

# Applied Machine Learning with R

Felix Heinrich

Breeding Informatics Group  
Department of Animal Science  
Georg-August University Göttingen

# Introduction to Machine Learning

## Tree Based Modeling

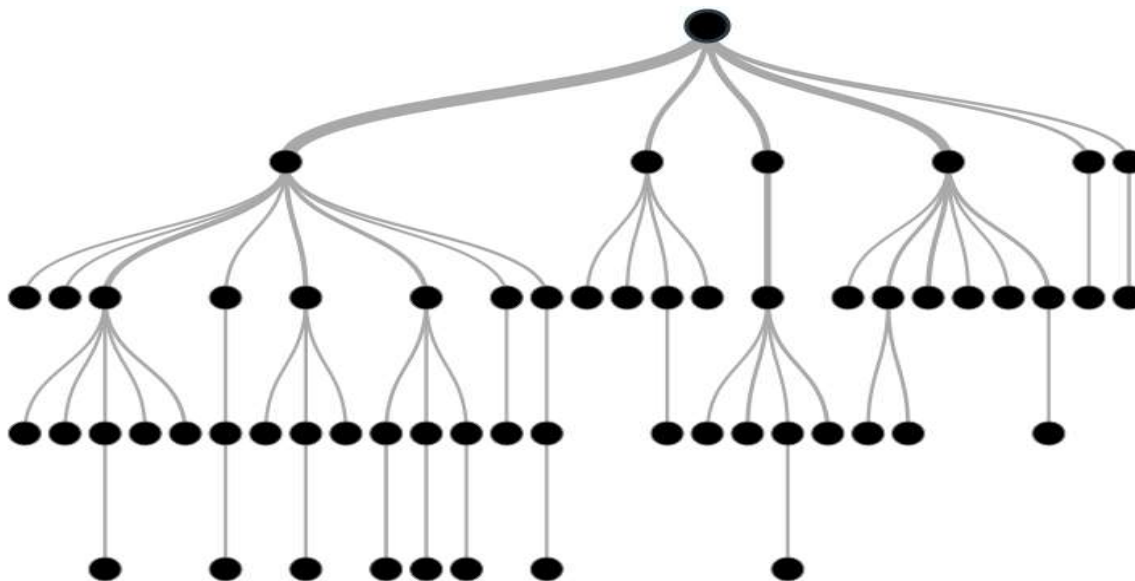
- one of the best and mostly used supervised learning methods
- empower predictive models with high accuracy, stability and ease of interpretation
- map non-linear relationships quite well
- adaptable at solving any kind of problem at hand (classification or regression).
- **Popular methods** used in all kinds of data science problems:
  - decision trees,
  - random forest,
  - gradient boosting

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

- works for both categorical and continuous input and output variables.

**Main idea:** Split the sample into two or more homogeneous sets (or sub-sample) based on most significant splitter / differentiator in input variables.



# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### **Example:**

Let us consider 30 students with three variables

- Gender (Boy/ Girl),
- Class( IX/ X)
- Height (5 to 6 ft).
- 15 out of them play cricket in their leisure time.

**Problem:** Create a model to predict who will play cricket during their leisure period?

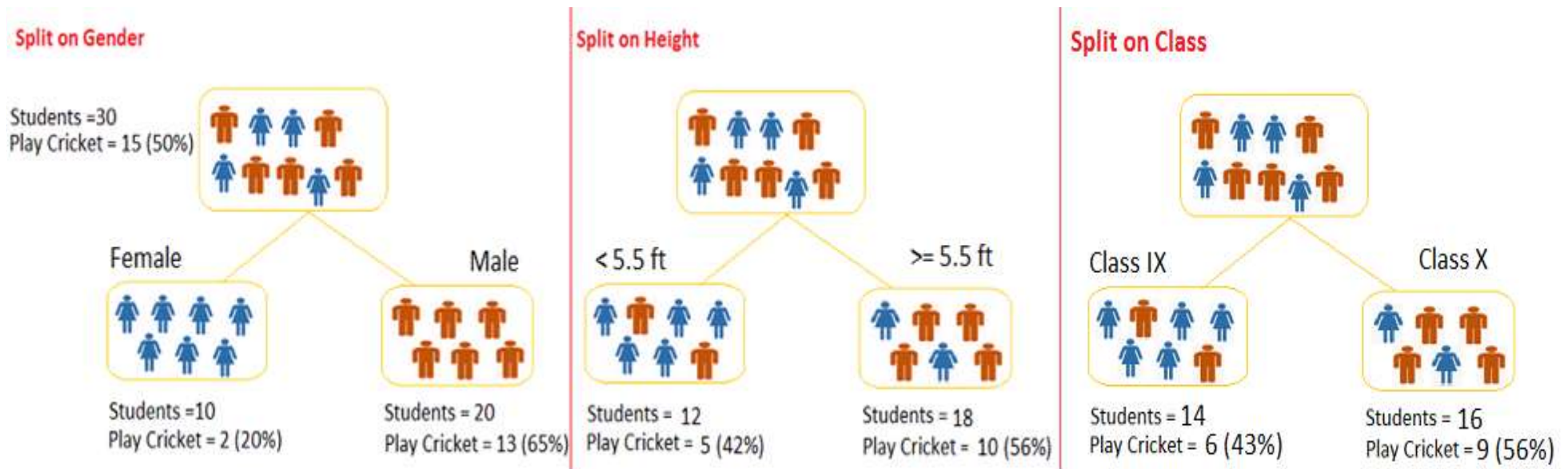
**Solution:** Segregate students who play cricket in their leisure time based on highly significant input variable among all three.

**Use decision tree:** It will segregate the students based on all values of the three variable and identify the variable, which creates the best homogeneous sets of students

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

**Decision tree** segregates the students based on all values of the three variable



In these decision trees, you can see that the variable **Gender** is able to identify the best homogeneous sets compared to the other two variables.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

Decision tree identifies the most significant variable and its value that gives the best homogeneous sets of population

Now the question which arises is:

➔ How does it identify the variable and the split?

To do this: decision tree uses various algorithms

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Important Terminology related to Decision Trees:

1. **Root Node:** It represents the entire population or sample and this further gets divided into two or more **homogeneous** sets.
2. **Splitting:** It is the process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called a decision node.
4. **Leaf/ Terminal Node:** Nodes which are not split are called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can see it as the opposite process of splitting.
6. **Branch / Sub-Tree:** A sub section of the entire tree is called branch or sub-tree.
7. **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 children of the parent node.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Important Terminology related to Decision Trees:

1. **Root Node:** It represents the entire population or sample and this further gets divided into two or more **homogeneous** sets.
2. **Splitting:** It is the process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called a decision node.
4. **Leaf/ Terminal Node:** Nodes which are not split are called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can see it as the opposite process of splitting.
6. **Branch / Sub-Tree:** A sub section of the entire tree is called branch or sub-tree.
7. **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 children of the parent node.



# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Important Terminology related to Decision Trees:

1. **Root Node:** It represents the entire population or sample and this further gets divided into two or more **homogeneous** sets.
2. **Splitting:** It is the process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called a decision node.
4. **Leaf/ Terminal Node:** Nodes which are not split are called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can see it as the opposite process of splitting.
6. **Branch / Sub-Tree:** A sub section of the entire tree is called branch or sub-tree.
7. **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 children of the parent node.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Important Terminology related to Decision Trees:

1. **Root Node:** It represents the entire population or sample and this further gets divided into two or more **homogeneous** sets.
2. **Splitting:** It is the process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called a decision node.
4. **Leaf/ Terminal Node:** Nodes which are not split are called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can see it as the opposite process of splitting.
6. **Branch / Sub-Tree:** A sub section of the entire tree is called branch or sub-tree.
7. **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 children of the parent node.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Important Terminology related to Decision Trees:

1. **Root Node:** It represents the entire population or sample and this further gets divided into two or more **homogeneous** sets.
2. **Splitting:** It is the process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called a decision node.
4. **Leaf/ Terminal Node:** Nodes which are not split are called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can see it as the opposite process of splitting.
6. **Branch / Sub-Tree:** A sub section of the entire tree is called branch or sub-tree.
7. **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 children of the parent node.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Important Terminology related to Decision Trees:

1. **Root Node:** It represents the entire population or sample and this further gets divided into two or more **homogeneous** sets.
2. **Splitting:** It is the process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called a decision node.
4. **Leaf/ Terminal Node:** Nodes which are not split are called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can see it as the opposite process of splitting.
6. **Branch / Sub-Tree:** A sub section of the entire tree is called branch or sub-tree.
7. **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 children of the parent node.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

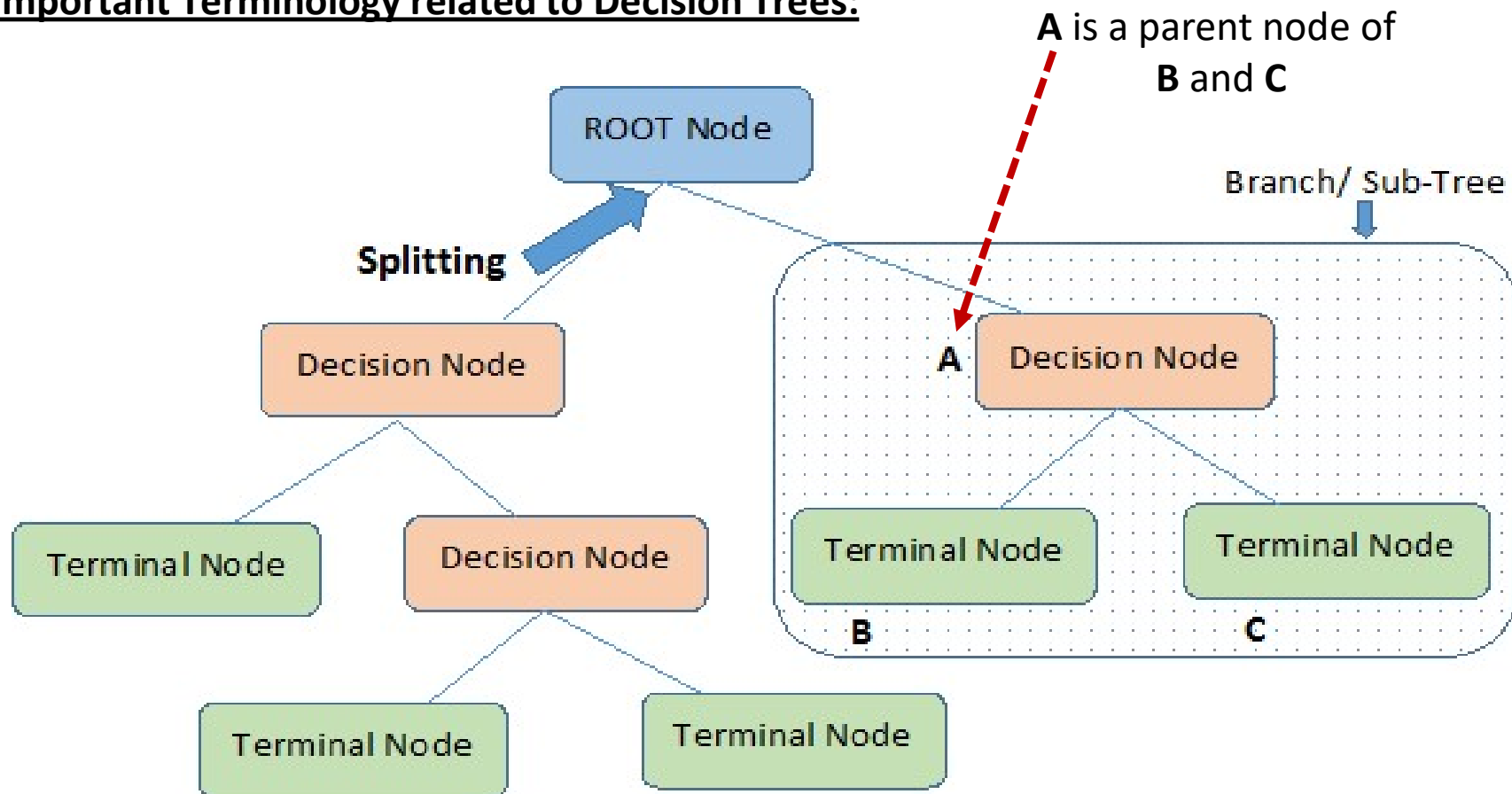
### Important Terminology related to Decision Trees:

1. **Root Node:** It represents the entire population or sample and this further gets divided into two or more **homogeneous** sets.
2. **Splitting:** It is the process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called a decision node.
4. **Leaf/ Terminal Node:** Nodes which are not split are called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can see it as the opposite process of splitting.
6. **Branch / Sub-Tree:** A sub section of the entire tree is called branch or sub-tree.
7. **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 children of the parent node.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Important Terminology related to Decision Trees:



# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Advantages of decision tree:

1. **Easy to Understand:** Its output is very easy to understand even for people from a non-analytical background. Its graphical representation is very intuitive and users can easily relate their hypothesis.
2. **Useful in Data exploration:** It is one of the fastest way to identify most significant variables and relation between two or more variables. It can also be used in data exploration stage.
3. **Less data cleaning required:** It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.
4. **Data type is not a constraint:** It can handle both numerical and categorical variables.
5. **Non-parametric Method:** Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Advantages of decision tree:

1. **Easy to Understand:** Its output is very easy to understand even for people from a non-analytical background. Its graphical representation is very intuitive and users can easily relate their hypothesis.
2. **Useful in Data exploration:** It is one of the fastest way to identify most significant variables and relation between two or more variables. It can also be used in data exploration stage.
3. **Less data cleaning required:** It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.
4. **Data type is not a constraint:** It can handle both numerical and categorical variables.
5. **Non-parametric Method:** Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.



# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Advantages of decision tree:

1. **Easy to Understand:** Its output is very easy to understand even for people from a non-analytical background. Its graphical representation is very intuitive and users can easily relate their hypothesis.
2. **Useful in Data exploration:** It is one of the fastest way to identify most significant variables and relation between two or more variables. It can also be used in data exploration stage.
3. **Less data cleaning required:** It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.
4. **Data type is not a constraint:** It can handle both numerical and categorical variables.
5. **Non-parametric Method:** Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Advantages of decision tree:

1. **Easy to Understand:** Its output is very easy to understand even for people from a non-analytical background. Its graphical representation is very intuitive and users can easily relate their hypothesis.
2. **Useful in Data exploration:** It is one of the fastest way to identify most significant variables and relation between two or more variables. It can also be used in data exploration stage.
3. **Less data cleaning required:** It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.
4. **Data type is not a constraint:** It can handle both numerical and categorical variables.
5. **Non-parametric Method:** Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Advantages of decision tree:

1. **Easy to Understand:** Its output is very easy to understand even for people from a non-analytical background. Its graphical representation is very intuitive and users can easily relate their hypothesis.
2. **Useful in Data exploration:** It is one of the fastest way to identify most significant variables and relation between two or more variables. It can also be used in data exploration stage.
3. **Less data cleaning required:** It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.
4. **Data type is not a constraint:** It can handle both numerical and categorical variables.
5. **Non-parametric Method:** Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.

# Introduction to Machine Learning

## What is a Decision Tree? How does it work?

### Disadvantages of decision tree:

1. **Over fitting:** Over fitting is one of the most practical difficulties for decision tree models. This problem gets solved by setting constraints on model parameters and pruning (discussed in detail later).
2. **Not as good for continuous variables:** While working with continuous numerical variables, decision tree loses information when it categorizes variables in different categories.

# Introduction to Machine Learning

## Regression Trees vs Classification Trees

- The terminal nodes (or leaves) lie at the bottom of the decision tree
- Decision trees are typically drawn upside down such that
  - roots are the tops
  - leaves are the bottom
- Both trees work almost similar to each other,
- There are primary differences & similarity between classification and regression trees:



# Introduction to Machine Learning

## Regression Trees vs Classification Trees

### Differences & similarities between classification and regression trees:

1. **Regression trees** are used when the dependent variable is continuous. **Classification trees** are used when the dependent variable is categorical.
2. **In case of regression trees**, the value obtained by terminal nodes in the training data is the **mean** response of observation falling in that region. Thus, if an unseen data observation falls in that region, we will make its prediction with the **mean value**.
3. **In case of classification tree**, the value (class) obtained by terminal node in the training data is the **mode** of observations falling in that region. Thus, if an unseen data observation falls in that region, we will make its prediction with the **mode value**.
4. **Both trees** divide the predictor space (independent variables) into distinct and non-overlapping regions.

# Introduction to Machine Learning

## Regression Trees vs Classification Trees

### Differences & similarities between classification and regression trees:

1. **Regression trees** are used when the dependent variable is continuous. **Classification trees** are used when the dependent variable is categorical.
2. **In case of regression trees**, the value obtained by terminal nodes in the training data is the **mean** response of observation falling in that region. Thus, if an unseen data observation falls in that region, we will make its prediction with the **mean value**.
3. **In case of classification tree**, the value (class) obtained by terminal node in the training data is the **mode** of observations falling in that region. Thus, if an unseen data observation falls in that region, we will make its prediction with the **mode value**.
4. **Both trees** divide the predictor space (independent variables) into distinct and non-overlapping regions.

# Introduction to Machine Learning

## Regression Trees vs Classification Trees

### Differences & similarities between classification and regression trees:

1. **Regression trees** are used when the dependent variable is continuous. **Classification trees** are used when the dependent variable is categorical.
2. **In case of regression trees**, the value obtained by terminal nodes in the training data is the **mean** response of observation falling in that region. Thus, if an unseen data observation falls in that region, we will make its prediction with the **mean value**.
3. **In case of classification tree**, the value (class) obtained by terminal node in the training data is the **mode** of observations falling in that region. Thus, if an unseen data observation falls in that region, we will make its prediction with the **mode value**.
4. **Both trees** divide the predictor space (independent variables) into distinct and non-overlapping regions.



# Introduction to Machine Learning

## Regression Trees vs Classification Trees

### Differences & similarities between classification and regression trees:

1. **Regression trees** are used when the dependent variable is continuous. **Classification trees** are used when the dependent variable is categorical.
2. **In case of regression trees**, the value obtained by terminal nodes in the training data is the **mean** response of observation falling in that region. Thus, if an unseen data observation falls in that region, we will make its prediction with the **mean value**.
3. **In case of classification tree**, the value (class) obtained by terminal node in the training data is the **mode** of observations falling in that region. Thus, if an unseen data observation falls in that region, we will make its prediction with the **mode value**.
4. **Both trees** divide the predictor space (independent variables) into distinct and non-overlapping regions.

# Introduction to Machine Learning

## Regression Trees vs Classification Trees

### Differences & similarities between classification and regression trees:

5. **Both the trees** follow a top-down greedy approach known as recursive binary splitting.
  - **top-down:** because it begins from the top of tree when all the observations are available in a single region and successively splits the predictor space into two new branches down the tree.
  - **greedy:** because the algorithm cares about only the current split, and not about future splits which will lead to a better tree.
6. **This splitting process** is continued until a user defined stopping criteria is reached. For example: we can tell the algorithm to stop once the number of observations per node becomes less than 50.
7. **In both cases**, the splitting process results in fully grown trees until the stopping criteria is reached.
  - But, the fully grown tree is likely to **overfit** data, leading to poor accuracy on unseen data. This brings '**pruning**'. Pruning is one of the techniques used to tackle overfitting.

# Introduction to Machine Learning

## Regression Trees vs Classification Trees

### Differences & similarities between classification and regression trees:

5. **Both the trees** follow a top-down greedy approach known as recursive binary splitting.
  - **top-down:** because it begins from the top of tree when all the observations are available in a single region and successively splits the predictor space into two new branches down the tree.
  - **greedy:** because the algorithm cares about only the current split, and not about future splits which will lead to a better tree.
6. **This splitting process** is continued until a user defined stopping criteria is reached. For example: we can tell the algorithm to stop once the number of observations per node becomes less than 50.
7. **In both cases**, the splitting process results in fully grown trees until the stopping criteria is reached.
  - But, the fully grown tree is likely to **overfit** data, leading to poor accuracy on unseen data. This brings '**pruning**'. Pruning is one of the techniques used to tackle overfitting.

# Introduction to Machine Learning

## Regression Trees vs Classification Trees

### Differences & similarities between classification and regression trees:

5. **Both the trees** follow a top-down greedy approach known as recursive binary splitting.
  - **top-down:** because it begins from the top of tree when all the observations are available in a single region and successively splits the predictor space into two new branches down the tree.
  - **greedy:** because the algorithm cares about only the current split, and not about future splits which will lead to a better tree.
6. **This splitting process** is continued until a user defined stopping criteria is reached. For example: we can tell the algorithm to stop once the number of observations per node becomes less than 50.
7. **In both cases**, the splitting process results in fully grown trees until the stopping criteria is reached.
  - But, the fully grown tree is likely to **overfit** data, leading to poor accuracy on unseen data. This brings '**pruning**'. Pruning is one of the techniques used to tackle overfitting.

# Introduction to Machine Learning

## How does a tree decide where to split?

- **The decision of making strategic splits heavily affects a tree's accuracy.**
  - The decision criteria is different for classification and regression trees.
- **Decision trees use multiple algorithms to decide how to split a node in two or more sub-nodes.**
  - The creation of sub-nodes increases the homogeneity of resultant sub-nodes
  - Decision tree splits the nodes on all available variables and then selects the split which results in the most homogeneous sub-nodes.
- **The algorithm selection is also based on the type of target variables.**
  - Four most commonly used algorithms in decision tree:
    1. Gini Index
    2. Chi-Square
    3. Information Gain
    4. Reduction in Variance

# Introduction to Machine Learning

## How does a tree decide where to split?

- **The decision of making strategic splits heavily affects a tree's accuracy.**
  - The decision criteria is different for classification and regression trees.
- **Decision trees use multiple algorithms to decide how to split a node in two or more sub-nodes.**
  - The creation of sub-nodes increases the homogeneity of resultant sub-nodes
  - Decision tree splits the nodes on all available variables and then selects the split which results in the most homogeneous sub-nodes.
- **The algorithm selection is also based on the type of target variables.**
  - Four most commonly used algorithms in decision tree:
    1. Gini Index
    2. Chi-Square
    3. Information Gain
    4. Reduction in Variance

# Introduction to Machine Learning

## How does a tree decide where to split?

- **The decision of making strategic splits heavily affects a tree's accuracy.**
  - The decision criteria is different for classification and regression trees.
- **Decision trees use multiple algorithms to decide how to split a node in two or more sub-nodes.**
  - The creation of sub-nodes increases the homogeneity of resultant sub-nodes
  - Decision tree splits the nodes on all available variables and then selects the split which results in the most homogeneous sub-nodes.
- **The algorithm selection is also based on the type of target variables.**
  - Four most commonly used algorithms in decision tree:
    1. Gini Index
    2. Chi-Square
    3. Information Gain
    4. Reduction in Variance

# Introduction to Machine Learning

## How does a tree decide where to split?

**Gini Index:** It says, if we randomly select two items from a population then they must be of the same class and the probability for this is 1 if the population is pure.

1. It works with categorical target variable “**Success**” or “**Failure**”.
2. Higher value of Gini → higher homogeneity.
3. CART (Classification and Regression Tree) uses Gini method to create **binary splits**.

### Steps to calculate Gini for a split

1. Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure ( $p^2 + q^2$ ).
2. Calculate Gini for split using weighted Gini score of each node of that split



# Introduction to Machine Learning

## How does a tree decide where to split?

**Gini Index:** It says, if we randomly select two items from a population then they must be of the same class and the probability for this is 1 if the population is pure.

1. It works with categorical target variable “**Success**” or “**Failure**”.
2. Higher value of Gini → higher homogeneity.
3. CART (Classification and Regression Tree) uses Gini method to create **binary splits**.

### Steps to calculate Gini for a split

1. Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure ( $p^2 + q^2$ ).
2. Calculate Gini for split using weighted Gini score of each node of that split

# Introduction to Machine Learning

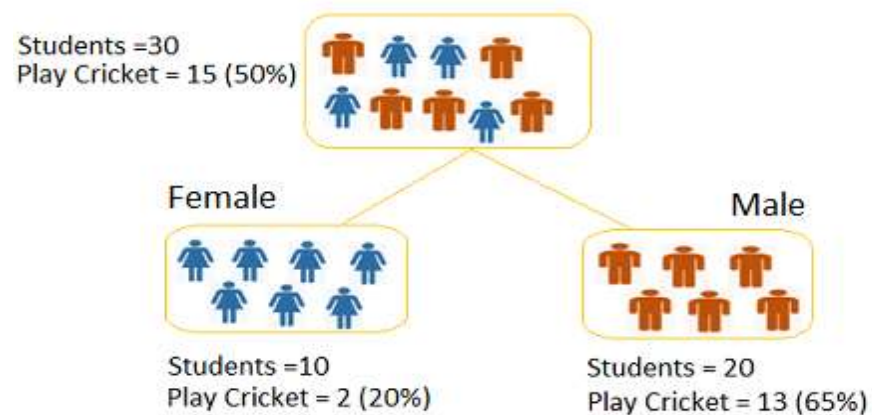
## How does a tree decide where to split?

### Gini Index:

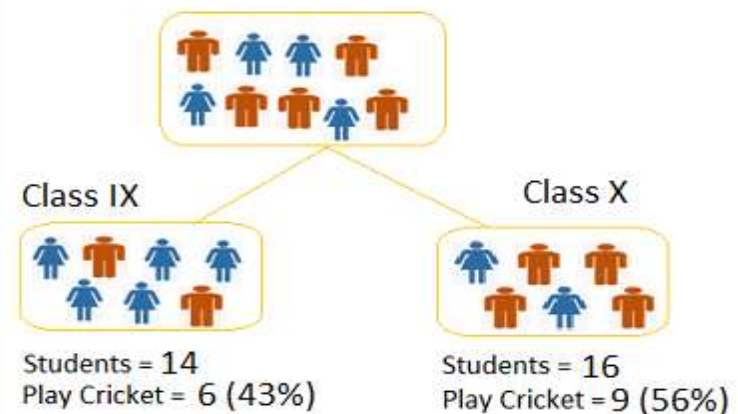
**Example:** Let's segregate the students based on target variable (playing cricket or not ).

- We split the population using two input variables **Gender** and **Class**.

#### Split on Gender



#### Split on Class

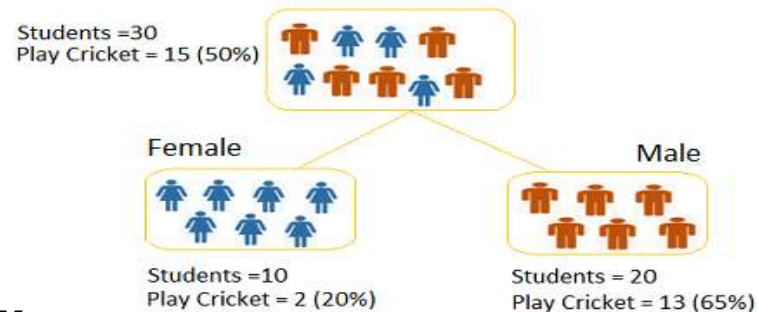


- Now, we want to identify which split is producing more **homogeneous** sub-nodes using **Gini index**.

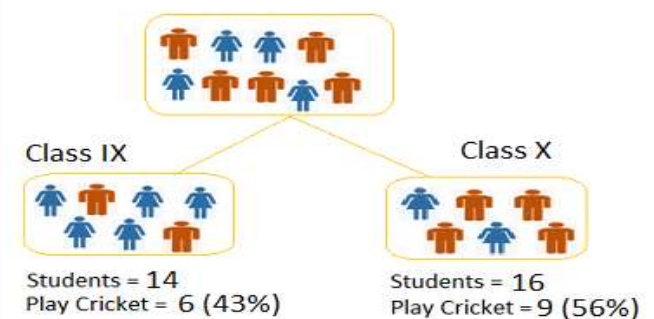
# Introduction to Machine Learning

## Gini Index Example:

### Split on Gender



### Split on Class



## Calculate Gini index

### Split on Gender:

1. Gini for sub-node Female  

$$= (0.2) * (0.2) + (0.8) * (0.8) = 0.68$$
2. Gini for sub-node Male  

$$= (0.65) * (0.65) + (0.35) * (0.35) = 0.55$$
3. Weighted Gini for Split Gender  

$$= (10/30) * 0.68 + (20/30) * 0.55 = \mathbf{0.59}$$

### Split on Class:

1. Gini for sub-node Class IX  

$$= (0.43) * (0.43) + (0.57) * (0.57) = 0.51$$
2. Gini for sub-node Class X  

$$= (0.56) * (0.56) + (0.44) * (0.44) = 0.51$$
3. Weighted Gini for Split Class  

$$= (14/30) * 0.51 + (16/30) * 0.51 = \mathbf{0.51}$$

**Decision:** Gini score for Split on **Gender** > Split on Class, hence, the node split will take place on **Gender**.

# Introduction to Machine Learning

## How does a tree decide where to split?

**Chi-Square:** It finds out the statistical significance between the differences of sub-nodes and parent node.

- It is measured by the sum of squares of standardized differences between observed and expected frequencies of the target variable.
  1. It works with categorical target variable “Success” or “Failure”.
  2. It can perform two or more splits.
  3. The higher the value of Chi-Square the higher the statistical significance
  4. Chi-Square of each node is calculated using formula,
$$\text{Chi-square} = ((\text{Actual} - \text{Expected})^2 / \text{Expected})^{1/2}$$
  5. The generated tree is called CHAID (Chi-square Automatic Interaction Detector)

# Introduction to Machine Learning

## How does a tree decide where to split?

### Chi-Square:

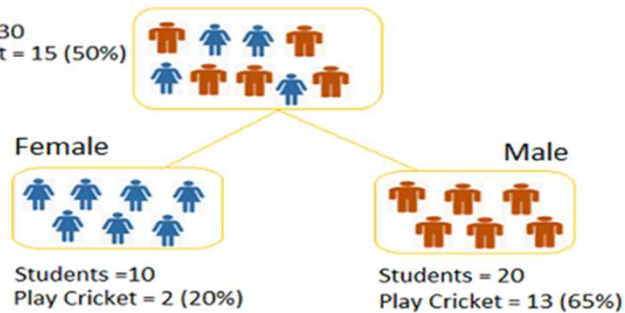
#### Steps to calculate Chi-square for a split:

1. Calculate Chi-square for individual node by calculating the deviation for Success and Failure both
2. Calculate Chi-square of Split using Sum of all Chi-square of Success and Failure of each node of the split

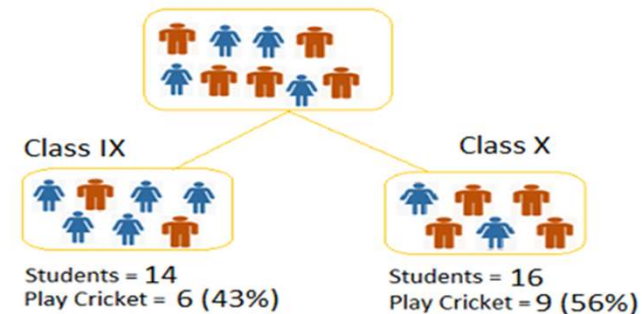
**Example:** Let's consider the same example that we have used to calculate Gini.

#### Split on Gender

Students = 30  
Play Cricket = 15 (50%)



#### Split on Class

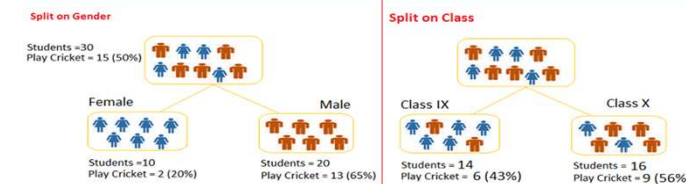


# Introduction to Machine Learning

## Chi-Square Example:

### Calculating Chi-square value for Gender

- First we are populating for node Female, populate the actual value for
  - “Play Cricket” and “Not Play Cricket”, these are **2** and **8**, respectively.
- Calculate expected value for “Play Cricket” and “Not Play Cricket”,
  - it would be **5** for both because the parent node has a probability of **50%** and we have to apply the same probability on **Female count (10)**.
- Calculate deviations by using formula, **Actual – Expected**.
  - It is for “Play Cricket” ( $2 - 5 = -3$ ) and for “Not play cricket” ( $8 - 5 = 3$ ).
- Calculate Chi-square of node for “Play Cricket” and “Not play Cricket”
  - formula =  $((\text{Actual} - \text{Expected})^2 / \text{Expected})^{1/2}$ .
- Follow similar steps for calculating Chi-square value for Male node.
- Now add all Chi-square values to calculate Chi-square for split Gender.
- ➔ We can use a table for the calculation



# Introduction to Machine Learning

## Chi-Square Example:

### Calculating Chi-square value for gender

#### Split on Gender

Students = 30  
Play Cricket = 15 (50%)

Female

Students = 10  
Play Cricket = 2 (20%)

Male

Students = 20  
Play Cricket = 13 (65%)

#### Split on Class

Class IX

Students = 14  
Play Cricket = 6 (43%)

Class X

Students = 16  
Play Cricket = 9 (56%)

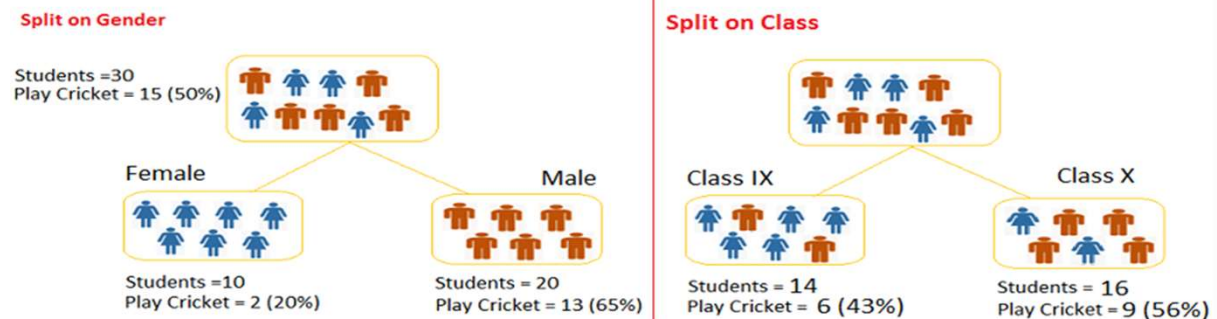
#### Split on Gender

Node	Play Cricket	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
Female	2	8	10	5	5	-3	3	1.34	1.34
Male	13	7	20	10	10	3	-3	0.95	0.95
Total Chi-Square								4.58	

# Introduction to Machine Learning

## Chi-Square Example:

### Calculating Chi-square value for Class



**Decision:** Chi-square also identifies that the **Gender** split is more significant compared to the **Class** split

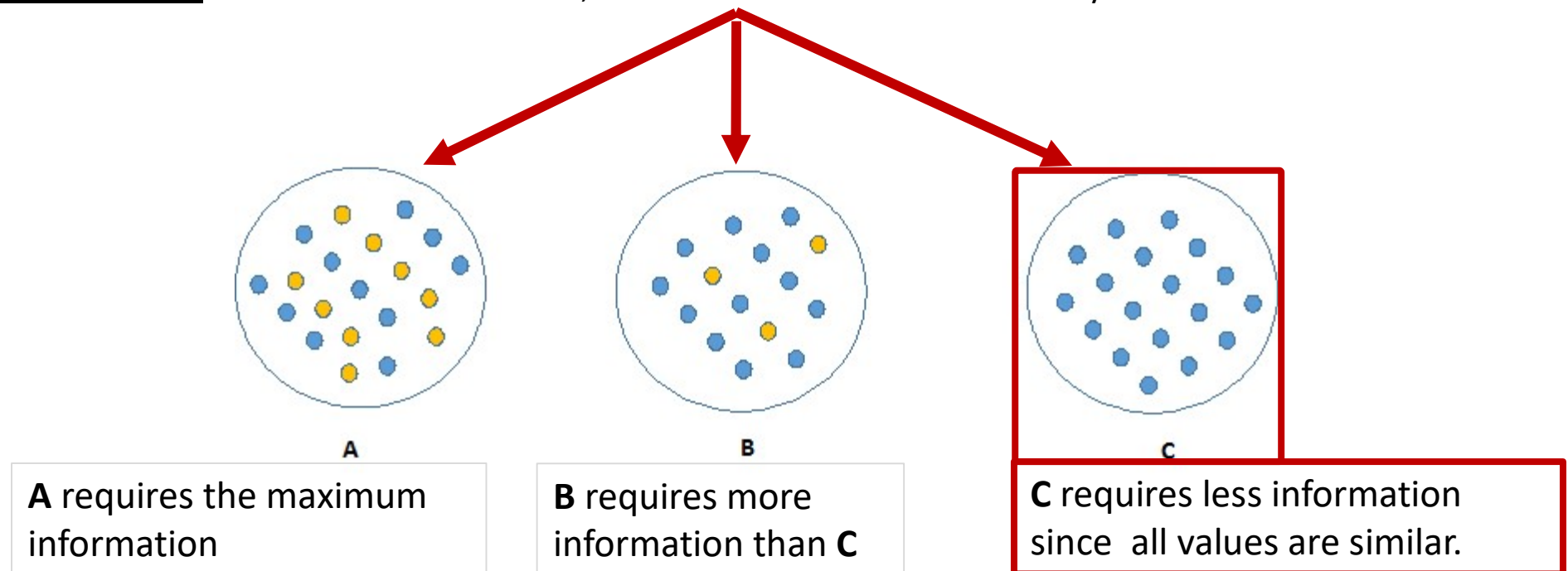
$$\text{Chi-square}(\text{Gender}) > \text{Chi-square}(\text{Class})$$



# Introduction to Machine Learning

## How does a tree decide where to split?

**Information Gain:** Consider the three nodes, which node can be described easily



**C** is a pure node, **B** is less impure and **A** is more impure.

# Introduction to Machine Learning

## How does a tree decide where to split?

### Conclusion for Information Gain:

- Less impure node requires less information to describe it
- 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
- if the sample is equally divided (50% – 50%), it has an entropy of one.
- If we consider the probability of success (p) and failure (q) in each node
  - Entropy =  $-p \log_2 p - q \log_2 q$
- Entropy is also used with categorical target variables.
  - It chooses the split which has the lowest entropy compared to parent node and other splits.
  - ***The smaller the entropy, the better it is.***

# Introduction to Machine Learning

## How does a tree decide where to split?

### Steps to calculate entropy for a split:

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

#### Split on Gender

Students = 30  
Play Cricket = 15 (50%)



Female



Students = 10  
Play Cricket = 2 (20%)

Male



Students = 20  
Play Cricket = 13 (65%)

#### Split on Class



Class IX



Students = 14  
Play Cricket = 6 (43%)

Class X



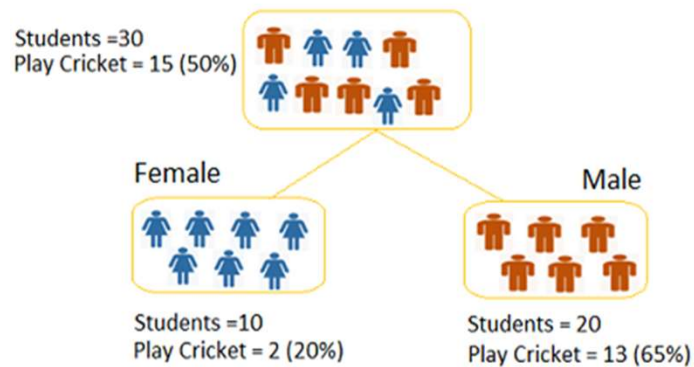
Students = 16  
Play Cricket = 9 (56%)

# Introduction to Machine Learning

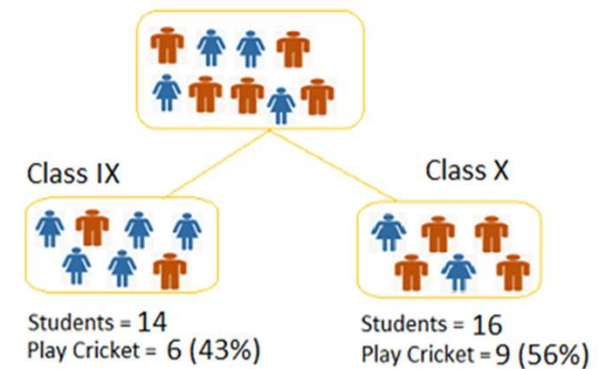
## How does a tree decide where to split?

### Calculate entropy for a split:

Split on Gender



Split on Class



1. Entropy for parent node =  $-(15/30) \log_2 (15/30) - (15/30) \log_2 (15/30) = 1$ .
2. Entropy for:
  - Female node =  $-(2/10) \log_2 (2/10) - (8/10) \log_2 (8/10) = \mathbf{0.72}$
  - Male node =  $-(13/20) \log_2 (13/20) - (7/20) \log_2 (7/20) = \mathbf{0.93}$
3. Entropy for split Gender = Weighted entropy of sub-nodes =  $(10/30) * 0.72 + (20/30) * 0.93 = \mathbf{0.86}$

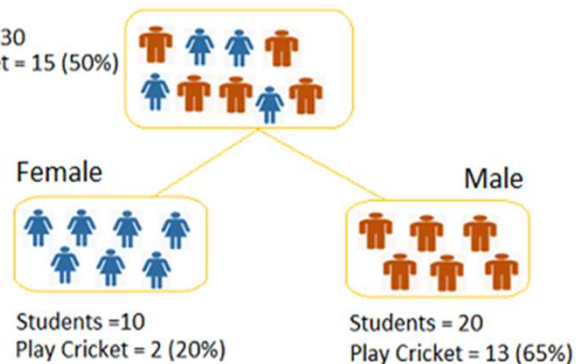
# Introduction to Machine Learning

## How does a tree decide where to split?

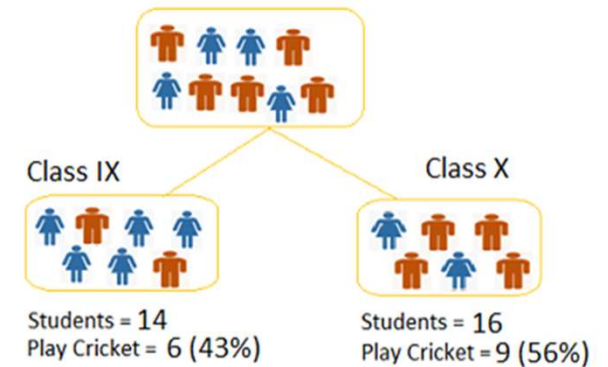
### Calculate entropy for a split:

#### Split on Gender

Students = 30  
Play Cricket = 15 (50%)



#### Split on Class



**Decision:** Entropy for *Split on Gender* is the lowest among all, so the tree will split on *Gender*.

We can derive information gain from entropy as **1 - Entropy**.

# Introduction to Machine Learning

## How does a tree decide where to split?

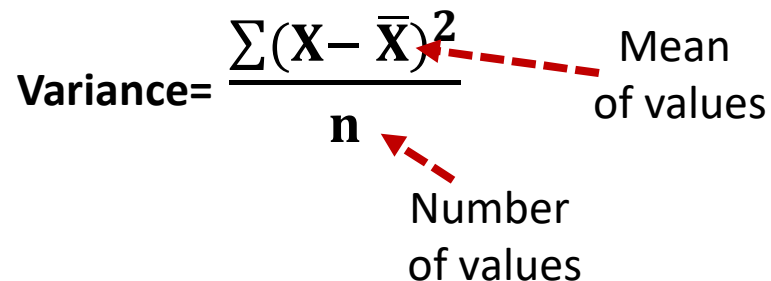
**Reduction in Variance:** used for continuous target variables (regression problems).

- This algorithm uses the standard formula of variance to choose the best split.
- The split with lower variance is selected as the criteria to split the population:

$$\text{Variance} = \frac{\sum (X - \bar{X})^2}{n}$$

Mean of values

Number of values



### Steps to calculate Variance:

1. Calculate variance for each node.
2. Calculate variance for each split as weighted average of each node variance.

# Introduction to Machine Learning

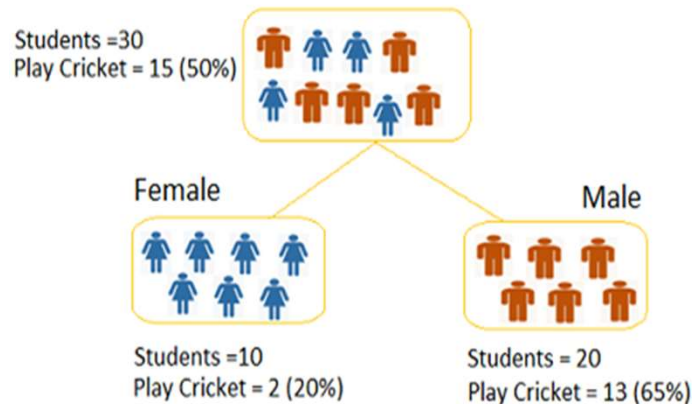
## How does a tree decide where to split?

### Example:

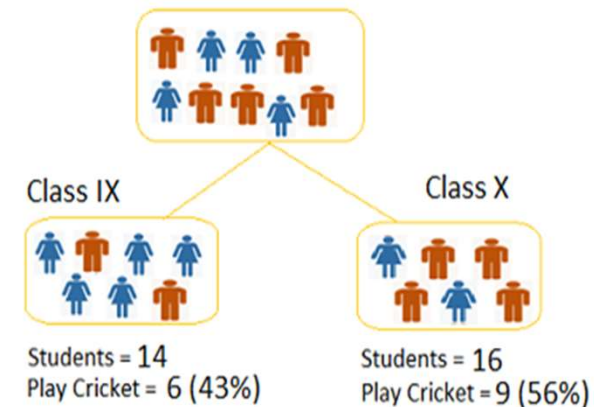
1. Assign numerical value
  - 1 for play cricket
  - 0 for not playing cricket
2. Apply Reduction in Variance and identify the best split:

- Variance for Root node,
  - **mean**  $\rightarrow ((15 \times 1) + (15 \times 0)) / 30 = 0.5$
  - **variance**  $\rightarrow ((15 \times (1-0.5)^2) + (15 \times (0-0.5)^2)) / 30 = \mathbf{0.25}$
- **Mean of Female node**  $= ((2 \times 1) + (8 \times 0)) / 10 = 0.2$  & **Variance**  $= (2 \times (1-0.2)^2 + 8 \times (0-0.2)^2) / 10 = 0.16$
- **Mean of Male Node**  $= ((13 \times 1) + (7 \times 0)) / 20 = 0.65$  & **Variance**  $= (13 \times (1-0.65)^2 + 7 \times (0-0.65)^2) / 20 = 0.23$
- Variance for Split Gender = Weighted Variance of Sub-nodes  $= (10/30) \times 0.16 + (20/30) \times 0.23 = \mathbf{0.21}$

Split on Gender



Split on Class



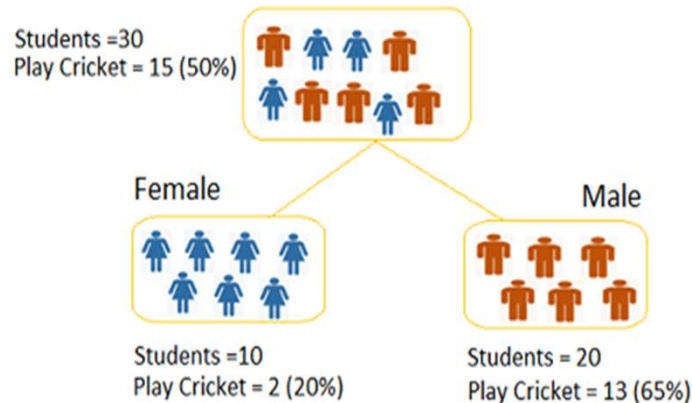
# Introduction to Machine Learning

## How does a tree decide where to split?

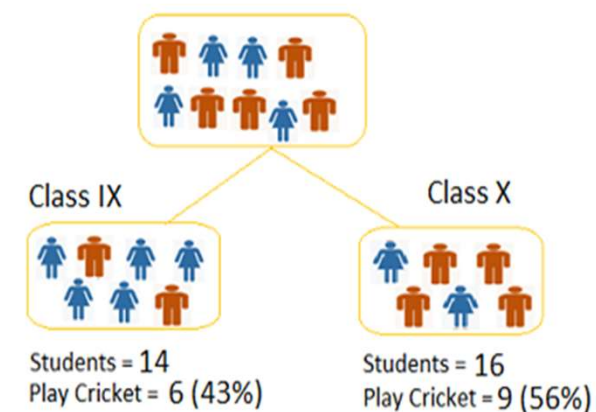
### Example:

1. Assign numerical value
  - 1 for play cricket
  - 0 for not playing cricket
2. Apply Reduction in Variance and identify the best split:

#### Split on Gender



#### Split on Class



- Mean of Class IX node =  $(6 \times 1 + 8 \times 0) / 14 = 0.43$  and Variance =  $(6 \times (1 - 0.43)^2 + 8 \times (0 - 0.43)^2) / 14 = 0.24$
- Mean of Class X node =  $(9 \times 1 + 7 \times 0) / 16 = 0.56$  and Variance =  $(9 \times (1 - 0.56)^2 + 7 \times (0 - 0.56)^2) / 16 = 0.25$
- Variance for Split Class =  $(14/30) \times 0.24 + (16/30) \times 0.25 = \mathbf{0.25}$



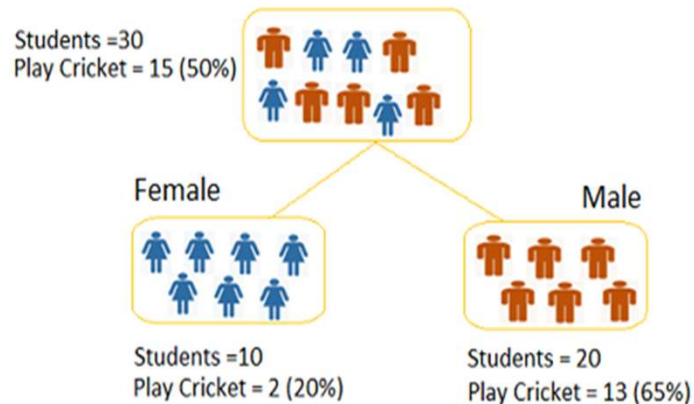
# Introduction to Machine Learning

## How does a tree decide where to split?

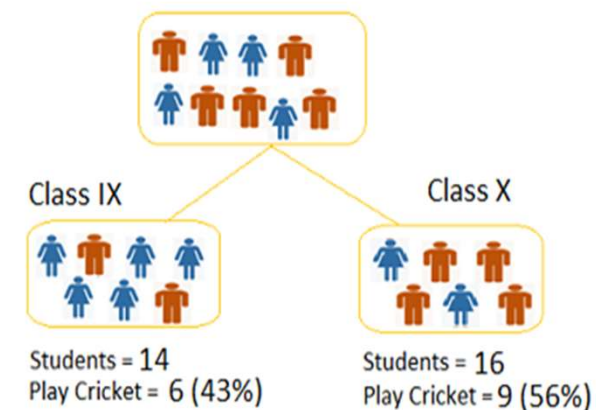
### Example:

1. Assign numerical value
  - 1 for play cricket
  - 0 for not playing cricket
2. Apply Reduction in Variance and identify the best split:

### Split on Gender



### Split on Class



**Decision:** Gender split has lower variance compared to parent node, so the split would take place on ***Gender*** variable.

# Introduction to Machine Learning

## Are tree based models better than linear models?

➔ If I can use,

➔ logistic regression for classification problems

➔ linear regression for regression problems,

Why is there a need to use decision trees?

**Actually, we can use any algorithm. It depends on the type of problem you are solving.**

❖ Some key factors to decide which algorithm to use:

- If the relationship between dependent & independent variable is well approximated by a linear model, linear regression will outperform tree based model.
- If there is a high non-linearity & complex relationship between dependent & independent variables, a tree model will outperform a classical regression method.
- If you need to build a model which is easy to explain to people,
  - A decision tree model will always do better than a linear model.
  - Decision tree models are even simpler to interpret than linear regression!

## Example:

Import the data file called “**animals.csv**” into the variable **mydata**

```
> mydata= read.table("animals.csv", header=TRUE, sep=";", stringsAsFactors = TRUE)
# Check the dimensions of the data
# View statistical summary of dataset
# View the complete data
```

### Process the dataset

- Divide **mydata** in two data frames as **dfTraining** (70%) and **dfTest** (30%)
- Create a **tree** for classifying the data based on the animal with **dfTraining**
  - (function in R: **tree**, **you should first install the library**)
- Check the created tree and its performance on the training data
- Plot the tree
- Create a **confusion matrix** for the predicted and true animal on **dfTest**

## Example:

### Process the dataset

- Divide **mydata** in two data frames as **dfTraining** (70%) and **dfTest** (30%)  
*>countTraining = round(nrow(mydata)\*0.7)*  
*>randomRows=sample(1:nrow(mydata), size= countTraining, replace=F)*  
*>dfTraining = mydata[randomRows,]*  
*>dfTest = mydata[- randomRows,]*
- Create a **tree** for classifying the data based on the animal with **dfTraining** (function in R: **tree**)
- Check the created tree and its performance on the training data
- Plot the tree
- Create a **confusion matrix** for the predicted and true animal on **dfTest**

## Example:

### Process the dataset

- Create a **tree** for classifying the data based on the animal with **dfTraining** (function in R: **tree**)  
*>library(tree) #library containing the tree function*  
*>myTree = tree(Animal~., dfTraining)*  
*#Create a tree with Animal as class attribute and all other attributes as variables*
- Check the created tree and its performance on the training data
- Plot the tree
- Create a **confusion matrix** for the predicted and true animal on **dfTest**

## Example:

### Process the dataset

- Check the created tree and its performance on the training data

```
>myTree
```

```
>summary(myTree)
```

```
# plot the complete dataset based on the variables that are actually used in tree construction (with ggplot)
```

```
> ggplot(data = mydata, aes(x=seq_along(Torso_7), y= Torso_7)) +geom_point(aes(color=Animal))
```

- Plot the tree
- Create a **confusion matrix** for the predicted and true animal on **dfTest**

# Decision Trees

## Example:

### Process the dataset

- Plot the tree

*>plot(myTree, type = "uniform") #type = "uniform" so that all branches are of the same length*

*>text(myTree)*

- Create a **confusion matrix** for the predicted and true animal on **dfTest**

## Example:

### Process the dataset

- Plot the tree

```
>plot(myTree, type = "uniform") #type = "uniform" so that all branches are of the same length  
>text(myTree)
```

- Create a **confusion matrix** for the predicted and true animal on **dfTest**

```
>myPred = predict(myTree, dfTest, type = "class")  
>table(myPred, dfTest$Animal)
```



## Exercise:

Import the data file called “**plantData.csv**” into the variable **mydata**

### Process the dataset

- Divide **mydata** in two data frames as **dfTraining** (70%) and **dfTest** (30%)
- Create a **tree** for classifying the data based on the species with **dfTraining** (function in R: **tree**)
- Check the created tree and its performance on the training data
- Plot the tree
- Create a **confusion matrix** for the predicted and true species on **dfTest**

## Exercise:

Import the data file called “**plantData.csv**” into the variable **mydata**

```
>mydata = read.table("plantData.csv", header = TRUE, sep = ",", stringsAsFactors = TRUE)
```

### Process the dataset

- Divide **mydata** in two data frames as **dfTraining** (70%) and **dfTest** (30%)
- Create a **tree** for classifying the data based on the species with **dfTraining** (function in R: **tree**)
- Check the created tree and its performance on the training data
- Plot the tree
- Create a **confusion matrix** for the predicted and true species on **dfTest**

## Exercise:

### Process the dataset

- Divide **mydata** in two data frames as **dfTraining** (70%) and **dfTest** (30%)  
*>countTraining = round(nrow(mydata)\*0.7)*  
*>randomRows=sample(1:nrow(mydata), size= countTraining, replace=F)*  
*>dfTraining = mydata[randomRows,]*  
*>dfTest = mydata[- randomRows,]*
- Create a **tree** for classifying the data based on the species with **dfTraining** (function in R: **tree**)
- Check the created tree and its performance on the training data
- Plot the tree
- Create a **confusion matrix** for the predicted and true species on **dfTest**

## Exercise:

### Process the dataset

- Create a **tree** for classifying the data based on the species with **dfTraining** (function in R: **tree**)

```
>myTree = tree(Species~., dfTraining)
```

- Check the created tree and its performance on the training data
- Plot the tree
- Create a **confusion matrix** for the predicted and true species on **dfTest**

## Exercise:

### Process the dataset

- Check the created tree and its performance on the training data

*>myTree*

*>summary(myTree)*

- Plot the tree
- Create a **confusion matrix** for the predicted and true species on **dfTest**

## Exercise:

### Process the dataset

- Plot the tree

```
>plot(myTree, type = "uniform")
```

```
>text(myTree)
```

- Create a **confusion matrix** for the predicted and true species on **dfTest**

## Exercise:

### Process the dataset

- Create a **confusion matrix** for the predicted and true species on **dfTest**  
*>table(myPred, dfTest\$Species)*