CSC 410: Big Data and Machine Learning, Fall 2022 Assignment 1

Assigned Monday August 29, due Sunday September 11. Max points: 100.

In this assignment, you will use and evaluate the decision tree and linear regression learning algorithms for various problems. Please (i) comment your code extensively so we will be able to read it and (ii) write the experiment report in ACM or IEEE format.

1. Learning problems

We will supply you with data for three binary classification problems. Please download the data.zip file from the Canvas system. Each problem will have the following files:

- A "problem.mat" file, which will have the examples and attributes. One attribute will be an " id" attribute which will be useful for identifying examples, but which you will not use when learning.
- A "problem.info" file, which gives additional information about the problem, such as how the data was generated. This is for your information only and does not affect the implementation in any way.

We have provided you with python code to read in this data and store it.

2. Implementation

You do not need to implement the algorithms by yourselves, but are supposed to understand the basic ideas and know how to use them. Basic Python programming skills are needed. Please use Python3, not Python2.

- Use *sklearn* as the machine learning library (It should be able to handle nominal and continuous attributes) and *NumPy* library for data representation in matrix.

- If not mentioned in the questions, the settings are default.

3. Decision Tree Learner (60 points)

1) Split each of the three datasets into training and testing subsets by the ratio 80/20.
    a. On each dataset, train the decision tree classifiers with *gini* as the node selection criteria, what is the accuracy of the classifier? If choosing *entropy* as the node selection criteria, what is the accuracy?
    b. For *voting*, what is the accuracy of the classifier when the depth is set to 1? How many leaves are in the tree? Visualize/plot the tree.
    c. For *spam*, plot the accuracy as the depth of the tree is increased. On the x-axis, choose depth values to test so there are at least five evenly spaced points. Does the accuracy improve smoothly as the depth of the tree increases? Can you explain the pattern of the graph?
2) Split the *volcanoes* dataset into training and testing subsets by the ratio 80/20, 70/30, 60/40 and 50/50, and report the accuracies. Which partition shows the highest precision?

4. Linear Regression Learner (40 points)

1) Split each of the three datasets into training and testing subsets by the ratio 80/20.
    a. What is the accuracy of Logistic Regression model on the different learning problems?
    b. Visualize/plot the Logistic Regression curve for *voting* problem. What are the upper and lower bounds of the curve?
    c. Can we use the normal Linear Regression model for the three learning problems? Explain.

4. Writeup

Prepare a writeup on your experiments by using any of the following template:

- ACM (https://www.acm.org/publications/proceedings-template/)
- IEEE (https://www.ieee.org/conferences/publishing/templates.html)

Write down any further insights or observations you made while implementing and running the algorithms, such as time and memory requirements, the complexity of the code, etc. Especially interesting insights may be awarded extra points.

You may also receive extra points for well-written code that uses good data structures and runs efficiently. Conversely, poorly written, or not following the ACM/IEEE format, or hard to understand and inefficient code will lose points.

5. What to turn in

You will turn in:

1. Your writeup, and
2. Your source code. You may include a readme if needed (e.g. if you wish to bring anything to my attention). Please ensure your code is well documented. **I will not be able to spend a lot of time debugging your code if it crashes during our testing.** Please note that I will test your code with other problems than those you have data for.

To turn in your code and writeup, use Canvas. Prepare a zip file with all your files and name it <yourname>_prog1.zip. **This zip file should only contain your writeup, source code and readme (if needed) and not executables/object files/data files/unmodified code/anything else, and must be timestamped by the due date to avoid a late penalty.**