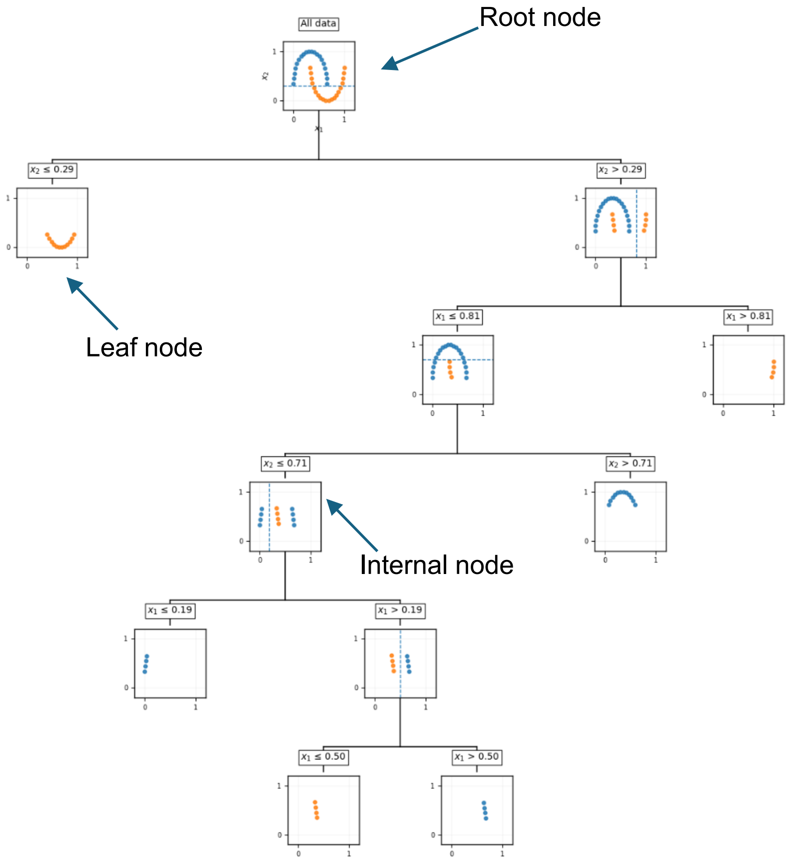
Introduction to Machine Learning

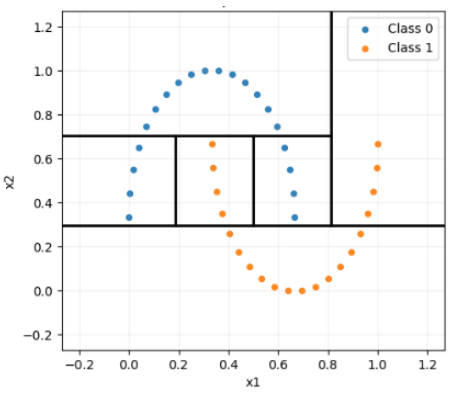
Lecture 14:

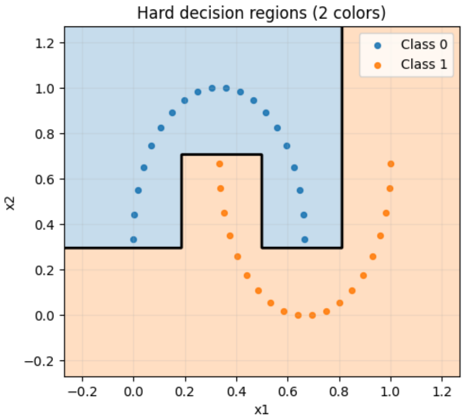
Decision Trees

Prof. Subrahmanya Swamy Peruru
Department of Electrical Engineering
IIT Kanpur

Decision Trees: Classification Demo





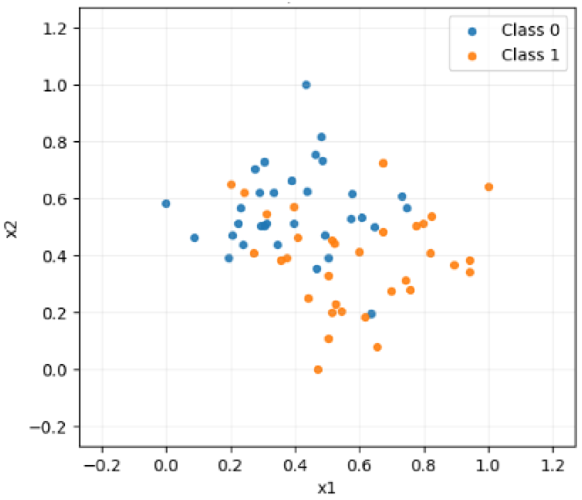


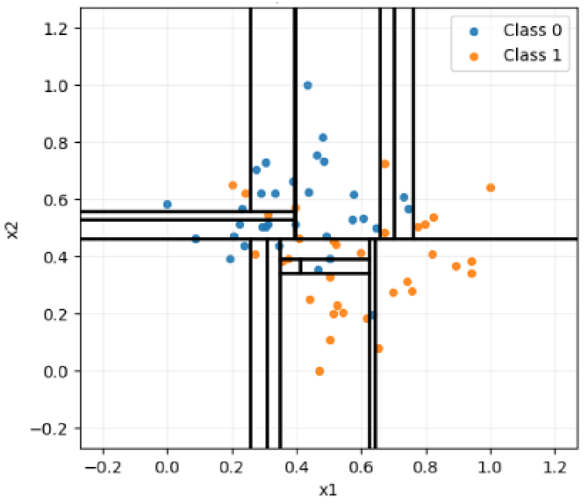
Demo Link

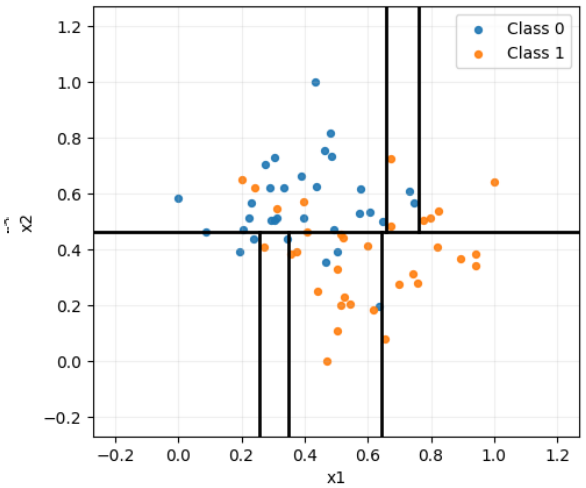
How do we choose the feature and threshold for splitting?

How is the prediction made at a leaf when the node is impure?

Leaf nodes (Regions) need not be pure





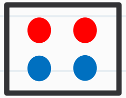


Aiming for pure leaf nodes (regions) may overfit

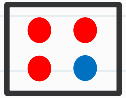
Often allow impure leaves and predict using the
- **majority class within each leaf**
- **empirical class probabilities**
- **Fit a classifier (e.g. logistic regression for each region separately)**

Measuring node impurity

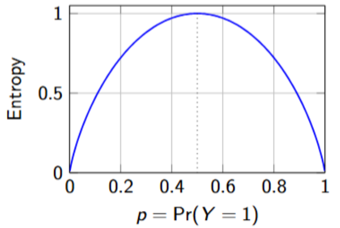
How impure is each node?

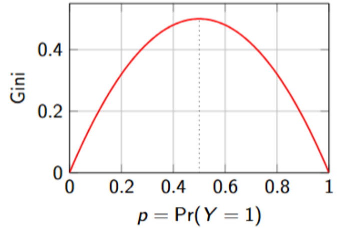


Node A



Node B





Entropy:

$$H(p) = -\sum_c p_c \log_2 p_c$$

Gini index:

$$G(p) = \sum_c p_c\,(1 - p_c)$$

Node A: 50 percent of each class $\Rightarrow$ $(p, 1-p) = (0.5,\ 0.5)$

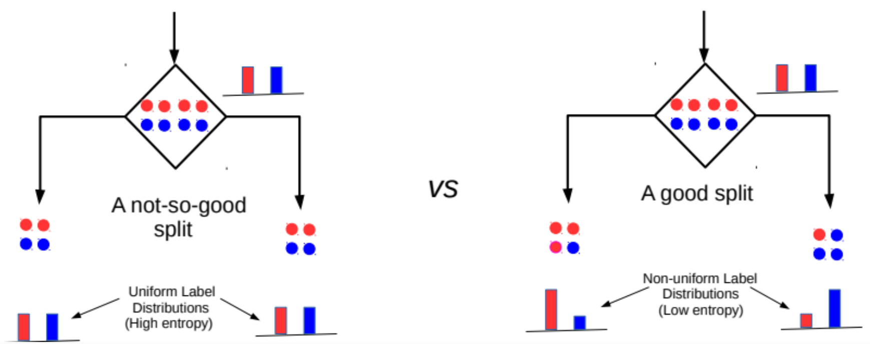
$$H = 1.0, \qquad G = 0.5$$

Node B: 75 percent of one class $(p, 1-p) = (0.75,\ 0.25)$

$$H \approx 0.811, \qquad G = 0.375$$

Key message: High impurity ⇒ High entropy or High Gini index

Measuring the Quality of a Split



$$\text{Split Quality} = \text{Impurity}(\text{parent}) - \sum_k \frac{n_k}{n} \text{Impurity}(\text{child}_k)$$

Impurity of parent minus **average impurity of child nodes**

Split quality is called as **information gain** if impurity is measured using **entropy.**

Fig credits: Prof. Piyush Rai

Information Gain: Example

Features: $X_1 \in \{\text{Red}, \text{Blue}\}$, $X_2 \in \{\text{Circle}, \text{Square}\}$, $Y \in \{0,1\}$.

Impurity at root node (measured through Entropy):
(#ones, #zeros) $= (6, 6) \Rightarrow p_1 = p_0 = 1/2$, so

$$H(S) = -1/2 \log_2 1/2 - 1/2 \log_2 1/2 = 1$$

Split with feature X_1 or X_2?

Which one reduces the node impurity? (Compute Information gain)

$$\mathrm{IG}(S, X_1) = H(S) - \sum_v \frac{|S_v|}{|S|} H(S_v).$$

Features:

$X_1 \in \{\text{Red}, \text{Blue}\}$, $X_2 \in \{\text{Circle}, \text{Square}\}$,

#	X_1	X_2	Y
1	Red	Circle	1
2	Red	Circle	1
3	Red	Circle	1
4	Red	Circle	1
5	Red	Circle	0
6	Red	Square	1
7	Red	Square	0
8	Blue	Circle	0
9	Blue	Square	1
10	Blue	Square	0
11	Blue	Square	0
12	Blue	Square	0

Information gain for split X_1

Features:

$X_1 \in \{\text{Red}, \text{Blue}\}, \qquad X_2 \in \{\text{Circle}, \text{Square}\},$

#	X_1	X_2	Y
1	Red	Circle	1
2	Red	Circle	1
3	Red	Circle	1
4	Red	Circle	1
5	Red	Circle	0
6	Red	Square	1
7	Red	Square	0
8	Blue	Circle	0
9	Blue	Square	1
10	Blue	Square	0
11	Blue	Square	0
12	Blue	Square	0

Split by X_1: (#ones, #zeros).

$S_{\text{Red}}: (5,2)$ (7 points), $S_{\text{Blue}}: (1,4)$ (5 points)

Entropy of Red node:

$$H(S_{\text{Red}}) = -\mathbf{5/7}\log_2 5/7 - \mathbf{2/7}\log_2 2/7 = 0.863$$

Entropy of Blue node:

$$H(S_{\text{Blue}}) = -\mathbf{1/5}\log_2 1/5 - \mathbf{4/5}\log_2 4/5 = 0.722$$

Weighted child entropy:

$$\sum_{v\in\{Red,Blue\}} \frac{|S_v|}{|S|} H(S_v) = \mathbf{7/12}\,(0.863) + \mathbf{5/12}\,(0.722) = 0.804$$

Information Gain (ID3 Algo): for feature X_1 with values v,

$$\text{IG}(S, X_1) = H(S) - \sum_v \frac{|S_v|}{|S|} H(S_v).$$

$$\boxed{\text{IG}(S, X_1) = 1 - 0.804 = 0.196}$$

Information gain for split X_2

Split by X_2: (#ones, #zeros).

$$S_{\text{Circle}}: (4,2) \quad (6 \text{ points}), \qquad S_{\text{Square}}: (2,4) \quad (6 \text{ points})$$

Entropy of each child:

$$H(S_{\text{Circle}}) = -4/6 \log_2 4/6 - 2/6 \log_2 2/6 = 0.918$$

$$H\left(S_{\text{Square}}\right) = -2/6\ \log_2 2/6 - 4/6\ \log_2 4/6 = 0.918$$

Weighted child entropy:

$$6/12\,(0.918) + 6/12\,(0.918) = 0.918$$

Information gain:

$$\boxed{\mathrm{IG}(S, X_2) = H(S) - \sum_v \frac{|S_v|}{|S|} H(S_v) = 1 - 0.918 = 0.082}$$

Choose the split with larger information gain:

$$IG(S, \boldsymbol{X_1}) = 0.196 > IG(S, \boldsymbol{X_2}) = 0.082 \quad \Rightarrow \quad \boxed{\textbf{Split on } \boldsymbol{X_1}.}$$

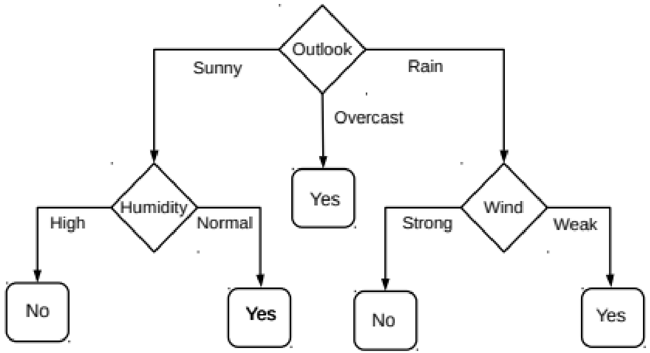
Features:

$X_1 \in \{\text{Red}, \text{Blue}\}$, $\quad X_2 \in \{\text{Circle}, \text{Square}\}$,

#	X_1	X_2	Y
1	Red	Circle	1
2	Red	Circle	1
3	Red	Circle	1
4	Red	Circle	1
5	Red	Circle	0
6	Red	Square	1
7	Red	Square	0
8	Blue	Circle	0
9	Blue	Square	1
10	Blue	Square	0
11	Blue	Square	0
12	Blue	Square	0

Exercise: Verify the split choices using Information Gain

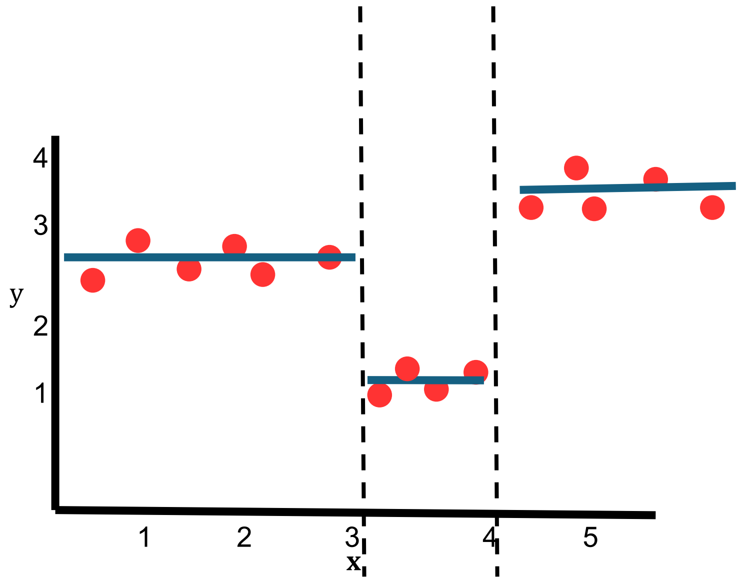
day	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

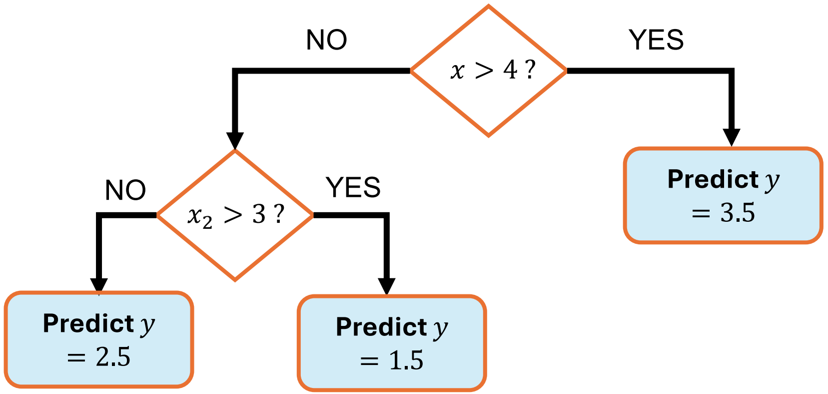


Compute the information gain at each node and verify the chosen split.

Fig credits: Prof. Piyush Rai

Decision Trees for Regression





Regression trees predict the mean in each region and choose splits that reduce variance.

Prediction for a region: **mean target value** of the samples in that region
Impurity measure: **variance** (equivalently, MSE or SSE)

Fig credits: Prof. Piyush Rai

Decision Trees: A Few Comments

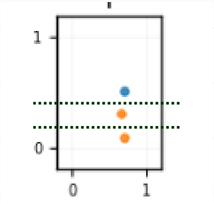
How to handle features with more than two values? Example: $X \in \{red,\ green,\ blue\}$, $Age \in \{1, 2, 3, \dots, 100\}$

- Multi-way splits (one for each value of the feature)
  - can favor features with many distinct values (addressed by gain ratio in C4.5 Algo)

- Binary splits with subsets of values in each branch (CART Algo)
  - Eg: $\{red\}\ vs\ \{green,\ blue\}$ (In principle, need to search over all possible binary portions of values)

How to decide threshold values for real-valued features?

- Sort the data points based on that feature and
- try candidate thresholds between consecutive points



When to stop splitting a node further (Pre-pruning)?

- If the node is a pure node
- If a certain pre-defined **maximum depth** is reached
- If the node has very **few points**
- If **gain (impurity reduction) is below a threshold**

Advantages

- Simple and easy to interpret
- Test time prediction is very fast.

Disadvantages

- Finding the globally optimal decision tree under depth or size constraints is NP-hard.
- Hence, decision trees are learned using greedy, top-down splitting heuristics.

Thank You