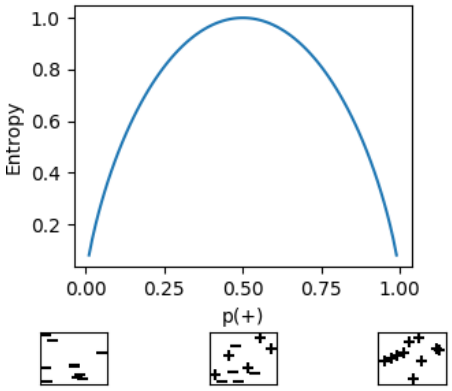
Decision Tree Construction

In this activity, we will explore the CART decision tree construction algorithm for classification. This approach works by first grouping all the data into a single root node. If all the examples in the node are from a single class, we are done. Otherwise, we iterate through each feature to see how well that feature splits/partitions the data by class, and then select the feature providing the best split (we will formalize the best split in the next activity). Child nodes are created using the best split, 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.

1. **Information Entropy**



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

$$-\sum_{i=1}^{c}\left(p_i(t)\ \log_2\ (p_i(t))\right)$$

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 $-$.

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

2. **Selecting Splits**

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

$$Gain(Split) = Entropy(Parent) - \left(\frac{m_{left}}{m}Entropy(Left) + \frac{m_{right}}{m}Entropy(Right)\right) \quad (1)$$

Where $m_{left/right}$ is the number of training examples within a child node and m 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 denoted Δ_{info}.

Here is a small 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

(a) Calculate the entropy of the full dataset.

(b) Calculate the gain associated with splitting on the **Home Owner** attribute.

(c) Calculate the gain associated with splitting on the **Marital Status** attribute. Consider two possible splits: single vs. married/divorced and single/married vs. divorced.

(d) Calculate the gain associated with splitting on **Annual Income** The table below is provided to help you keep track of the possible split points for the Annual Income 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											

(e) What is the best overall split point and what is the information gain associated with that split?

3. **Run Time Analysis of Tree Construction**

(a) Consider the run time cost of evaluating the splits for a single continuous feature like *annual income*. For a dataset of size n, how many operations does it take? You can consider the cost of computing the entropy for a single split point as $O(1)$.

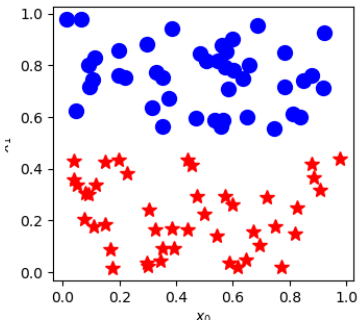
(b) Given a dataset with d attributes, what is the cost of a determining a single split, considering all possible attributes?

(c) What is the overall cost of the greedy decision tree construction algorithm, assuming a reasonably balanced tree?

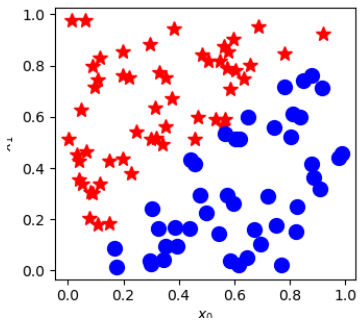
(d) What is the worst-case cost of classifying a point using a balanced decision tree?

4. Example Data and Applicability of Decision Trees

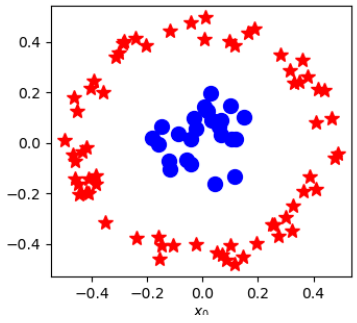
Considering the case that only *binary* splits are possible (each node can have at most two children), consider how decision trees divide up the feature space. Below are three sets of training data with 2 features. Make some general comments on how decision trees might perform given this input data.



a



b



c

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many pages to expect for this document.