Decision Trees

Introduction

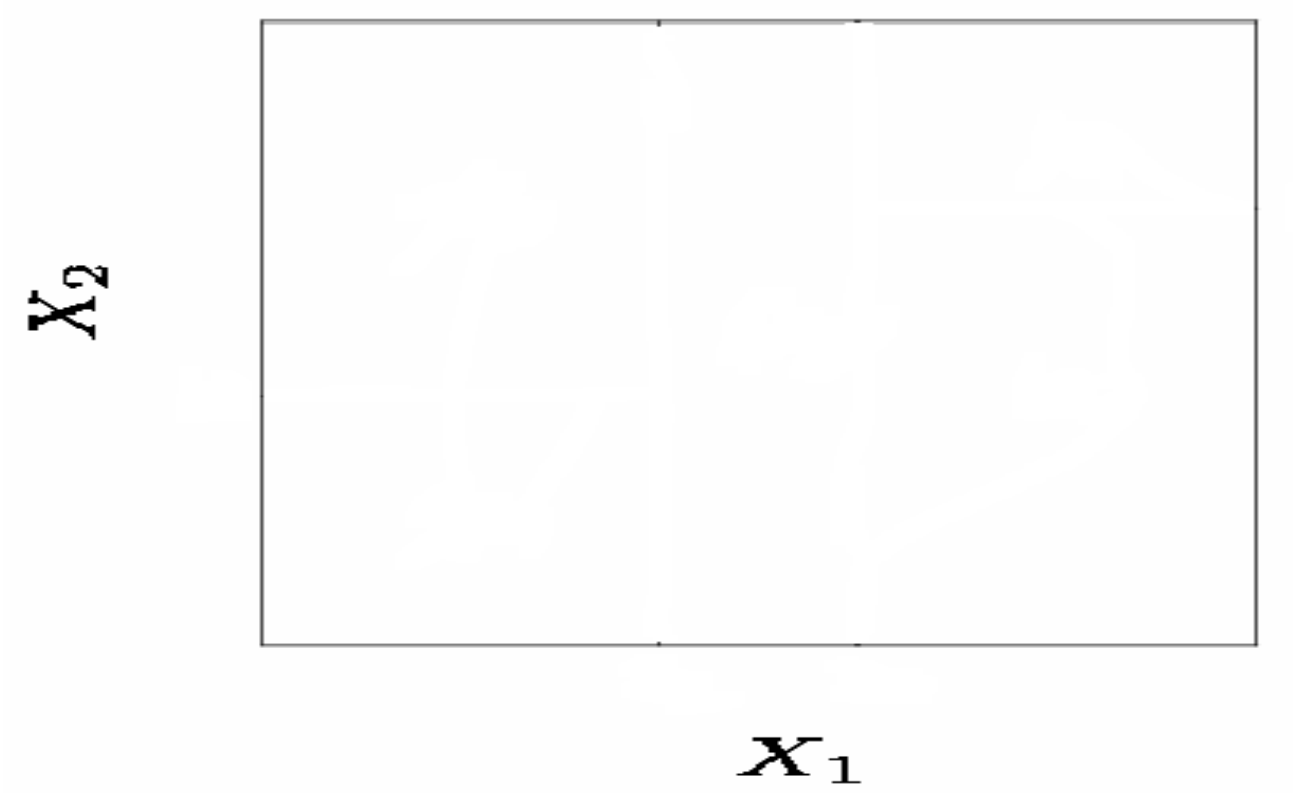
- We now discuss *tree-based* methods.
- Note that our main goal is to predict a target variable based on several input variables.
- Decision trees can be applied to both regression and classification problems.

Introduction

- Thus there are of two main types:
  1. Classification tress used when the predicted outcome is a categorical variable.
  2. Regression tress used when the predicted outcome is a quantitative variable.
- The term **Classification And Regression Tree (CART)** analysis is a popular umbrella term used to refer to both of the above methods.

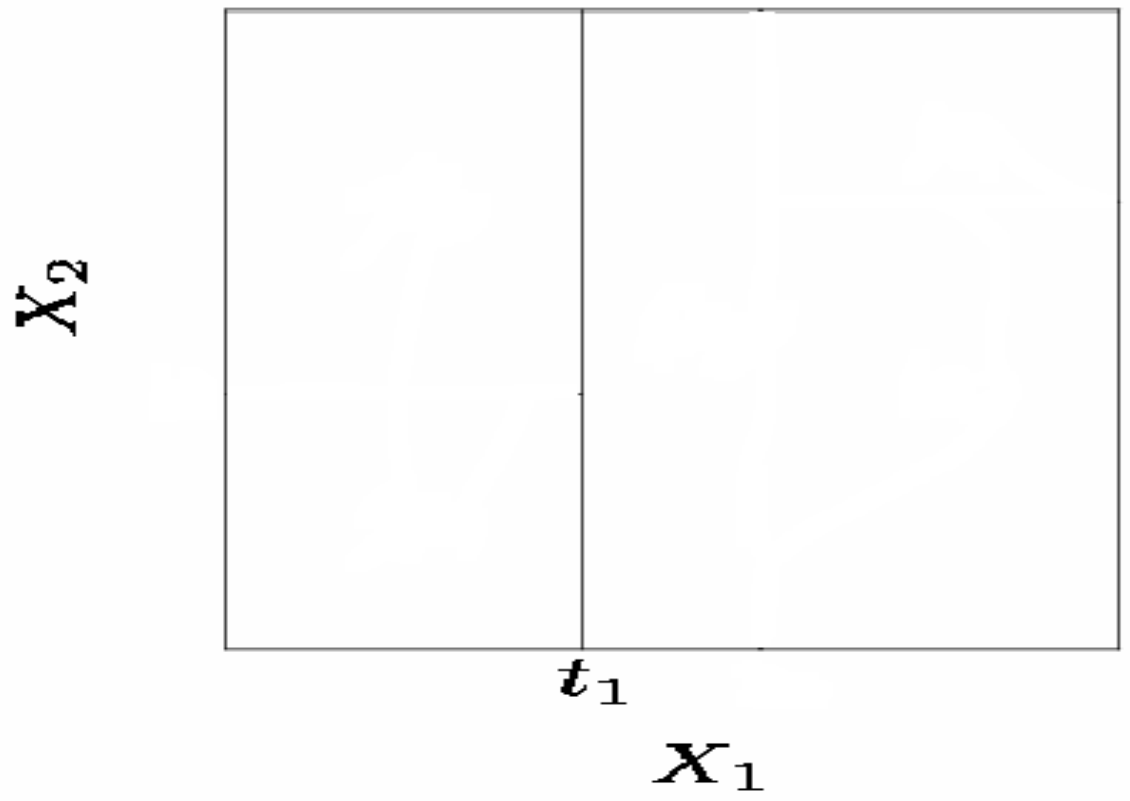
The General View

- Typically we create the partitions by iteratively splitting one of the X variables into two regions.



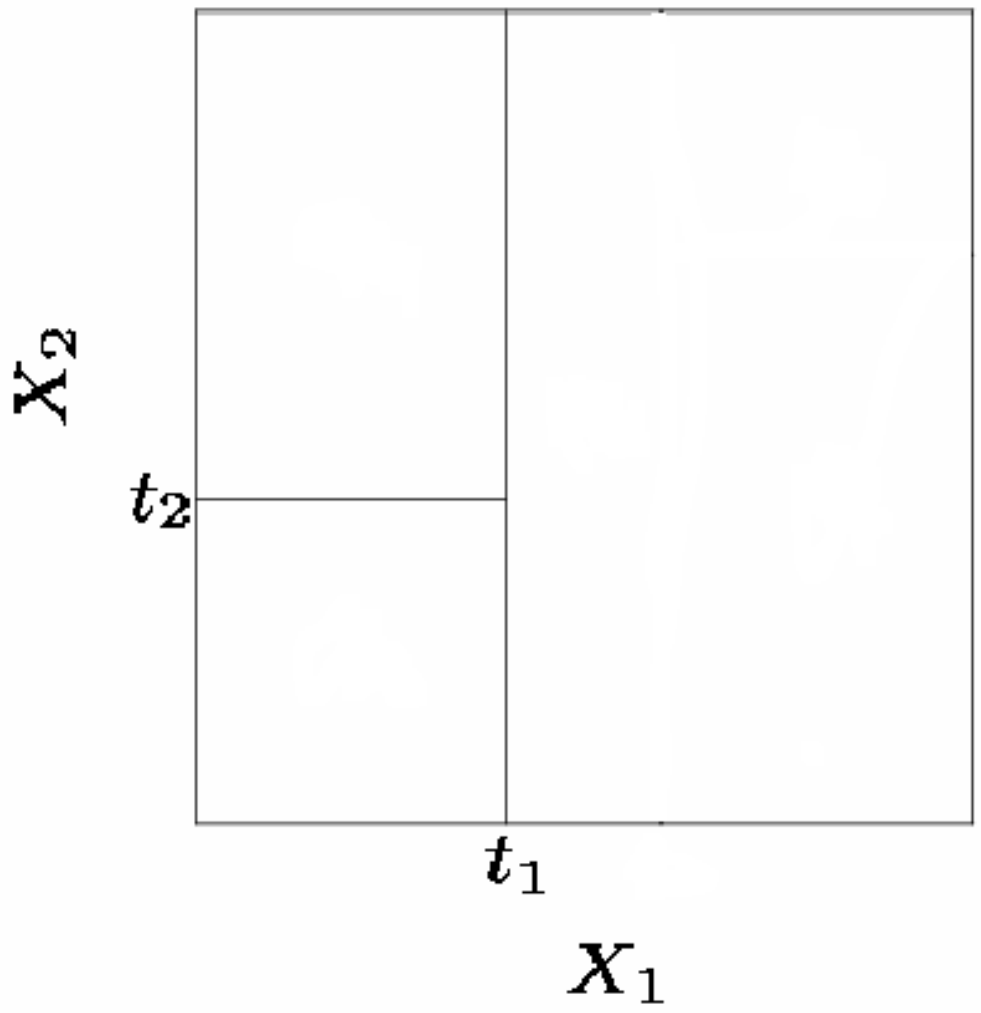
The General View

1. First split on $X_1 = t_1$.



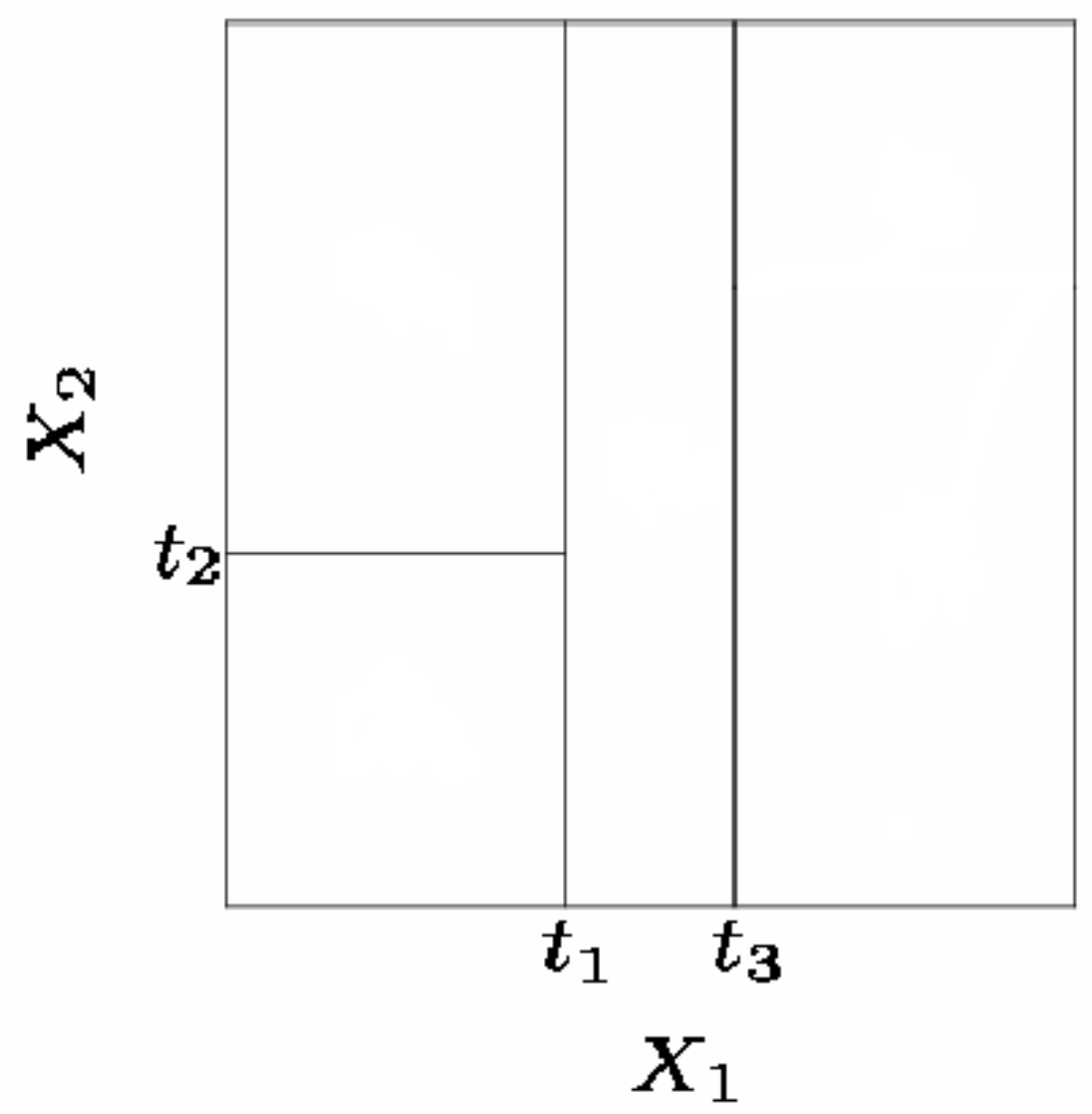
The General View

1. First split on $X_1 = t_1$.
2. If $X_1 \leq t_1$, split on $X_2 = t_2$.



The General View

1. First split on $X_1 = t_1$.
2. If $X_1 \leq t_1$, split on $X_2 = t_2$.
3. If $X_1 > t_1$, split on $X_1 = t_3$.



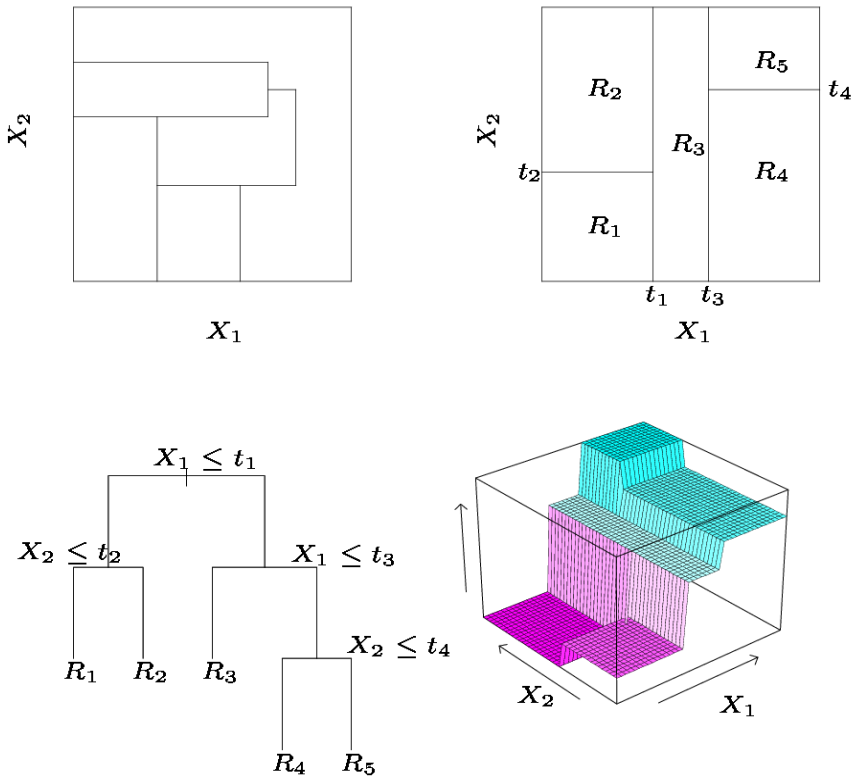
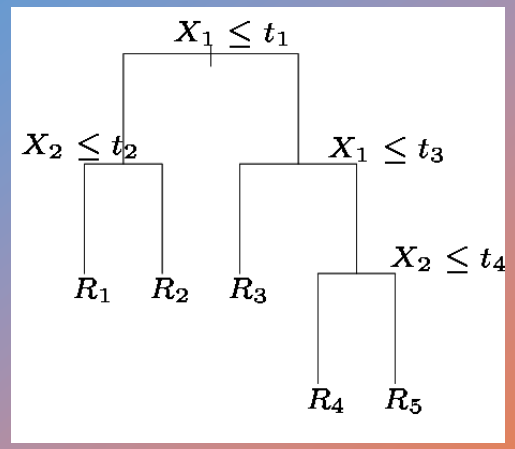
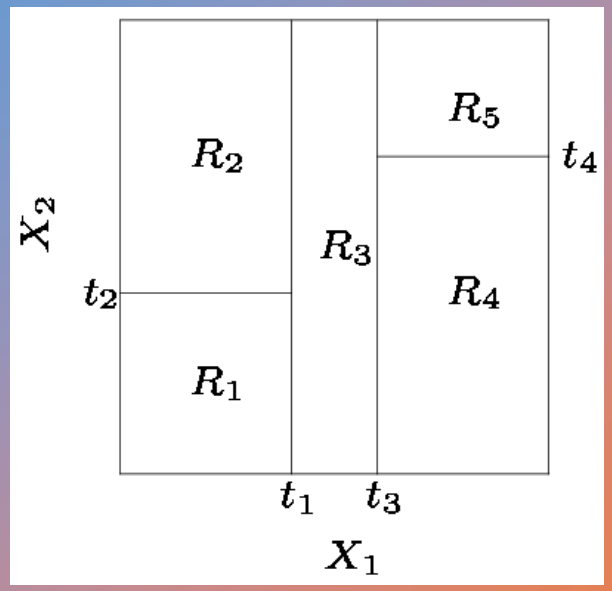


Figure 9.2: *Partitions and CART. Top right panel shows a partition of a two-dimensional feature space by recursive binary splitting, as used in CART, applied to some fake data. Top left panel shows a general partition that cannot be obtained from recursive binary splitting. Bottom left panel shows the tree corresponding to the partition in the top right panel, and a perspective plot of the prediction surface appears in the bottom right panel.*

The General View

- 1. First split on $X_1 = t_1$.
- 2. If $X_1 \leq t_1$, split on $X_2 = t_2$.
- 3. If $X_1 > t_1$, split on $X_1 = t_3$.
- 4. If $X_1 > t_3$, split on $X_2 = t_4$.





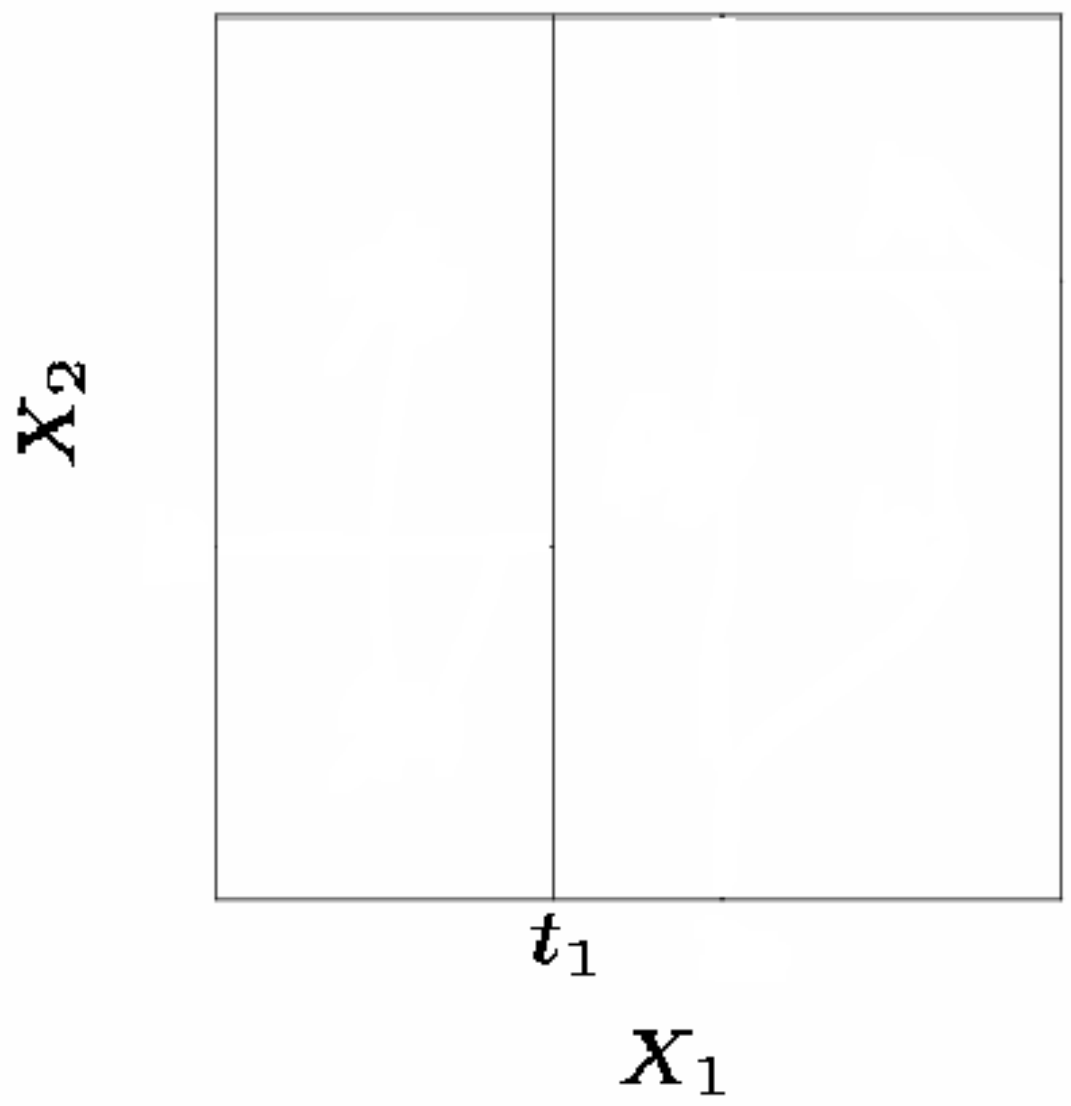
The General View

- When we create partitions like this, we can always represent them using a tree-like structure.
- This tree-like representation provides a very simple way to explain the model to a non-expert!!!

Where to split?

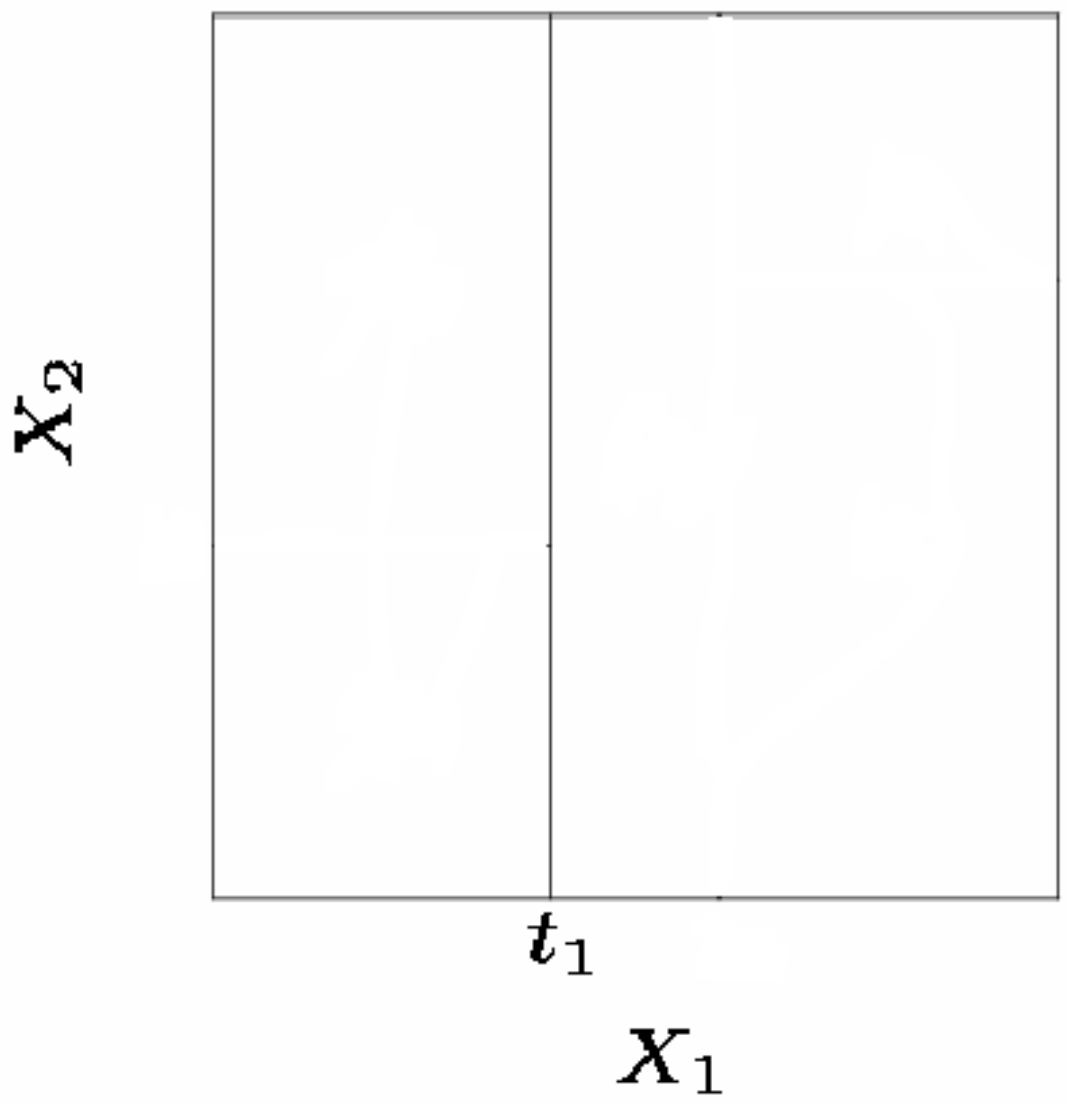
- We consider splitting into two regions, $X_j \leq s$ and $X_j > s$ for all possible values of s and $j = 1,2$.
- We then choose the s and j that results in the lowest MSE on the training data.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$



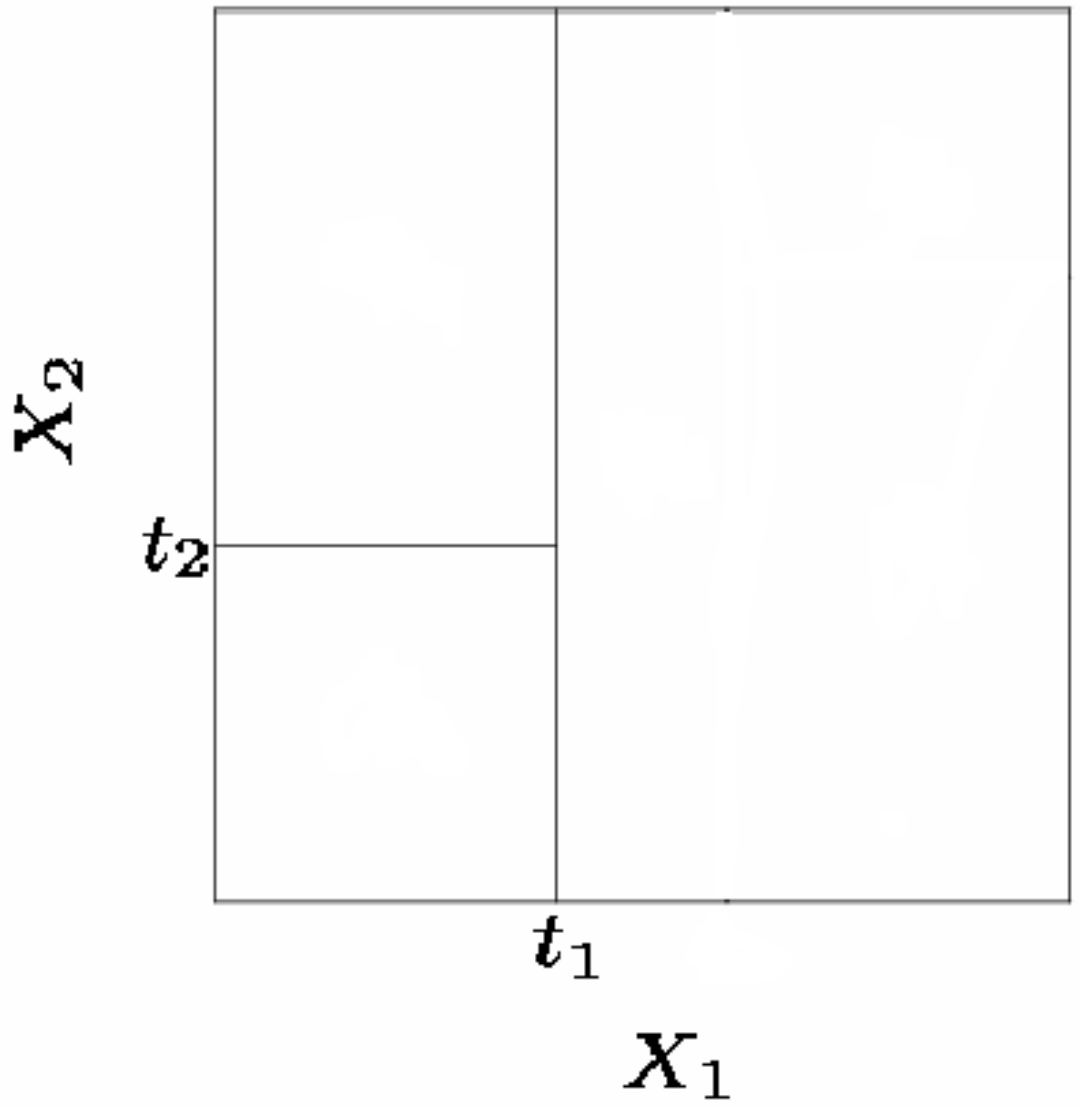
Where to split?

- Here the optimal split was on X_1 at point t_1.
- We now repeat the process looking for the next best split except that we must also consider whether to split the first region or the second region up.
- Again the criteria is smallest MSE.



Where to split?

- The optimal split was the left region on X_2 at point t_2.
- This process continues until our regions have too few observations to continue e.g. all regions have 5 or fewer points.



Important Terminology related to Decision Trees

Root Node: It represents entire population or sample and this further gets divided into two or more homogeneous sets.

Splitting: It is a process of dividing a node into two or more sub-nodes.

Decision Node: When a sub-node splits into further sub-nodes, then it is called decision node.

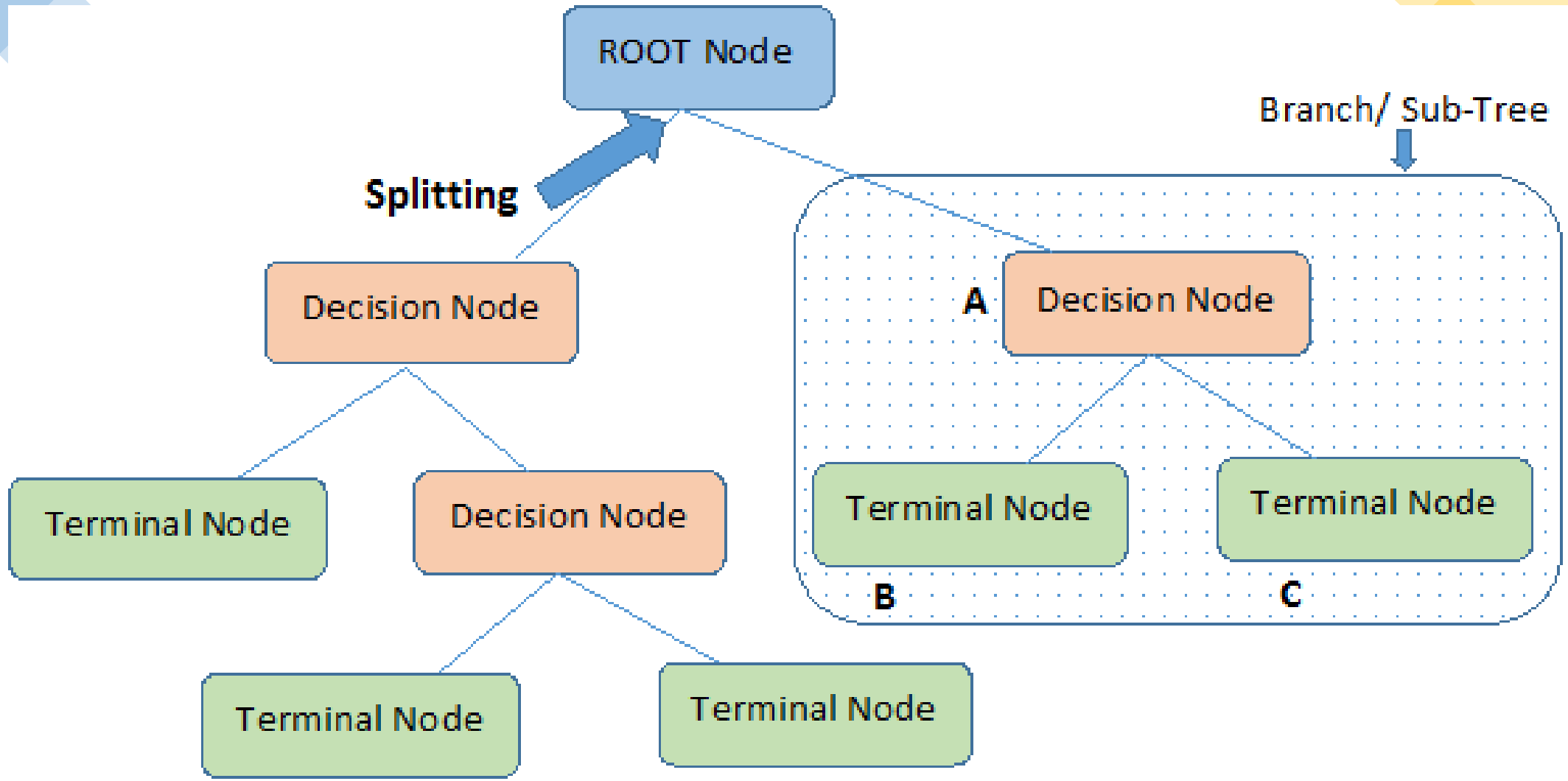
Leaf/ Terminal Node: Nodes do not split is called Leaf or Terminal node.

Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.

Branch / Sub-Tree: A sub section of entire tree is called branch or sub-tree.

Parent and Child Node: A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

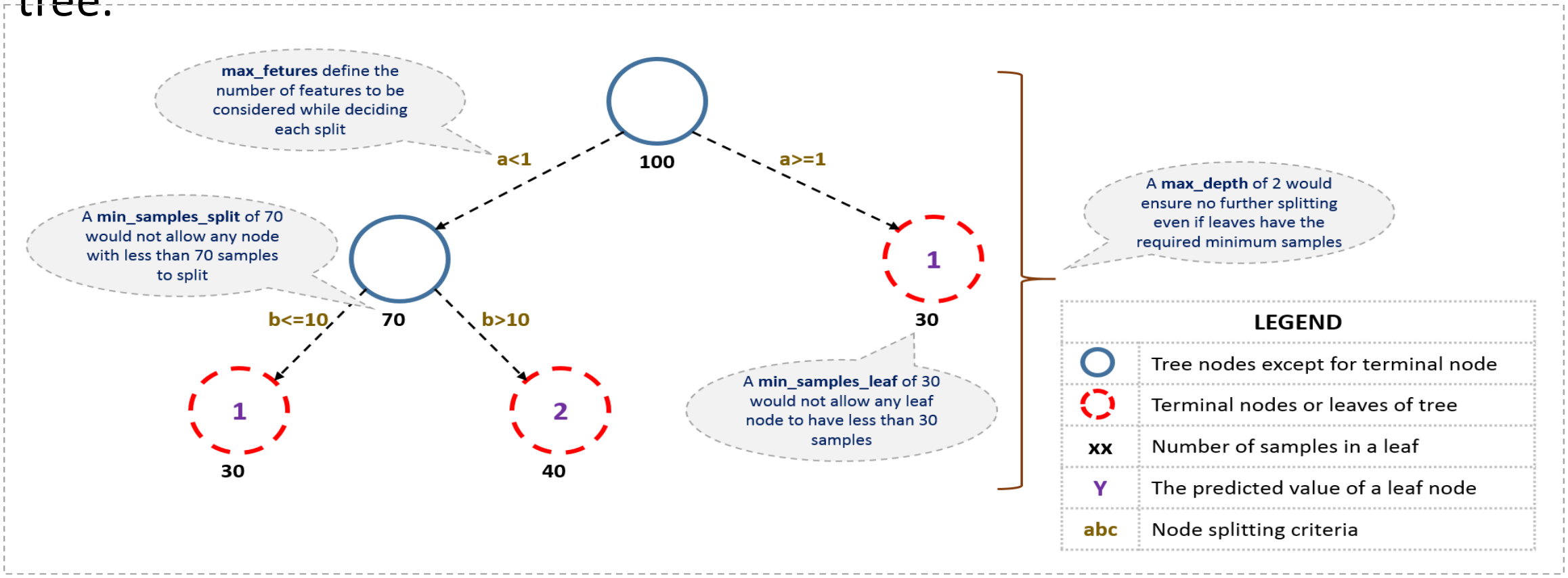
These are the terms commonly used for decision trees. As we know that every algorithm has advantages and disadvantages, below are the important factors which one should know.



Note:- A is parent node of B and C.

Setting Constraints on Tree Size

- This can be done by using various parameters which are used to define a tree. First, lets look at the general structure of a decision tree:



Tree Pruning

- Grow a very large tree and then prune it back in order to obtain a subtree.
- The **cptable** in the fit contains the mean and standard deviation of the errors in the cross-validated prediction against each of the geometric means, and these are plotted by this function.
- A good choice of cp for pruning is often the **leftmost value for which the mean lies below the horizontal line.**

- We need a way to select a small set of subtrees for consideration.
- This is done through **cost complexity** pruning, or weakest link pruning.

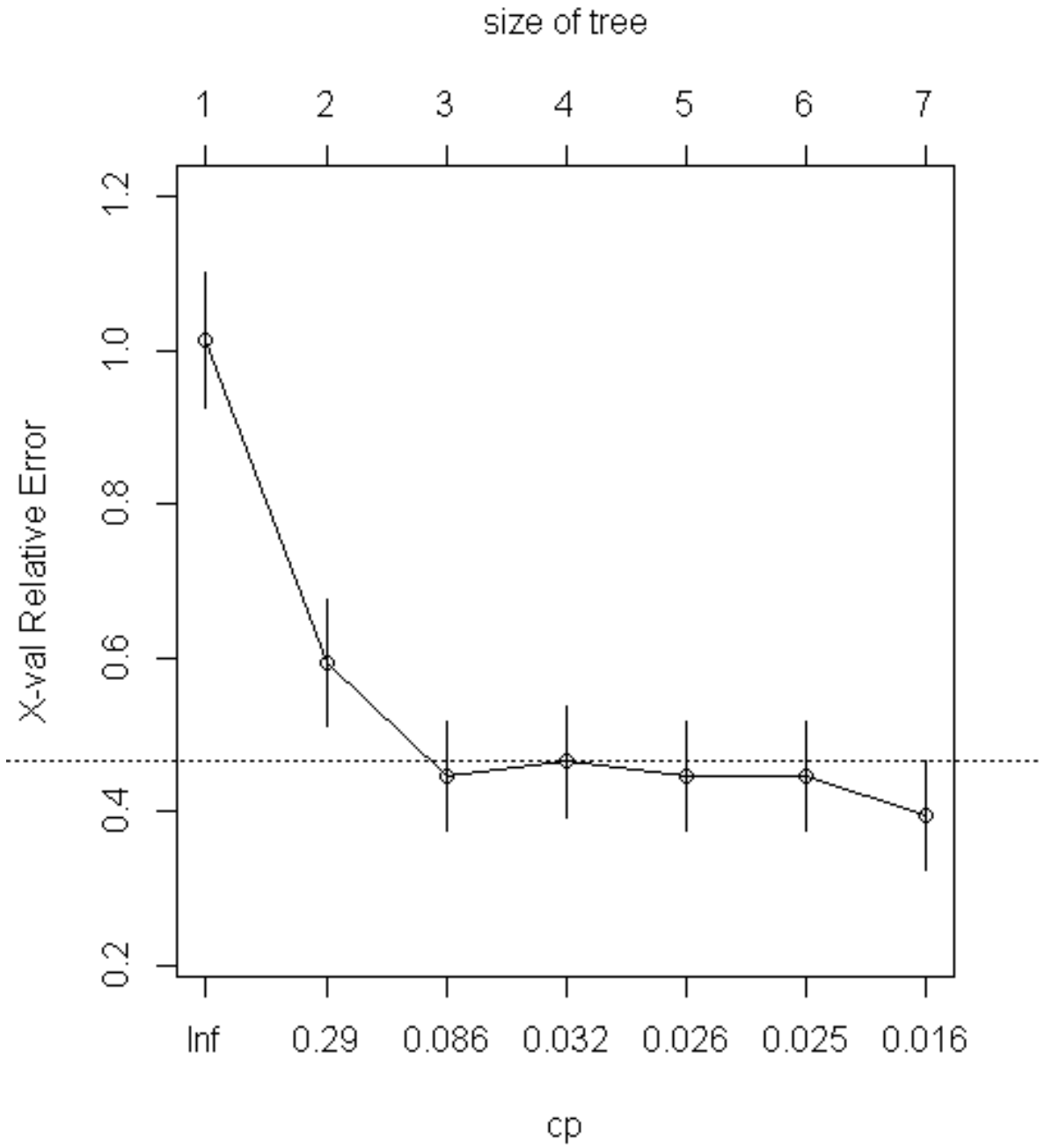
Tree Pruning

- In **rpart package**, controlled by the **complexity parameter (cp),** which imposes a penalty to the tree for having two many splits. The default value is 0.01. The higher the cp, the smaller the tree.
- A too small value of cp leads to overfitting and a **too large cp** value will result to a **too small tree**. Both cases decrease the predictive performance of the model.
- An optimal cp value can be estimated by testing different cp values and using **cross-validation approaches** to determine the corresponding prediction accuracy of the model.
- The best cp is then defined as the one that maximize the cross-validation accuracy
- Pruning can be easily performed in the caret package workflow, which invokes the rpart method for automatically testing different possible values of cp, then choose the optimal cp that maximize the cross-validation (“cv”) accuracy, and fit the final best CART model that explains the best our data.

Cross-validation on tree method

complexity parameter (cp) =

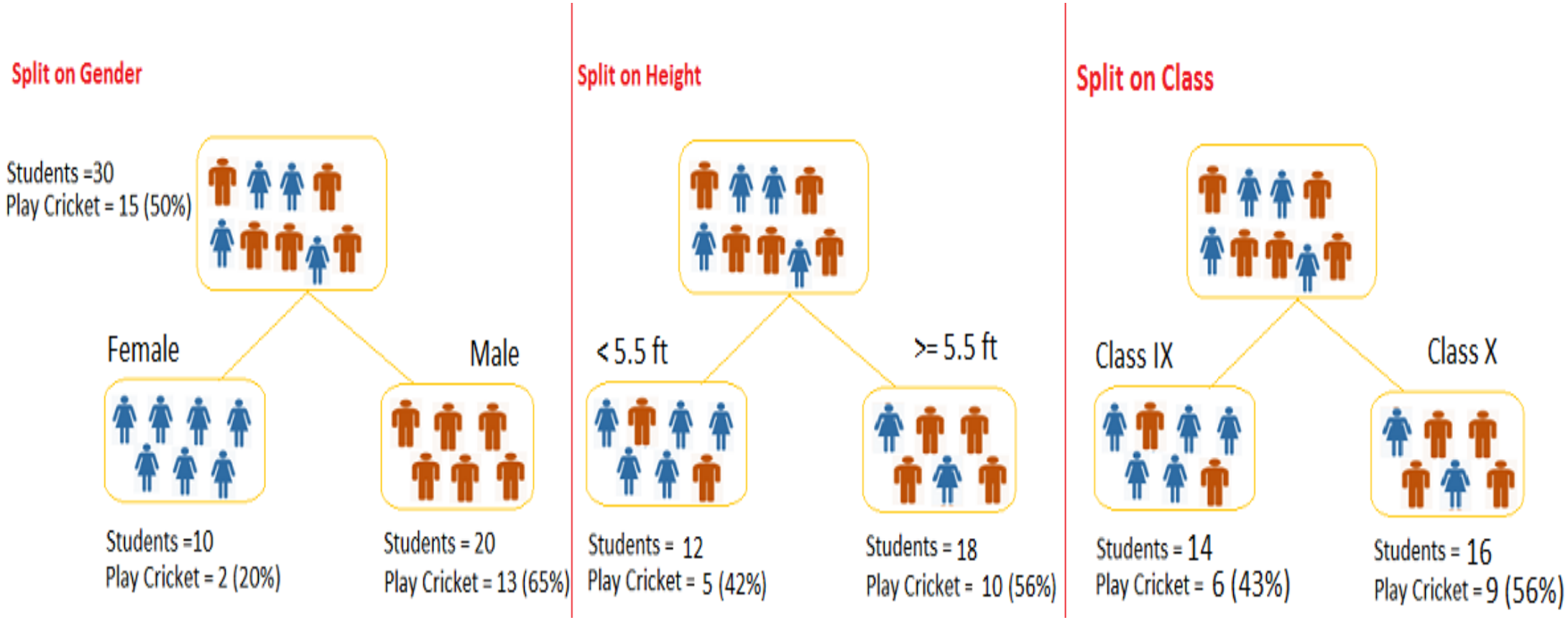
$$\sum_{\text{Terminal Nodes}} Misclass_i + \lambda * (Splits)$$



Classification Tree

Example:-

- Let’s say we have a sample of 30 students
- with three variables Gender (Boy/ Girl), Class(IX/ X) and Height (5 to 6 ft). 15 out of these 30 play cricket in leisure time.
- Now, I want to create a model to predict who will play cricket during leisure period?
- In this problem, we need to segregate students who play cricket in their leisure time based on **highly significant input variable** among all three.
- The Decision tree will segregate the students based on all values of three variable and identify the variable,
- which creates the best homogeneous sets of students (which are heterogeneous to each other).

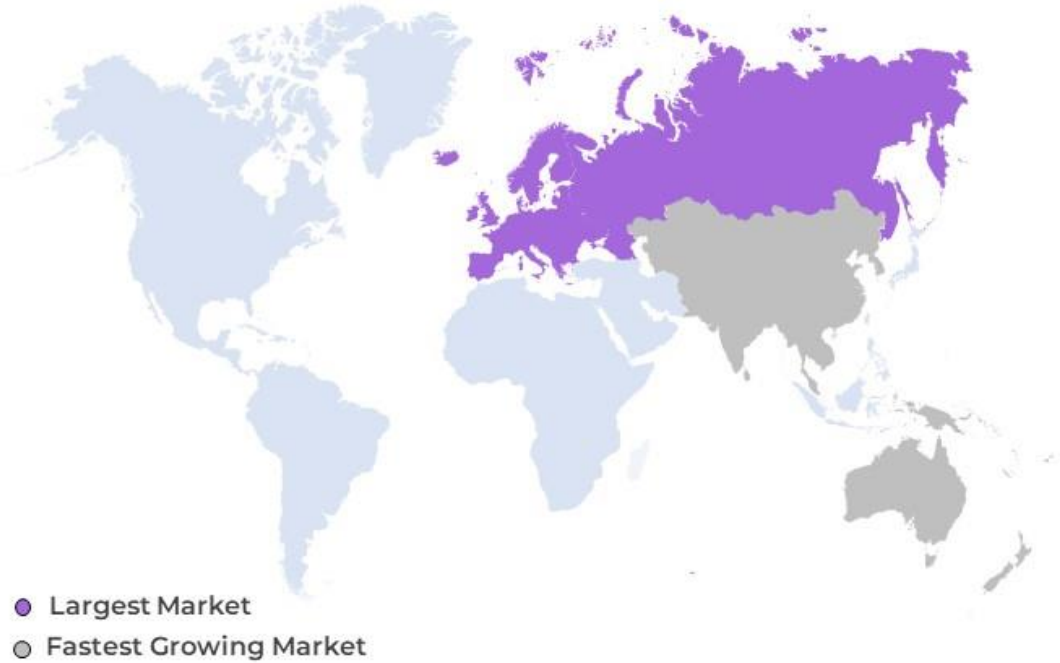


- As mentioned above, decision tree identifies the most significant variable and its value that gives best homogeneous sets of population.
- Now the question which arises is, how does it identify the variable and the split?

Baby Car Seat Market

Trends, By Region, 2023-2030

KINGS RESEARCH



- Largest Market
- Fastest Growing Market

$2135.61 Million

Europe Market Size, 2022

6.78%

Europe Market CAGR (2023-2030)

Source: kingsresearch.com

Carseats Data Set

- A data set containing sales of child car seats at 400 different stores.
- A data set with 400 observations on the following 11 variables.
- The variables are as follows:

1. Sales: Unit sales (in thousands) at each location.

2. CompPrice: Price charged by competitor at each location

3. Income: Community income level (in thousands of dollars)

4. Advertising: Local advertising budget for company at each location (in thousands of dollars)

5. Population: Population size in region (in thousands)

Carseats Data Set

- 6. Price: Price company charges for car seats at each site
- 7. ShelveLoc: A factor with levels “Bad”, “Medium” and “Good” indicating the quality of the shelving location.
- 8. Age: Average Age of the local population
- 9. Education: Education level at each location
- 10. Urban: A factor with levels “No” and “Yes” to indicate whether the store is in an urban or rural location
- 11. US: A factor with levels “No” and “Yes” to indicate whether the store is in the US or not.

Converting Regression problem to Classification Problem

Carseats Data Set

We now recode “Sales” as binary variable.

We create a dummy variable “High”, which takes on a value “Yes” if the sales exceed 8 units and “No” otherwise.

We will model “High” with the help of ten predictors.

Classification Tree

A classification tree is to make a prediction for a categorical response rather than continuous one.

In a regression tree, the predicted response for an observation is given by the average response of the training observations that belong to the same terminal node.

In a classification tree, we predict that each observation belongs to the most commonly occurring class of the training observations in the region to which it belongs.

Classification Tree

- The tree is grown in the same manner as with a regression tree
- Minimizing MSE no longer makes sense. A natural alternative is classification error rate.
- The classification error rate is simply the fraction of the training observations in that region that do not belong to the most common class.
- There are several other different criteria available as well, such as the **“gini index”** and **“cross-entropy”.**

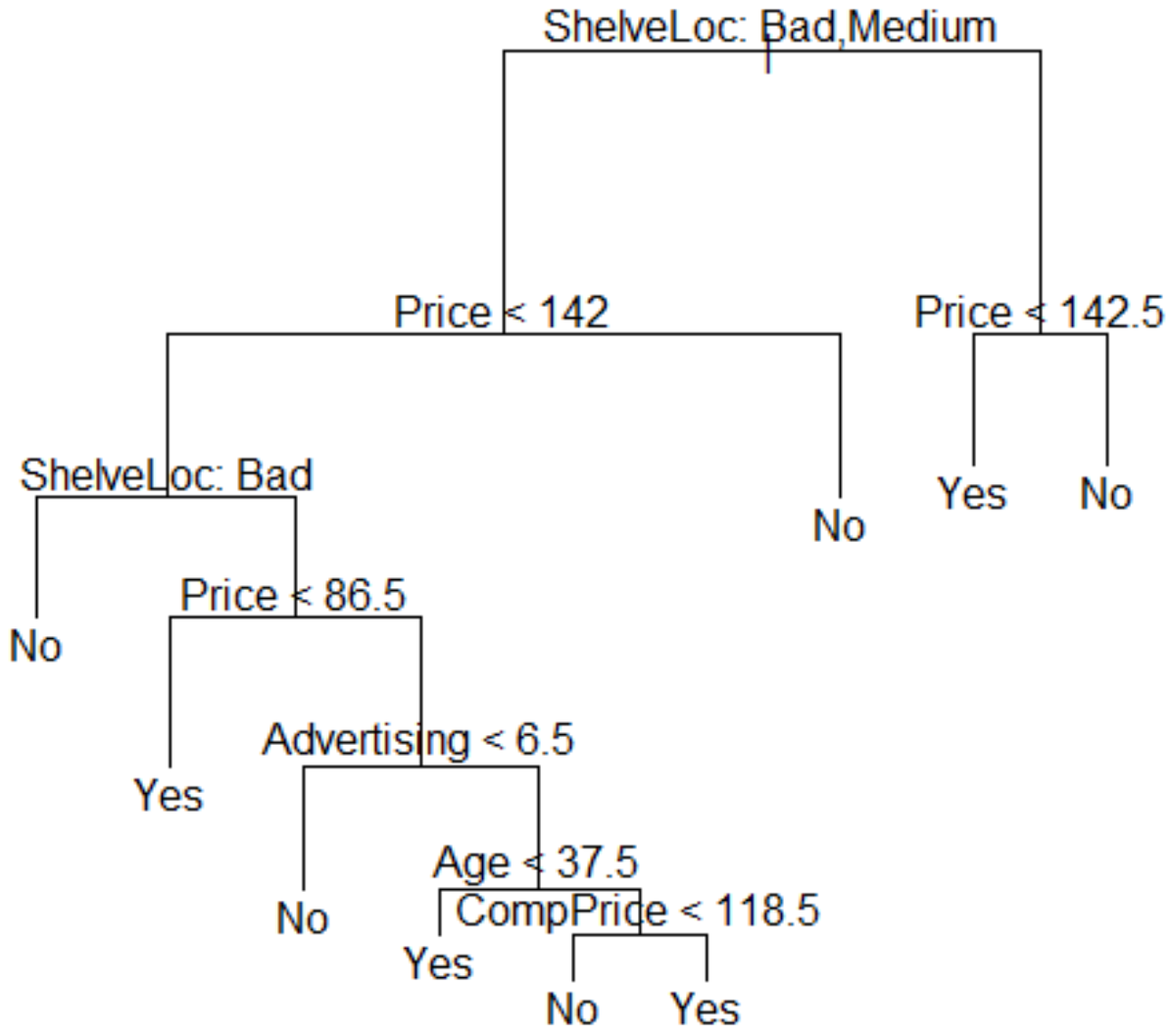
Carseats Data Set

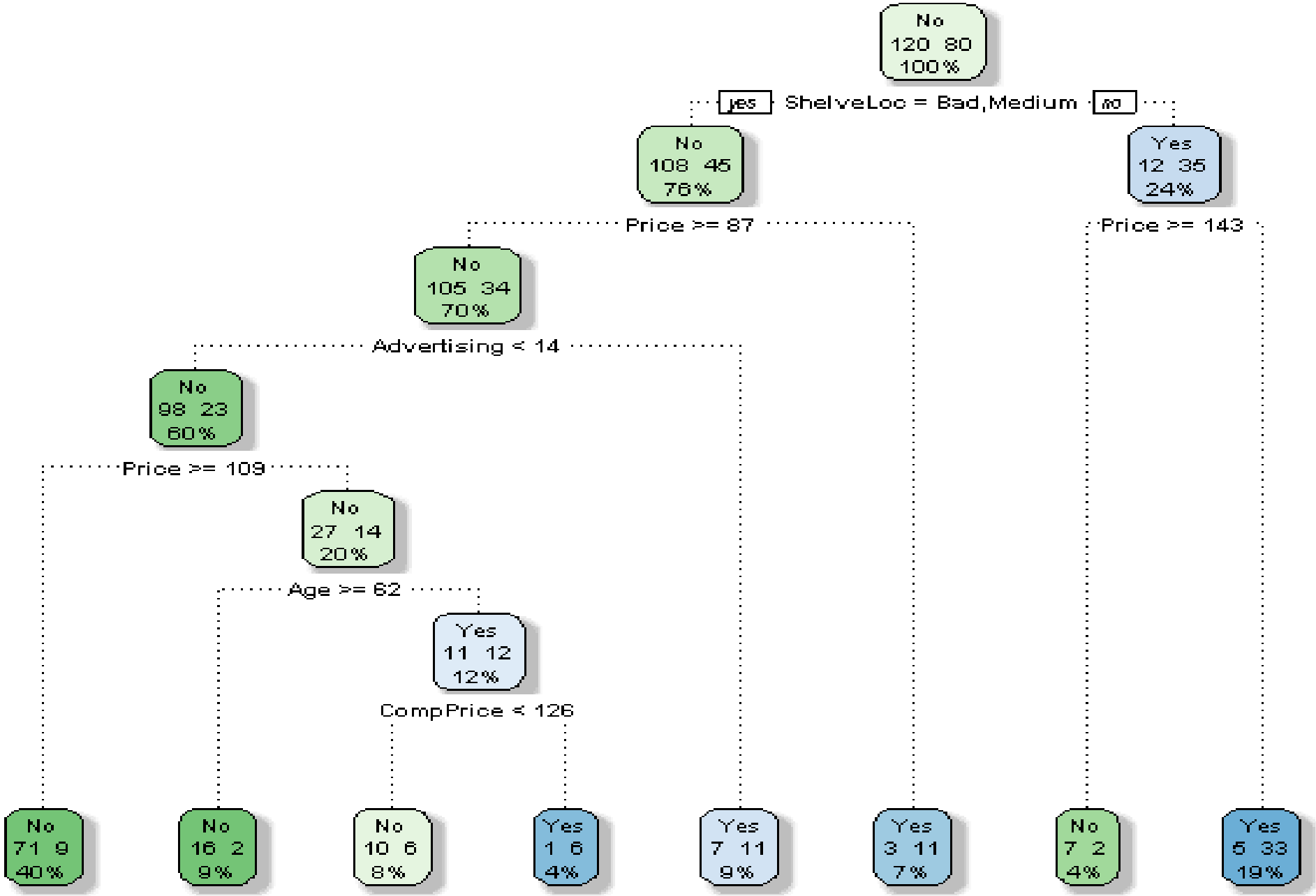
We split the observations into a training data set and a test data set.

Both the training set and the test set contain 200 observations.

We next build a tree using the training set, and then evaluate its performance based on the test data.

Pruned Tree


ShelveLoc: Bad,Medium
Price < 142
Price < 142.5
ShelveLoc: Bad
No
Yes
No
No
Price < 86.5
Yes
Advertising < 6.5
No
Age < 37.5
Yes
CompPrice < 118.5
No
Yes


No
120 80
100%
yes
ShelveLoc = Bad,Medium
no
No
108 45
76%
Yes
12 35
24%
Price >= 87
Price >= 143
No
105 34
70%
Advertising < 14
No
98 23
60%
Price >= 109
No
27 14
20%
Age >= 62
Yes
11 12
12%
CompPrice < 126
No
71 9
40%
No
16 2
9%
No
10 6
8%
Yes
1 6
4%
Yes
7 11
9%
Yes
3 11
7%
No
7 2
4%
Yes
5 33
19%

	True High status			
Predicted High Status		No	Yes	Total
	No	?	?	?
	Yes	?	?	?
	Total	?	?	?

Confusion Matrix based on Test Data

- $Sensitivity = \frac{?}{?} = \%$
- $Specificity = \frac{?}{?} = ?\,\%$
- $Total\ Error\ Rate = \frac{?}{?} = ?\,\%$

	True High status		
Predicted High Status	No	Yes	Total
No	88	28	116
Yes	28	56	84
Total	116	84	200

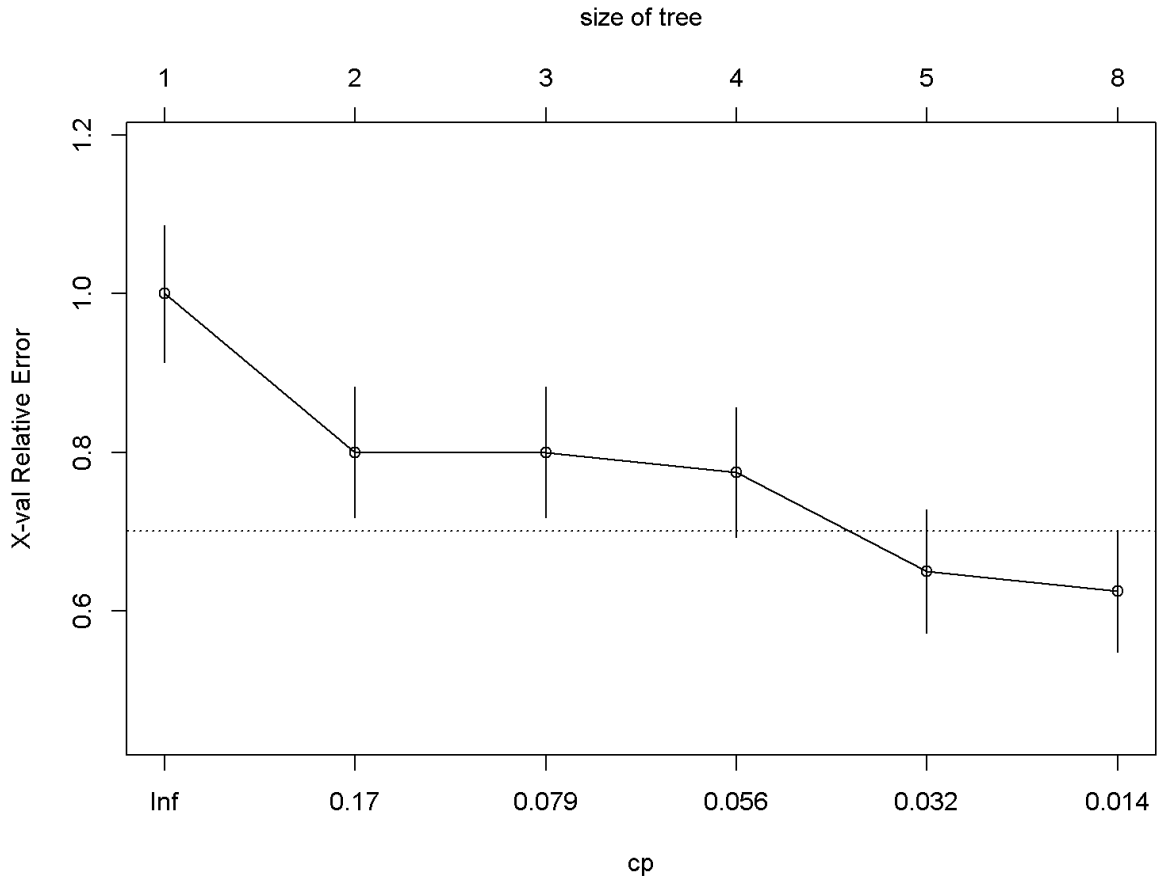
Confusion Matrix based on Test Data

- $Sensitivity = \frac{56}{84} = 66.67\%$
- $Specificity = \frac{88}{116} = 75.86\%$
- $Total\ Error\ Rate = \frac{56}{200} = 28\%$

Cross Validation

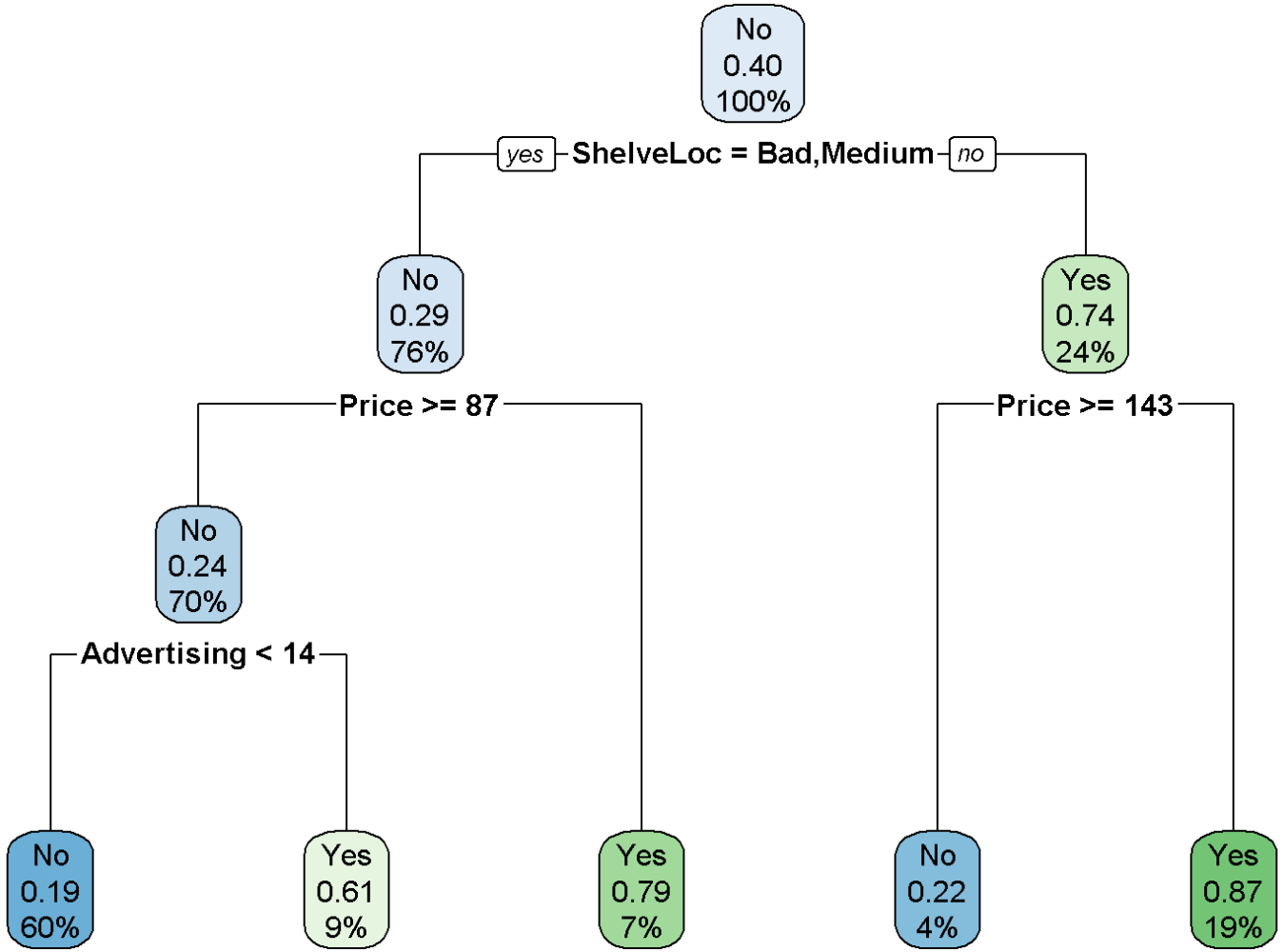
- We now consider whether pruning the tree leads to a better performance.
- We decide the optimal level of tree complexity using cross-validation.

Cross Validation



We select based on CP.

Pruned Tree


No
0.40
100%
yes
ShelveLoc = Bad,Medium
no
No
0.29
76%
Yes
0.74
24%
Price >= 87
Price >= 143
No
0.24
70%
Advertising < 14
No
0.19
60%
Yes
0.61
9%
Yes
0.79
7%
No
0.22
4%
Yes
0.87
19%

	True High status			
Predicted High Status		No	Yes	Total
	No	?	?	?
	Yes	?	?	?
	Total	?	?	?

Confusion Matrix based on Test Data

- $Sensitivity = \frac{?}{?} = \%$
- $Specificity = \frac{?}{?} = ?\,\%$
- $Total\ Error\ Rate = \frac{?}{?} = ?\,\%$

	True High status			
Predicted High Status		No	Yes	Total
	No	94	24	118
	Yes	22	60	82
	Total	116	84	200

Confusion Matrix based on Test Data for Pruned Tree

- $Sensitivity = \frac{60}{84} = 71.43\%$
- $Specificity = \frac{94}{116} = 81.03\%$
- $Total\ Error\ Rate = \frac{46}{200} = 23\%$

True Positive and False Positive Rate

- As we have seen above, varying the classifier threshold changes its true positive and false positive rate.
- These are also called the *sensitivity* and one minus the *specificity* of our classifier.
- To make the connection with the epidemiology literature, we may think of "+" as the "disease" that we are trying to detect, and "–" as the "non-disease" state.
- To make the connection to the classical hypothesis testing literature, we think of "–" as the null hypothesis and "+" as the alternative (non-null) hypothesis.
- In the context of the Default data, "+" indicates an individual who defaults, and "–" indicates one who does not.

True Positive and False Positive Rate

	Predicted Class			
True Class		- Or Null	+ or Non-null	Total
	- Or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

True Positive and False Positive Rate

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity
Pos. Pred. value	TP/P*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N*	

Training Error Rate and Test Error Rate

- The misclassification error rate calculated earlier with the optimal threshold was 13.81%.
- However, we have used the same data to train and test our model.
- In reality, this error rate is in fact the *training error* rate.
- In order to assess the accuracy of the model, we should first fit a model using a part of the data and then should examine the performance on the "hold-out" data.
- This error rate is called the *test error* rate.
- Next we have used 80% of the observations to fit the model and 20% of observations are kept aside for validating the model.

Deciding the Optimal Threshold

- How can we decide which threshold value is best?
- Such a decision generally depends on *domain knowledge*, such as detailed information about the costs associated with defaulting.
- However, some common approaches involve maximizing the Sensitivity or Specificity.
- Another approach available is to maximize the $Youden\ Index\ (Sens.+Spec.-1)$ should be closer to 1. (see http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2749250/).
- The R library "Epi" determines the threshold by maximizing the sum of specificity and sensitivity.

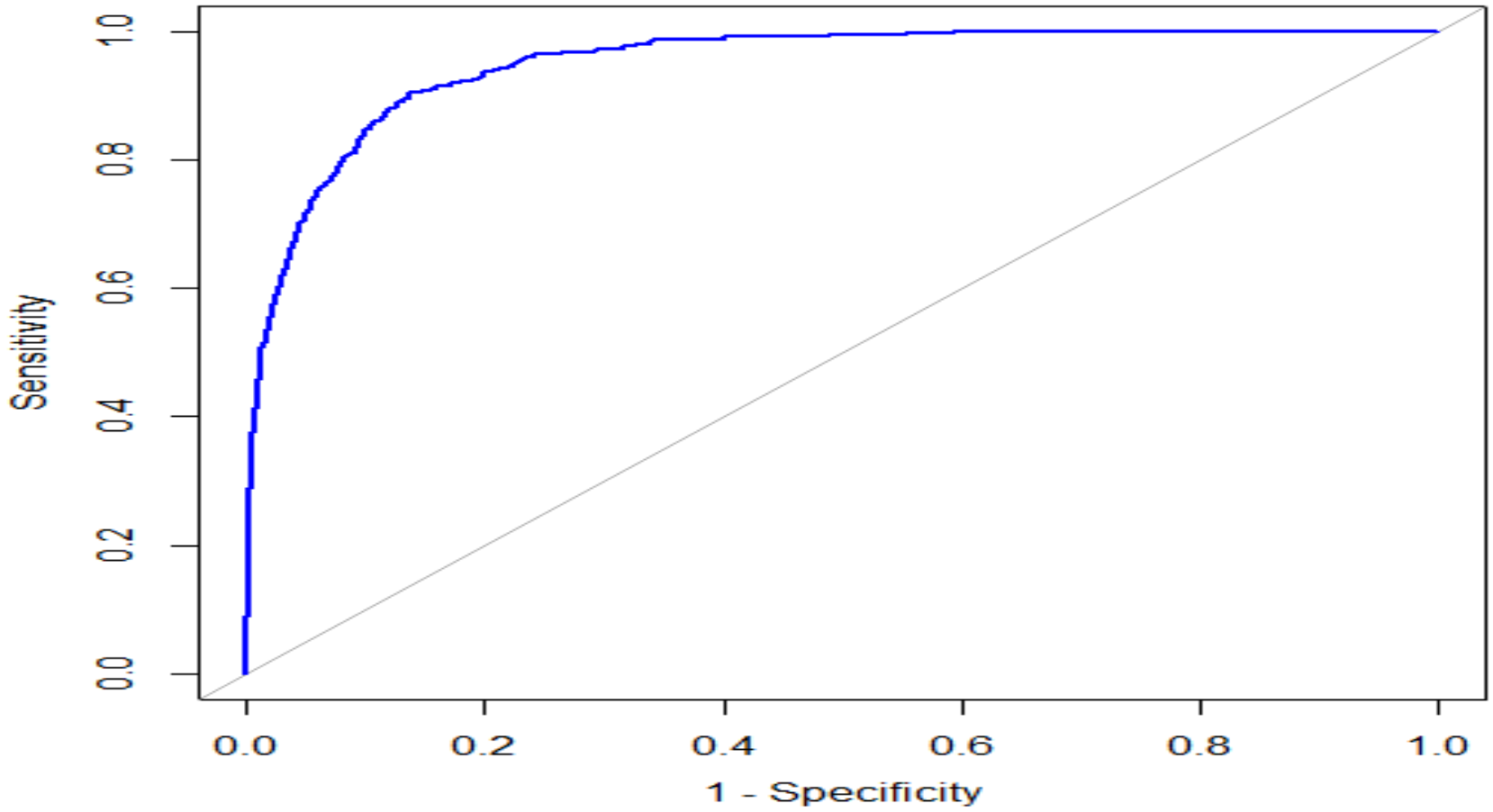
ROC Curve

- The *ROC curve* is a used for simultaneously displaying two types of errors for all possible thresholds.
- The name “ROC” comes from communications theory. It is an acronym for *receiver operating characteristics*.
- The overall performance of a classifier, summarized over all possible thresholds, is given by the *area under the (ROC) curve* (AUC).
- An ideal ROC curve should touch the top left corner, so the larger the AUC the better the classifier.

ROC Curve

Threshold Point	Sensitivity	Specificity	1 – Specificity
0.0	1.000	0.000	1.000
0.1	0.745	0.942	0.056
0.2	0.610	0.971	0.029
0.3	0.508	0.986	0.014
0.4	0.402	0.992	0.008
0.5	0.315	0.996	0.004
0.6	0.243	0.998	0.002
0.7	0.171	0.999	0.001
0.8	0.090	1.000	0.000
0.9	0.030	1.000	0.000
1.0	0.000	1.000	0.000

ROC Curve

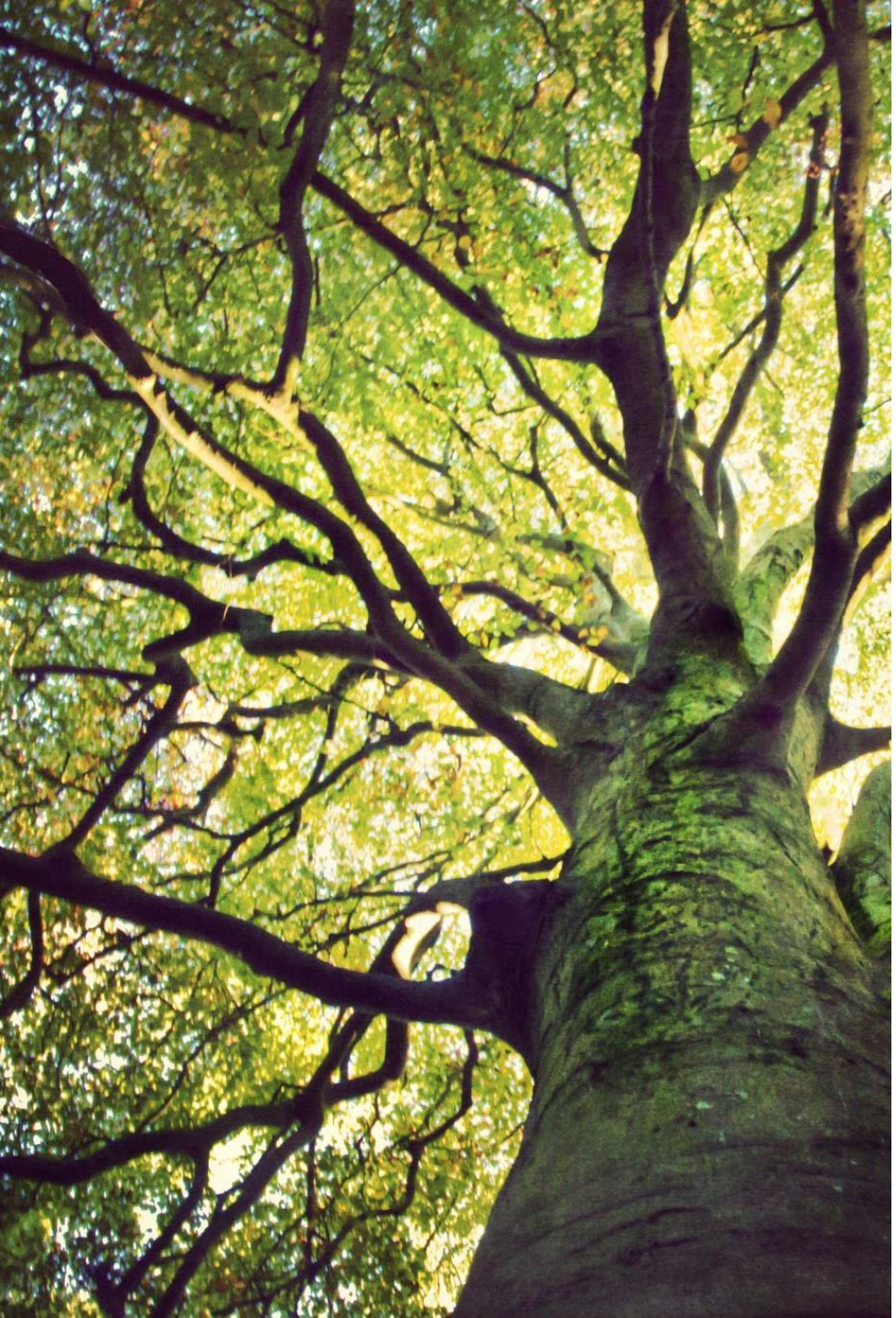


General Rule

AUC	Decision
$AUC = 0.5$	No Discrimination
$0.7 \leq AUC < 0.8$	Acceptable Discrimination
$0.8 \leq AUC < 0.9$	Excellent Discrimination
$AUC \geq 0.9$	Outstanding Discrimination

Entropy, information gain,

gini index, CHI-SQUARE AUTOMATIC INTERACTION DETECTION (CHAID)



Steps in Entropy and information gains: Example of bank loan

- The first step in constructing a decision tree is to choose the most informative attribute.
- A common way to identify the most informative attribute is to use entropy-based methods, which are used by decision tree learning algorithms such **as ID3 (or Iterative Dichotomiser 3) and C4.5.**
- **T**he entropy methods select the most informative attribute based on two basic measures:
    - ***Entropy,*** which measures the *impurity* of an attribute
    - ***Information gain,*** which measures the *purity* of an attribute

Entropy

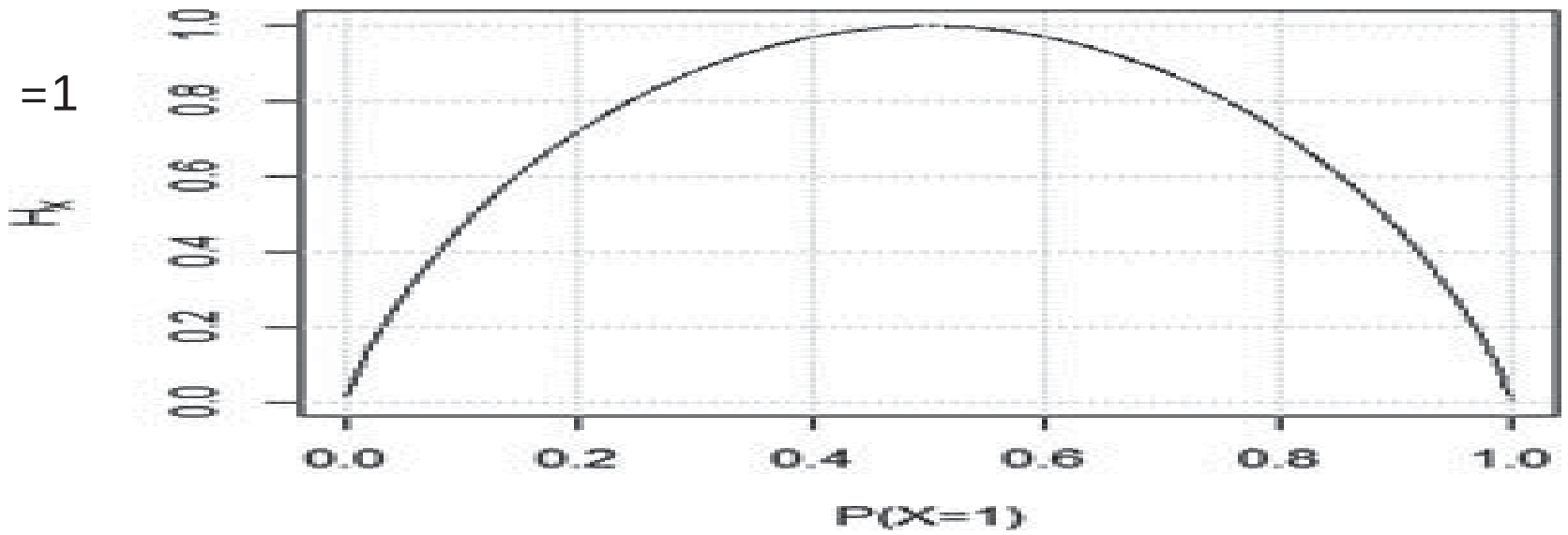
- less impure node requires less information to describe it. And more impure node requires more information.
- Information theory is a measure to define this degree of disorganization in a system known as ***Entropy.***
- If the sample is completely homogeneous, then the entropy is zero.
- And if the sample is an equally divided (50% – 50%), it has entropy of one.

Entropy

Given a class X and its label $x \in X$, let $P(x)$ be the probability of x . H_x ,the entropy of X , is defined as shown in

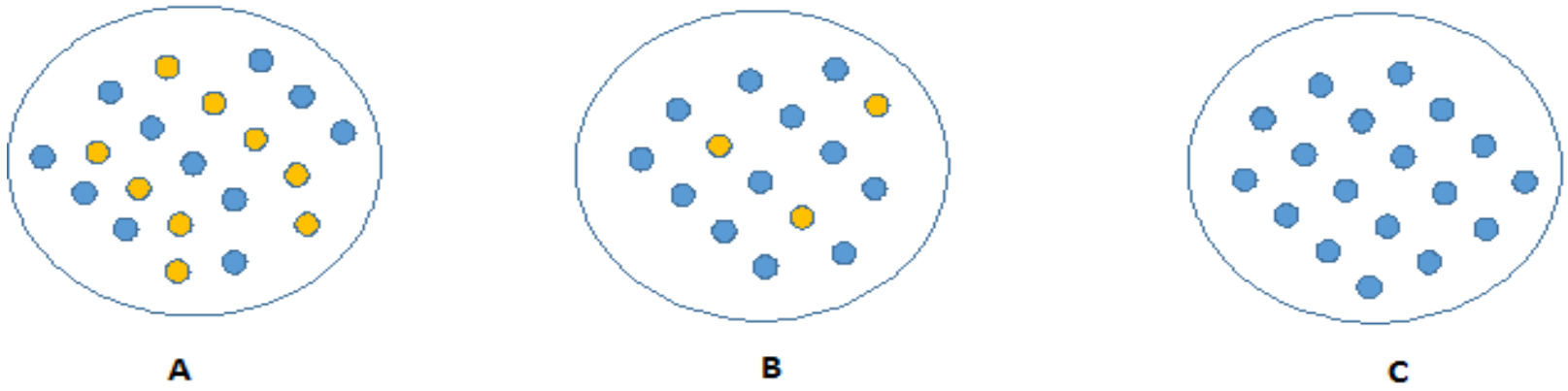
$$H_X = -\sum_{\forall x \in X} P(x) \log_2 P(x)$$

$H_x = -(0.5 \times \log_2 0.5 + 0.5 \times \log_2 0.5) = 1$



Information Gain:

- Look at the image below and think which node can be described easily. I am sure, your answer is C because it requires less information as all values are similar. On the other hand, B requires more information to describe it and A requires the maximum information. In other words, we can say that C is a Pure node, B is less Impure and A is more impure.



Steps to calculate entropy for a split

- Calculate entropy of parent node
- Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.

Example: bank loan subscription

- **Feature:**
  - *poutcome*
  - *contact*
  - *housing*
  - *job*
  - *education*
  - *marital*
  - *loan*
  - *default*

The General Algorithm

- the objective of a decision tree algorithm is to construct a tree *T* from a training set *S*. If all the records in *S* belong to some class *C* (*subscribed* = yes, for example), or if *S* is sufficiently pure (greater than a preset threshold), then that node is considered a leaf node and assigned the label *C*. The ***purity*** of a node is defined as its probability of the corresponding class.

  For example, in Figure, the root
  P(subscribed = yes)=1− 1789/2000 =10.55%; ***therefore, the root is only 10.55% pure on the*** subscribed = yes ***class. Conversely, it is 89.45% pure on the*** subscribed = no ***class.***

Entropy

- For the bank marketing scenario, the output variable is *subscribed*. The base entropy is defined as entropy of the output variable, that is $H_{subscribed}$.
- As seen previously, $P(subscribed = yes) = 0.1055$ and $P(subscribed = no) = 0.8945$. According to Equation of entropy, the base entropy
- $$H_{subscribed} = -0.1055 \cdot \log_2 0.1055 - 0.8945 \cdot \log_2 0.8945 \approx 0.4862.$$

Conditional Entropy

The next step is to identify the conditional entropy for each attribute. Given an attribute X, its value *x* , its outcome Y**,** and its value y, conditional entropy $H_{Y|X}$ is the remaining entropy of *Y* given *X* , formally defined as,

$$H_{Y|X} = \sum_{x} P(x) H(Y|X = x)$$

$$= -\sum_{\forall x \in X} P(x) \sum_{\forall y \in Y} P(y|x) \log_2 P(y|x)$$

Consider the banking marketing scenario, if the attribute `contact` is chosen, *X* = {cellular, telephone, unknown}. The conditional entropy of `contact` considers all three values.

Information gain

- information gain of an attribute *A* is defined as the difference between the base entropy and the conditional entropy of the attribute, as shown

$$InfoGain_A = H_S - H_{S|A}$$

- In the bank marketing example, the information gain of the *contact* attribute is show,

$$InfoGain_{contact} = H_{subscribed} - H_{contact|subscribed}$$

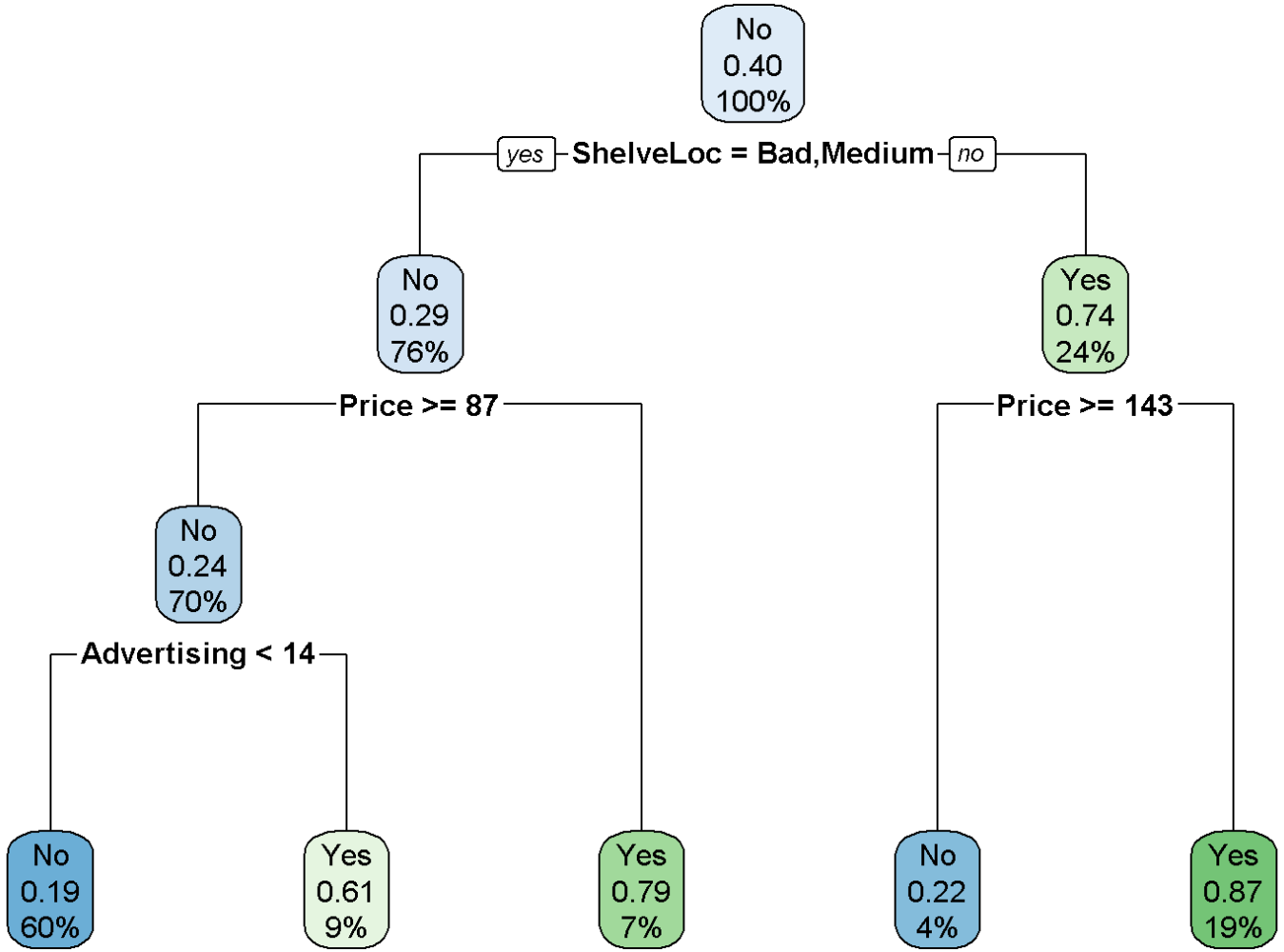
$$= 0.4862 - 0.4661 = 0.0201$$

Calculating Information Gain of Input Variables for the First Split

Attribute	Information Gain
poutcome	0.0289
contact	0.0201
housing	0.0133
job	0.0101
education	0.0034
marital	0.0018
loan	0.0010
default	0.0005

```
> ig<-information_gain(High~.-Sales,data = Carseats, type = "info
gain")
> #dfI<-data.frame(at=IG$attributes, ga=IG$importance)
> ig[order(ig$importance, decreasing = T), ]
     attributes  importance
6     ShelveLoc 0.099734918
5         Price 0.050617166
3   Advertising 0.045205973
7           Age 0.024651408
10           US 0.017117003
9         Urban 0.001950963
1     CompPrice 0.000000000
2        Income 0.000000000
4    Population 0.000000000
8     Education 0.000000000
```

Pruned Tree


No
0.40
100%
yes
ShelveLoc = Bad,Medium
no
No
0.29
76%
Yes
0.74
24%
Price >= 87
Price >= 143
No
0.24
70%
Advertising < 14
No
0.19
60%
Yes
0.61
9%
Yes
0.79
7%
No
0.22
4%
Yes
0.87
19%

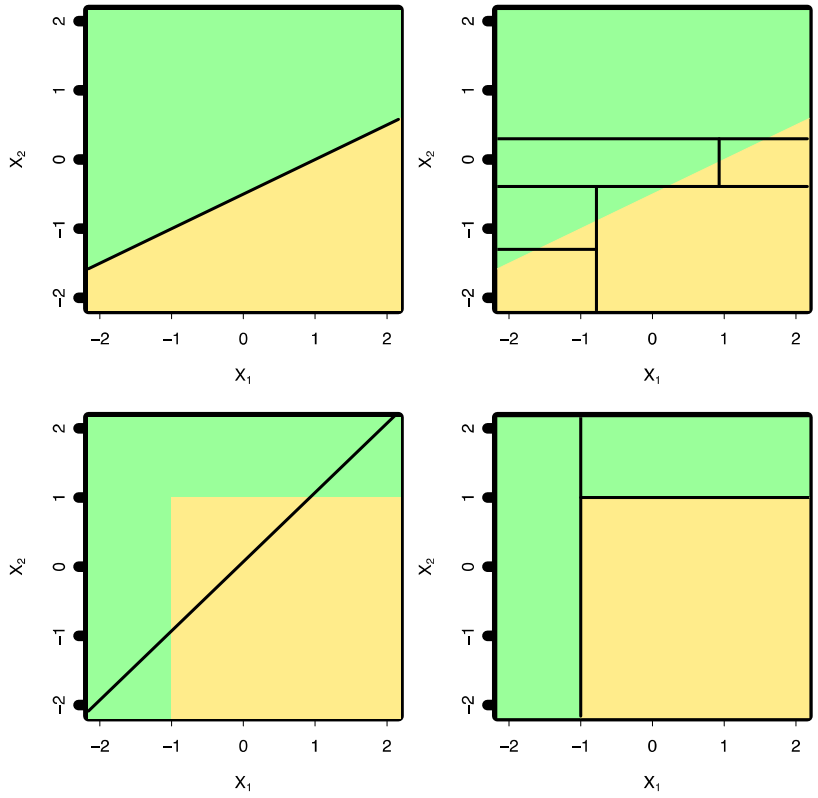
Tree Vs Linear model??

Trees vs. Linear Models

- Which model is better?
  - If the relationship between the predictors and response is linear, then classical linear models such as linear regression would outperform regression trees
  - On the other hand, if the relationship between the predictors is non-linear, then decision trees would outperform classical approaches

Trees vs. Linear Model: Classification Example

- Top row: The true decision boundary is linear
  - Left: linear model (Better)
  - Right: decision tree
- Bottom row: The true decision boundary is non-linear
  - Left: linear model
  - Right: decision tree (Better)



Advantages and Disadvantages of Decision Trees

- Advantages:
  - Trees are very easy to explain to people (even easier than linear regression).
  - Trees can be plotted graphically, and hence can be easily communicated even to a non-expert.
  - They work fine for both classification and regression problems.

- Disadvantages:
  - Trees don’t have the same prediction accuracy as some of the more flexible approaches available in practice.