# Decision Trees

### OBJECTIVES

By the end of this lesson, you will have bragging rights for:

- Describing decision trees
- Explaining how decision trees work, for both classification and regression
- Visually representing a decision tree

### WHAT IS A DECISION TREE?

Tree-like models of decisions that can be used for both **classification** and **regression** purposes in Machine Learning. This means that we can use decision trees for data containing either continuous or categorical variables, or both.

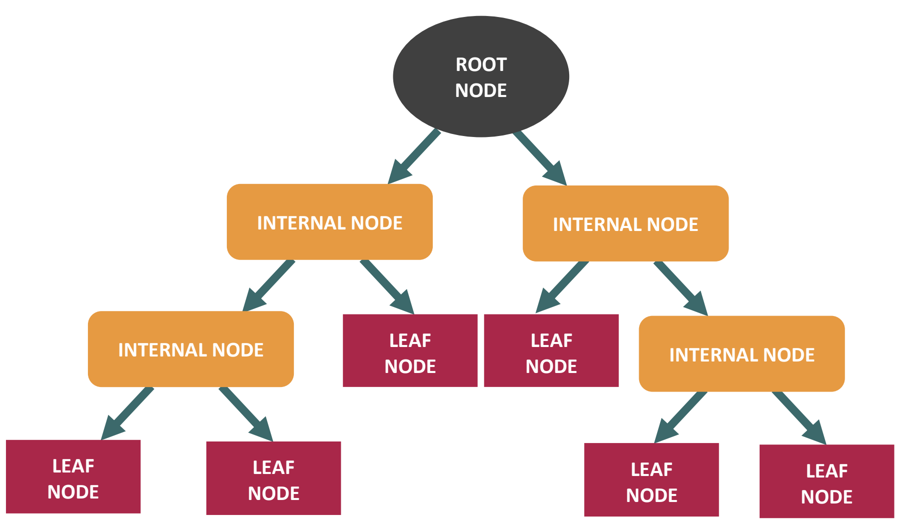
Decision trees are:

- *Non-parametric*: unlike OLS and logistic regressions, decision trees do not require either parameter for the model or underlying assumptions of the distribution of the data and its errors. A decision trees model is constructed based on the observed data.
- *Hierarchical*: the model is defined by a series of questions that eventually lead to a target label for each observation used

### COMPONENTS OF A DECISION TREE

**<u>NOTE</u>:** To better understand these components, think of decision trees as upside-down trees.

- *Branches*: the arrows connecting the nodes from the top to the bottom of the tree
- *Root node*: the starting point of the tree from the top. It contains questions that need to be answered. It had no incoming branch, but two or more outgoing branches.
- *Internal nodes*: also contain additional questions that follow those asked in the root/previous node. They have one incoming branch and split into two or more outgoing branches
- *Leaf nodes*: contain a target label. They have one incoming branch but no outgoing branches. Like an actual tree, the leaf nodes are always at the end of the branch.



## HOW DO WE GROW A DECISION TREE?

**Example:**

The dating scene in 2018 has been quite a struggle. People don't get you no matter how hard you try. You decided to do what any regular person would do: join Tinder! After a month of many left swipes and a few unsuccessful right swipes, you began to think something was wrong. You started to that maybe you're swiping right for the wrong reasons and decide to embark on a soul-searching journey of what makes you swipe right to prospects. Tinder wants to help you with your dating life, so they offered to give you data on your activities on the app for the last month and the profile of each prospect.

You have the following variables in your data set:

- Cute: whether or not you thought the prospect was cute (1: Yes, 0: No)[1]
- Job: whether or not the prospect filled in the job section in their profile (1: Yes, 0: No)
- Smoker: whether or not the subject is a smoker
- Swipe: which direction you swiped (1: Right, 0: No). *This the target variable*.

To grow an efficient decision tree, we would need a **greedy algorithm** that is able to efficiently select an optimal feature for each node.[2] The algorithm goes through the following steps:

- For the root node, it considers each feature in the data set as a candidate and divides them into smaller groups (which go into the internal nodes) based on their splits
- It calculates the cost on accuracy for each split and selects the feature with the least cost in accuracy.
  - For regression, cost is calculated by summing the difference between the actual target and predicted target for each observation (i.e. Tinder prospect)
  - For classification, we use the **Gini Index** to measure **purity**. A node is 100% pure if all the observations in the node have the same target value. The algorithm will select the feature that produces the lowest Gini Index.

$$\text{Gini Index} = \mathbf{1 - \sum_{i=1}^{J} p_i^2}$$ where $J$ is the number of classes and $p_i$ is fraction of items in class $i$.

- It does this recursively, further creating smaller and smaller nodes by splitting the previous node

**<u>NOTE</u>:** This means that the feature in the root node is the best predictor for the target variable.

## HOW DO WE MAKE THE TREE STOP GROWING?

In our example, we have just three features for the decision tree. In practice, we are more likely to have more complex data sets with a large set of features. Complex trees (that cause overfitting) are more likely to occur if we allow the tree to grow continuously. We can stop the tree from growing by setting one or more of the following constraints in the algorithm:

- Set a minimum number of observations a leaf node needs
- Set a maximum depth for the tree (i.e. the longest path from the root node to the leaf node)

## ADDITIONAL RESOURCE

- http://www.jmlr.org/papers/volume11/ben-haim10a/ben-haim10a.pdf
- https://www.hackerearth.com/practice/algorithms/greedy/basics-of-greedy-algorithms/tutorial/

---

[1] For this example, assume that you also have data on whether or not your prospects are cute.

[2] Refer to the slides from the lecture for a visual representation of the decision tree.