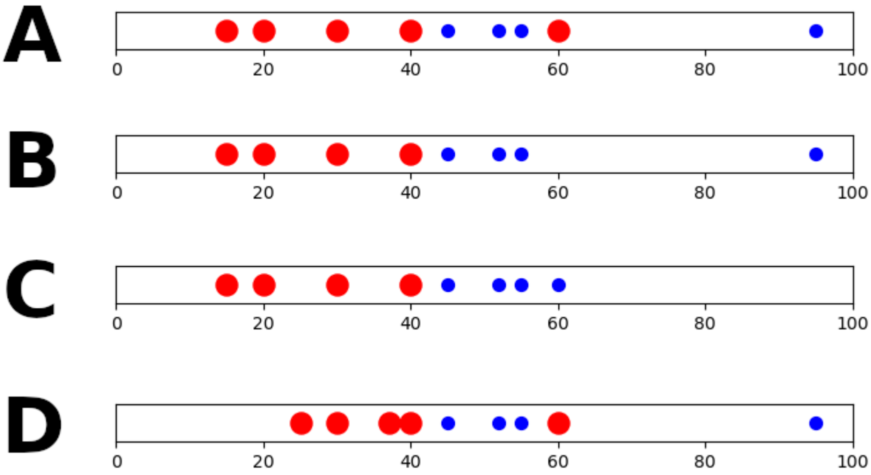
CS5228 – Tutorial 6

Classification & Regression II (Tree-Based Models)

1. **Effects of data distribution on Decision Trees.** The figure below shows 4 distributions of the values for a single feature. The color and shape of the dots reflect the class label. Since we only have two colors/sizes, the example application is a binary classification task.



(a) Assume a Decision Tree classifier that only performs binary splits. For each data distribution A-D, where (approximately) would the classifier split the values into 2 child nodes?

Solution:

- For all for distributions, the threshold for the best split would be around 42.5

(b) Let's assume our data points only have this one feature. Just by looking at the data distributions A-D for this single feature, what can we say about the "look" of the final decision tree (without any pre or post-pruning)?

Solution:

- Distributions B and C require only this single split to yield child nodes without any impurity. So for this branch of the Decision Tree, the learning algorithm stops.

- Distributions A and D require both 2 additional splits to ensure nodes without any impurity.

Note that this result might vary when considering more than this single feature.

(c) Given the results from (a) and (b), summarize how the distribution of feature values affects the training of a Decision Tree.

Solution:

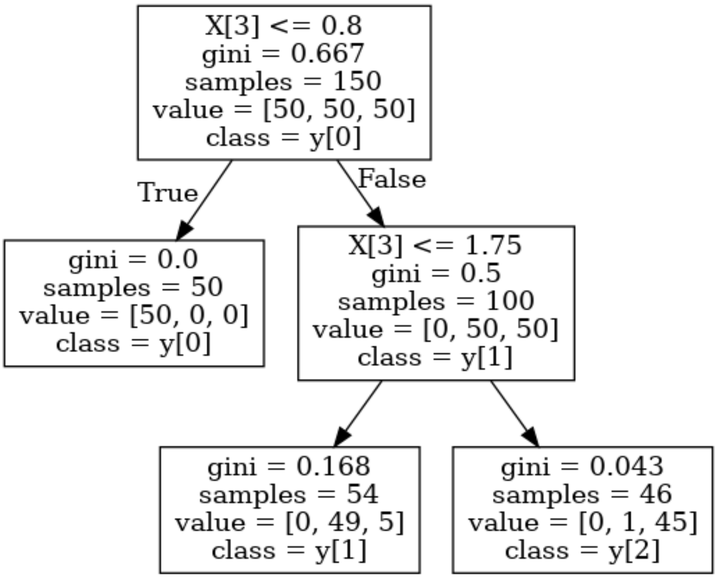
- As long as an outlier does not affect the "order" of data points with respect to their class labels, the outlier won't affect the split regarding the resulting child nodes. So in this regard, Decision Trees are somewhat robust against outliers.
- Even if the order of data points w.r.t. their class labels is preserved, the exact threshold where to perform the split still vary, of course
- Distributions A and require both 2 additional splits to ensure nodes without any impurity.
- If an outlier does affect the order data points w.r.t. their class labels, then this will generally result in a different split. And since single Decision Trees (not Tree Ensembles) are quite sensitive to changes in the dataset, such outliers do have an impact on the resulting Decision Tree.

(d) You can now add a single data point of Class **Blue** into Distribution A? Where would you place this new data point to maximize the negative effect on the resulting Decision Tree in terms of the required splits?

Solution:

- Placing the new data point of Class **Blue** between two data Points of Class **Red** would require 2 additional splits. Any other placement only in 1 split at worst (e.g., at value 5).
- An interesting special case is to place the new data point of Class **Blue** directly on top of an existing data point of Class **Red**. This would not result in more than 2 additional splits, but it would mean that the Decision Tree can no longer perfectly fit the data any longer.

2. **Interpreting Decision Trees.** The Decision Tree shown below has been trained over the IRIS Dataset with a maximum depth of 2. Recall that each data sample has 4 numerical features (all measures in centimeters), and is labeled with 1 out of 3 classes.



(a) What insights can you get from this Decision Tree?

Solution:

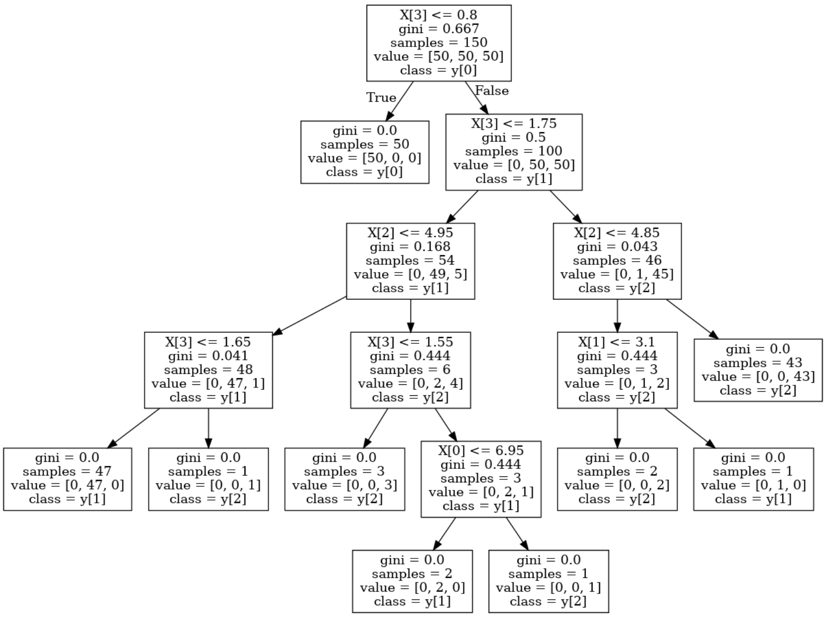
- Feature 3 is chosen for the first split at the root, it is thus the strongest predictor
- The samples of Class 0 are easiest to identify since the root split is sufficient to separate them from the samples of the other 2 classes.
- Side note: This example shows that a child may be split using the same feature as the parent node.

(b) Assume you train the Decision Tree without any restrictions on its maximum depth. Given the Decision Tree above, which statement can you make about the full Decision Tree.

Solution:

- The first 2 splits will naturally be identical since the specification of a maximum depth does not affect the algorithm for finding the best split.
- The full Decision tree will be deeper/higher since the child nodes of the ”restricted” tree above are still impure.
- It’s not really possible to tell how many more splits are needed as it heavily depends on the data distribution which is not sufficiently expressed in the Decision Tree.

- At the very minimum, 2 more splits are required to separate the data samples of 2 impure child nodes in the restricted tree.
- In the worst case, for each impure child node, O(N) more splits are needed where N is the number of data samples in that child node
- The Decision Tree below shows to full Decision Tree trained over the IRIS dataset


X[3] <= 0.8 gini = 0.667 samples = 150 value = [50, 50, 50] class = y[0]
True
False
gini = 0.0 samples = 50 value = [50, 0, 0] class = y[0]
X[3] <= 1.75 gini = 0.5 samples = 100 value = [0, 50, 50] class = y[1]
X[2] <= 4.95 gini = 0.168 samples = 54 value = [0, 49, 5] class = y[1]
X[2] <= 4.85 gini = 0.043 samples = 46 value = [0, 1, 45] class = y[2]
X[3] <= 1.65 gini = 0.041 samples = 48 value = [0, 47, 1] class = y[1]
X[3] <= 1.55 gini = 0.444 samples = 6 value = [0, 2, 4] class = y[2]
X[1] <= 3.1 gini = 0.444 samples = 3 value = [0, 1, 2] class = y[2]
gini = 0.0 samples = 43 value = [0, 0, 43] class = y[2]
gini = 0.0 samples = 47 value = [0, 47, 0] class = y[1]
gini = 0.0 samples = 1 value = [0, 0, 1] class = y[2]
gini = 0.0 samples = 3 value = [0, 0, 3] class = y[2]
X[0] <= 6.95 gini = 0.444 samples = 3 value = [0, 2, 1] class = y[1]
gini = 0.0 samples = 2 value = [0, 0, 2] class = y[2]
gini = 0.0 samples = 1 value = [0, 1, 0] class = y[1]
gini = 0.0 samples = 2 value = [0, 2, 0] class = y[1]
gini = 0.0 samples = 1 value = [0, 0, 1] class = y[2]

(c) How would your answer for (a) and (b) change if all the input features would have been standardized before training the Decision Tree.

Solution:

- Apart from the specific thresholds, nothing would change and the Decision Trees would look the same.
- Decision Trees do not take the interaction between features into account, so any form of normalization/standardization to make features "equally important" are not needed.
- Standardization does not affect the "order" of data samples with respect to their values, which in turn does not affect the algorithm for finding the best split (cf. Task 1)

(d) Assume someone gives you the optimal Decision Tree, i.e., optimal in the sense that it results in the highest accuracy (or any other suitable metric). What can we say about the root node of this optimal Decision Tree?

Solution:

- As the training algorithm for a Decision Tree is a heuristic, there is a non-zero probability that the optimal Decision Tree performs the first split not w.r.t. Feature 1.
- However, since Feature 3 is such a strong predictor, the probability that Feature 3 is used for the first split in the optimal Decision Tree is arguably very high.