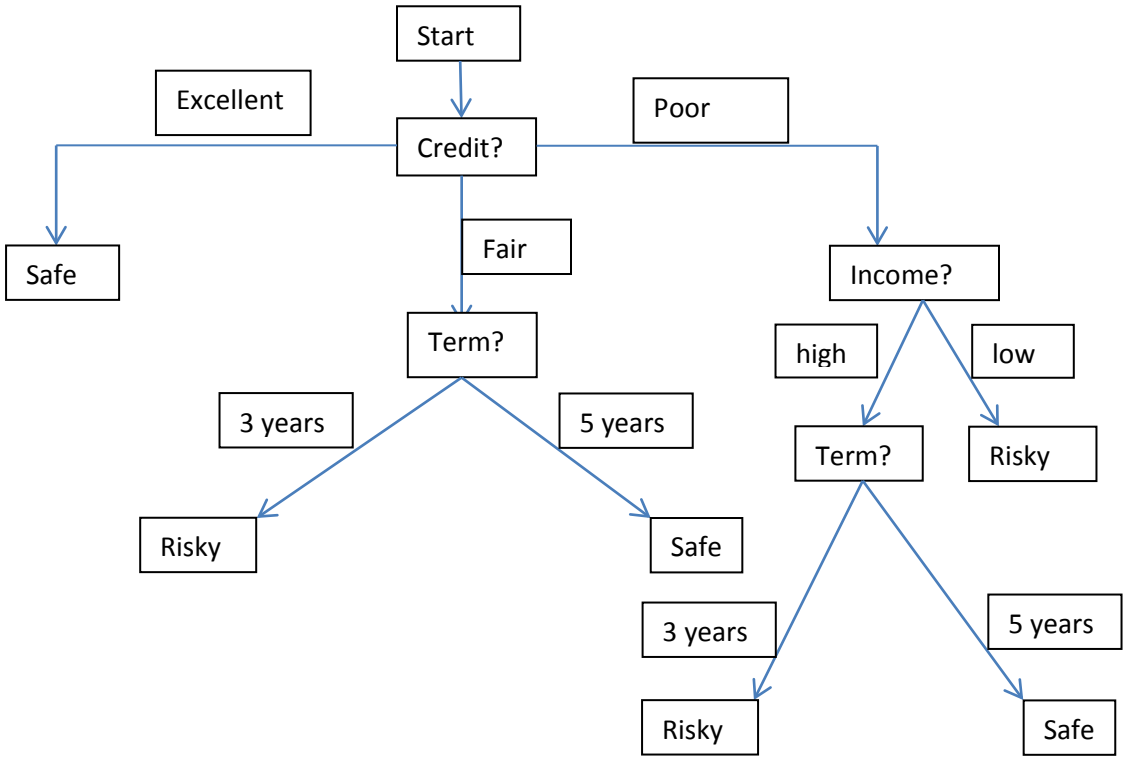
- Decision Trees



- Quality metric: Classification error

$$Error = \frac{\#\ incorrect\ predictions}{\#\ examples}$$

- Selecting what features to come next in an order can be a really hard problems and lead to explanation in many combinations (NP-hard problem).

- A high-level outline of the learning algorithm for decision trees:

Step 1: Start with an empty tree.

Step 2: Split on a feature.

Step 3: Make predictions.

Step 4: Recursion.

- Greedy decision tree learning

Step 1: Start with an empty tree.

Step 2: Select a feature to split data.

For each split of the tree:

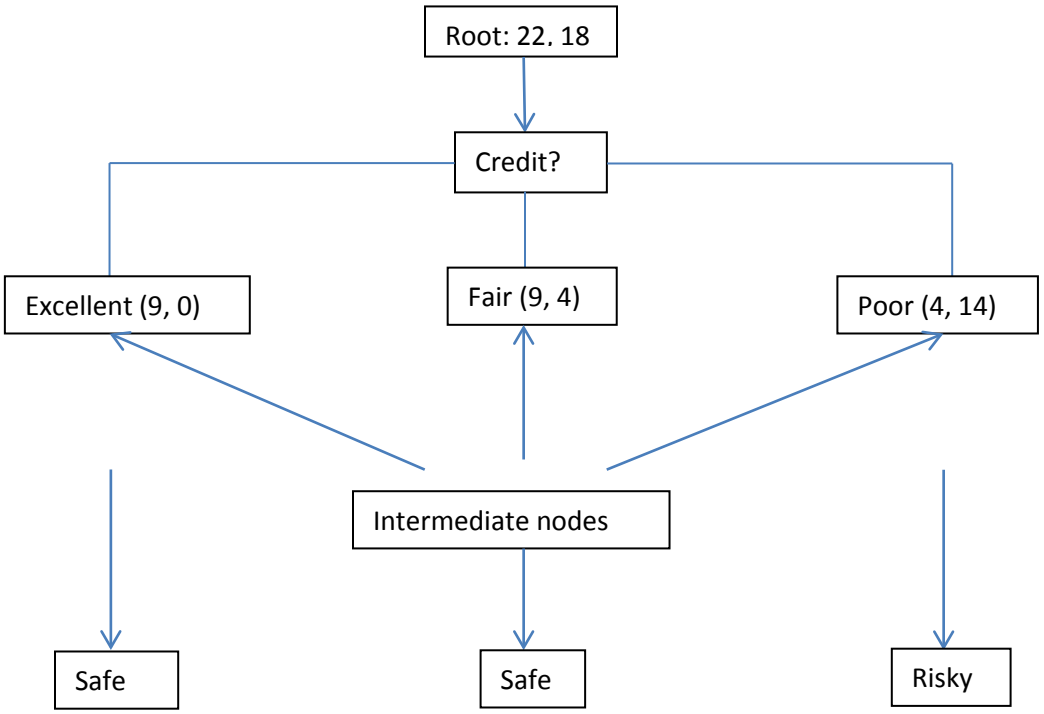
Step 3: If nothing more to split, make predictions.

Step 4: Otherwise, go to Step 2 and continue (recurse) on this split.

- What feature to split on?

Feature split learning = Decision stump learning

- Visual notation: Intermediate nodes



For each intermediate node, set $\hat{y} = majority\ value$

- How do we select the best feature?

Intuitively, a better split is one that gives you the lowest classification error.

- Calculating classification error

Step 1: $\hat{y} = class\ of\ majority\ of\ data\ in\ node.$

Step 2: Calculate classification error of predicting $\hat{y}$ for this data.

- Feature split selection algorithm

- Given a subset of data M (a node in a tree)

- For each feature $h_i(x)$:

1) Split data of M according to feature $h_i(x)$;

2) Compute classification error split.

- Choose feature $h^*(x)$ with lowest classification error.

- When to stop recursing:

- Stopping condition 1: All data agrees on y.

- Stopping condition 2: Already split on all features.

- Decision tree prediction algorithm.

- Predict (tree_node, input)

If current tree_node is a leaf:

  Return majority class of data points in leaf

Else:

  Next_node = child node of tree_node whose feature value agrees with input

  Return predict (next_node, input)