Introduction to Machine Learning (CS 135)
Assignment 05 (40 points)
Due on Gradescope by 11:59 PM, Saturday, 20 November 2021

For this assignment, you will make modifications to a Python notebook (`hw05.ipynb`) that has been supplied. You will complete the various required sections outlined below in that notebook. When you are done, generate a PDF version of that notebook, with all results (figures, printed results, etc.) included, for submission to Gradescope. You will also submit the raw notebook source, two PDF images generated by your program (but not embedded in the worksheet itself), and a `COLLABORATORS.txt` file, via a separate link, as described below.

Adding the `graphviz` library

Before you start the assignment, you should add one additional library to your Python install; this will allow you to visualize decision trees in graphical format. You will only need to do the following once; after these steps are performed, these tools will be available, along with all the other ones we have been using, simply by activating the environment as usual.

1. Activate your environment as you normally would:

```
conda activate ml135_env_sp21
```

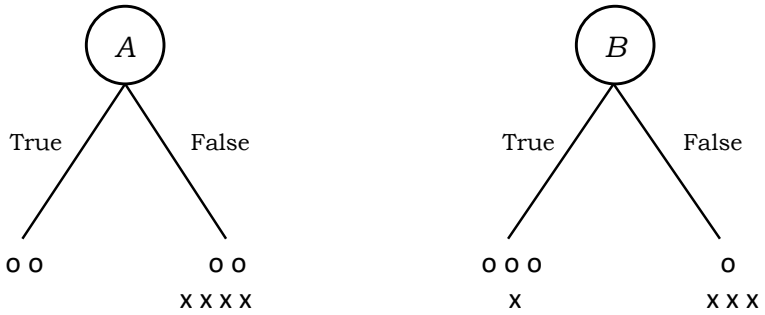
2. From within the environment, install `graphviz`:

```
conda install python-graphviz
```

Once these steps are completed, you will be able to use the new tools from inside the COMP 135 environment, in notebooks and other Python code.

Decision Trees (33 points)

You will examine the use of decision trees for classification, along with different heuristics for evaluating which features to choose when building such trees.



1. (*15 pts.*) The diagram above shows the results of splitting a simple data-set according to two binary features (A and B). The data-set consists of eight entries, of two different types (`o` and `x`). You will compute and display the results of computing the two feature-heuristics seen in the readings and in class lecture notes.

(a) (5) Compute the values for each feature, based upon the counting heuristic discussed in the reading (Daumé). Print out the features in order from best to worst, along with the heuristic (correctness) value for that feature, using the format:

```
feature_name: num_correct/total_data
```

(b) (6) Compute the values for each feature, based upon the information-theoretic heuristic discussed in lecture. Print out the features in order from best to worst, along with the heuristic (gain) value for that feature, to 3 decimal places of precision, using the format:

```
feature_name: information_gain
```

(c) (4) Discuss the results: if we built a tree using each of these heuristics, what would happen? What does this mean?

2. (*6 pts.*) We have provided some data on abalone, a widespread shellfish.* The input data consists of a number of features of abalone, as shown in the following table, while the output is the number of rings found in the abalone shell:

column	type	unit	description
`is_male`	binary		`1 ==` male; `0 ==` female
`length_mm`	numeric	mm	longest shell measurement
`diam_mm`	numeric	mm	shell diameter, perpendicular
`height_mm`	numeric	mm	height of shell
`whole_weight_g`	numeric	gram	weight (entire)
`shucked_weight_g`	numeric	gram	weight (meat)
`viscera_weight_g`	numeric	gram	weight (guts)
`shell_weight_g`	numeric	gram	weight (dried shell)

In addition, we have supplied a simplified version of the data, where each input-feature has been converted to a binary value (either above average value for that feature (`1`), or not `0`), and the output value $y \in \{0, 1, 2\}$ no signifies a *Small*, *Medium*, or *Large* number of rings; the data has also been simplified down to only four features of the original eight. Each data-set is broken up into `x` and `y` sets already, for both training and testing.

Your code will explore these data sets, computing the two heuristics for the simplified data, and classifying both sets using decision trees.

(a) (3) Compute the counting-based heuristic for the features of the ***simplified*** abalone data. Print out the features in order, using the same format as before.

(b) (3) Compute the information-theoretic heuristic for the features of the ***simplified*** abalone data. Print out the features in order, using the same format as before.

*Original data: Warwick J. Nash, et al. (1994) https://archive.ics.uci.edu/ml/datasets/Abalone

3. (*12* pts.) You will use `sklearn` decision trees on both versions of the abalone data-set:

   https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

   (a) (8) For each data-set, create the classifier using the `criterion='entropy'` option, which uses the same information-theoretic heuristic discussed in lecture. After building the model, you can use its `score()` function to get its accuracy on each of the testing and training portions of the data. Print out these values, being clear which value is which.

   In addition, export the two trees as PDF images, using the `export_graphviz()` and `render()` functions.[†] When done, you should be able to open those images (they will be in the directory with your active notebook file) to examine them.

   (b) (4) Discuss the results you have just seen. What do the various accuracy-score values tell you? How do the two trees that are produced differ? Looking at the outputs (leaves) of the simplified-data tree, what sorts of errors does that tree make?

Code submission (7 points)

1. (*2 pts.*) Submit the source code (`hw05.ipynb`) to Gradescope.

2. (*2 pts.*) Submit the two PDF images of trees that were generated by your decision-tree visualizations. Each PDF file should be named to indicate which tree it shows (the full data-set or the simplified one).

3. (*3 pts.*) Along with your code, submit a completed version of the `COLLABORATORS.txt` file. An example has been provided, which you should edit appropriately to include:

   - Your name.
   - The time it took you to compete the assignment.
   - Any resources you used to complete the assignment, including discussions with the instructor, TA's, or fellow students, and any online or offline resources consulted. If you did not need to consult any outside resources, you can say so.
   - A brief description of what parts, if any, of the assignment caused you to seek help.

[†]See the user guide for sample code: https://scikit-learn.org/stable/modules/tree.html