Assignment 3 of IntroAI
Due date: Dec 6, 2022, 11:55 pm.
Weight: 8% of the final marks

Important Notes:

1. This assignment must be completed with an individual effort. You can refer resources on the web or discuss with fellow students, but the writing and answering the questions must be your own effort. Cross-checking your answers using software will be performed to detect possible cheating.
2. These questions may be somewhat similar to the potential questions in the midterm and final exams, thus careful thinking and completion of these questions would be very helpful in order to do well in exams, and to understand the materials of the course well.
3. You only need to answer 8 questions (and we will only mark the first 8, if you submit solutions to more questions). As such, we may add more questions after the assignment is posted to give you more selection. **Please specify which questions you are answering in your submission.**

Total 8%. Each question is 1% (unless otherwise noted)

The following questions are drawn from the questions in Chapter 18 of the textbook. For some questions, you may need to make some realistic assumptions. Write down your assumptions clearly.

18.1

18.2

18.3

18.6. Note: often you do not need to use log to determine which node has the maximum information gain; often intuition is sufficient. (In the assignment and exams, plug the numbers in the formula, and you may use intuition to determine which attribute has the maximum info gain to be selected as the root of the subtree). Add: use the tree built to predict a new (test) example with values of A1-A3 to be 0, 1, 1, respectively. What class (y) the tree would predict?

18.6 continued: Apply k-NN algorithm with k=1 and k=3 to predict a new example with values of A1-A3 to be 0, 1, 1, respectively? If there is a tie, break the tie randomly.

18.6 continued: Apply the Naive Bayes learning algorithm using this small dataset on a new example with values of A1-A3 to be 0, 1, 1, respectively? When the probability is 0, use a small number (such as 0.05) to substitute 0. (If we don't have time to cover Naive Bayes, you can search the Web - it is a very simple algorithm).

18.6 continued: Treating the output y as another attribute (A4), use k-means on this small dataset with attributes A1-A4 with k=2. Iterate the algorithm for at least 3 steps to see if it is converged.

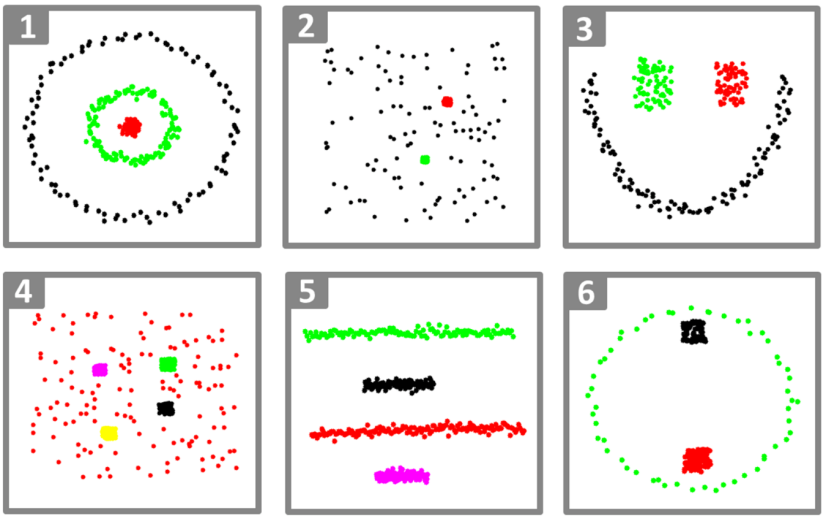
18.6 continued: Continue from the question directly above, use the Agglomerative Clustering algorithm to make clusters of data. Show the dendrogram produced.

18.7

18.12

18.19

The following questions use this figure below:



Programming with supervised learning (2% - counted as two 1% questions):
Generate simulated data similar to the 6 figures above with different colors indicating different classes, as "ground truth". As the data is 2-D, visualize the data to make sure they are approximating the figures. Randomly generate and sample the data to form the training set (of various sizes), validation set, and test set. As data are "freely" generated, your validation and test sets can be large to get reliable estimates of predictive accuracy. Program the k-NN algorithm and use cross-validation to find the best k for each problem (best k as producing smallest predictive accuracy on the validation set). Report the test accuracy for such k-NN as the most reliable estimate of the predictive accuracy of your model for the future unseen test data.

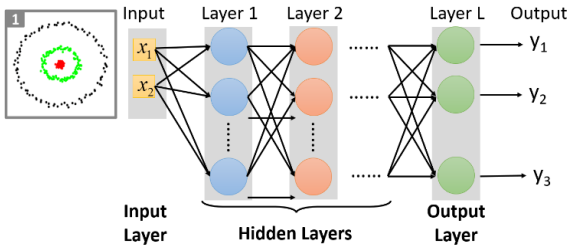
Comparing your k-NN above with decision tree learning algorithm (2%):
A learning algorithm is usually called "better", if with the training data of the same size (say 10, 100, 1000, etc), algorithm A predicts more accurately than algorithm B, on the test data. You can use existing libraries or open source codes (such as here), or write your own codes, to apply the decision algorithm on the data generated above. Compare the decision tree and k-NN to see which one performs better on each of the 6 problems.

Question on clustering (2%):
For the 6 problems above (note: the colors indicate the intended clustering results; the data have no labels or cluster indicators to begin with. There is no "ground truth" for the clustering problem), program k-Means and the Agglomerative algorithms to see what clusters will be produced. Explain in what cases the intended clusters (as colored) may be produced. (Of course in high-dimensional complex data, the "intended clusters" are unknown to us).

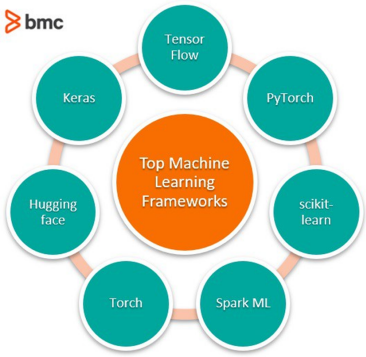
Question on noisy data (2%):
This question asks you to study the behavior of supervised learning algorithms when the data contain the so-called "noise" (sometimes called outlier or anomaly). Choose two problems (from the 6 problems

above), and generate the “ground truth” data with various kinds and levels of “noise” (for example, with probability of 5%, 10%, 20%, the label is modified). Apply decision tree and k-NN on the training data (which contain “noise”). Observe if “overfitting” happens. (Overfitting is a phenomenon where the model is trained to have very small (0%) training error but the test error actually increases). Try to use the validation set (which also contains “noise”) to choose the best hyper-parameters (such as pruning levels for decision trees, k for k-NN). Report the test accuracy for such models as the most reliable estimate of the predictive accuracy of your model for the future unseen test data.

Question on (deep) neural networks (2%):



Choose a dataset such as the one above, and use an existing deep neural network framework or library to build and train a (deep) neural network for this classification task. You can try different numbers of hidden layers, and different numbers of units in each hidden layer. They are the hyper-parameters of the neural networks. Usually the more “complex” the pattern to be learned, the more layers and more units would be needed, but there is no “formula” for the right or best answers. One good news is that researchers often find that large deep neural networks do not “overfit” the data often (unlike DT and k-NN) so you can try a few large neural networks. You can try to see if “overfitting” happens in your networks with validation sets, and report final test accuracy on the test set. This question is very open-ended. You can use any tool you find suitable, including the following most popular deep learning frameworks.



Report
produce the highest accuracy on the test set (which also contains “noise”). Explain your results.

To submit: submit your answer in one PDF file via OWL. You could include drawing and handwriting etc in the file but make sure they must be clearly viewable or marks can be deducted.