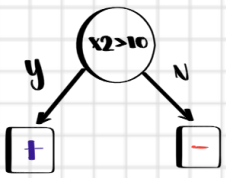
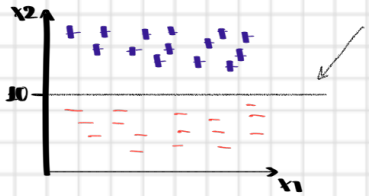
Lecture 11 - 22/10/24

Decision Trees

- A **supervised** learning method
- Use an **if-else** structure to define a decision boundary
  - Each **focuses** on a **single feature**





Decision boundary!

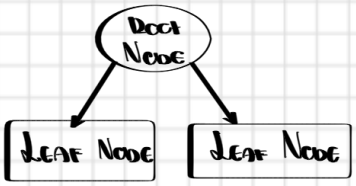
We wanna divide the feature space into distinct, non-overlapping regions or "strata"

- Ask Question **Recursively** to create a tree
- **Several** decision regions can predict the same value
- In regression trees the output label y label is the **mean**

Good Characteristics:

- **Interpretability** → Easy explainable
- **Efficiency** → Save only the tree not the data, O(log(n)) for tree operations.

- Every decision leads to a binary split



↳ this is called **STUMP**. → It is a one-level tree

- Splits can be both:
  - Binary split
  - Multi-way split

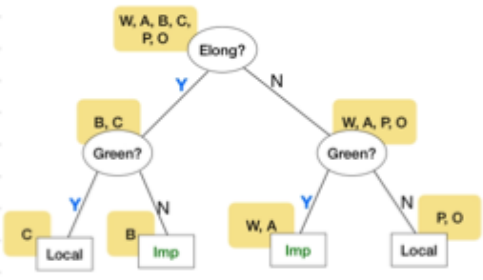
- There are **three types** of **input** features:
  - Numerical
  - Categorical
  - Mixed

} which all lead to → Classification trees / Regression trees

	Elongate	Big	Green	Class
Watermel	N	Y	Y	Imported
Apple	N	N	Y	Imported
Banana	Y	N	N	Imported
Cucumber	Y	N	Y	Local
Pumpkin	N	Y	N	Local
Orange	N	N	N	Local

This tree works perfectly, but, there are multiple trees that achieve the goal...

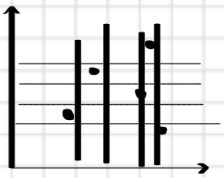
So, which one to choose?



This looks better!

How to train a decision tree...

- How do we build a good decision tree?
  - A balanced tree is good!
  - We use heuristic aiming for good trees
    - use greedy algo ~ at each node, make a split that best divides on specific criterion
      - Try all possible splits for the current node
      - Use heuristic to measure how good each split is
      - Pick best split for that node
      - Repeat three steps above recursively for the childs



Criterion for evaluating quality of a split (in regression trees)

- How good is split S on these data points?
  - Pretend we've used S to divide data points at the node
  - Calculate the labels $\hat{y}$ of the two resulting regions (the mean) → $\hat{y}_{R_1}$ $\hat{y}_{R_2}$
  - Calculate RSS of data point at the node, summed over both regions.

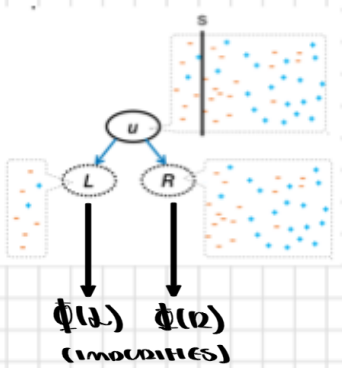
$$RSS = \sum_{i:x_i \in R_1} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2} (y_i - \hat{y}_{R_2})^2$$

    - RSS is used as the criterion for evaluating a split for a regression tree!
  - Pick the split with the minimum RSS for that node

Criterion for evaluating quality of a split (in classification trees)

- How good is split S on these data points?
  - Pretend we've used S to divide the data points at node u
  - Split creates two child, node L, node R
  - Concept of Impurity:

    A measure of how mixed the data is wrt class members.

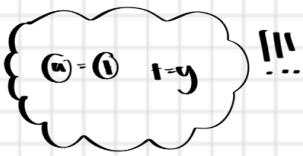


$\phi(L)$ $\phi(R)$
(impurities)

- What is then the Quality of split S?
  - we cannot just sum up the impurities of the two childs!
  - Use weights:

N_L = data points in left
N_R = data points in right
N_u = data points at u

L: $P_L = N_L / N_u$ R: $P_R = N_R / N_u$



- Combine the impurities using weighted avg's:

$$P_L \phi(t_L) + P_R \phi(t_R)$$

Best split is the one that minimizes this!

OR

maximize $\phi(t) - (P_R \phi(t_R) + P_L \phi(t_L))$

Stop if there is no split that gives us more info.

Impurity function

- As said it is a measure of how mixed data is
- Φ can be defined as a function of posterior probabilities:

$$\Phi(p_0, p_1, \dots, p_{K-1})$$

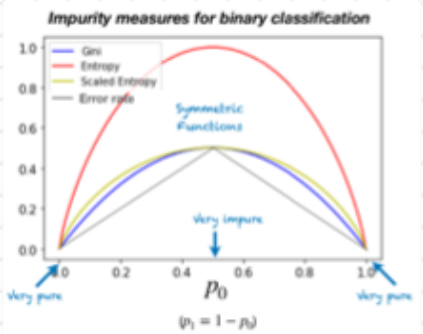
- Estimated posterior probability for class k at node t:

$$p_k = \hat{p}(k|t) = \frac{N_k}{N_t}$$

- Φ is max only when all p_k are equal.
- Φ is minimum if all points are in the same class.
- Φ is symmetric, i.e. it doesn't care about the order of input probs.

- Impurity functions:
  - Gini impurity: $\Phi(i) = 1 - \sum_{k=1}^{K} p_k^2$
  - Entropy: $\Phi(i) = -\sum_{k=1}^{K} p_k \log p_k$
  - ...
  - Classification Error Rate

not a great one

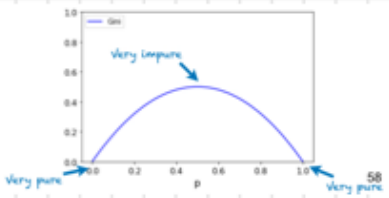


Gini Index

- $p_k = \hat{p}(k|t) = \frac{N_k}{N_t}$

$$G(t) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

For a binary classification: $G(t) = 2p(1-p)$



Entropy

$$p_k = \hat{p}(k|t) = \frac{N_k}{N_t} \qquad H(t) = -\sum_{k=1}^{K} p_k \log p_k$$

Entropy of random vars is the avg of "uncertainty" inherent to the vars possible outcomes

When everything is uniform: Entropy maximized

We wanna minimize entropy by every split!