Content Learning Objectives

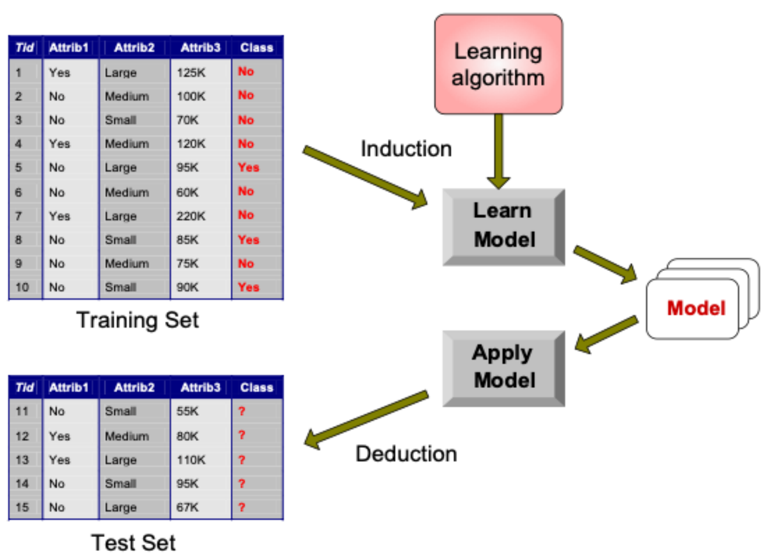
After completing this activity, students should be able to:

- Define the task of classification
- Define and calculate entropy and information gain
- Compute performance of a classifier and test for overfitting
- Compare classifier performance

Activity 1 : Supervised Learning

Given a set of example data, which we will call *training data*, the goal is to learn a function/model that will compute a target value for new data. This is known as **supervised** learning. An example of supervised learning is predicting the sales price of a home. Example data for this task is shown in the table below.

SQFT	# of bedrooms	zip code	Sales price
1000	2	22999	160,000
2000	4	20111	450,000
2200	3	90210	850,000



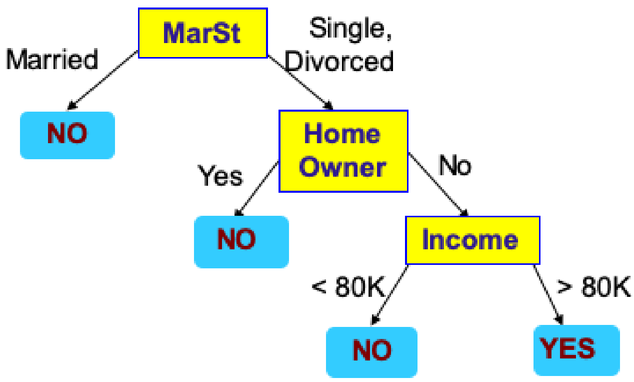
Introduction to Data Mining. Pang-Ning Tan et. al. 2019.

When the predicted value is a real number (sales price), we call this *regression*. When the predicted value is a label, this is called *classification*, for example, identifying an email as **spam** or **not spam**. When only two labels are possible (as is the case with spam), this is called *binary classification*. A general approach to classification to shown on the left.

1. List 3 possible uses of **regression**.

2. List 3 possible uses of **classification**.

Activity 2 : Tree Structures for Classification



A tree structure can be created for classification. For example, the tree on the left is used to predict whether someone who borrows money from a bank will default on the loan.

3. Use the tree to classify a newly arrived example with the following features:

   - Homeowner = no
   - Marital Status = married
   - income = 80K

   Trace the pathway through the tree. What label (yes or no) does the tree assign for defaulted borrow?

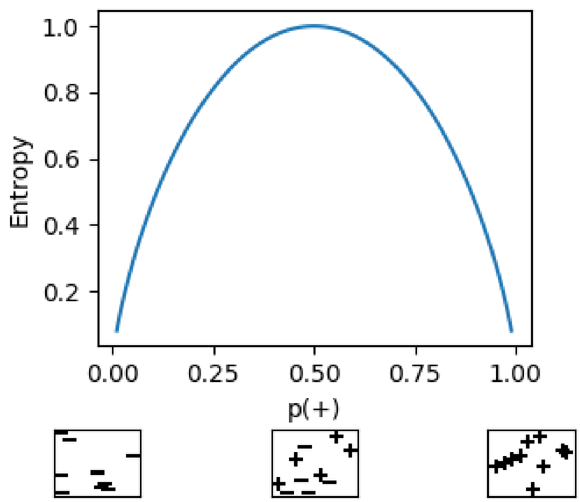
Decision Tree Construction

This type of classifier is known as a **decision tree**. How do we build this structure from training data? The goal is to build a tree that both labels the training data accurately and also performs well on data that the model has not seen. Performing well on data that was not used to build the model is important, and is known as **generalization**.

There are usually many possible trees that will perfectly classify the training data. It is impossible to know, in advance, which tree will have the best performance on unseen data points. A valuable principle here, and throughout machine learning, is **Occam's Razor**, the principle that, all else being equal, the simpler explanation is likely to be best. In the case of decision trees, the "simpler explanation" corresponds to *smaller* trees, either in terms of height, or the total number of nodes.

You may not be surprised to learn that the problem of building an optimal decision tree is NP-Complete. Following a common theme, we will build a high-quality tree using a greedy approach known as **Hunt's Algorithm**. This approach works by first grouping all the data into a single node. If this node contains only one class label, we are done. Otherwise, it iterates through each feature to see how well that feature splits/partitions the data, and picks the best feature. Child nodes are created using that partition, and the algorithm is applied recursively to those nodes. There is no guarantee that this greedy strategy will result in an optimal tree, but in practice, it works well.

Activity 3 : Decision Boundaries for Decision Trees



The greedy part of this algorithm is selecting the "best" split for partitioning the data into nodes that are as homogenous as possible. One measure of homogenaity is *entropy*. Entropy is defined as follows:

$$-\sum_{i=0}^{c-1}\Big(p_i(t)\ \log_2\ (p_i(t))\Big) \qquad (1)$$

Where $p_i(t)$ is the relative frequency of class i in node t. The figure on the left shows how entropy changes based on a binary classification problem of $+$ and $-$.

4. For a binary classifier, calculate the entropy of the SET (contents of a node) where 13 examples are of the class "+" and 20 examples are of class "−".

Activity 4 : Define how to Partition/Split Data

Following the example from out textbook, our dataset has the following features.

- Homeowner (domain = {True, False})
- Married (domain = {Single, Married, Divorced})
- Income (domain = $[0,\infty]$)
- Defaulted borrower (domain = {Yes, No})

First, we define how we will split data. Next, we will evaluate each split.

For **nominal types**, we can evaluate a multiway split (each nominal value has its own child) or make binary splits. **Ordinal** values can also be split this way, but care should be taken to maintain the order property (one split could be "small, medium" and another "large", but you would avoid splits of "small, large" and then medium).

Continuous attributes are usually handled by taking adjacent attributes, computing their midpoint, and using that as the split point. For each such position, the entropy can be computed and the split with the *lowest* weighted entropy is selected.

5. Identify the type of each attribute in the above description (write your answer to the right of each feature).

The goal of a partitions/split is to maximum the difference in homogeneity between the parent node and its children. This is known as *gain*, or in the case of entropy, *information gain*.

$$Gain(Split) = Entropy(Parent) - \sum \frac{N(v_j)}{N} Entropy(Child) \tag{2}$$

Where $N(v_j)$ is the number of training examples within a child node and N is the is the number of examples in the parent node that is being split. Each entropy calculation is *weighted* by the number of examples it contains (and the weights sum to 1). This gain is sometimes notated by Δ_{info}.

Here is the full training set:

Home Owner	Martial Status	Annual Income	Defaulted Borrower
Yes	Single	120,000	No
No	Married	100,000	No
Yes	Single	70,000	No
No	Single	150,000	Yes
Yes	Divorced	85,000	No
No	Married	80,000	Yes
No	Single	75,000	Yes

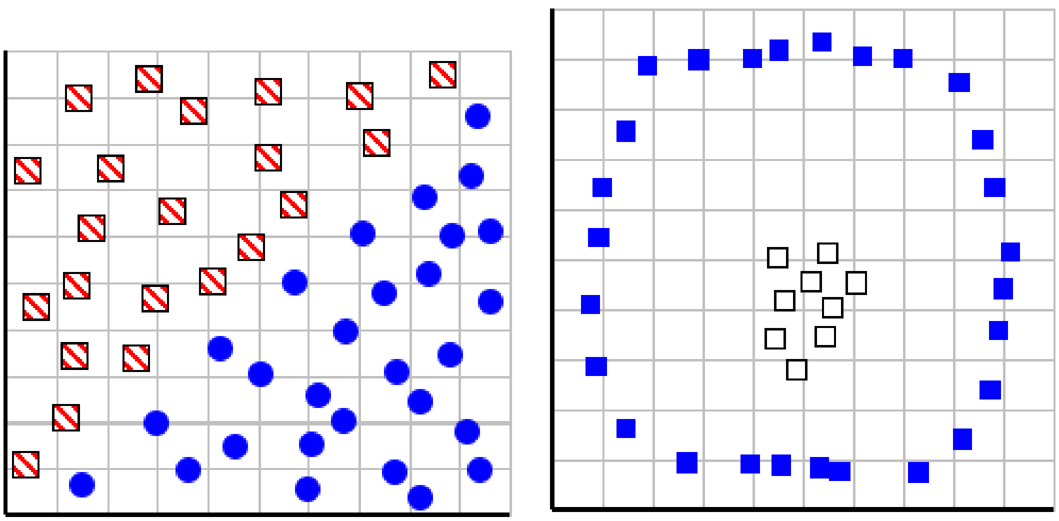
6. For the example data provided, identify the best first split point (from a greedy perspective) and calculate the information gain from the root node (all the data) to the child nodes with respect to the split point. Show your work for each split point and identify the best split. The table below is provided to help you keep track of the possible split points for the Annual Income attribute. For the purposes of this example, assume a three-way split for the Marital Status attribute.

Class	No	Yes	Yes	No	No	No	Yes
	Annual Income (In thousands)						
	70	75	80	85	100	120	150

Split Points	72.5		77.5		82.5		92.5		110		135	
	$\leq$	$>$	$\leq$	$>$	$\leq$	$>$	$\leq$	$>$	$\leq$	$>$	$\leq$	$>$
Yes	0	3	1	2								
No	1	3	1	3								
Weighted Entropy	.857											

Activity 5 : Example Data and Applicability of Decision Trees

Considering the case that only *binary* splits are possible (each node can have at most two children), consider how decision trees divide up the space. Below is a set of training data with 2 features. Make some general comments on how decision trees might perform given this input data.



a b

Activity 6 : Quantifying Your Model's Performance

7. We will be discussing several types of classification algorithms in this class. While the accuracy and error rate of your classifier are certainly important ways to judge quality, discuss other factors that you should weigh when evaluating these algorithms.