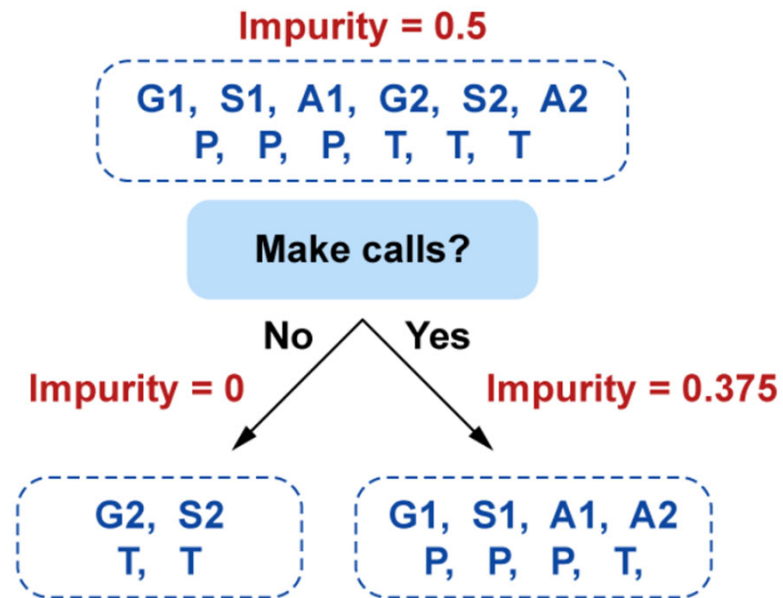


Machine Learning

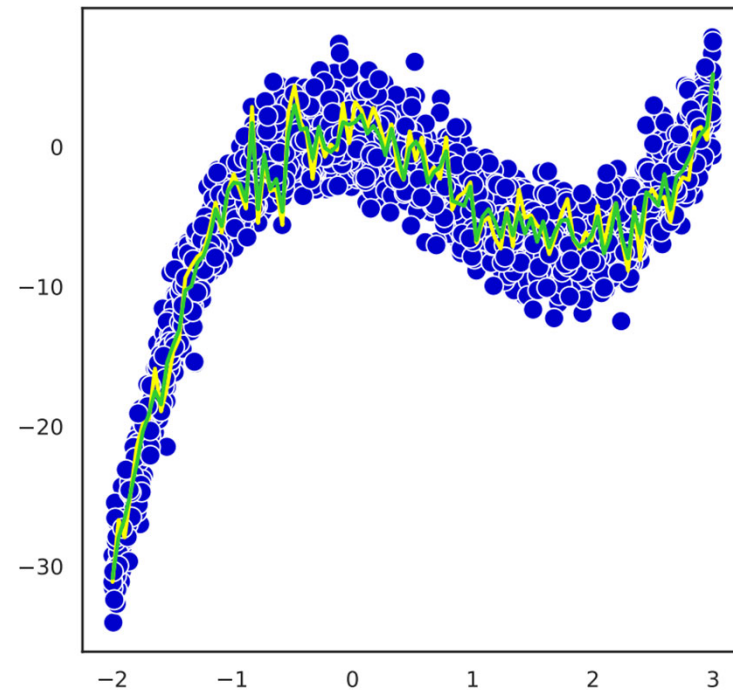
Decision trees

Jian Liu

Part 1: Decision tress



Part 2: Ensemble tress

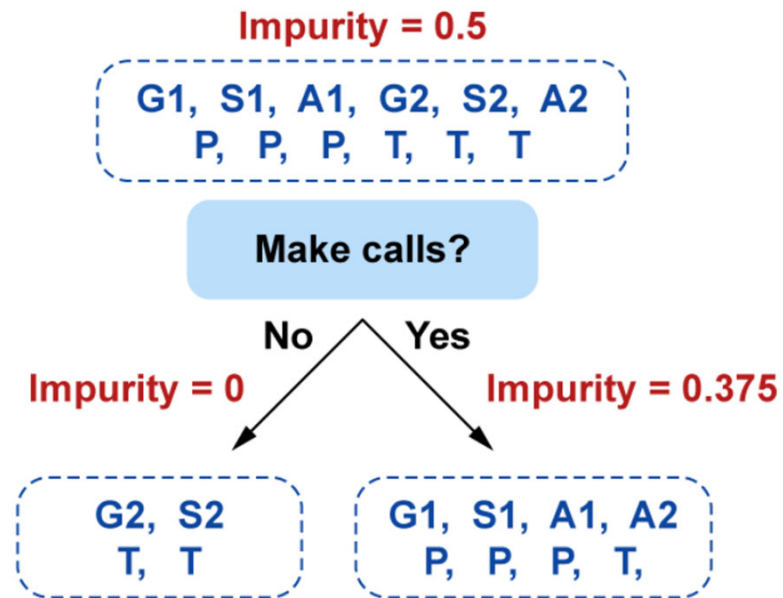


Machine Learning

Decision trees

Jian Liu

Part 1: Decision tress



What is Classification?

- Classification is a form of supervised learning
- Goal: assign each sample in a dataset to one or more pre-defined categories or *classes*



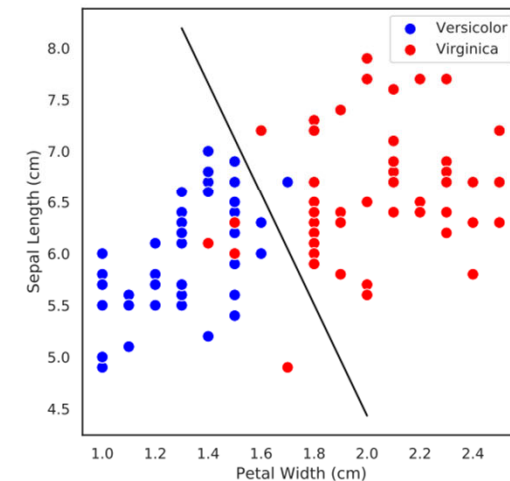
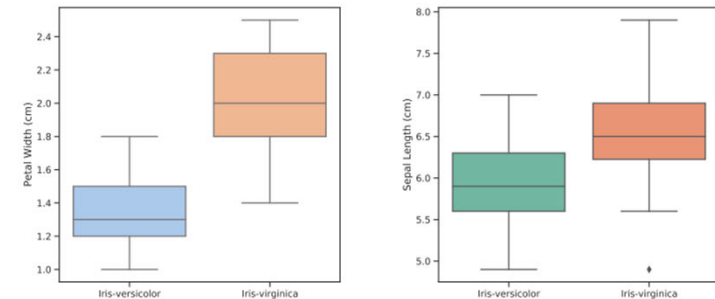
Figure: Classify samples into two distinct species of Iris flower

Source: Wikipedia; Creative Commons License CC BY-SA 3.0

- Typical pipeline: collect data, preprocessing, extract features, **train classifiers** and **predict using trained classifiers (inference)**

What is Classification?

- Another way of looking at it - approximate a decision boundary given some features
- Extract best (most discriminative) features from each flower image
 - Reduce data and dimensionality
 - Ex: Petal length, petal width, sepal length, sepal width, etc.
- Each sample can be represented by a set of features \mathbf{x} with associated class label (species) y



What is Classification?

- More formally classification can be defined as:
- Given a dataset \mathbb{X} of N samples where each sample is represented by a feature set $\mathbf{x}_{i=1\dots N} \in \mathbb{X}$; and its corresponding discrete class label y_i denoting its membership to a specific class $\mathcal{C}_{k=1\dots K}$, we want to:

Learn a model that maps each feature set \mathbf{x}_i to its class label y_i by approximating the decision boundary(ies) that best separates samples belonging to each $\mathcal{C}_{k=1\dots K}$

Classification Task	Feature Set	Class Labels
Sorting music into genres	Features derived from audio signals	Rock, Jazz, Blues
Categorising emails	Features derived from text data in emails	Spam, non-spam
Triaging COVID-19 patients according to severity of infection	Features derived from chest X-rays or CT images	High, moderate, low

Classification Techniques

Base classifiers

- Decision Trees
- Nearest-neighbour
- Perceptron
- Logistic regression
- Multi-layer perceptron (Neural Networks)
- Support vector machines
- Gaussian Processes
- Naive Bayes
- Bayesian belief networks
-

Ensemble classifiers

- Bagging (applied to trees - Random Forests)
- Boosting (applied to trees - gradient tree boosting, XGBoost)
- Mixture of Experts
-

Decision trees

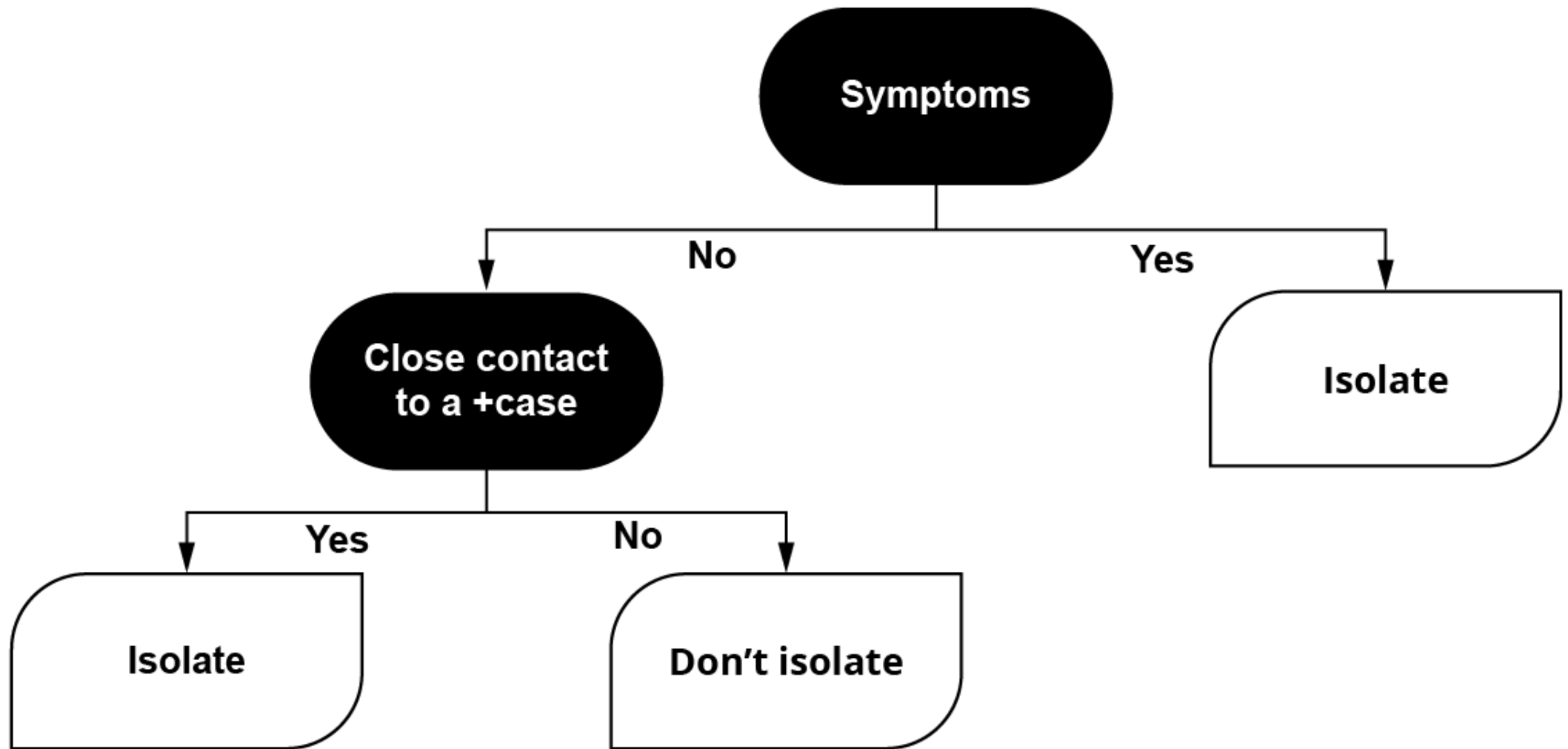


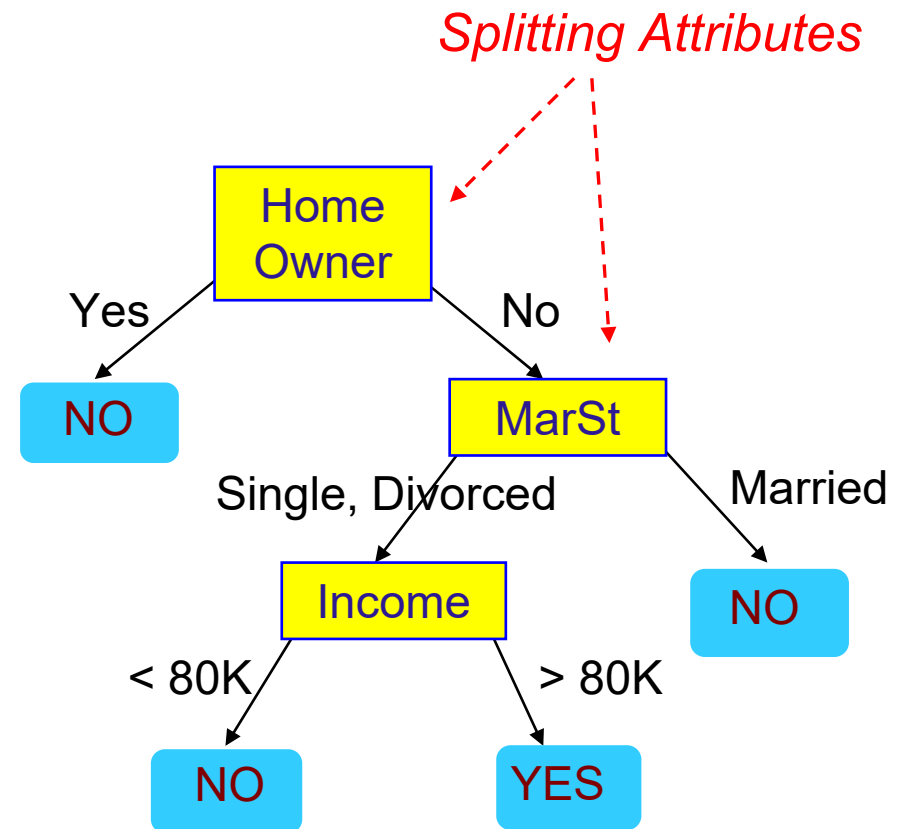
Diagram of a decision tree (DT) with a binary structure

Example of a Decision Tree

categorical categorical continuous class

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

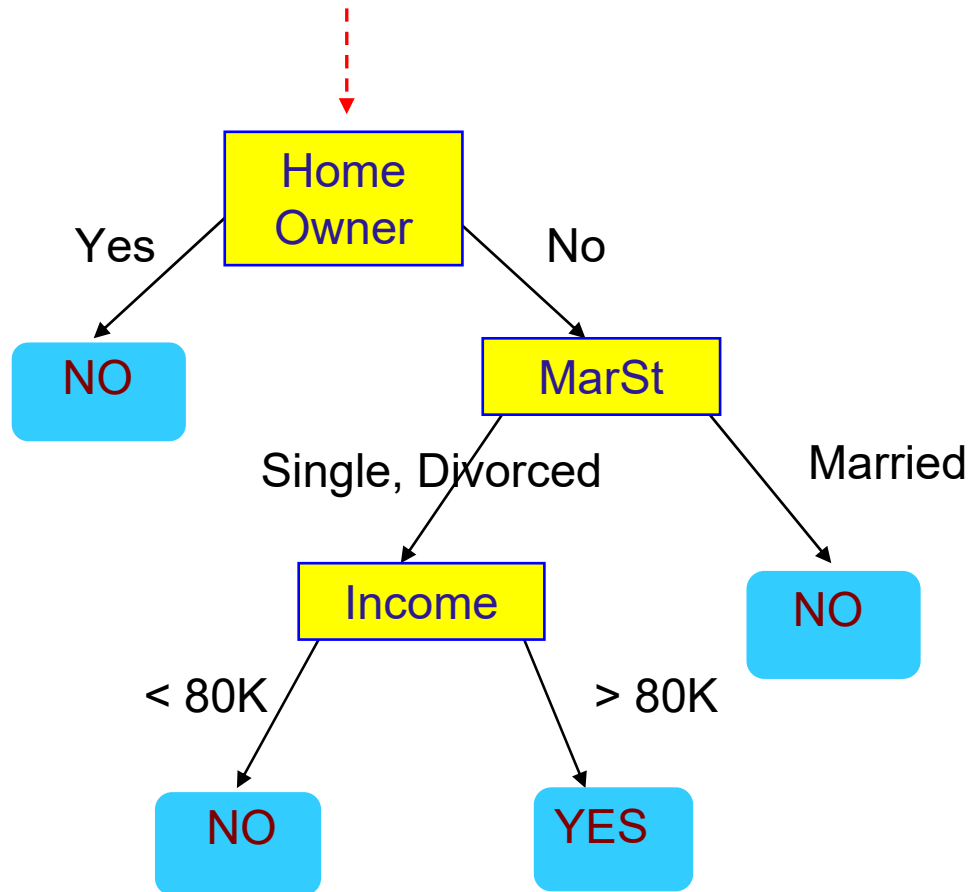
Training Data



Model: Decision Tree

Apply Model to Test Data

Start from the root of tree.



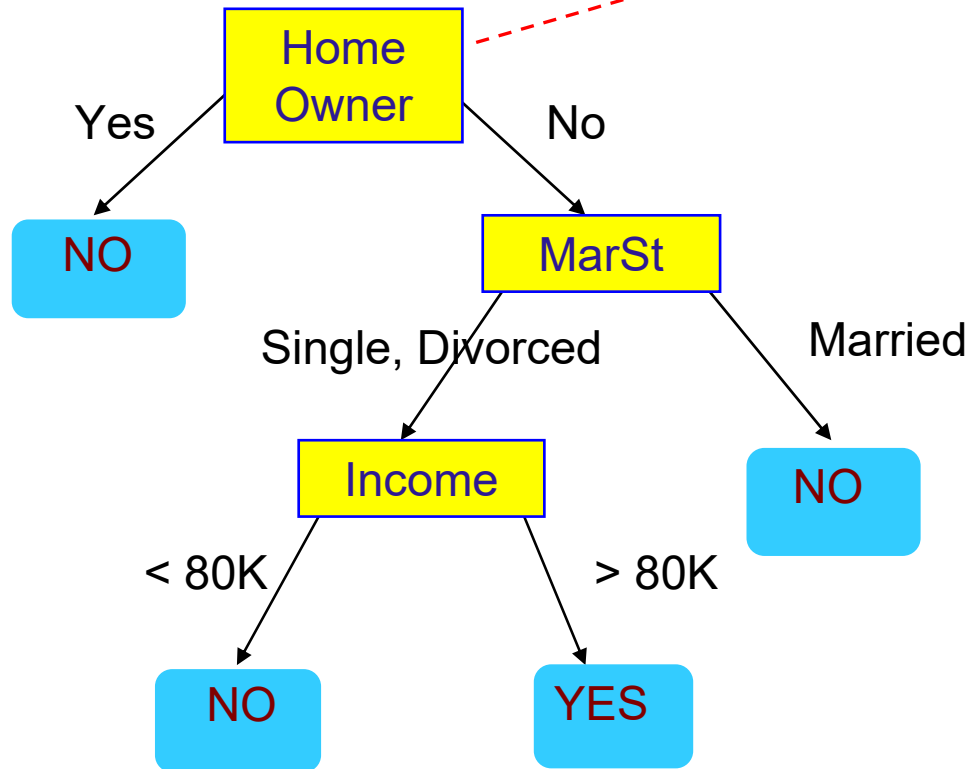
Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Apply Model to Test Data

Test Data

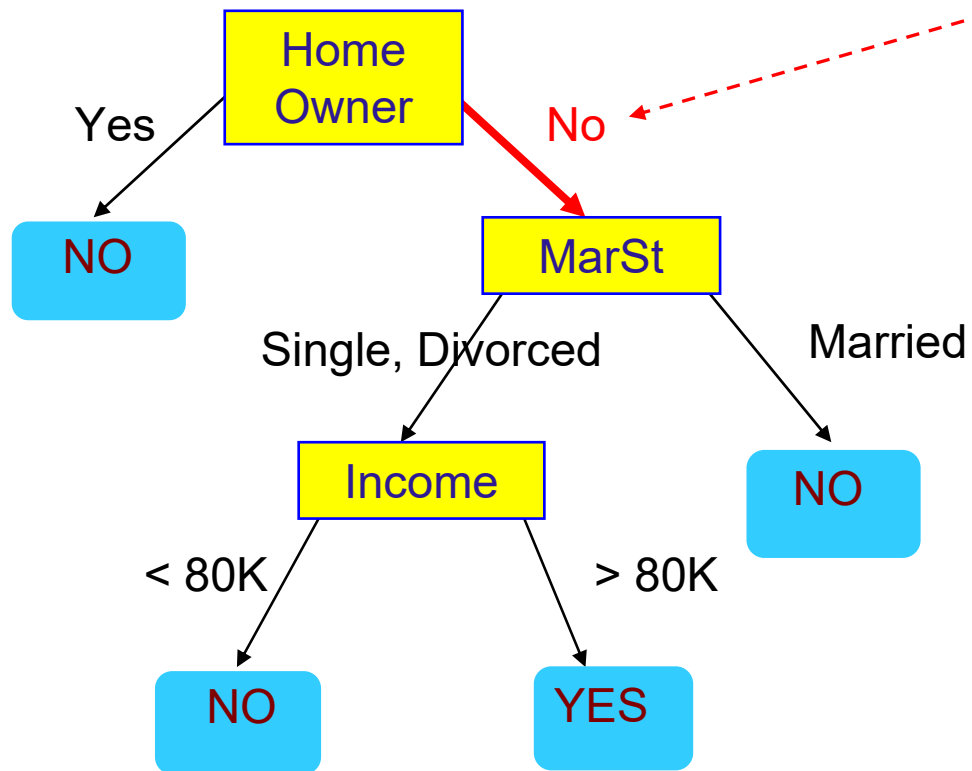
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

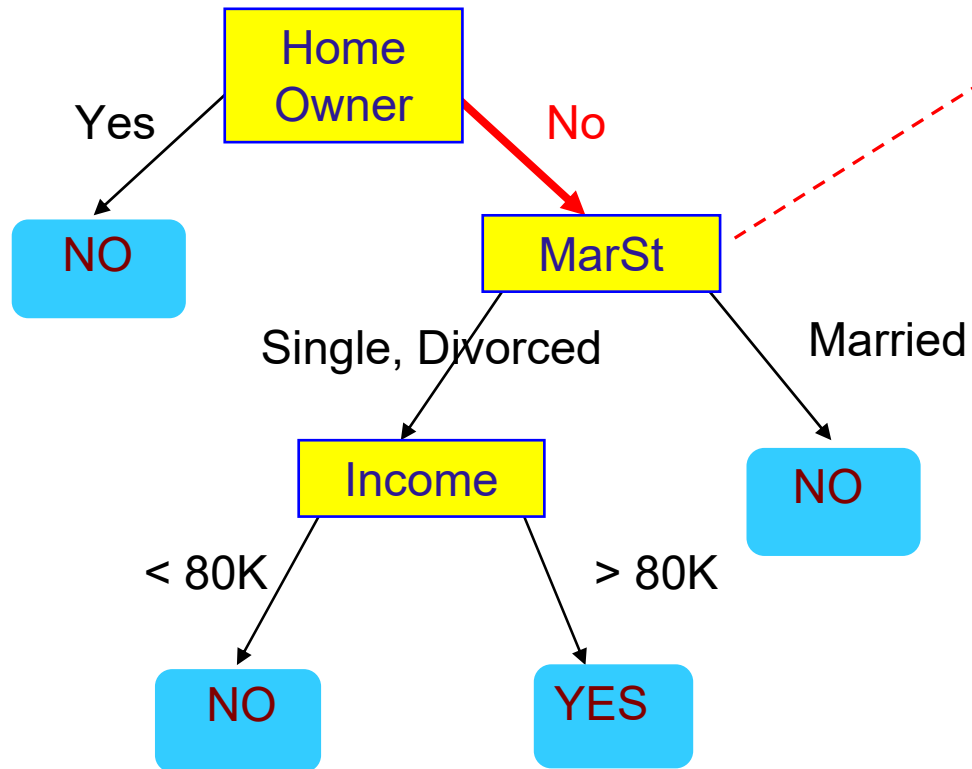
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

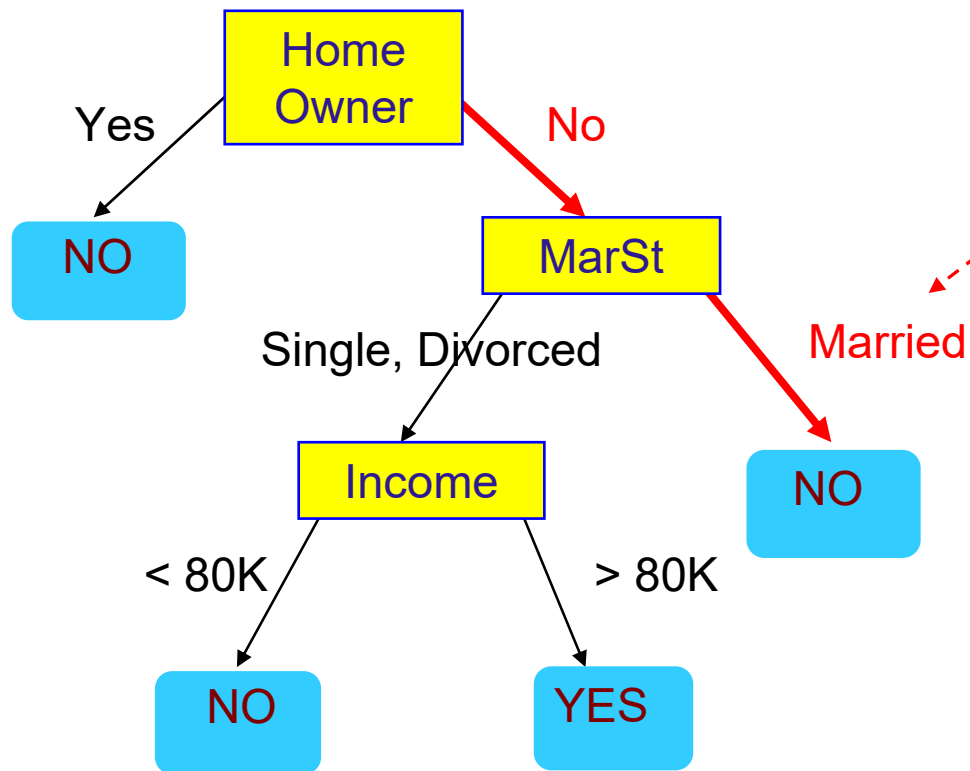
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

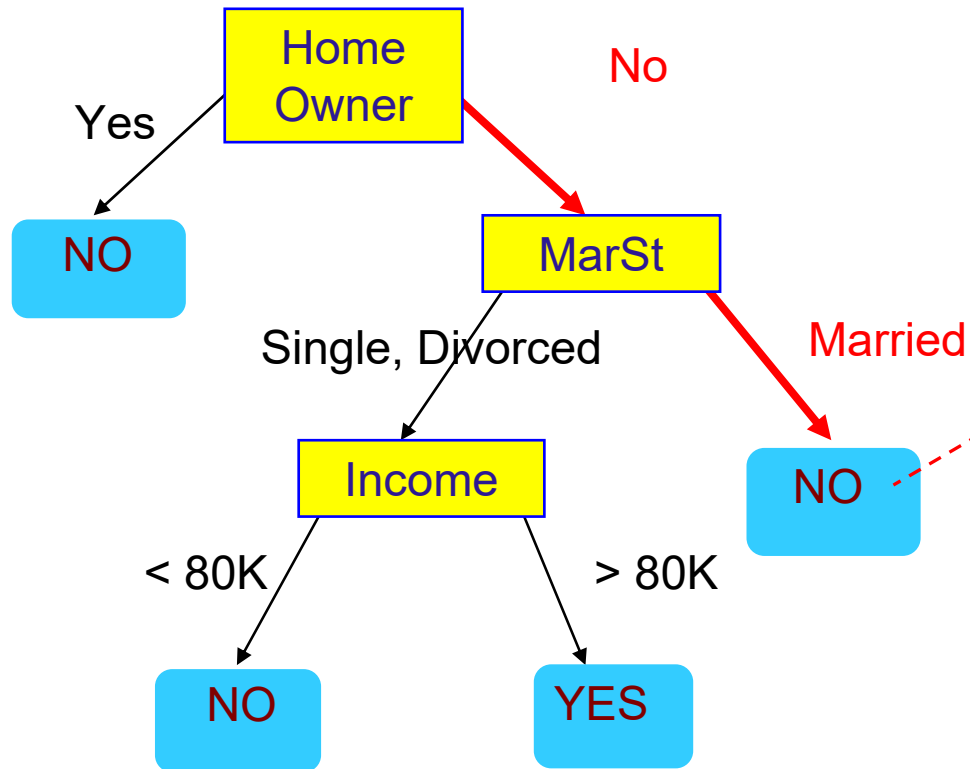
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Assign Defaulted to
"No"

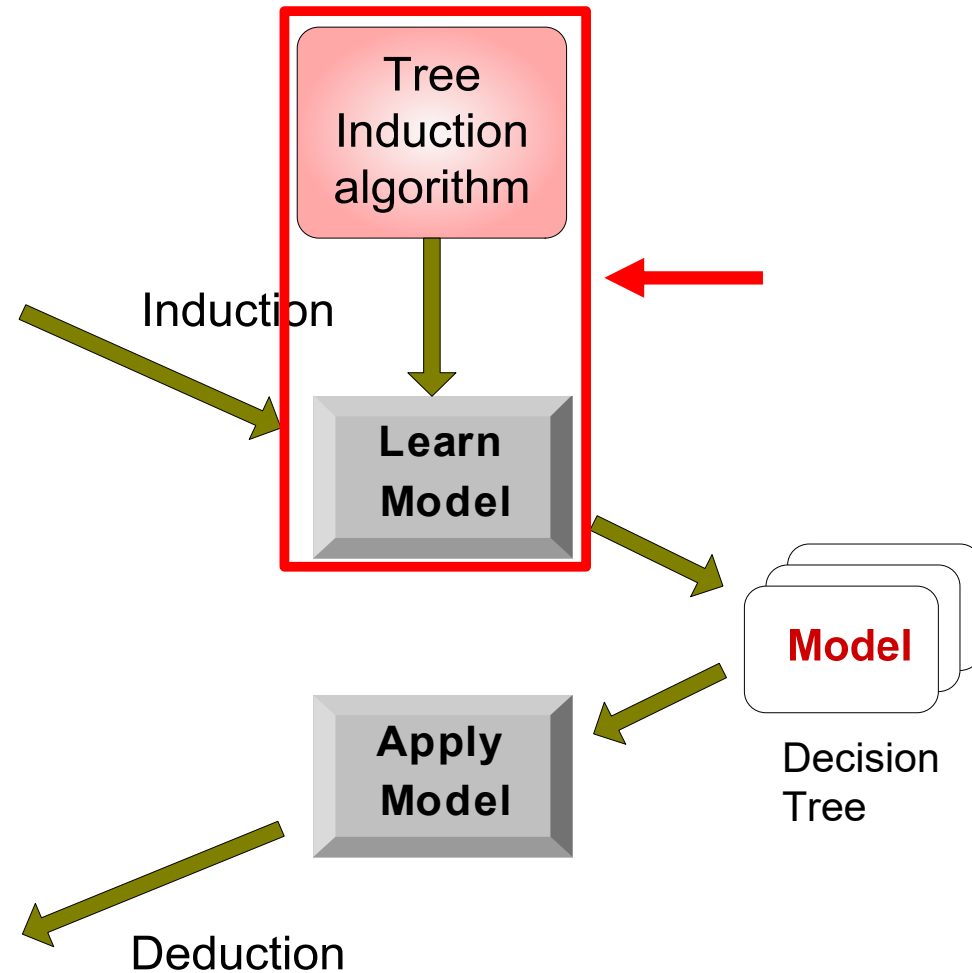
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Methods for Expressing Test Conditions

Depends on attribute types

- Binary
- Nominal
- Ordinal
- Continuous

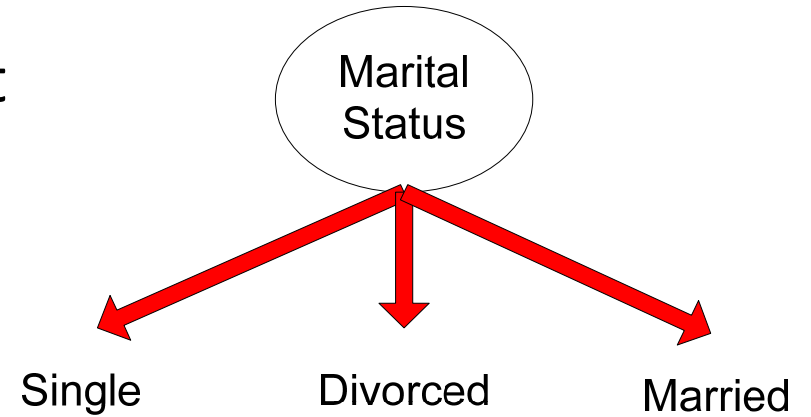
Depends on number of ways to split

- 2-way split
- Multi-way split

Test Condition for Nominal Attributes

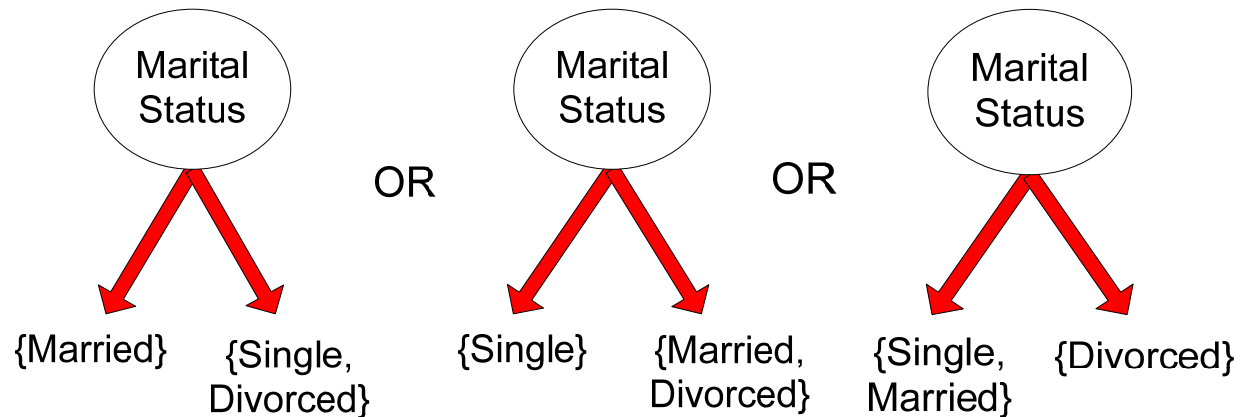
□ Multi-way split:

- Use as many partitions as distinct values.



□ Binary split:

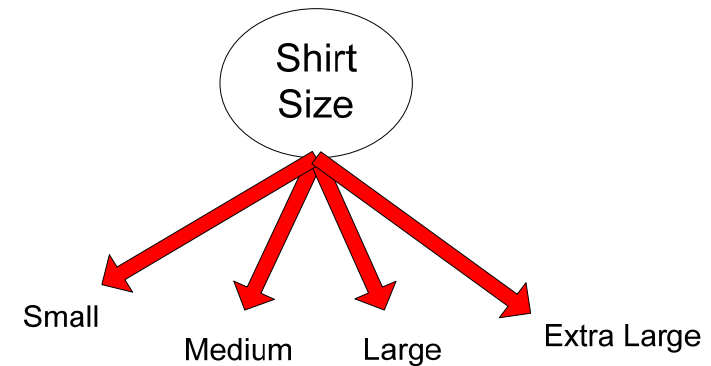
- Divides values into two subsets



Test Condition for Ordinal Attributes

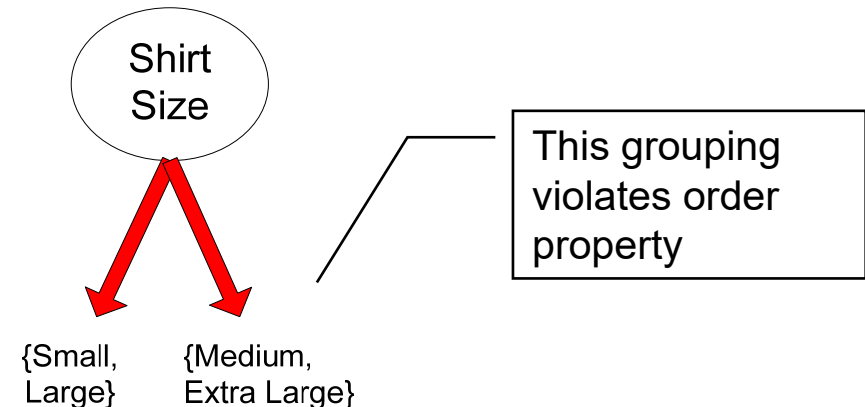
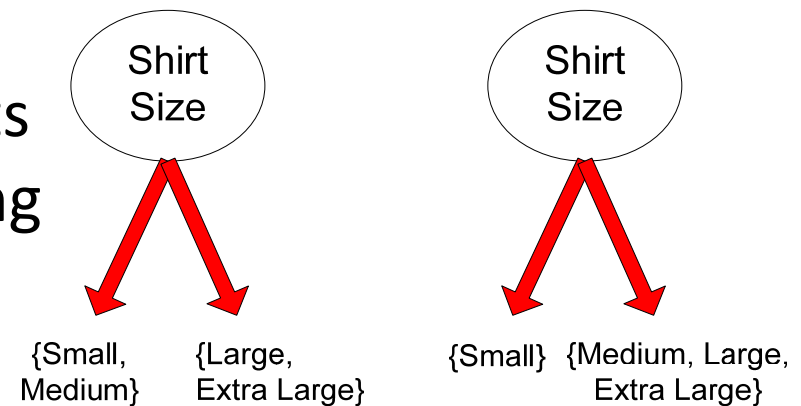
I Multi-way split:

- Use as many partitions as distinct values

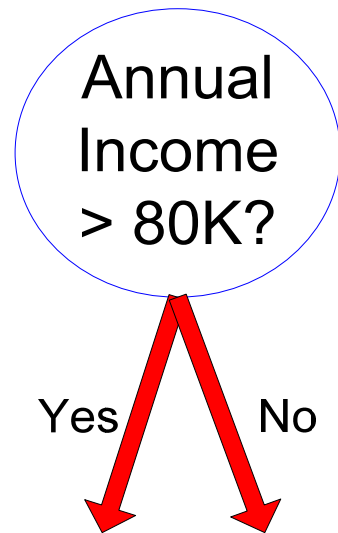


I Binary split:

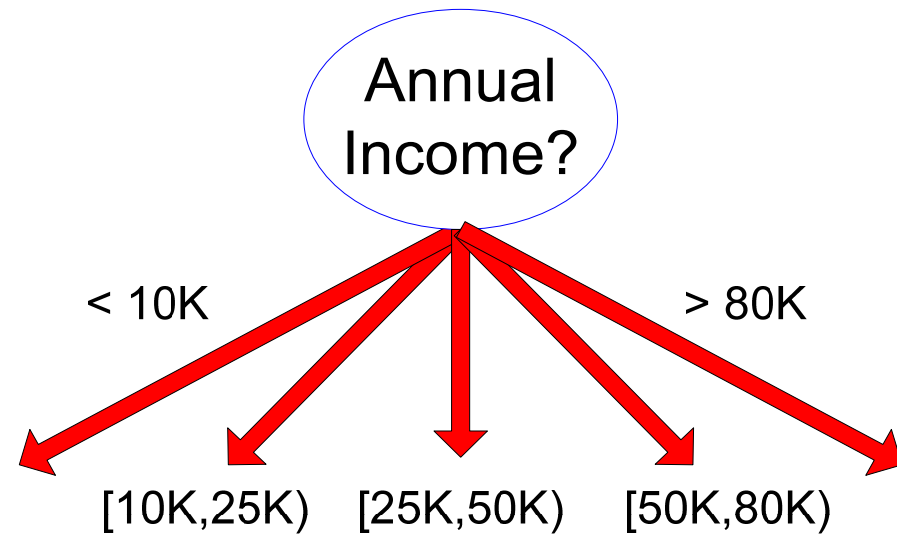
- Divides values into two subsets
- Preserve order property among attribute values



Test Condition for Continuous Attributes



(i) Binary split



(ii) Multi-way split

Splitting Based on Continuous Attributes

Different ways of handling

Discretization to form an ordinal categorical attribute

Ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.

Static – discretize once at the beginning

Dynamic – repeat at each node

Binary Decision: $(A < v)$ or $(A \geq v)$

consider all possible splits and finds the best cut

can be more compute intensive

Example

The dataset allows us to classify a device that is not part of the dataset to find out whether it is a phone or a tablet.

In the dataset, the possible values for the features and the class are as follows:

Screen size = {6, 7, 8}, Makes calls = {Yes, No}, Class = {Phone, Tablet}

Device	Screen size	Makes calls	Classification
G1	6 inches	Yes	Phone
S1	6 inches	Yes	Phone
A1	7 inches	Yes	Phone
G2	7 inches	No	Tablet
S2	7 inches	No	Tablet
A2	8 inches	Yes	Tablet

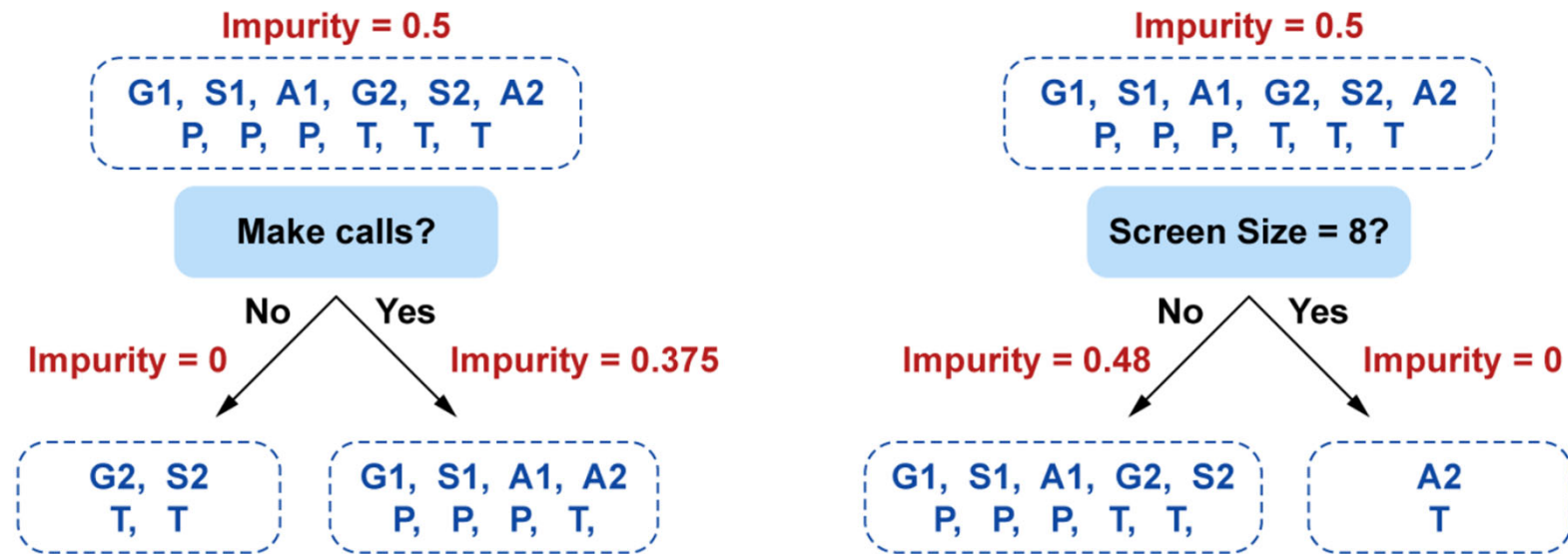
‘screen size’: we have phones that are 6 and 7 inches and we have tablets that are also 7 or 8 inches.

‘makes calls’: **all** phones ‘make calls’ while **most** tablets do not ‘make calls’.

This suggests that if we were to make decisions about a device class being a phone or a tablet, we should first look at the ‘makes calls’ attribute and if it is ‘no’ then it is a ‘tablet’ if it is ‘yes’ then it is most likely a ‘phone’

Create an algorithm

CART (Classification and Regression Tree): Step 1

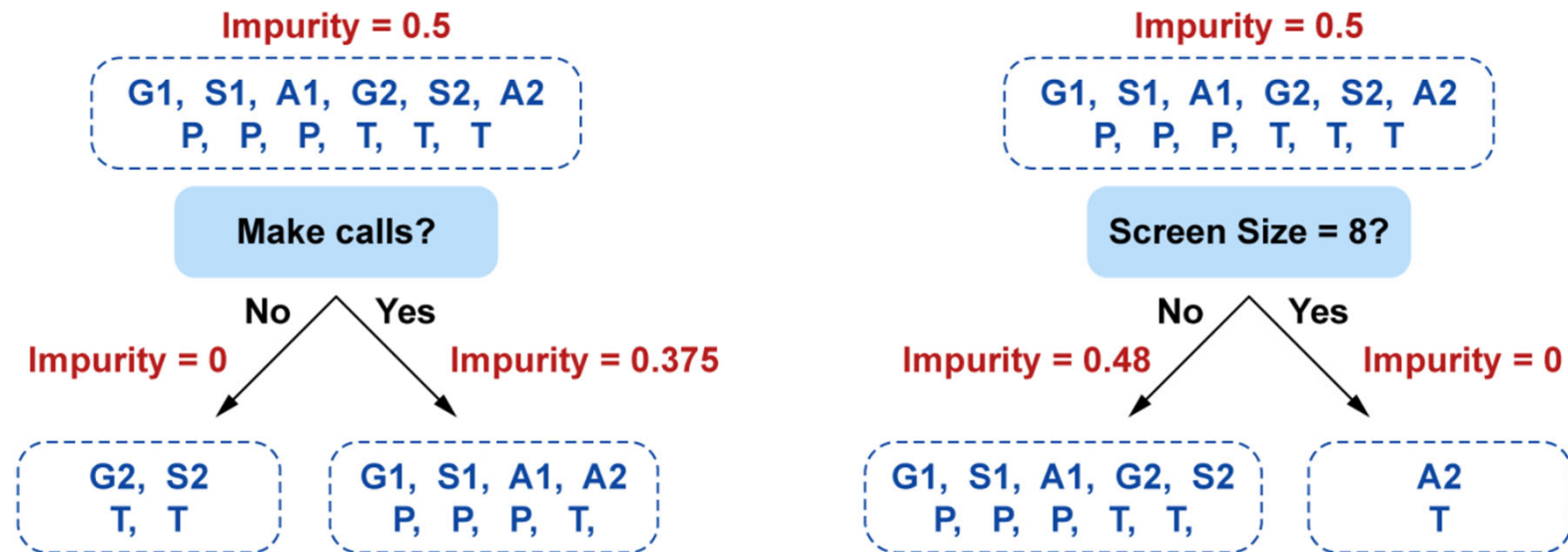


To quantify the quality of each split, we use the **Gini Impurity** of each **node data**.

The Gini impurity describes how pure or mixed the data labels in a node are.

The purer the data the closer Gini is to 0, while the more mixed the data in the node the closer Gini is to 0.5.

CART (Classification and Regression Tree): Step 1



The **Information gain of a split** refers to how much information we gain by choosing one of the possible splits.

If we would like to decide which of the above two splits is better, then we just calculate the information gain and we choose the one that has the highest information gain.

Information Gain = Impurity of parent-weighted average impurity of children

$$\text{Info Gain('makes calls')} = 0.5 - \left(\frac{2}{6} \times 0 + \frac{4}{6} \times 0.375 \right) = 0.25$$

$$\text{Info Gain('screen size=8')} = 0.5 - \left(\frac{5}{6} \times 0.48 + \frac{1}{6} \times 0 \right) = 0.1$$

Gini impurity

We take the probability of one label in the node and we multiply it with the probability of the other label (or sum of other labels probabilities if we have multi-class dataset). And we do that again for the second label and so on.

$$p(Tab) = \frac{3}{6} = 0.5 \quad p(Pho) = \frac{3}{6} = 0.5.$$

$$\begin{aligned} \text{Gini Impurity (set)} &= p(Tab)[1 - p(Tab)] + p(Pho)[1 - p(Pho)] \\ &= p(Tab) + p(Pho) - [p^2(Tab) + p^2(Pho)] \end{aligned}$$

However we have $p(Tab) + p(Pho) = 1$, therefore:

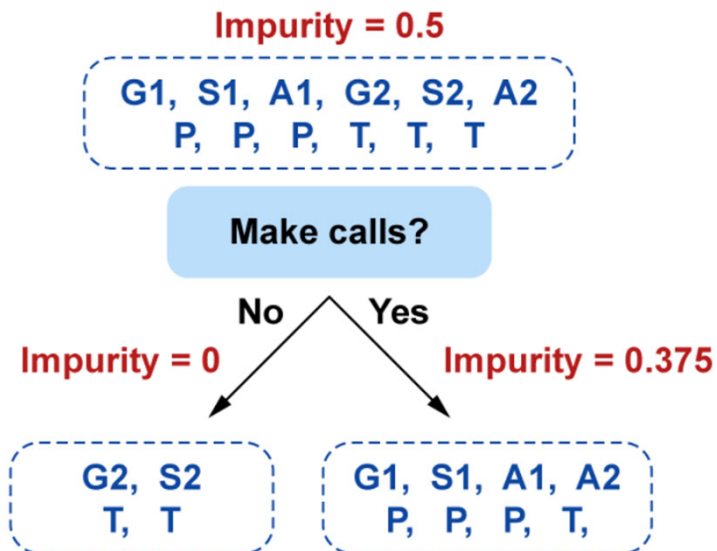
$$\text{Gini Impurity(set)} = 1 - [p^2(Tab) + p^2(Pho)]$$

$$\text{Gini Impurity (node set)} = 1 - \sum_{c=1}^I p^2(i)$$

Before split:

$$\text{Impurity} \left(\begin{matrix} G_1 S_1 A_1 G_2 S_2 A_2 \\ PPPTTT \end{matrix} \right) = 1 - \left[\left(\frac{3}{6} \right)^2 + \left(\frac{3}{6} \right)^2 \right] = 0.5$$

Gini impurity



After spilt:

IMPURITY FOR CHILDREN OF 'MAKES CALLS':

Left (No)

$$\text{Gini Impurity} \left(\begin{matrix} G_2 S_2 \\ TT \end{matrix} \right) = 1 - \left[\left(\frac{2}{2} \right)^2 + (0)^2 \right] = 0$$

Right (Yes):

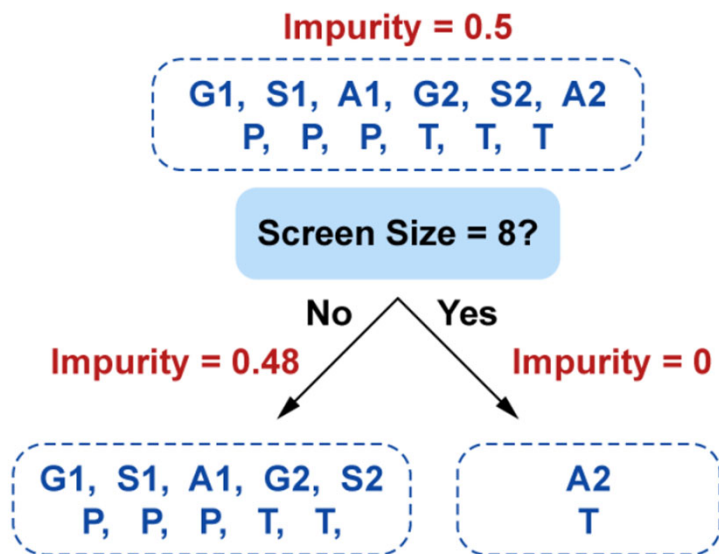
$$\text{Gini Impurity} \left(\begin{matrix} G_1 S_1 A_1 A_2 \\ PPPT \end{matrix} \right) = 1 - \left[\left(\frac{1}{4} \right)^2 + \left(\frac{3}{4} \right)^2 \right] = 0.375$$

Info gain

Info Gain('makes calls')

$$= 0.5 - \left(\frac{2}{6} \times 0 + \frac{4}{6} \times 0.375 \right) = 0.25$$

Gini impurity



After spilt:

IMPURITY FOR CHILDREN OF 'SCREEN SIZE =8':

Left (No)

$$\text{Gini Impurity} \left(\begin{matrix} G_1 S_1 A_1 G_2 S_2 \\ P P P T T \end{matrix} \right) = 1 - \left[\left(\frac{3}{5} \right)^2 + \left(\frac{2}{5} \right)^2 \right] = 0.48$$

Right (Yes):

$$\text{Gini Impurity} \left(\begin{matrix} A_2 \\ T \end{matrix} \right) = 1 - \left[\left(\frac{1}{1} \right)^2 + (0)^2 \right] = 0$$

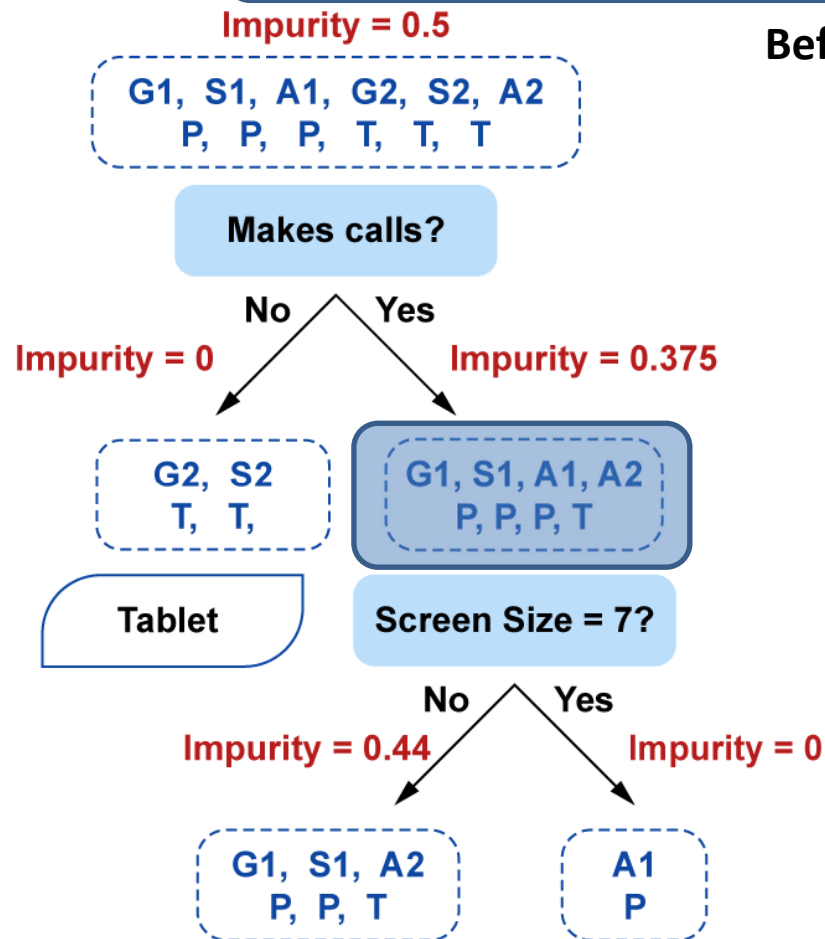
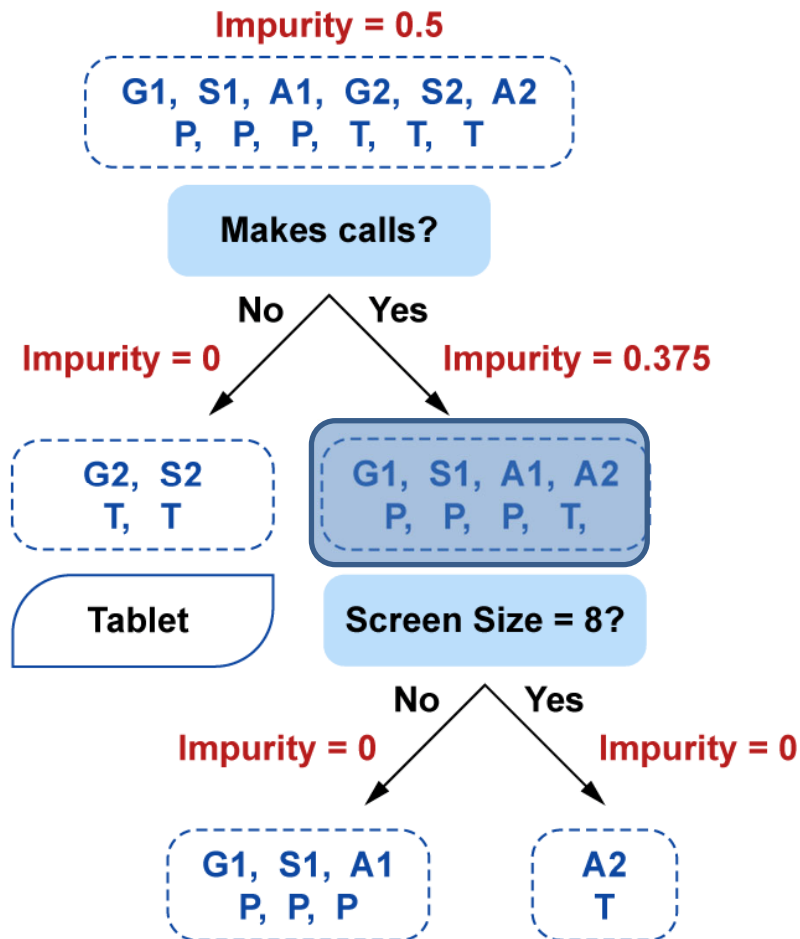
Info gain

$$\begin{aligned} \text{Info Gain('screen size' = 8')} \\ = 0.5 - \left(\frac{5}{6} \times 0.48 + \frac{1}{6} \times 0 \right) = 0.1 \end{aligned}$$

Step 2 of CART algorithm

Next, the CART algorithm will convert the branch on the left of the 'makes calls' into a leaf since it's a pure node (all of its data points are of class 'tablet'). The right hand side node is a mixture of 3 'phones' and a 'tablet'.

$$\text{Gini Impurity} \left(\begin{matrix} G_1 S_1 A_1 A_2 \\ PPPT \end{matrix} \right) = 1 - \left[\left(\frac{3}{4} \right)^2 + \left(\frac{1}{4} \right)^2 \right] = 0.375$$

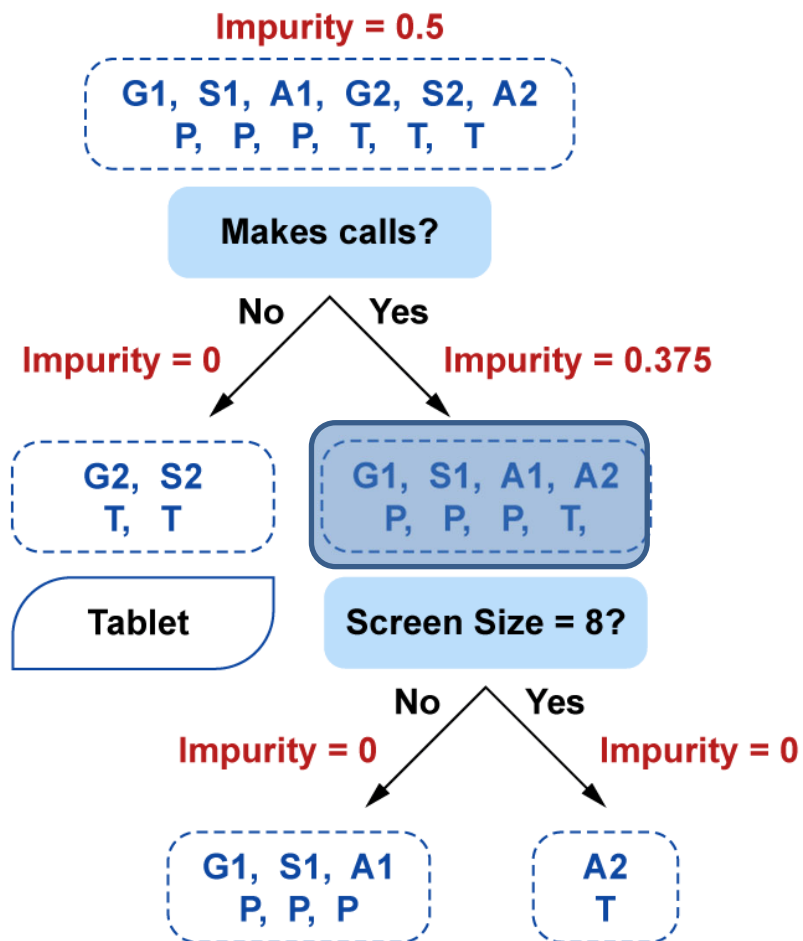


Before split

Step 2 of CART algorithm

Next, the CART algorithm will convert the branch on the left of the 'makes calls' into a leaf since it's a pure node (all of its data points are of class 'tablet'). The right hand side node is a mixture of 3 'phones' and a 'tablet'.

After split



IMPURITY OF CHILDREN OF 'SCREEN SIZE=8'

Left (No):

$$\text{Gini Impurity} \left(\begin{matrix} G_1 S_1 A_1 \\ PPP \end{matrix} \right) = 1 - \left[\left(\frac{3}{3} \right)^2 + (0)^2 \right] = 0$$

Right (Yes):

$$\text{Gini Impurity} \left(\begin{matrix} A_2 \\ T \end{matrix} \right) = 1 - \left[(0)^2 + \left(\frac{1}{1} \right)^2 \right] = 0$$

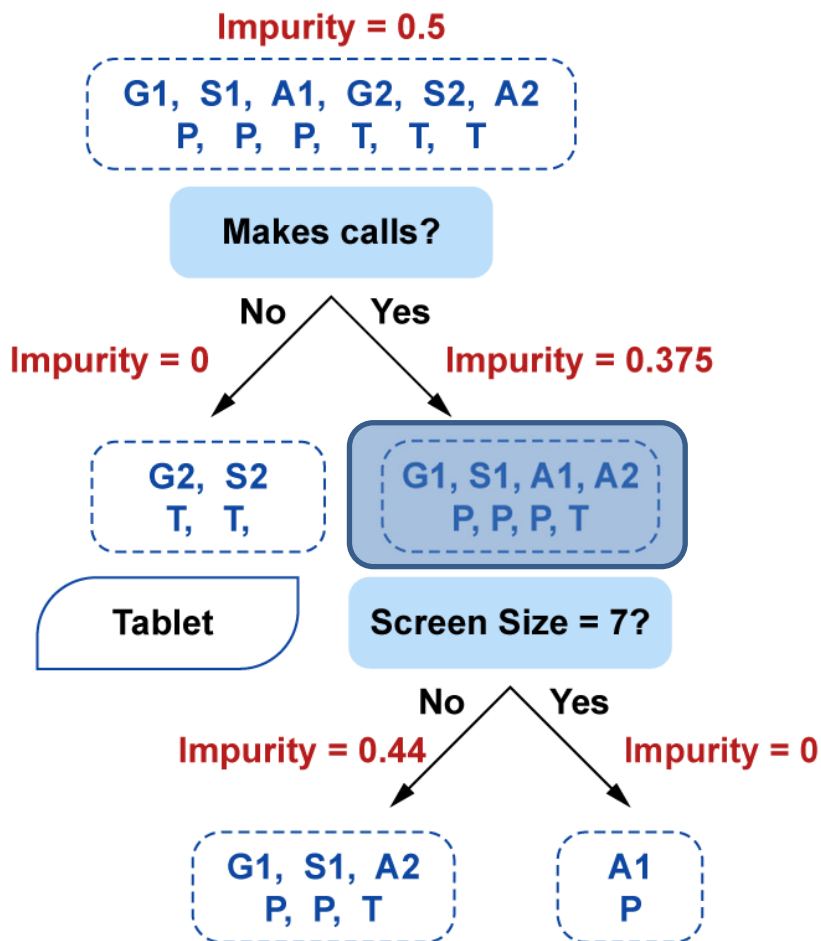
Info gain:

$$\begin{aligned} & \text{Info Gain('screen size = 8')} \\ &= 0.375 - \left(\frac{3}{4} \times 0 + \frac{1}{4} \times 0 \right) \\ &= 0.375 \end{aligned}$$

Step 2 of CART algorithm

Next, the CART algorithm will convert the branch on the left of the 'makes calls' into a leaf since it's a pure node (all of its data points are of class 'tablet'). The right hand side node is a mixture of 3 'phones' and a 'tablet'.

After split



IMPURITY OF CHILDREN OF 'SCREEN SIZE=7'

Left (No):

$$\text{Gini Impurity} \left(\begin{matrix} G_1 S_1 A_2 \\ P P T \end{matrix} \right) = 1 - \left[\left(\frac{2}{3} \right)^2 + \left(\frac{1}{3} \right)^2 \right] = 0.44$$

Right (Yes):

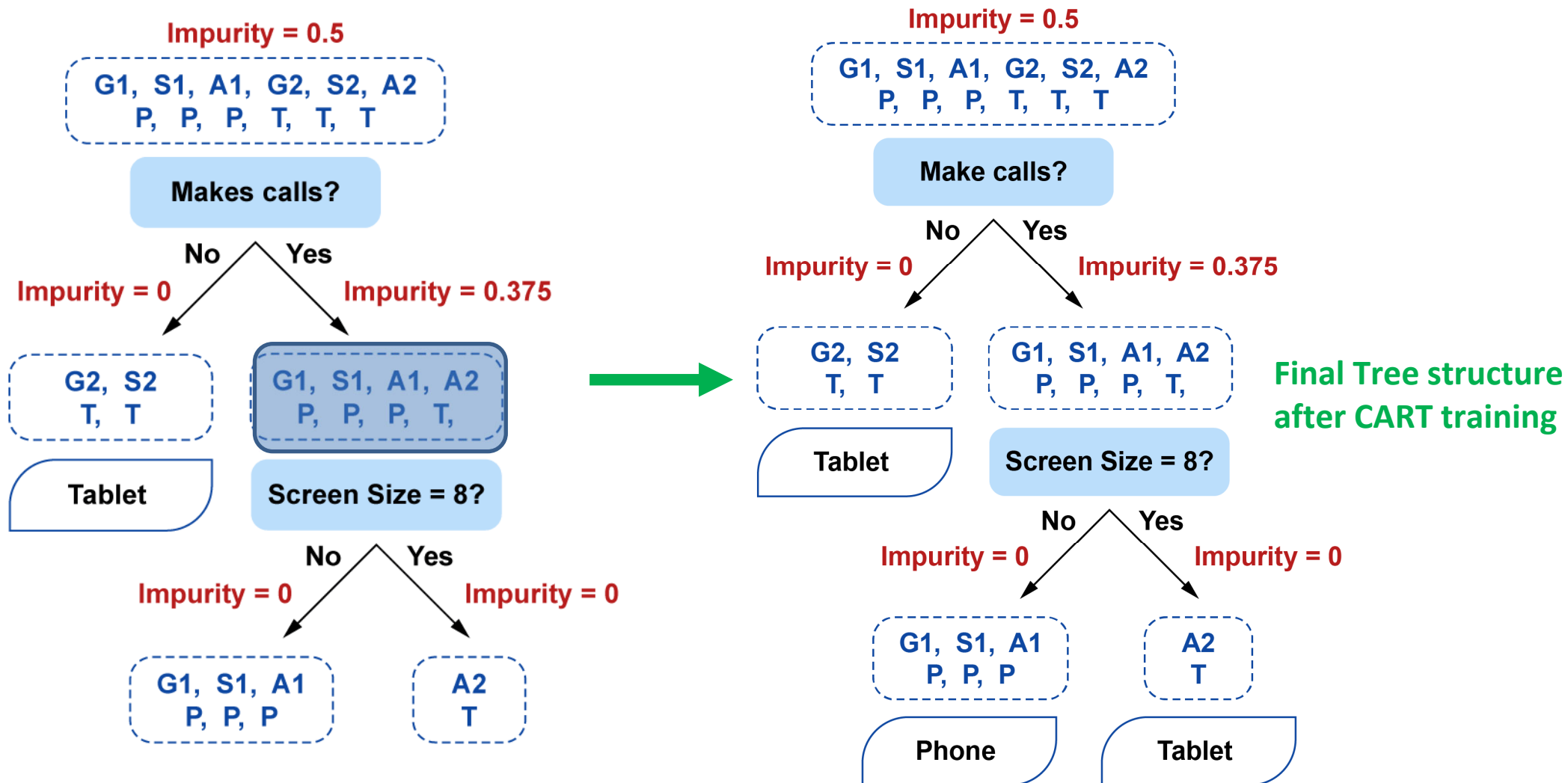
$$\text{Gini Impurity} \left(\begin{matrix} A_1 \\ P \end{matrix} \right) = 1 - \left[\left(\frac{1}{1} \right)^2 + (0)^2 \right] = 0$$

Info gain:

$$\begin{aligned} \text{Info Gain('screen size = 7')} \\ &= 0.375 - \left(\frac{3}{4} \times 0.44 + \frac{1}{4} \times 0 \right) \\ &= 0.0416 \end{aligned}$$

Step 2 of CART algorithm

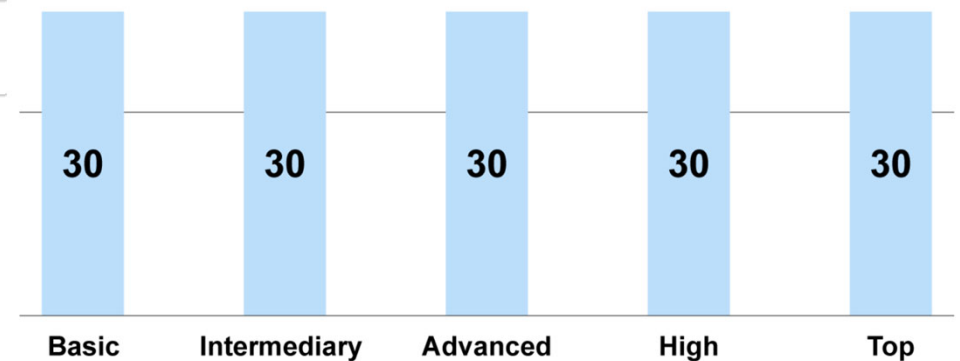
Next, the CART algorithm will convert the branch on the left of the 'makes calls' into a leaf since it's a pure node (all of its data points are of class 'tablet'). The right hand side node is a mixture of 3 'phones' and a 'tablet'.



Discretising continuous variables

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125	No
2	No	Married	100	No
3	No	Single	70	No
4	Yes	Married	120	No
5	No	Divorced	95	Yes
6	No	Married	60	No
7	Yes	Divorced	150	No
8	No	Single	85	Yes
9	No	Married	75	No
10	No	Single	90	Yes

Annual income ranges £K	Annual income Bands
[18, 48[Basic
[48, 78[Intermediary
[78, 108[Advanced
[108, 130[High
[130, 160[Top
[160, [Exec



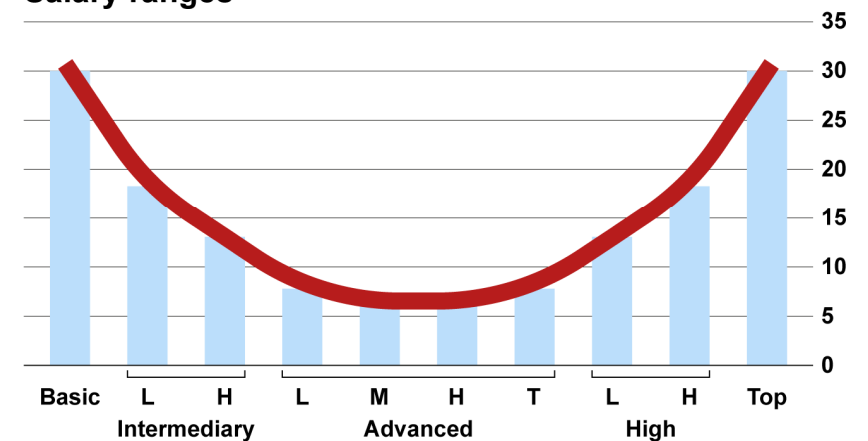
Income bands uniformly distributed

Discretising continuous variables

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125	No
2	No	Married	100	No
3	No	Single	70	No
4	Yes	Married	120	No
5	No	Divorced	95	Yes
6	No	Married	60	No
7	Yes	Divorced	150	No
8	No	Single	85	Yes
9	No	Married	75	No
10	No	Single	90	Yes

Annual Income Ranges £K	Annual income category
[18, 48[Basic
[48, 66[Intermediary L
[66, 78[Intermediary H
[78, 86[Advanced L
[86, 93[Advanced M
[93, 100[Advanced H
[100, 108[Advanced T
[108, 112[High L
[112, 130[High H
[130, 160[Top

Salary ranges

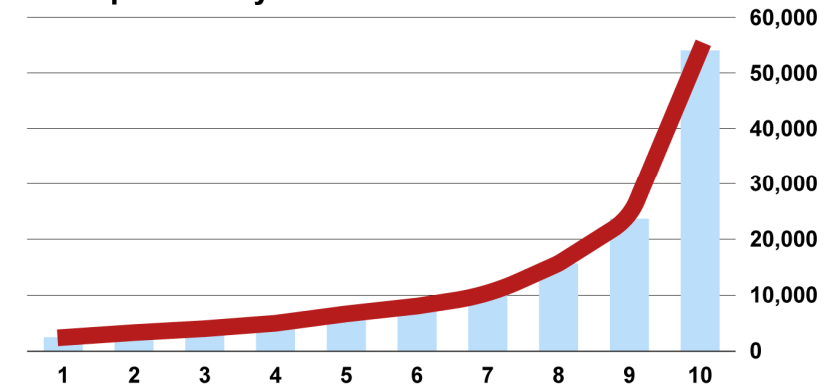


Discretising continuous variables

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125	No
2	No	Married	100	No
3	No	Single	70	No
4	Yes	Married	120	No
5	No	Divorced	95	Yes
6	No	Married	60	No
7	Yes	Divorced	150	No
8	No	Single	85	Yes
9	No	Married	75	No
10	No	Single	90	Yes


Annual income ranges £K	Annual income increment £K	Annual income category
54,900	2,300	Basic
57,700	2,800	Intermediary L
61,000	3,300	Intermediary H
65,000	4,000	Advanced L
70,200	5,200	Advanced M
76,800	6,600	Advanced H
86,000	9,200	Advanced T
98,600	12,600	High L
121,000	22,400	High H
175,000	54,000	Exec

UK top 10 salary increments



Discretising continuous variables

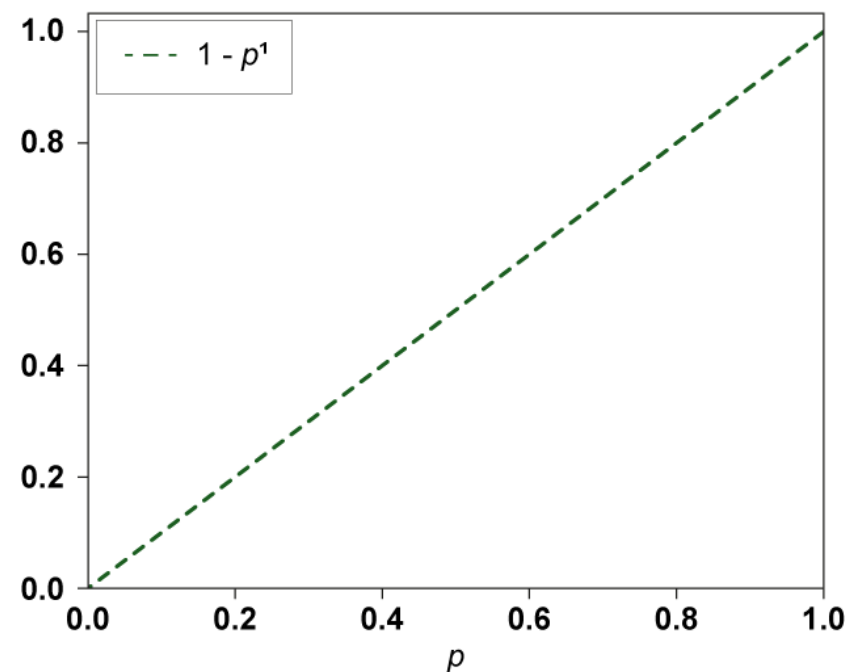
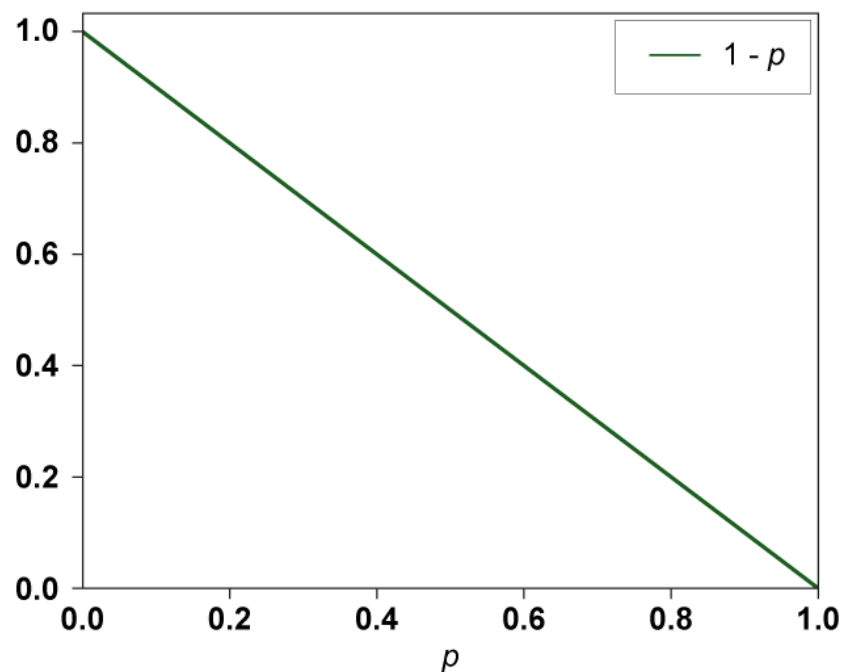
ID	Home owner	Marital status	Annual income	Defaulted borrower	Possible splits for annual income
6	No	Married	60	No	
3	No	Single	70	No	65
9	No	Married	75	No	72.5
8	No	Single	85	Yes	80
10	No	Single	90	Yes	87.5
5	No	Divorced	95	Yes	92.5
2	No	Married	100	No	97.5
4	Yes	Married	120	No	110
1	Yes	Single	125	No	122.5
7	Yes	Divorced	150	No	137.5



Gini Index

For a 2-class problem:

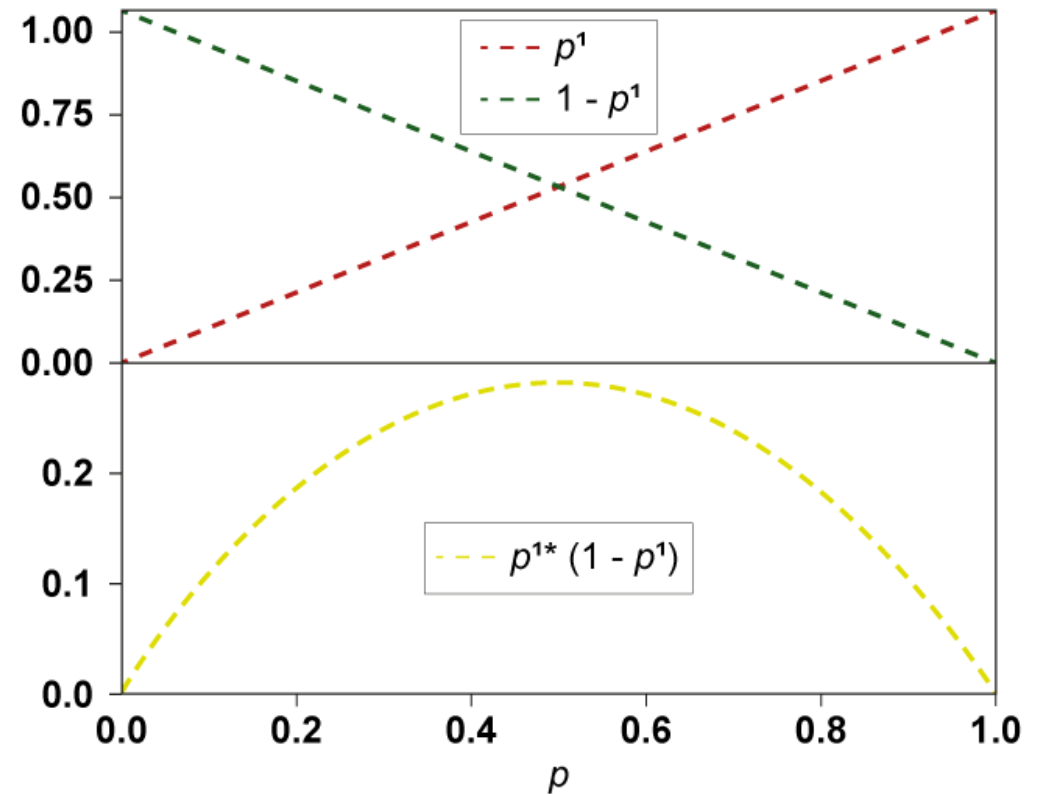
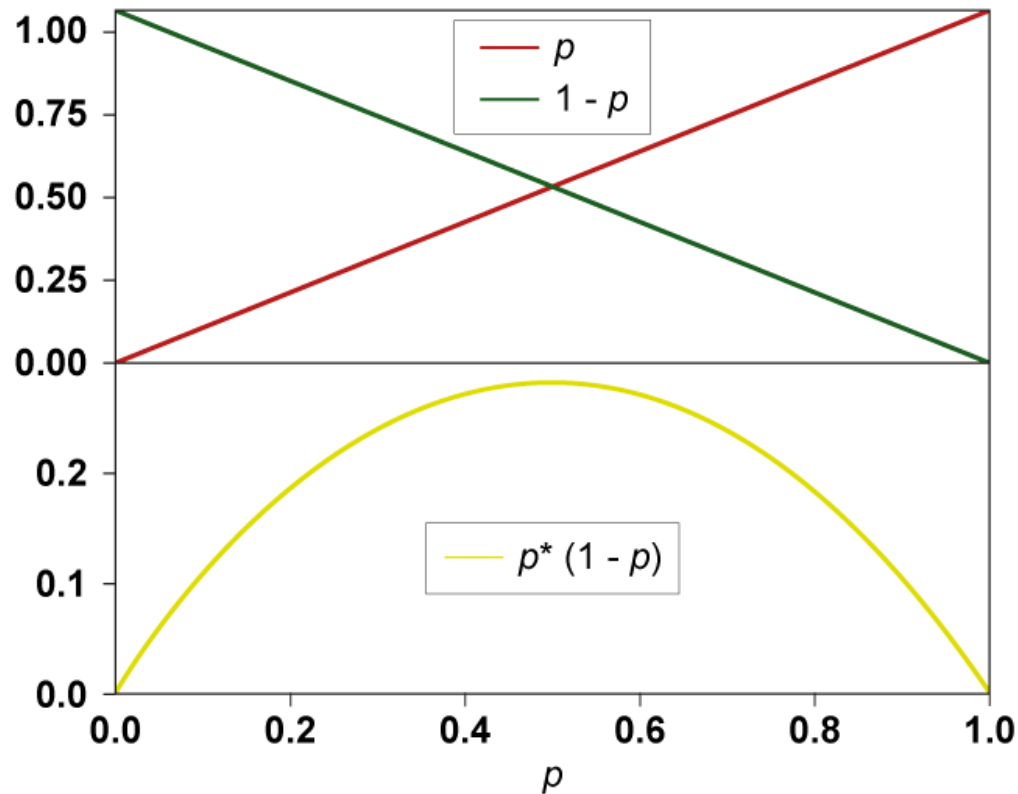
- When the probability p is low, the Gini is low. Hence, we simply include p in Gini formula.
- When the probability p is high, the Gini is low. Hence, we include the term $1-p$ in the Gini formula.
- To take into account both of the points above, the Gini index should include the term $p(1-p)$.



Gini Index

For a 2-class problem:

- Gini impurity for class C1 = $p(1-p)$, C1 has probability p .
- Same for C2 with p'



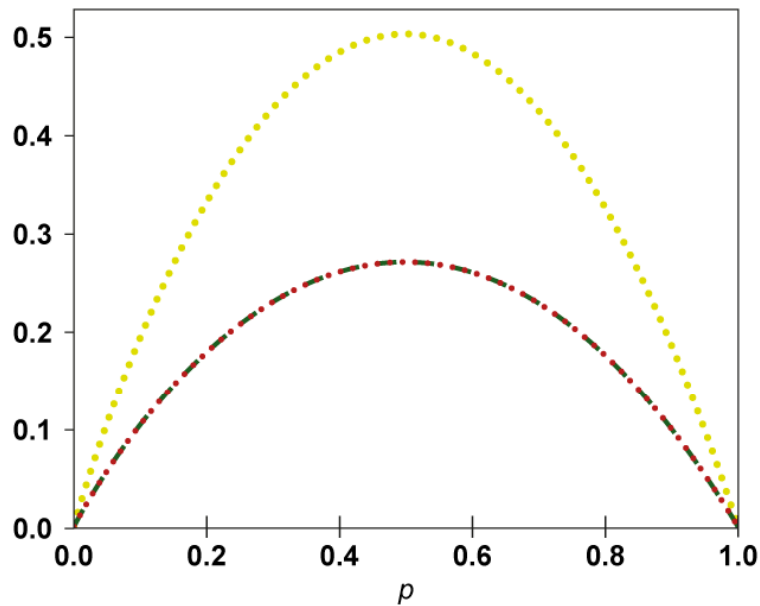
Gini Index

For a 2-class problem:

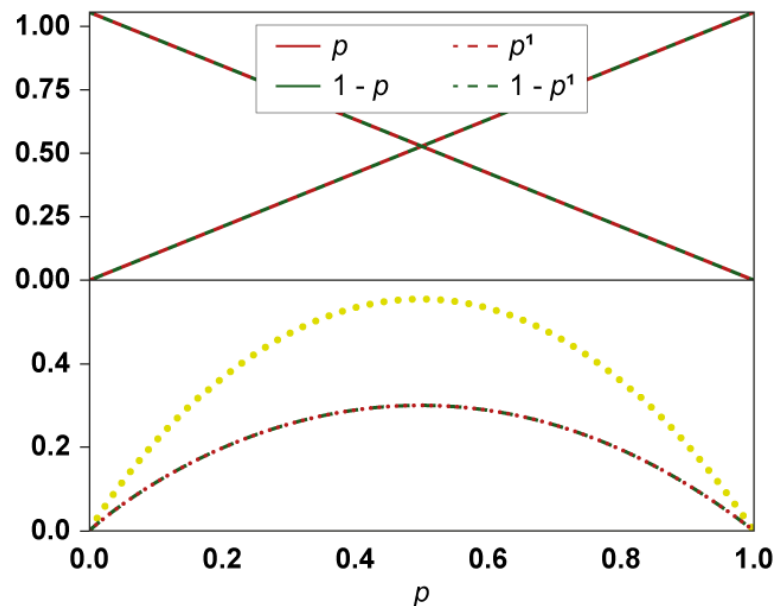
$$\text{Gini} = p(1 - p) + p'(1 - p')$$

In the case of Gini impurity it is helpful to realise that $p + p' = 1$ hence:

$$\text{Gini} = p(1 - p) + p'(1 - p') = (p + p') - (p^2 + p'^2) = 1 - (p^2 + p'^2)$$



$\text{Gini}(C1) = -p \log(p)$
 $\text{Gini}(C2) = -p' \log(p')$
 $\text{Gini}(C1, C2) = p^*(1 - p) + p'^*(1 - p')$



$\text{Gini}(C1) = -p \log(p)$
 $\text{Gini}(C2) = -p' \log(p')$
 $\text{Gini}(C1, C2) = p^*(1 - p) + p'^*(1 - p')$

Gini Index

For a 2-class problem:

$$\text{Gini} = p(1 - p) + p' (1 - p')$$

In the case of Gini impurity it is helpful to realise that $p + p' = 1$ hence:

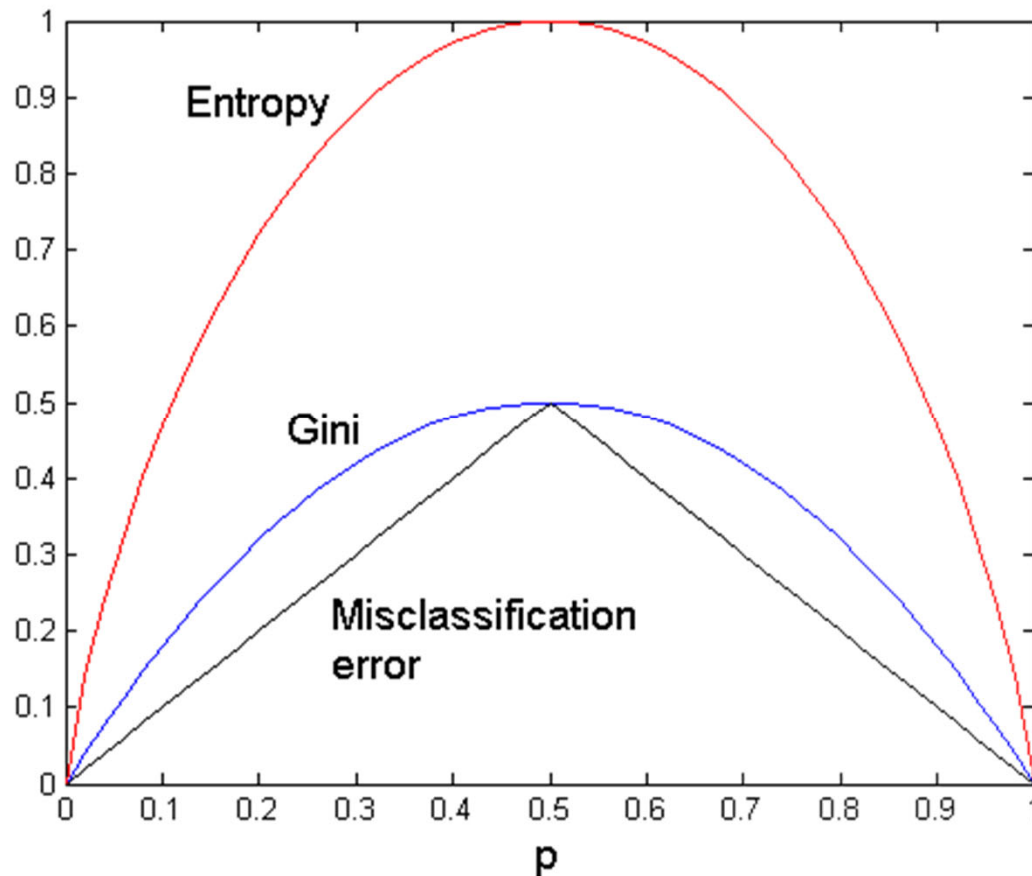
$$\text{Gini} = p(1 - p) + p' (1 - p') = (p + p') - (p^2 + p'^2) = 1 - (p^2 + p'^2)$$

For a K-class problem:

$$\text{Gini} = \sum_{i=1}^K p_i (1 - p_i) = 1 - \sum_{i=1}^K p_i^2$$

Comparison among Impurity Measures

For a 2-class problem:



$$\text{Classification error} = 1 - \max(p_i)$$

when the two classes have a probability of 0.5
(remember if $p=0.5$ then $1-p=0.5$)

then the entropy is maximal =1.

It fades away when one of the classes has high probability

the higher the probability the lower the entropy until it reaches 0, when the probability of either classes is 1 (same for one of the classes probability is close to 0 the other would be close to 1).

Entropy

For a 2-class problem:

For Class 1 with probability p , we want to make sure that:

1. When the probability p is low, the Entropy is low. Hence, we simply include p in Entropy formula at the same time.

2. When the probability p is high, the Entropy is low. Hence, we include the term $-\log(p)$ in the Entropy formula.

Note that $\log(p) \leq 0$ because $p \leq 1$. Hence $-\log(p) \geq 0$.

Note that $-\log(p)$ is monotonically decreasing function.

Note that the base of \log is normally 2 but any can do as long as we are consistent.

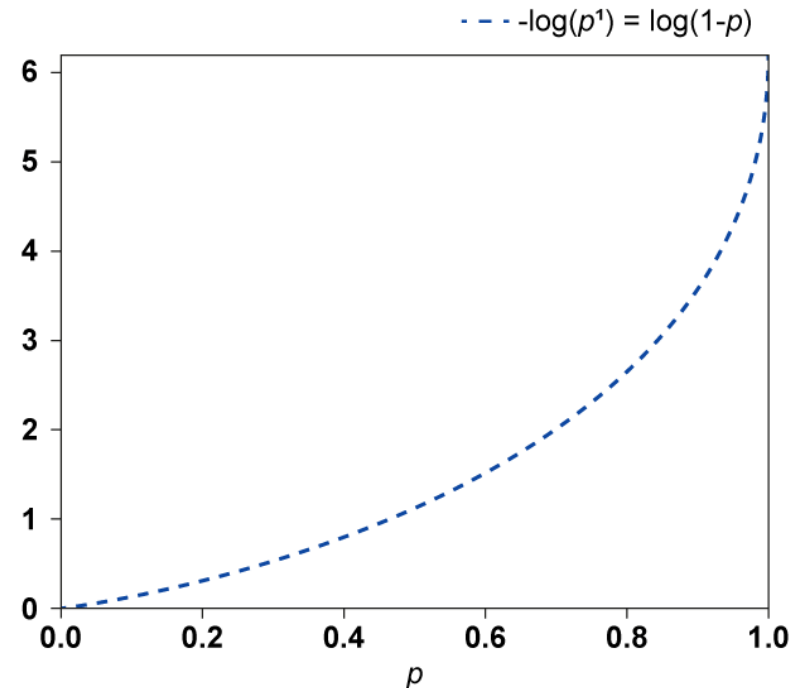
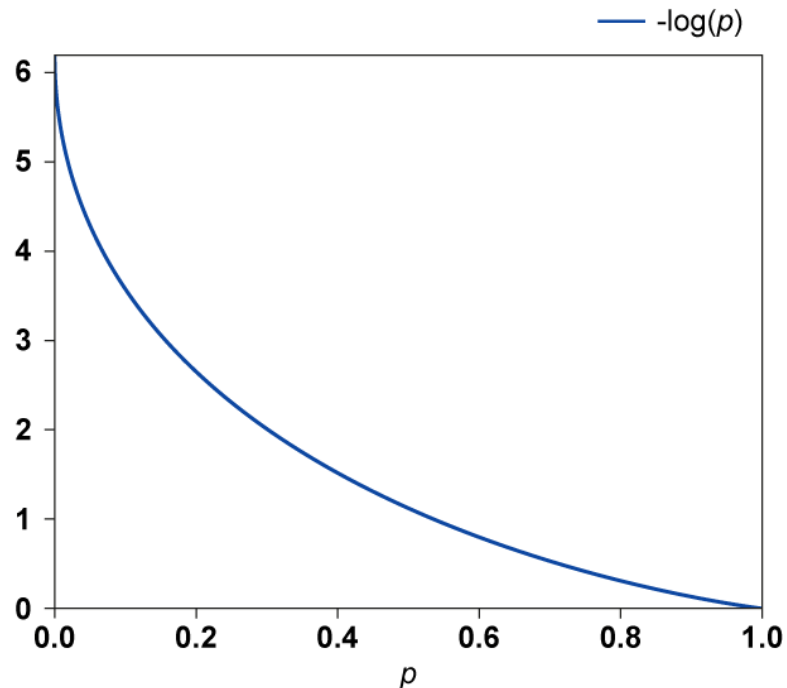
Entropy

For a 2-class problem:

The behaviour of $-\log(p)$ for class C1 and $-\log(p')$ for class C2 can be seen below.

When the probability increases $-\log(p)$ decreases but it is still non-negative.

Note that $-\log(p')$ is monotonically increasing function and is non-negative as well.

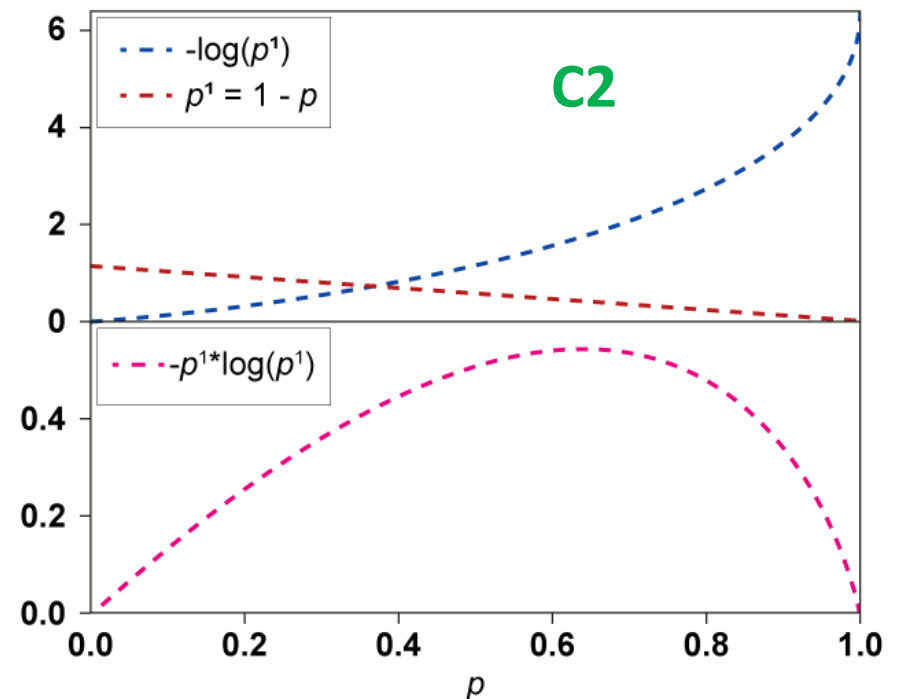
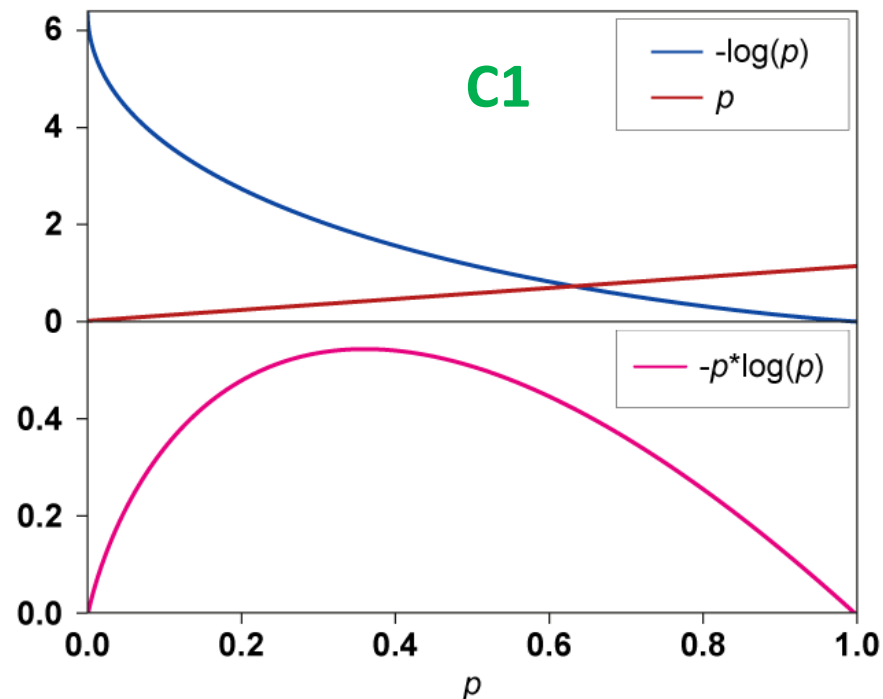


Entropy

For a 2-class problem:

To take into account both of the points above:

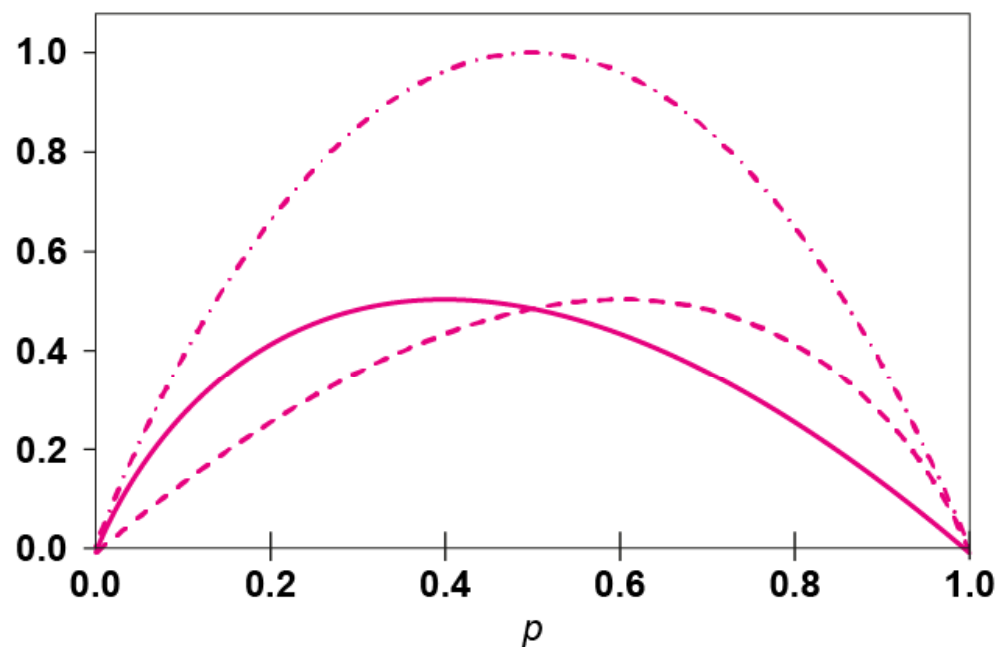
- the entropy for class C1 is $-\log(p)$,
- since we have two classes then we also need a similar term for the second class C2.
- given that C2 has a probability $p'=1-p$, its entropy is $(1-p)\log(1-p)$.



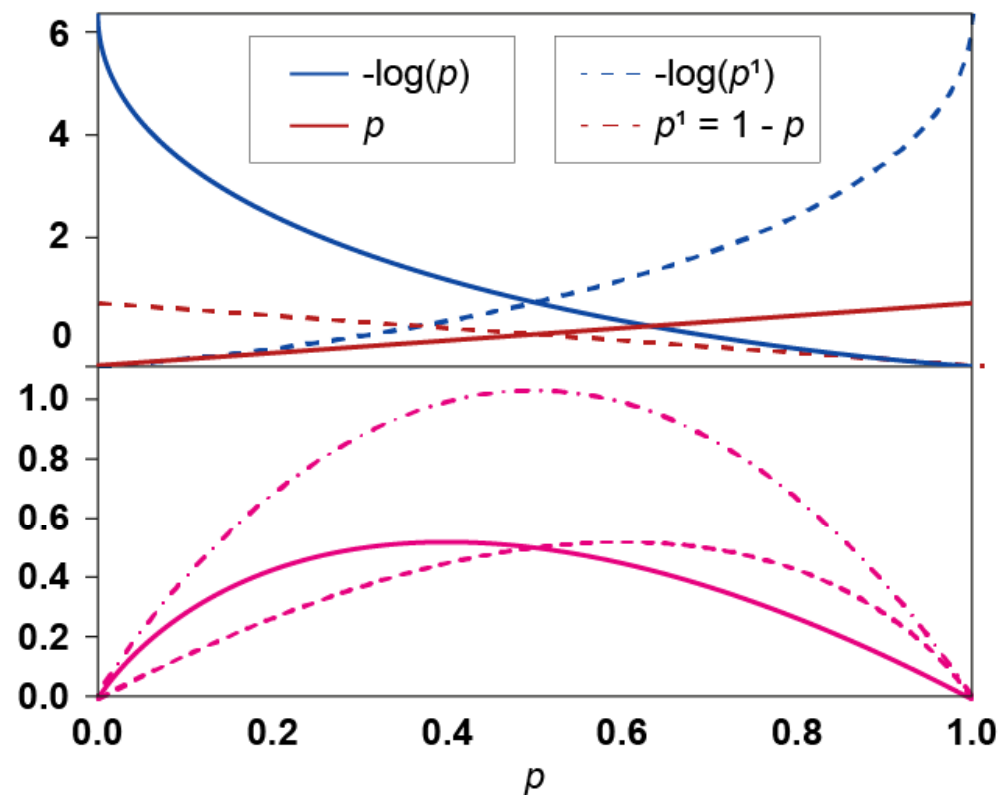
Entropy

For a 2-class problem:

$$\text{Entropy} = -p \log p - p' \log p'$$
$$\text{Entropy} = -p \log p - (1 - p) \log(1 - p)$$



— Entropy(C1) = $-p \log(p)$
- - Entropy(C2) = $-p' \log(p')$
· · Entropy(C1, C2) = $-p \log(p) - p' \log(p')$



— Entropy(C1) = $-p \log(p)$
- - Entropy(C2) = $-p' \log(p')$
· · Entropy(C1, C2) = $-p \log(p) - p' \log(p')$

Entropy

For a 2-class problem:

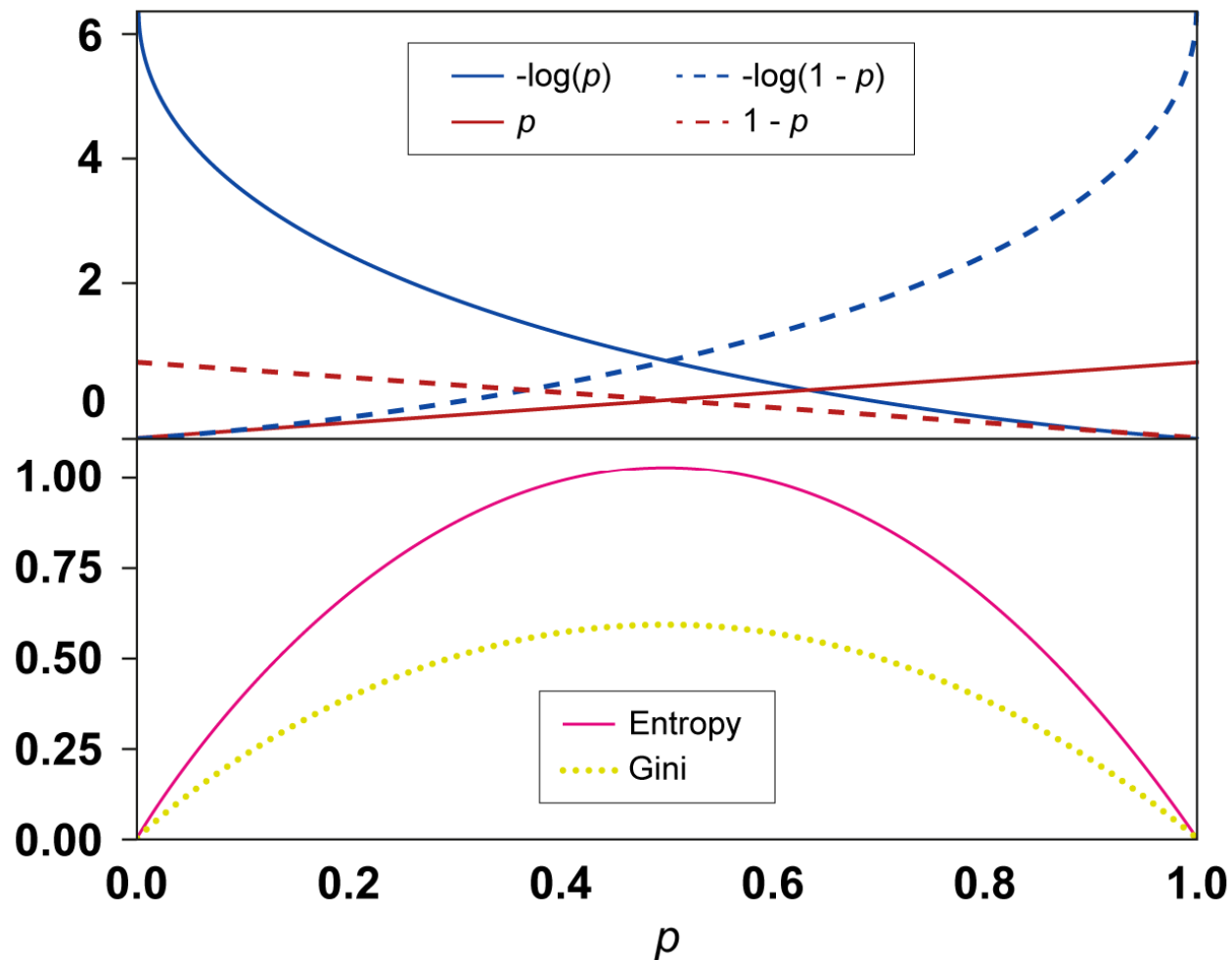
$$\begin{aligned}\text{Entropy} &= -p \log p - p' \log p' \\ \text{Entropy} &= -p \log p - (1 - p) \log(1 - p)\end{aligned}$$

For a K-class problem:

$$\text{Entropy} = - \sum_{i=1}^K p_i \log p_i$$

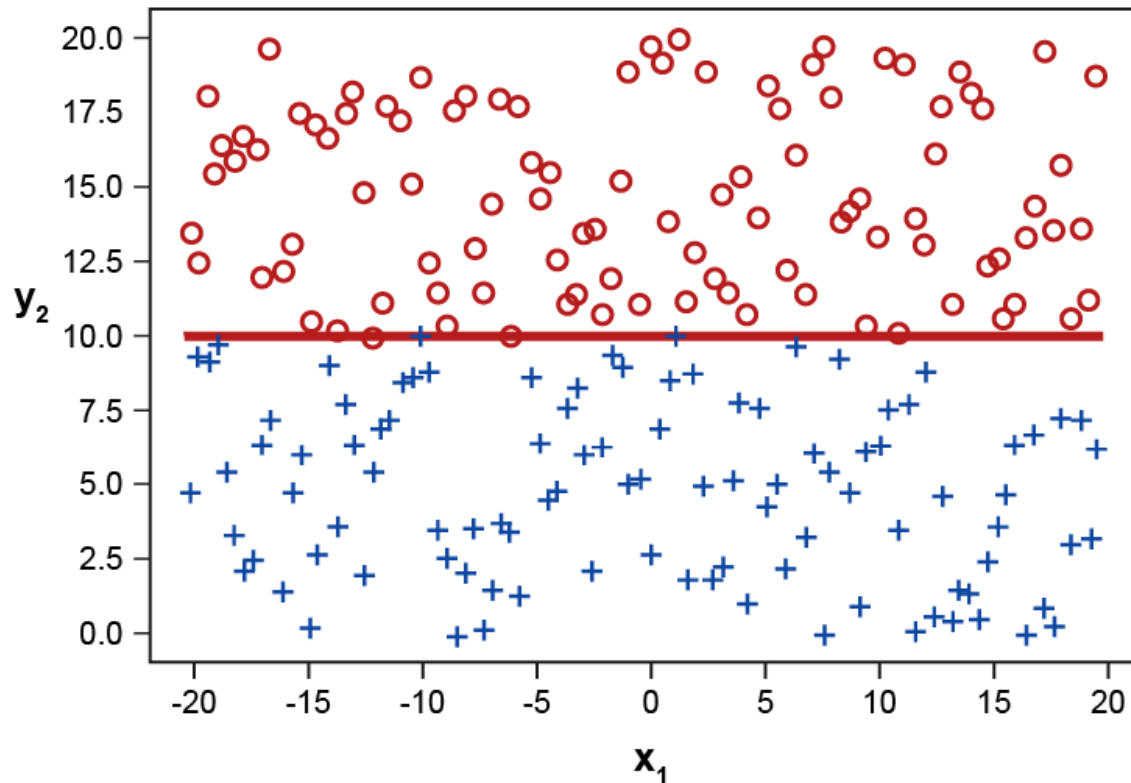
Gini vs Entropy

both produce consistent trees and have a similar behaviour albeit having different ranges



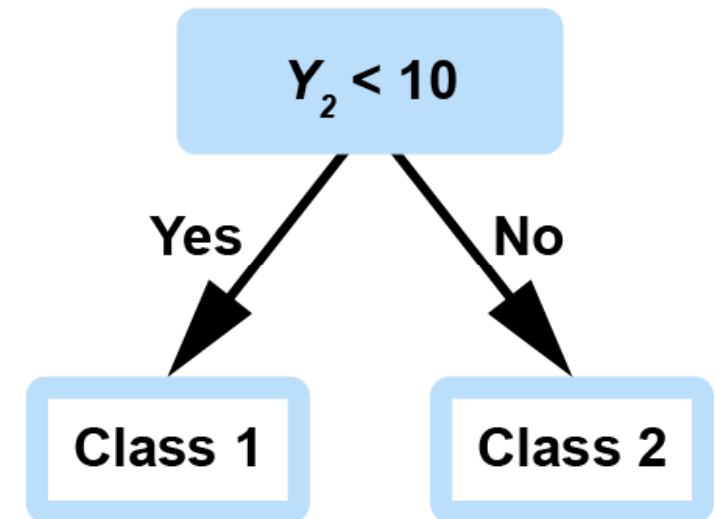
Decision boundaries of decision trees

Binary class balanced data with linearly separable decision boundary



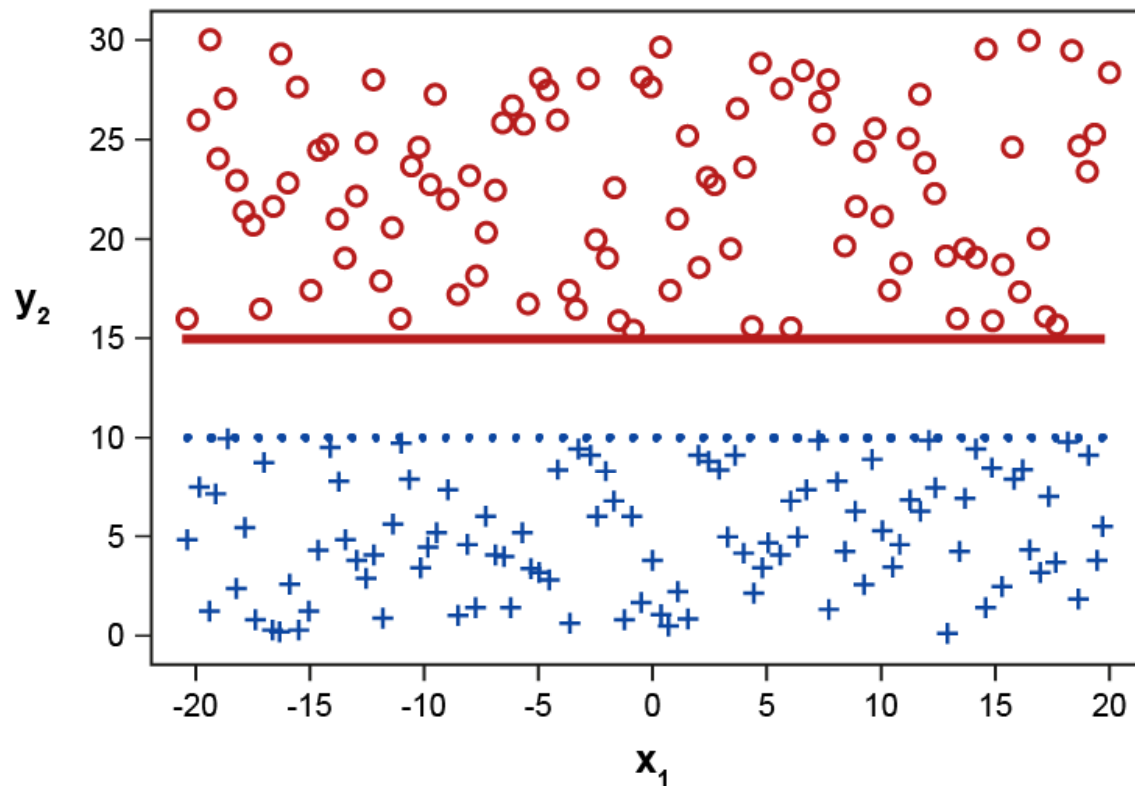
Key

— Decision boundary + Class 1 o Class 2



Decision boundaries of decision trees

Binary class balanced data with large margin linearly separable decision boundaries



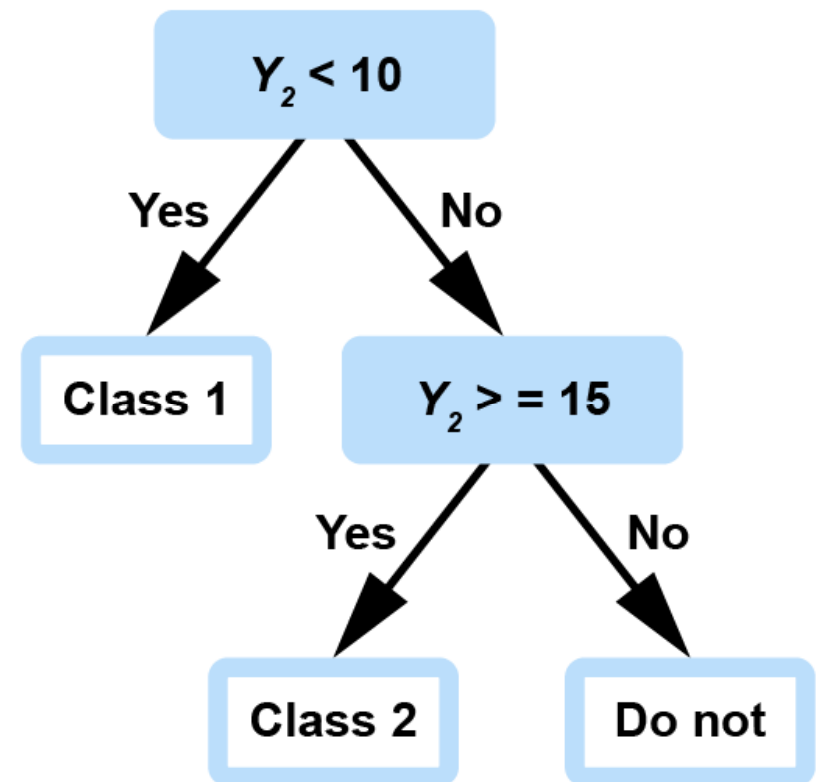
Key

..... Decision boundary 1

————— Decision boundary 2

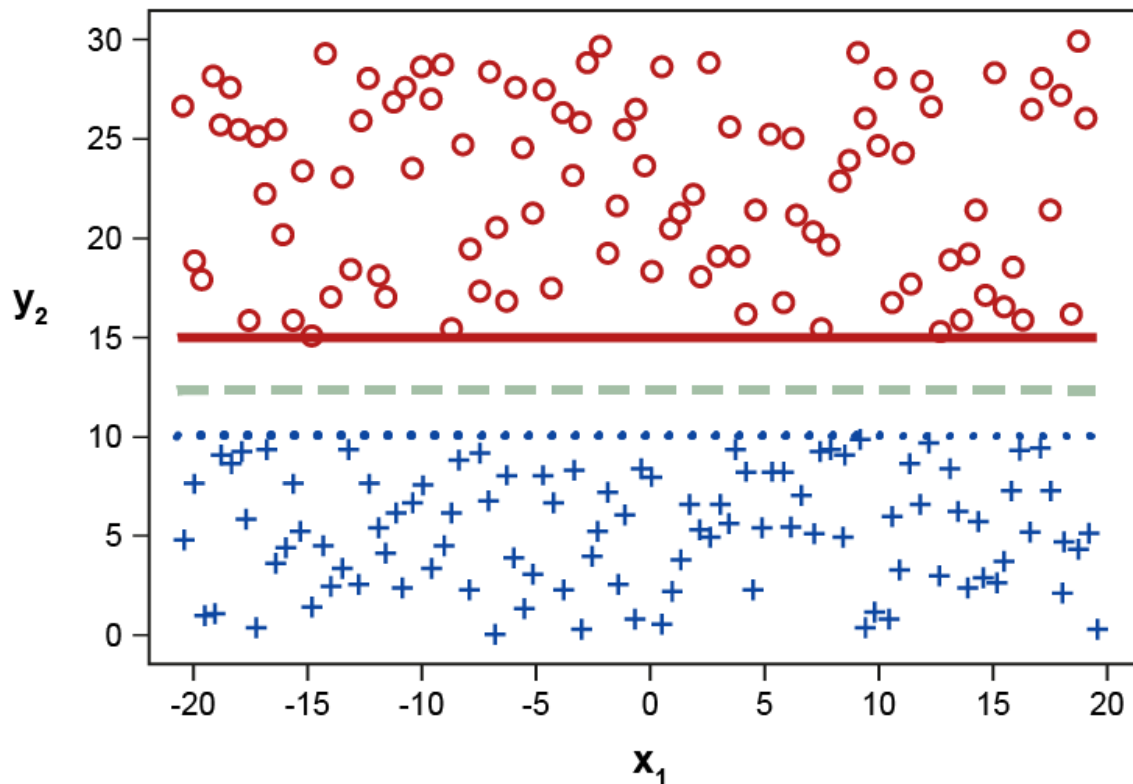
+ Class 1

o Class 2



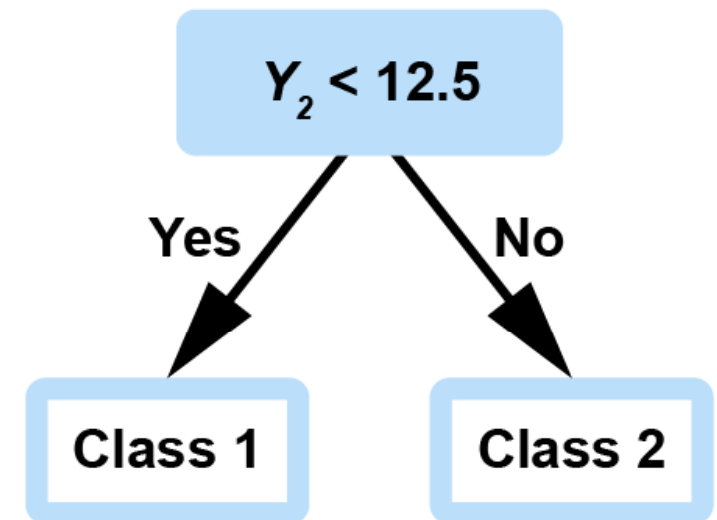
Decision boundaries of decision trees

Binary class balanced data with large margin linearly separable decision boundaries



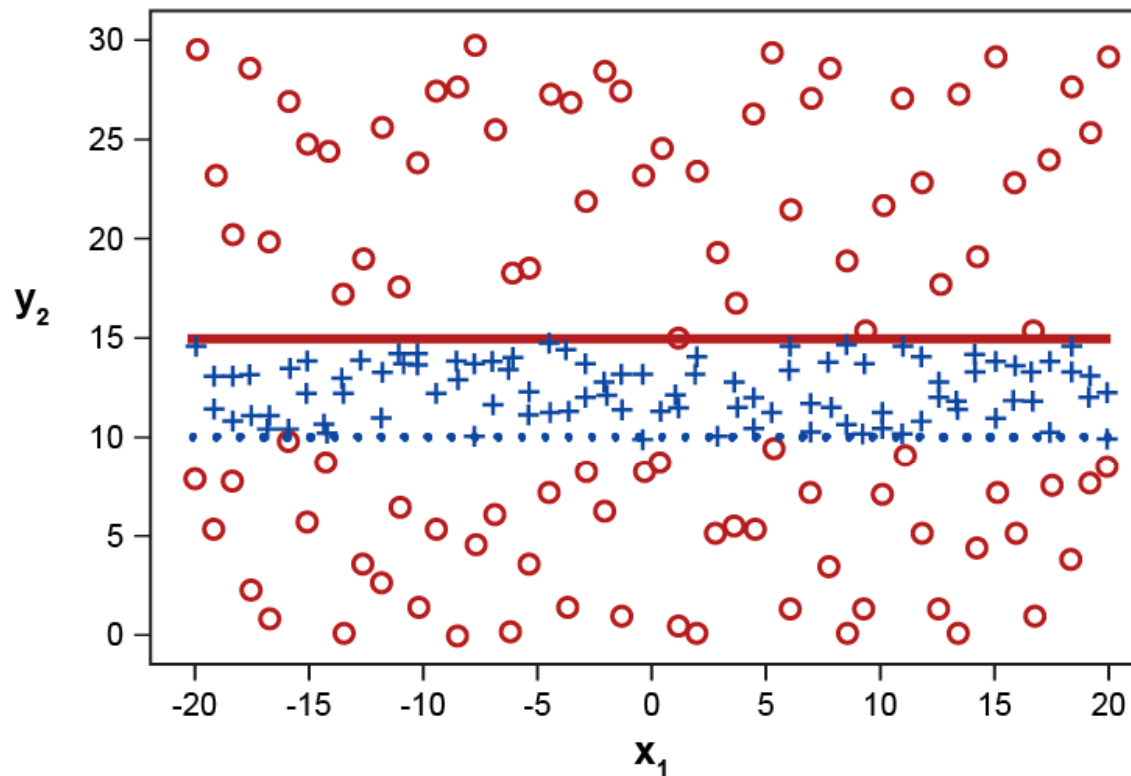
Key

- • Decision boundary 1
- + Class 1
- Decision boundary 2
- o Class 2
- - - Decision boundary 3



Decision boundaries of decision trees

Binary class balanced data with non-linearly separable decision boundaries



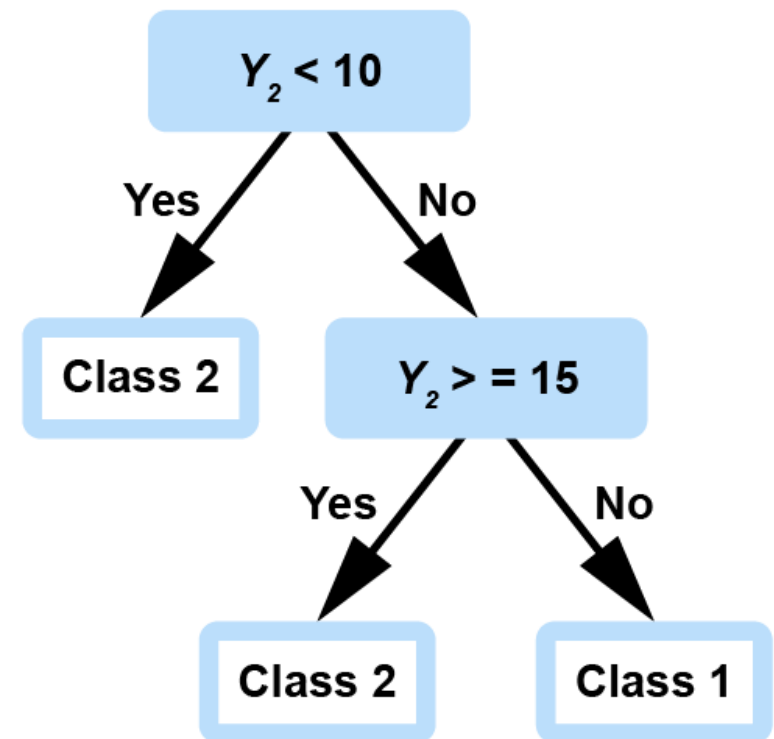
Key

..... Decision boundary 1

———— Decision boundary 2

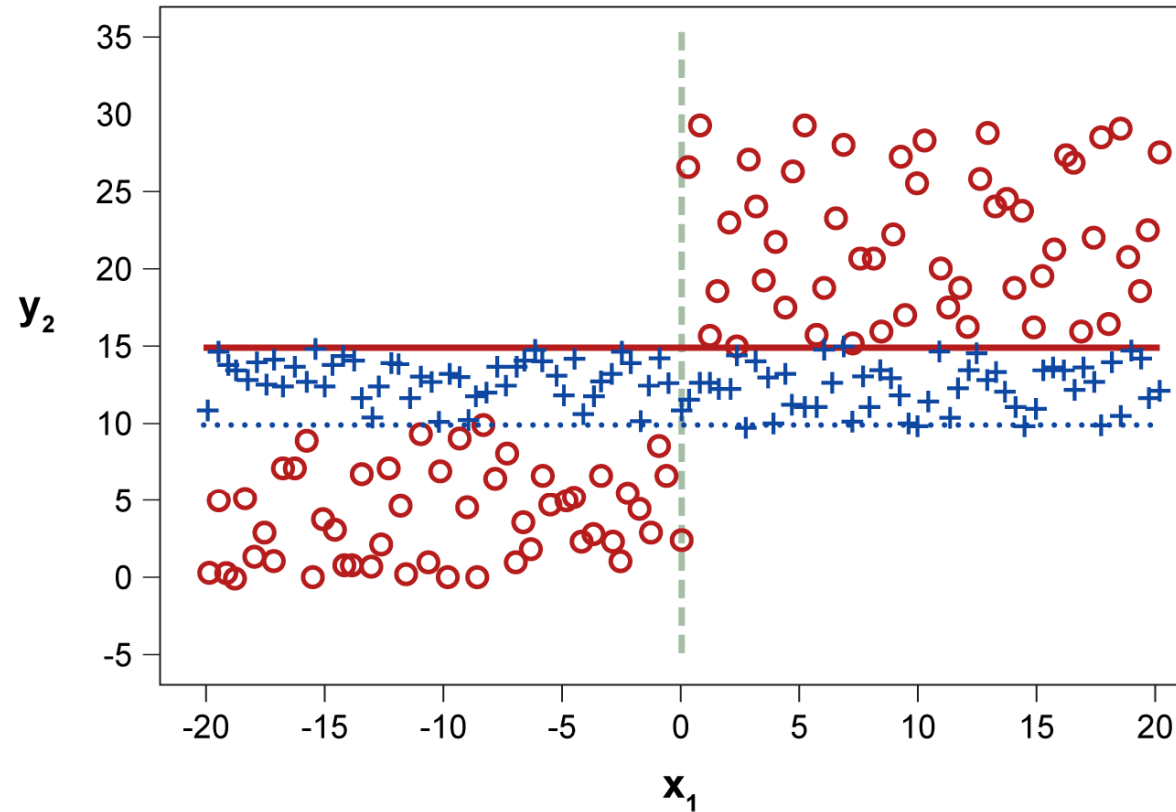
+ Class 1

o Class 2



Decision boundaries of decision trees

Binary class balanced data with non-linearly separable decision boundaries quartiles

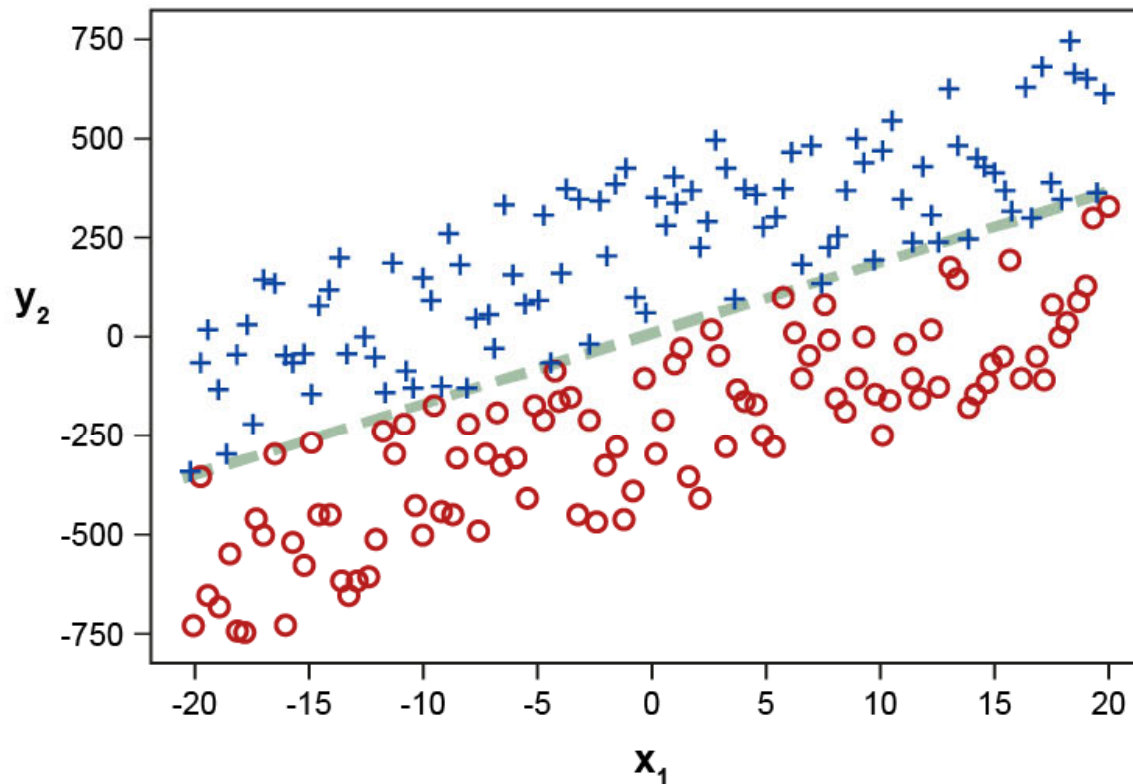


Key

- Decision boundary 1
- Decision boundary 2
- - - Decision boundary 3
- + Class 1
- Class 2

Decision boundaries of decision trees

Binary class balanced data with linearly separable diagonal decision boundary

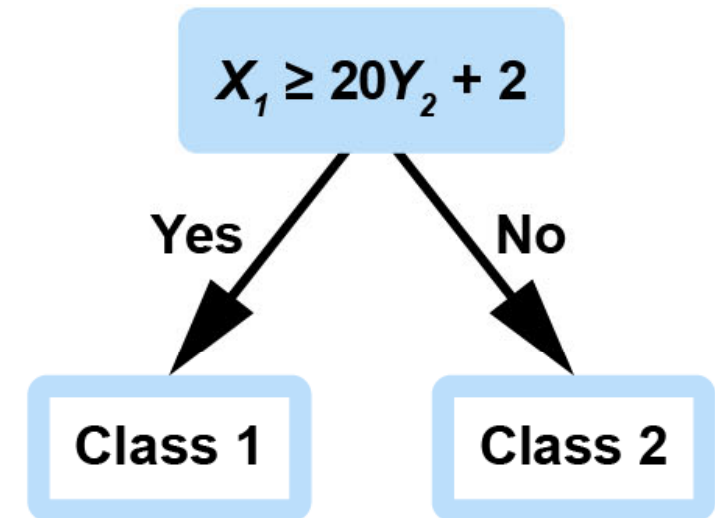


Key

— — — Decision boundary

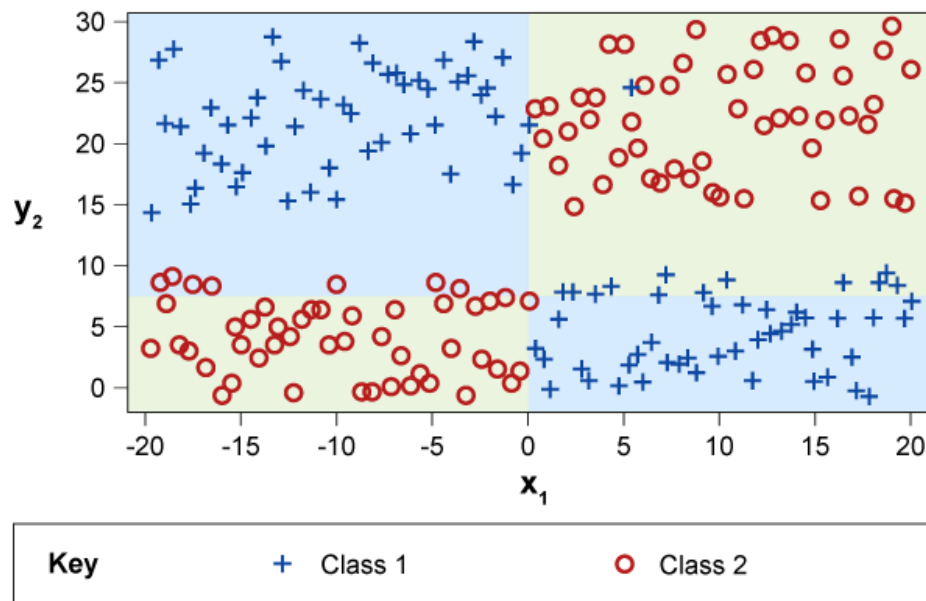
+ Class 1

o Class 2

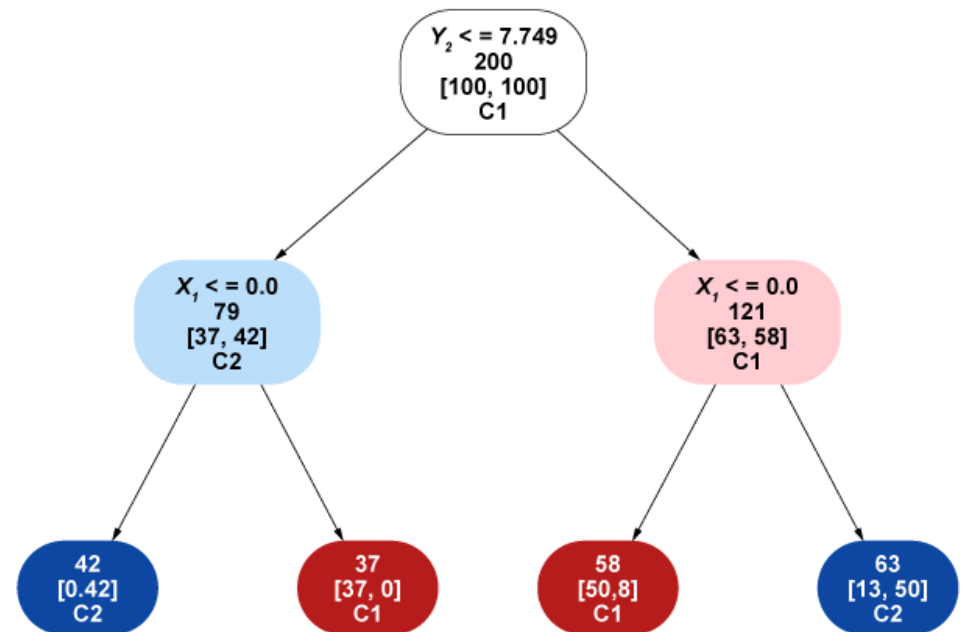


Decision boundaries of decision trees

Sandwich class 1 boundary with class 2 quartered boundaries

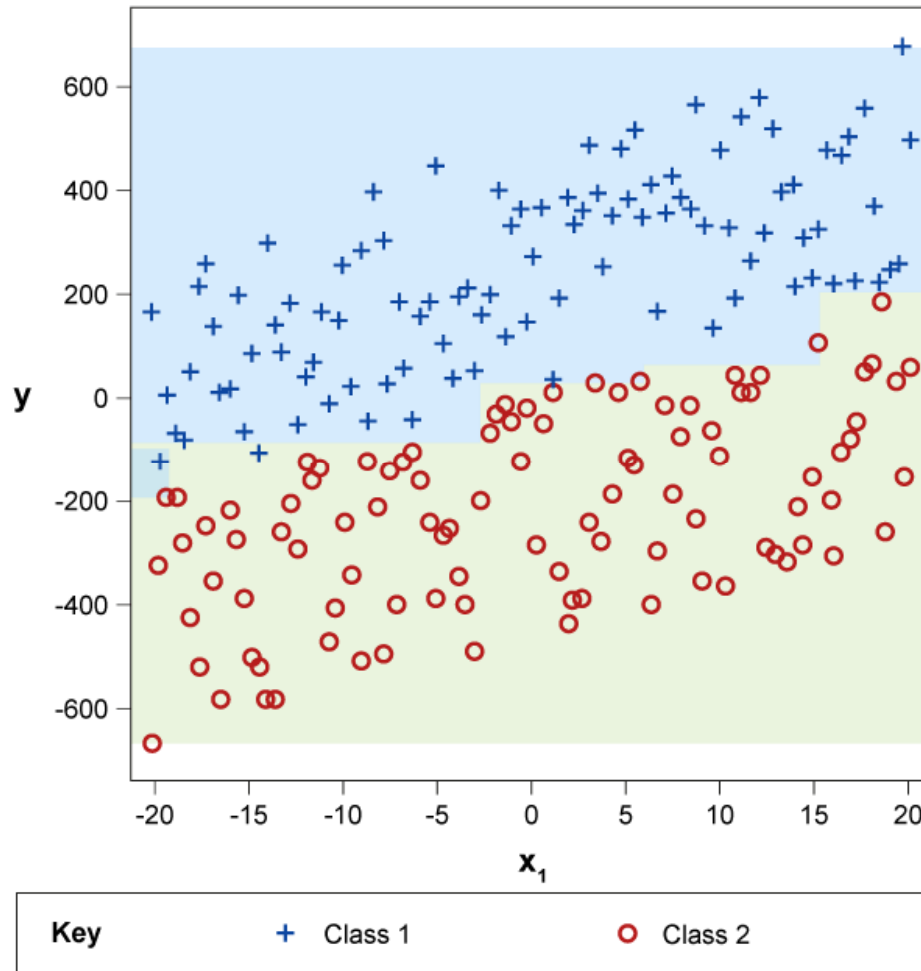


Decision tree (depth = 2)

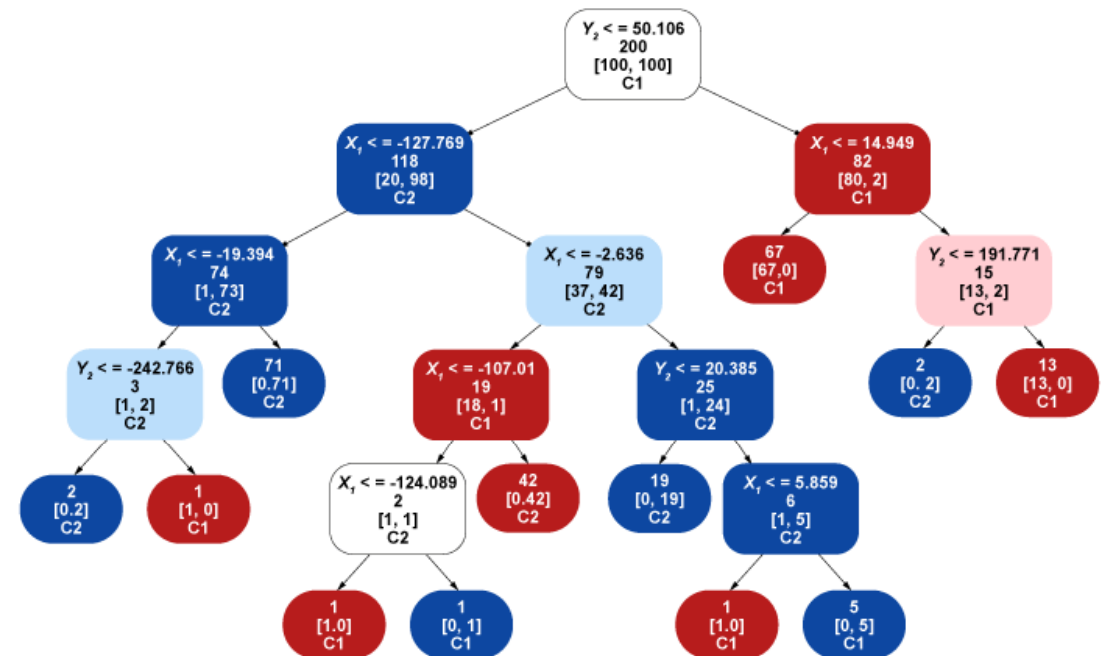


Decision boundaries of decision trees

Diagonal classes boundaries



Decision tree (depth = 5)



The DT is not very efficient, as the CART deals with one feature at a time in its conditions.

Decision Trees: Pros & Cons

- DTs are simple to implement and easy to interpret
- Can handle mixed feature types (continuous and/or categorical) and missing features
- Can be used for binary and multi-class classification, and regression
- DTs have high variance, i.e. sensitive to small changes in the data
- Not very good at handling continuous valued data
- Can easily overfit to your data - in practice need small but informative trees (i.e. regularisation)

Decision Trees: Pros & Cons

- Vanilla DTs are however not competitive with other supervised learning approaches
- But, DTs when ensembled are competitive - random forests and gradient boosting
- Ensembling methods grow and combine multiple trees to provide a consensus prediction (i.e. reduce variance and bias)
- Improvement in prediction accuracy with ensembling techniques is often at the price of interpretability