Machine Learning - Assignment 1 (Due: 4-Oct-2024)

This assignment consists of 4 parts: part 1 and part 2 (subdivided into parts 2.1 and 2.2) collectively account for 85% of the total grade, part 3 contributes 15/100 and part 4 is additional work specifically for the graduate students. Points are awarded only as specified. No part points will be given unless explicitly stated. Please ensure that all programming components of the assignment are completed using Python 3.

Please submit your python script (.py file) and report (PDF file) on D2L. Please note that if you submit your file in some other format besides those stated above, your mark will at most be 60%. Name your documents appropriately:
report_Firstname_LastName.pdf
script_ Firstname_LastName.py
""

For this first assignment, several blanks (____________) have been included to guide you. However, in no way these blanks are representative of the number of commands, parameters, length of what to input, or anything else. You may write as little or as much code as what is required to achieve the goals.

PART 1:
We use the 'breast cancer Wisconsin dataset' which records the measurements for breast cancer cases. There are two classes, benign and malignant. Your goal is to perform the following tasks:

- Fetch the breast cancer Wisconsin dataset and get the data features and associated target classes
- Get familiar with the data: how many instances are in the dataset? How many features describe these instances?
- Split the dataset into training and test sets: 40% of the instances should go to the test set, 60% of the instances should go to the training set. Use the holdout method to split the data and ensure that the training and test sets contain approximately the same percentage of instances of each target class.
- Build a decision tree classifier model (use the entropy criteria to split a node and ensure that only nodes with 6 or more training instances can be split).
- Verify the accuracy of the model using the test data.
- Visualize the tree that you built.

PART 2:
PART 2.1

- Visualize how the training and test errors vary as a function of the maximum depth allowed when building the decision tree.

PART 2.2

- Find the best maximum depth and the minimum number of samples to split a node when building a decision tree. For this, the training set will be subdivided into a training set and a validation set.
- Visualize the decision tree built with the best maximum depth and minimum number of samples to split a node hyperparameters.

PART 3:

The goal is to manually grow a decision tree using the provided training set in Table 1. You will compute the information gain at each step of the process using the entropy formula and decide which attribute to split the data on.

You are asked to build a decision tree with the following specifications:

- A binary tree should be created, i.e.: each node of the tree has, at most, two children.
- The classification decision at each leaf node is decided using the majority class in the class distribution of training instances reaching the leaf.

We make use of the training set from Table 1 to grow the decision tree. The instances are described by a binary attribute called ‘x1’, and an ordinal categorical attribute called ‘x2 (i.e. the values of this attribute have an order property (0<1<2)).

x1	**x2**	**y**
0	0	T
0	0	T
1	0	T
1	0	T
0	1	F
0	1	F
1	1	T
1	2	T
0	2	F
1	2	F

Table 1

Question 1: What is the entropy of this collection of training examples with respect to the target class? (3 points)

Question 2: What are the different options for the first split when constructing your decision tree? (3 points)

Question 3: For each potential first split option, compute the information gain. Only provide the results, there is no need to provide your calculations (3 points).

Question 4: Build the complete decision tree based on the given specifications and training set. The representation of the tree should adhere to the style used in the lecture notes of this course. (6 points)

PART 4:

Graduate students enrolled in CPS 8318 are required to work on a project of their own. The grade received for parts 1 and 2 will count for 80% of their total grade. Part 2 will count for the remaining 20% of the grade.

Choose a practical dataset (as opposed to the example ones we used in class) with a reasonable size from one of the following sources (other sources are also possible, e.g., Kaggle):

- UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets.php.

- KDD Cup challenges, http://www.kdd.org/kdd-cup.

Download the data, read the description, and use a decision tree classifier approach to solve a classification problem as best as you can. Write up a report of approximately 1 page, in which you briefly describe the dataset (e.g., the size – number of instances and number of attributes, what type of data, source), the problem, the approaches that you tried and the results. You can use any appropriate libraries.

Your tasks are:
1. research how to pre-process data on your own, if needed by the dataset you chose.
2. to report on which attributes are most important for your classifier
(hint: the feature that gives you the most information gain about the class labels).
3. to report on anything else inventive you can think to do, but the above 2 tasks would be enough.

Marking: 50% of the grade you will receive for part 3 will be for the write-up and 50% for the results. In the write-up, cite the sources of your data and ideas, and use your own words to express your thoughts. If you have to use someone else's words or close to them, use quotes and a citation. The citation is a number in brackets (like [1]) that refers to a similar number in the references section at the end of your paper or in a footnote, where the source is given as an author, title, URL or journal/conference/book reference. Grammar is important. Concerning the 50% for results, elaborate on what (if any) manipulations you did, what are the results for the algorithms you tried, and what else you tried.

Submit the python script (.py file(s)) with your redacted document (PDF file) on the D2L site. If the dataset is not in the public domain, you also need to submit the data file.

Ensure you did not miss any step from the .py script. Here is how Part 1 and Part 2 are graded:

Line #	Points
16	4
19	4
26	4
31	4
32	4
35	4
38	1
41	4
47	1
48	1
50	1
51	1
54	1
56	1
58	1
60	1
62	1
64	1
67	3
68	1
69	1
70	1
72	4
82	6
84	6
85	4
86	4
87	6
91	4
95	2
98	4