Python. You can choose whether you want to use Python 2 or 3.
- Use entropy and information gain to determine the optimal splits.
- Explain any implementation choices you had to make in the final report.
- Include the training set error for each synthetic dataset in the final report.

Visualize your classifiers (20 points)

Write a function that creates a visualization of the data set and the output of your best decision tree. Recall that supervised learning methods are approximating a function, so we can sample their value anywhere in feature space. Your function will display a graphic that shows the training data as a scatter plot with the decision tree approximation as a background.

Here is an example:



Your approach must meet the following criteria:

- show the training data and clearly distinguish between the class labels: you can color code the labels or use different markers
- show the function approximation: to do this, write a function that grid samples your prediction function in the area of feature space around the training data and use this to construct an image. You can use this image as a backdrop for your sample points.

Your writeup must include visualizations for each of your best decision trees for the synthetic data (so a total of 4 visualizations). Make sure you title each figure with the dataset name.

Classify Real Data on Video Game Sales (30 Points)

Extend your decision tree to work on a real dataset of video game sales and ratings. As before, create a method that estimates a decision tree of maximum depth 3. Chances are, you will be able to reuse most of the code written for the first part of this assignment, and I encourage you to do so. You are also allowed to test on this training data as well. Remember, this is typically a bad thing to do, but it is okay for this assignment.

Details:

- You may not use a high-level function for decision tree fitting.
- This dataset contains a mixture of data types and even has some missing data. You can handle continuous data in this data similarly to how you handled it for the synthetic

datasets; however, be aware that some dimensions contain nominal data. For handling missing data, there are many different possibilities. For example, you could create a new *not available* label and add that to the dataset.

- As mentioned at the start of class, all code should be written in Python. You can choose whether you want to use Python 2 or 3.
- Use entropy and information gain to determine the optimal splits.
- Explain any implementation choices you had to make in the final report.
- Include the training set error for this dataset in the final report.

Presentation (20 points)

Your report must be complete and clear. A few key points to remember:

- Complete: the report does not need to be long, but should include everything that was requested.
- Clear: your grammar should be correct, your graphics should be clearly labeled and easy to read.
- Concise: I sometimes print out reports to ease grading, don't make figures larger than they need to be. Graphics and text should be large enough to get the point across, but not much larger.
- Credit (partial): if you are not able to get something working, or unable to generate a particular figure, explain why in your report. If you don't explain, I can't give partial credit.

Bonus (5 points, required for graduate students)

Improve your decision tree implementation by using cross validation to choose the optimal maximum tree depth for each synthetic dataset. Are the depths different for different datasets? If so, why do you think this happens?

You may use either leave-one-out or a fold-based method. You **must** include a description of your algorithm in your writeup that would be sufficient for someone to reimplement your method.