CS434 Assignment 2
Due: Oct 18\textsuperscript{th} in class

\textit{Submission instruction:}

Please submit through TEACH both your code and report. Your submission should include both the source code (a readme file for how to run) and your report. You need to form a team of up to \textbf{three} students to work on this part of the assignment. There is no specific requirement on which programming language you use, but your code should provide comments and be easily understandable. Writing clean and easy to understand code will be rewarded.

The dataset provided for Part I constitutes of 30 features and is a matrix of Nx31 dimension. The first column of the test and train data is the true class label of the samples and the following 30 columns are the 30 features. This dataset belongs to Wisconsin Diagnostic Breast Cancer and classes are Diagnosis (+1= malignant and -1= benign).

\textbf{Part I.} The task is to implement the \textit{K-nearest neighbor} algorithm. In particular, you need to:

- Perform model selection using \textit{leave-one-out cross-validation} to select the best $K$ for the given learning task using the provided training data. Please consider the following range of $K$ values: 1, 3, 5, \ldots, 15. (This is a suggested range, feel free to explore more possible $k$ values). What is your choice of $K$?
- For each possible value of $K$, please compute the following: 1) training error 2) \textit{leave-one-out cross-validation error} on the training set; and 3) testing error on the provided test data. Plot these errors as a function of $K$.
- Discuss what you observe as the relationship between these three different errors.

\textbf{Part II.} Implement the top-down greedy induction algorithm for learning decision trees. In particular, given a set of training examples your implementation should learn: (1) a decision stump, i.e. a decision tree with only a single test; and (2) a fully grown decision tree. Please Use information gain as the selection criterion for building the decision tree. Test your implementation on the monk data set that is provided to you.

The original monk dataset is a synthetically generated dataset that was created for a comparison study on different machine learning algorithms. For those of you who are interested, this dataset (and two other related data sets), and the comparison results are described in the following paper:

The MONK’s Problem – A performance comparison of different learning algorithms by a group of machine learning researchers.
The particular dataset provided for this implementation assignment is the a revised monk 1 dataset. There are 6 features, the meanings of which are provided below:

\[
\begin{align*}
    x_1 & : \text{head\_shape} & \in & \text{round, square, octagon} \\
    x_2 & : \text{body\_shape} & \in & \text{round, square, octagon} \\
    x_3 & : \text{is\_smiling} & \in & \text{yes, nc} \\
    x_4 & : \text{holding} & \in & \text{sword, balloon, flag} \\
    x_5 & : \text{jacket\_color} & \in & \text{red, yellow, green, blue} \\
    x_6 & : \text{has\_tie} & \in & \text{yes, nc}
\end{align*}
\]

Note that the provided data use different numerical values to represent the different feature values. For example, feature 1 in the data has three different values 1, 2 and 3, representing ‘round’, ‘square’, and ‘octagon’ respectively. For this assignment, please do not worry about the bias for preferring multi-nominal features (features that when tested creates more than two branches) and just use the simple information gain (instead of the normalized gain ratio) to evaluate all features.

For this particular data set, the class label is a concept defined by the following rule:

\[(\text{has\_tie} = \text{yes}) \& (\text{head\_shape} \neq \text{body\_shape})\]

That is if \(X_6 = 1\) and \(X_1 \neq X_2\), then \(y = 1\) (positive), otherwise \(y = 0\) (negative).

In your report, please provide

1) Both the learned stump and the learned decision tree. To help grading easier, please provide for each selected test its information gain.
2) The training and testing error rate of the learned stump and full decision tree.

Finally, please answer the following questions:

1. Given the formula for generating the class label, please provide a (as compact as possible) decision tree that will correctly classify all training examples.
2. Do you expect the top-down greedy information algorithm to learn this optimal tree? Note that your answer should be general, not depending on the specific training set you use in this assignment.

Note that you could produce a decision stump by pruning the fully learned decision tree and create the leave nodes under the root test. However, I strongly suggest you to create a separate function that learns decision stump directly. This is because this function will be used as a base function for future assignments.