Machine Problem 1: Decision Tree with Categorical and Real Valued Attributes

In this machine problem, you will implement a decision tree to classify records with both categorical and real valued attributes.

**Step 1: Choose a dataset from UCI repository.**
A collection of datasets can be downloaded at [https://archive.ics.uci.edu/ml/datasets.html](https://archive.ics.uci.edu/ml/datasets.html). You need to choose a dataset for classification task with both categorical and real attributes, e.g., Abalone dataset. You can also use whatever dataset in your research with mixed types of attributes.

**Step 2: Split the dataset into training and test sets.**
You need to split the dataset based on K-fold cross validation (K must be no smaller than 2).

**Step 3: Implement a decision tree.**
Starting from a root node, you will gradually grow a decision tree by choosing an attribute at each node. The attribute can be randomly chosen or a best attribute with a largest reduction in impurity or training error is chosen by a brute-force trial of all candidate attributes. Each attribute can be chosen only one time through the tree, or multiple times, at your own discretion. A stop criterion can be based on a predefined threshold of tree depth or a satisfactory level of training error, or combined.

**Step 4: Report the performance.**
In each fold of cross-validation, report the classification accuracy, precision, recall and F1 measure on the test set. Finally, we will get an average and standard deviation of these metrics over K-fold.

**Step 5: Write the report.**
Your report should contain detailed explanation of each above step. Finally, you should analyze if you observe any overfitting of decision tree, as well as what causes the overfitting, too deep tree or over-reduced training error?

Submit your report and source code to cap6676ucf@gmail.com by October 10th. You can use whatever programming language for implementation.

You might need the following result to split the real attribute at a decision tree node.

**Appendix: A closed form solution to the optimal split point for real attributes**

Suppose you have a training subset reaching at a node of decision tree, denoted by \( \{(x_i, y_i)\}_{i = 1}^n \).
where \( x_i \) is the value of a real attribute for record \( i \), and \( y_i \in \{+1, -1\} \) is a binary class label for the corresponding record. Then our goal is to find a classifier which can separate two classes on the real axis of the attribute as much as possible. For this purpose, we define a linear classifier \( f(x) = wx + b \) as a mapping from attribute to class labels. We fit this function to the given training subset by least square error, i.e.,

\[
\min_{(w,b)} \sum_i (f(x_i) - y_i)^2
\]

Or equivalently in a vector format,

\[(wx + b \mathbf{1} - \mathbf{y})^T (wx + b \mathbf{1} - \mathbf{y})\]

Where \( x \) is the vector of all attributes in the training subset, and \( y \) is the corresponding vector of class labels. Take derivative to \( w \) and \( b \), and set the result to zero, we have

\[
(wx + b \mathbf{1} - \mathbf{y})^T x = 0
\]

\[
(wx + b \mathbf{1} - \mathbf{y})^T \mathbf{1} = 0
\]

By solving the above linear equations, we solve \( w \) and \( b \). Then the optimal split point can be set to \( f(x_{\text{split}}) = wx_{\text{split}} + b = 0 \), i.e., \( x_{\text{split}} = -\frac{b}{w} \).

Note that: the above linear classifier is found by minimizing the regression error, i.e., the deviation between the predicted value of \( f \) and the true label. However, in classification task, we do not care about the predicted value. Instead we only care about the indicated label from the prediction, i.e., the sign of linear function suggests the class label of the record. In this sense, directly minimizing the regression error is not the best choice for classification task.