CAP 6610 - Machine Learning (Spring 2024)
Project 1 - Due Date: Friday, 2/23/2024 11:59 pm

For this project, you are writing Python programs to implement two decision tree classifiers, namely, Quinlan's C4.5 and Breiman's CART as defined in our lecture. Once implemented, they will be experimented on the Credit Approval dataset from the University of California Irvine Machine Learning Repository.

1 Input Dataset

We will use the UCI Credit Approval (`https://archive.ics.uci.edu/ml/datasets/Credit+Approval`) dataset for this project. This dataset concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. There is a good mix of attributes – continuous, categorical with small numbers of values, and categorical with larger numbers of values. There are also a few missing values. (Detailed information about this dataset are in the crx.names file that you may download from the link above.)

It contains about 690 examples over 15 attributes, with each example labeled either positive ("+") or negative ("-"). Among the 15 attributes, 9 of them are categorical and 6 are continuous. Therefore, your decision tree classifiers should build on both types of attributes.

In this dataset, there are 44.5% positive examples and 55.5% negative examples. For your experiments, this dataset is randomly split for you to a training set of 80% (550 examples) and a test set of the remaining 20% (140 examples). Both have similar distribution: 44% positive and 56% negative for the training set, and 45% positive and 55% negative for the test set.

Both training.data and test.data can be downloaded from Canvas and you will use them for this project.

2 Missing Values

As aforementioned, there are missing values in the dataset. In total, there are 37 examples with one or more missing values, out of which 31 are in the training set and 6 in the test set.

To handle these missing values in the training set, you will set them to the *median* value calculated in the training set. To handle those in the test set, you will set them using the same median values you obtained looking at the training set.

In case the missing value is of the categorical attribute, you still will use sort the values alphabetically before picking the median.

3 Minimum Requirements

1. The program should implement Quinlan's C4.5 and Breiman's CART decision tree learning algorithms using the template shown in dt.pdf, where the IMPORTANCE method would be based on gain ratio and Gini index, respectively.
2. Handle missing values as instructed in the last section.
3. For both C4.5 and CART, the program should implement a 10-fold cross validation process over the training set, pick the best model with the highest F_1 score, and run it finally on the test set and report the F_1 score on it.
4. Based on the results on test set, compare C4.5 and CART.
5. Note that, in order to ensure same randomness across all project submissions, the 10 folds used in the cross validation shall be done sequentially: examples 1–55 to fold 1, examples 56–110 to fold 2, . . . , and examples 496–550 to fold 10.
6. Lastly, any built-in model in *sklearn* package is **disallowed** for this project.

4 Deliverables

Zip the following to [your-last-name]_Project1.zip and submit to Canvas.

1. A directory that contains all your Python programs that are may be .py or .ipynb.
2. A README file that contains instructions to run your Python programs.
3. A PDF report that describes what your program does and explains the results on the Credit Approval dataset for both training (10-fold cross validation) and test datasets.