# Intro to decision trees

Decision trees ask a question, then classify some value in a way. The classification could be categorical, or numeric by saying its in the range of or something. Order of questions of the values in questions on either side of a tree don’t have to be equal. The top of the tree is the root node, internal nodes are the nodes that ask some question, leaf nodes are the nodes that have some answer or classification.

To find the first variable we look at how each variable separates the different data points. Basically we want to find which variable separates things the best. The root node should be what separates the data best.

Diagram

Description automatically generated

A screenshot of a phone

Description automatically generated with low confidence

Impure leaf nodes mean that data from more than one classification ends up in them. In the heart disease example if the person does have chest pain that leaf node is impure because there are 105 yes and 39 no with respect to actually having heart disease who have chest pains. We want to measure the impurity, one way to do this is with a Gini coefficient.

Gini impurity = 1 – (probability of yes)^2 – (probability of no)^2

Ie heart disease = 1 – ( 105 / (105 + 39))^2 – ( 39 / ( 105 + 39 ))^2 = 0.395

Then to calculate gini impurity of that variable we take the fraction of people in leaf node 1 times the gini impurity of leaf node 1 + the fraction of people in leaf node 2 times the gini impurity of leaf node 2

To calculate latter nodes we would then take all the data points that have ended up in this leaf node and all the variables that have yet to be considered, we would select the one with the lowest weighted average gini impurity (aka the one that gives us the best sperability) and add it to the tree in this position.

We also want to check that the gini impurity of adding another decision onto that branch has a lower gini coefficient than the existing one on that branch. If the gini is not lower then this node becomes a leaf node that stops separating data.

Diagram, timeline

Description automatically generated

For continuous variable

Sort the data by value, lowest to highest. Then calculate the average weight for all adjacent data. Then calculate the impurity value for all the average values. The lowest impurity occurs at the best cutoff value to use.

For ranked data we calculate impurity scores for all possible ranks.

For multiple choice data we calculate an impurity score for all combinations of the data. If the data was color choice of r g or b then we have 6 options as we have all the individual choices or the situations of either red or blue, red or green, blue or green.

# Feature selection or missing data

If the impurity is lower without separation, we might make it a leaf node.

In a decision tree if a variable never reduces the impurity it will never be used in a tree and therefore not be needed in our algorithm. We could also make a threshold so that the impurity reduction must be large enough be considered in the tree. This helps in the prevention of overfitting.

For missing data we could find a column with the highest correlation and then use that to determine what value to fill the missing value with. If the data was continuous we could use the correlation table as well, then we could do a linear regression on the two continuous variables to fill the value. If the other column is discrete we could use the mean of that value.