Inf620 - Lecture Decision Tree

Depto de Informática - UFV

Ricardo Ferreira
ricardo@ufv.br

2025

Introduction

- Class Material (click here for Colab)
- **Review**: Supervised Learning with KNN and Naive Bayes
- **Problems**: Classification and Regression
- Decision Tree Technique:
    - What is a tree?
    - How trees are built
    - Classification and Regression

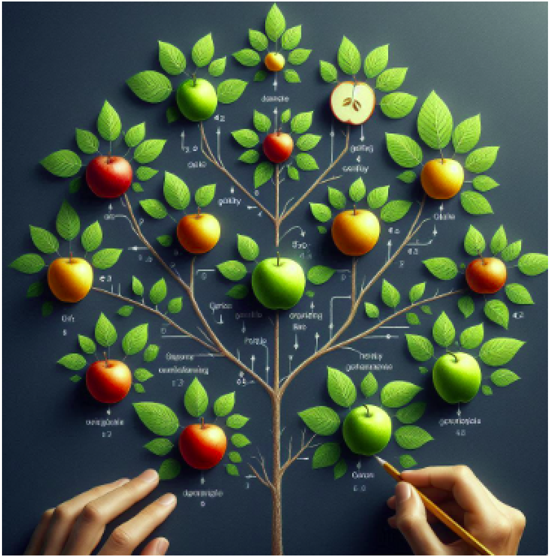
Review of KNN and Naive Bayes

- **Definition**: KNN - Supervised method used for classification and regression.
  - Simple
  - Classifies the test point based on the majority of neighbors (for classification) or average of neighbors (for regression).
- **Naive Bayes**: Probability
  - Classification
  - Faster than KNN, not always better
  - Independent variables

Additional Material

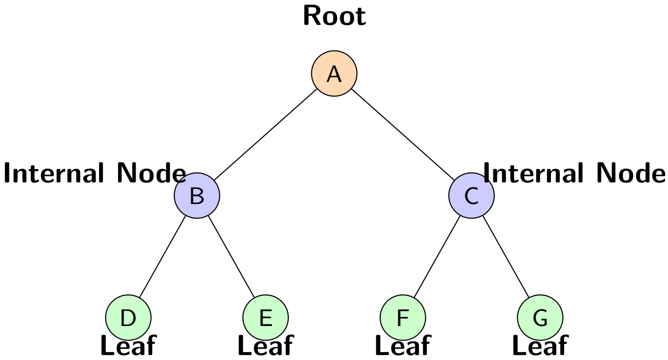
- Decision Tree - How it works (Machine Learning) - Video lecture (click here)
- Decision Trees Raphael Campos - blog (click here)
- Part 4 - Decision Tree, Random Forest and Gradient Boosting - Now or never! - Video lecture (click here)

Tree



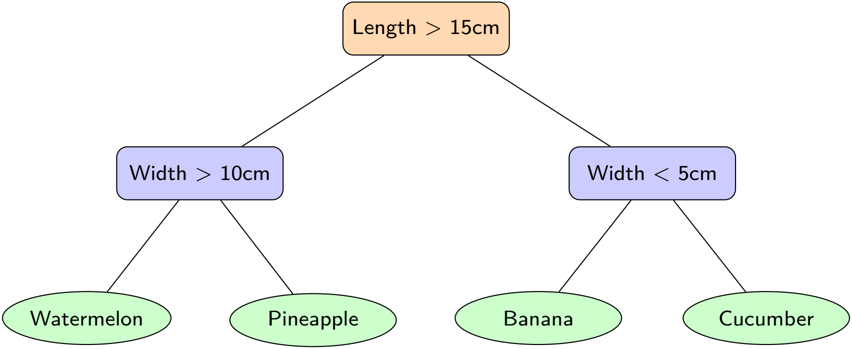
Tree in Computing

Decision Tree Terminology



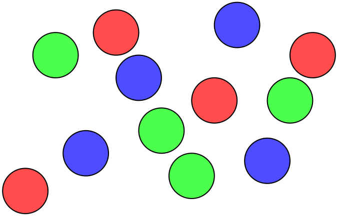
- Root (A): Initial node of the tree
- Internal Nodes (B, C): Nodes with children
- Leaves (D, E, F, G): Nodes without children

Fruit Classification by Dimensions



- Root: First decision based on length
- Decisions: Based on fruit width
- Fruits: Final classification

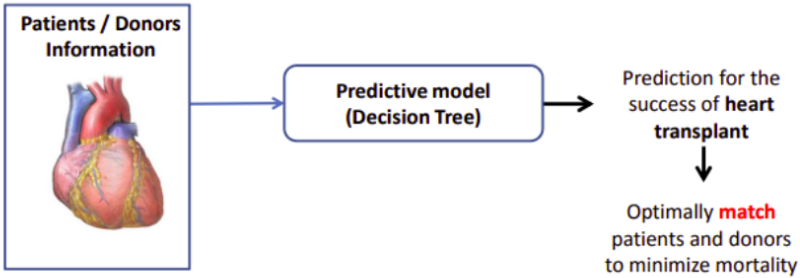
How to Separate Objects



Decision Tree: Overview

- **Definition:** A classification model that uses a tree structure for decision making
- **Main characteristics:**
  - Splits data into subsets based on conditions on the attributes
  - Hierarchical structure with decision nodes and class leaves
  - Easily interpretable and visually intuitive

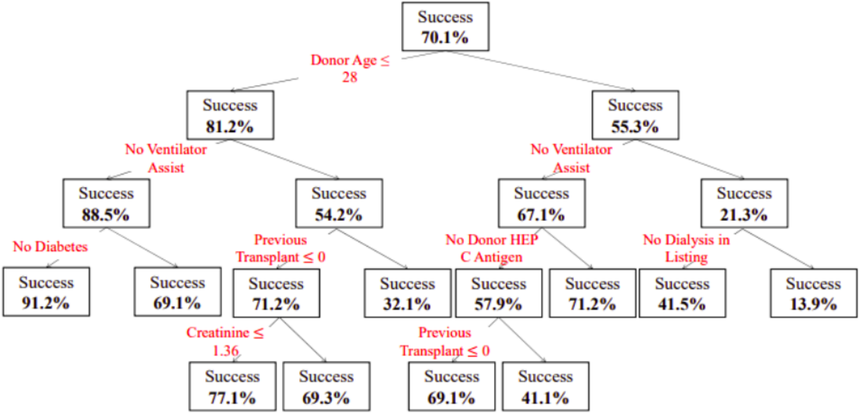
Exemplo de Árvore de Decisão



Type	Explanation	Note
Patient	56,716 patients (heart transplant patients)	follow-up until they died
Feature	141 Features (84 Continuous / 57 Binary)	From 1986 to 2015
Label	Dead: 16,986 Patients (29.95%) Alive: 39,730 Patients (70.05%)	

Clique aqui maiores informações

Decision Tree Example

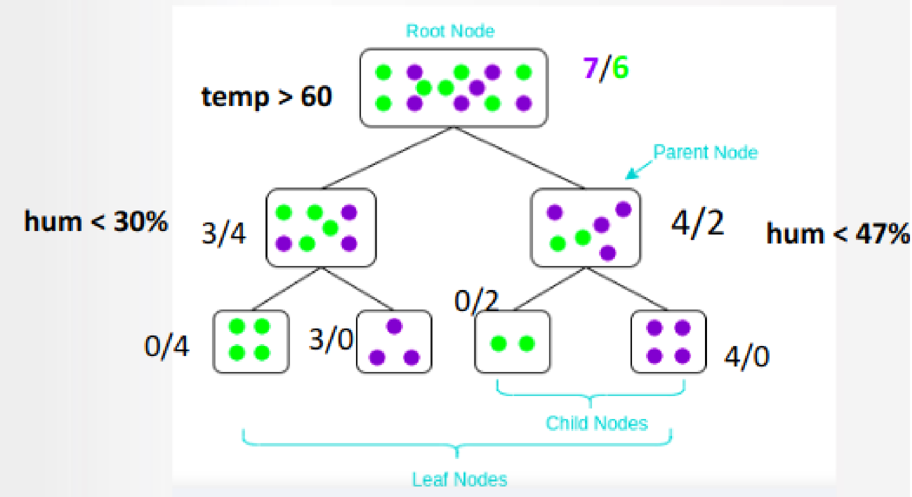


Click here for more information

Advantages of Decision Trees

- **Easy to understand** - Explainability
- Graphical representation is very intuitive
- Useful in data exploration
- Helps identify the most significant variables
- Requires less data cleaning
- Relatively unaffected by outliers and missing values
- Data type is not a restriction
  - Can handle numerical and categorical variables
- Non-parametric method
  - Works with different distributions and classifier structures

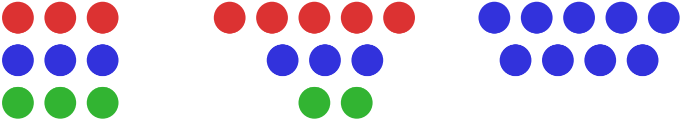
Decision Tree Example



How to classify?

Entropy in Decision Trees

Scenario A (High Entropy) | Scenario B (Medium Entropy) | Scenario C (Low Entropy)



Observations:

- Scenario C groups objects of the same class;
- C is a pure node, B is less impure, and A is the most impure
- Entropy measures the degree of disorder

Example 1: Entropy with 8 letters A

- Letter distribution: A A A A A A A A
- Probabilities: $p(A) = 1.0$
- **Entropy Calculation:**

$$H = -\sum p_i \log_2 p_i = -(1.0 \cdot \log_2 1.0) = 0$$

- **Entropy**: $H = 0$ (maximum certainty)

Example 2: Entropy with 4 A, 2 B, 1 C, 1 D

- Letter distribution: A A A A, B B, C, D
- Probabilities:

$$p(A) = 0.5, \quad p(B) = 0.25, \quad p(C) = 0.125, \quad p(D) = 0.125$$

- **Entropy Calculation:**

$$H = -\sum p_i \log_2 p_i =$$

$$= -(0.5 \log_2 0.5 + 0.25 \log_2 0.25 + 0.125 \log_2 0.125 + 0.125 \log_2 0.125)$$

- **Entropy**: $H \approx 1.75$

Example 3: Entropy with 2 A, 2 B, 2 C, 2 D

- Letter distribution: A A, B B, C C, D D
- Probabilities:

$$p(A) = 0.25, \quad p(B) = 0.25, \quad p(C) = 0.25, \quad p(D) = 0.25$$

- **Entropy Calculation:**

$$H = -\sum p_i \log_2 p_i =$$

$$= -(0.25 \log_2 0.25 + 0.25 \log_2 0.25 + 0.25 \log_2 0.25 + 0.25 \log_2 0.25)$$

- **Entropy**: $H = 2.0$ (maximum uncertainty for 4 equally likely events)

Entropy: Measuring Purity

- **Definition of Entropy:**
  - A measure of uncertainty or randomness in a probability distribution.
  - Entropy is calculated as:

    $$H = -\sum p_i \log_2 p_i$$

    where p_i is the probability of each class in the set.
- **Purity of a Set:**
  - Low entropy indicates a "pure"set (higher concentration of a single class).
  - High entropy suggests the set is "impure"(more uniform distribution among classes).
- **Application:**
  - Used in machine learning algorithms, such as Decision Trees, to determine the best data splits.
  - The goal is to minimize entropy after each split, increasing the purity of the resulting subsets.

Dataset Example and Entropy Calculation

Manufacturing	Mileage	Test Drive	Purchase
Recent	Low	Yes	Yes
Recent	High	Yes	Yes
Old	Low	No	No
Recent	High	No	No

Entropy Calculation with 4 samples

- Frequencies: **Yes:** 2 occurrences, **No:** 2 occurrences
- Probabilities:
  - $p_{\text{Yes}} = \frac{2}{4} = 0.5$
  - $p_{\text{No}} = \frac{2}{4} = 0.5$
- **Class Entropy (Purchase):**

$$-(0.5 \cdot \log_2 0.5 + 0.5 \cdot \log_2 0.5) = 1.0$$

Dataset Example - Column: Manufacturing

Manufacturing	Mileage	Test Drive	Purchase
Recent	Low	Yes	Yes
Recent	High	Yes	Yes
Old	Low	No	No
Recent	High	No	No

Information Gain (IG) for Manufacturing

- Recent: 2 yes, 1 no
- Not Recent: 1 no
- $$E_{recent} = -\left[\frac{2}{3}\log_2\left(\frac{2}{3}\right) + \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right] = -[-0.3900 - 0.5283]$$
  $$\approx 0.9183$$
- $IG = 1 - \frac{3}{4} \cdot E_{recent} - \frac{1}{4} \cdot E_{not} = 1 - \frac{3}{4} \cdot 0.9184 - \frac{2}{4} \cdot 1 = 0.69$

Dataset Example - Column: Mileage

Manufacturing	Mileage	Test Drive	Purchase
Recent	Low	Yes	Yes
Recent	High	Yes	Yes
Old	Low	No	No
Recent	High	No	No

Information Gain (IG) for Mileage

- Low: 1 yes, 1 no
- Not Low: 1 yes, 1 no
- $IG = 1 - \frac{2}{4} \cdot E_{low} - \frac{2}{4} \cdot E_{notLow} = 1 - \frac{2}{4} \cdot 1 - \frac{2}{4} \cdot 1 = 0$

Dataset Example - Column: Test Drive

Manufacturing	Mileage	Test Drive	Purchase
Recent	Low	Yes	Yes
Recent	High	Yes	Yes
Old	Low	No	No
Recent	High	No	No

Information Gain (IG) for Test Drive

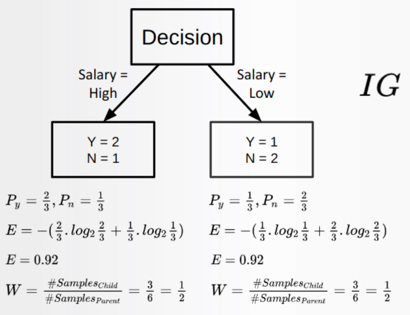
- Passed Test: 2 yes
- Did Not Pass: 2 no
- $IG = 1 - \frac{2}{4} \cdot E_{passed} - \frac{2}{4} \cdot E_{not} = 1 - \frac{2}{4} \cdot 0 - \frac{2}{4} \cdot 0 = 1$

Should I Accept the Job?

Salary	Localization	Tasks	Decision
High	Far	Interesting	Yes
Low	Near	Not Interesting	No
Low	Far	Interesting	Yes
High	Far	Not Interesting	No
High	Near	Interesting	Yes
Low	Far	Not Interesting	No

Salary?

Salary	Localization	Tasks	Decision
High	Far	Interesting	Yes
Low	Near	Not Interesting	No
Low	Far	Interesting	Yes
High	Far	Not Interesting	No
High	Near	Interesting	Yes
Low	Far	Not Interesting	No

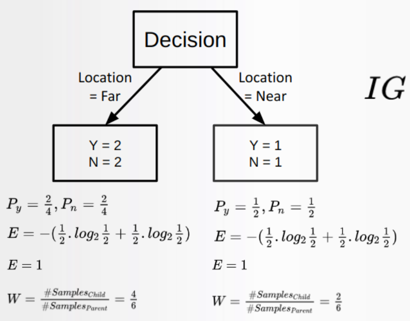


$$IG = 1 - \frac{1}{2}0.92 - \frac{1}{2}0.92 = 0.08$$

, small gain...

Location?

Salary	Localization	Tasks	Decision
High	Far	Interesting	Yes
Low	Near	Not Interesting	No
Low	Far	Interesting	Yes
High	Far	Not Interesting	No
High	Near	Interesting	Yes
Low	Far	Not Interesting	No

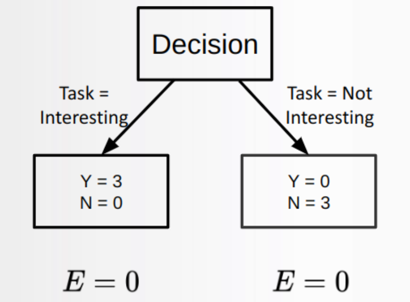


$$IG = 1 - \frac{4}{6}1 - \frac{2}{6}1 = 0$$

, no gain...

Tasks?

Salary	Localization	Tasks	Decision
High	Far	Interesting	Yes
Low	Near	Not Interesting	No
Low	Far	Interesting	Yes
High	Far	Not Interesting	No
High	Near	Interesting	Yes
Low	Far	Not Interesting	No



$$IG = 1 - \frac{3}{6}0 - \frac{3}{6}0 = 1$$

, best gain!

Gini Coefficient

Example: Identify if a person has Heart Disease

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes

$$Gini\ impurity = 1 - (P_y)^2 - (P_n)^2$$

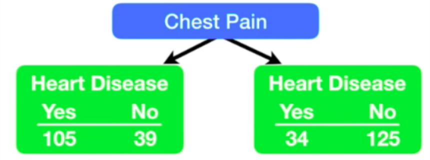
Two classes .....
but could be multiple classes

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

Click here for StatQuest

Gini Coefficient

Gini impurity



$$Gini\ impurity = 1 - (P_y)^2 - (P_n)^2$$

$$GI = 1 - \left(\frac{105}{105+39}\right)^2 - \left(\frac{39}{105+39}\right)^2$$

$$GI = 0.395$$

$$GI = 1 - \left(\frac{34}{34+125}\right)^2 - \left(\frac{125}{34+125}\right)^2$$

$$GI = 0.336$$

Weighted average

$$GI = \left(\frac{144}{144+159}\right) 0.395 + \left(\frac{159}{144+159}\right) 0.336$$

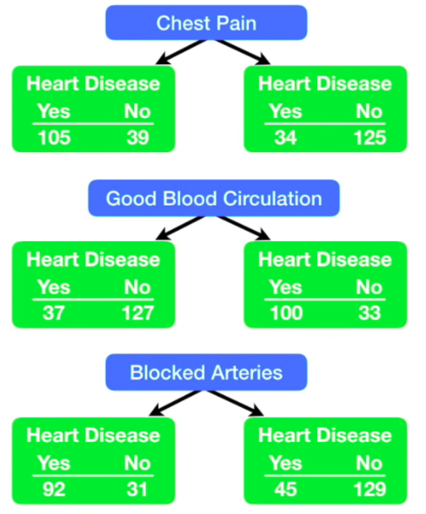
$$GI = 0.364$$

Click here for StatQuest

Gini Coefficient

Gini impurity

- Good Blood Circulation presents the lowest GI
  - Divides best the sample
  - Root node for our decision tree



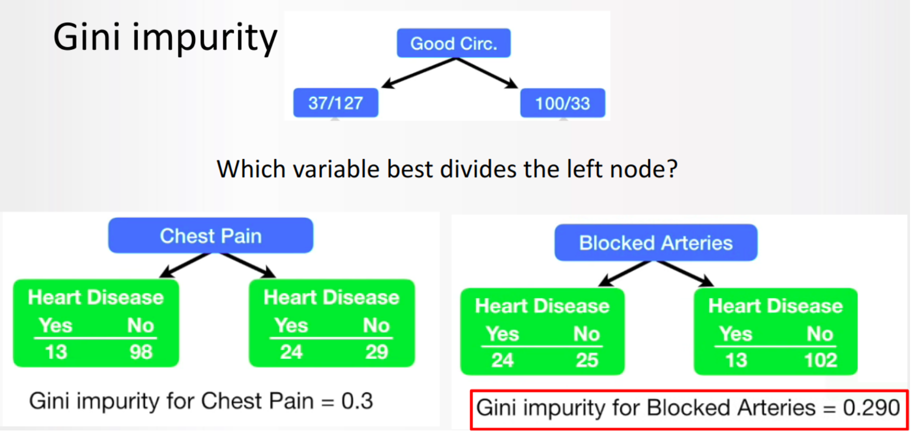
GI = 0.364

$\boxed{\text{GI = 0.360}}$

GI = 0.381

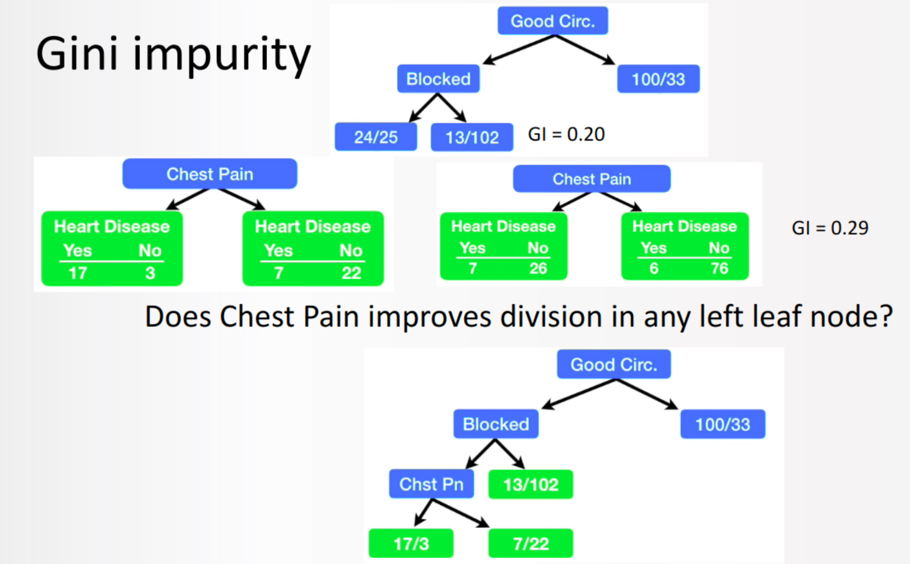
Click here for StatQuest

Gini Coefficient



Click here for StatQuest

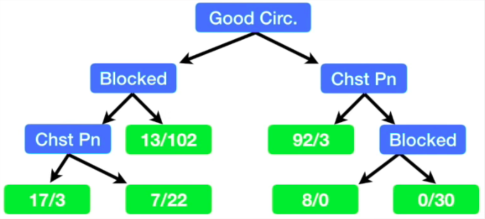
Gini Coefficient



Click here for StatQuest

Gini Coefficient

- Repeat for the right nodes
  1. Calculate all the Gini impurity scores
  2. If the node itself has the lowest score, it becomes the leaf node
  3. If separating the data results in an improvement, than pick the separation with the lowest impurity value

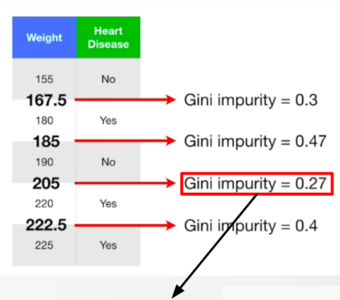


Click here for StatQuest

Numerical Values

- Numerical data
  - How do we determine what's the best weight to use to divide the patients?
    1. Sort the patients by weight
    2. Calculate the average weight for all adjacent patients
    3. Calculate the impurity values for each average weight

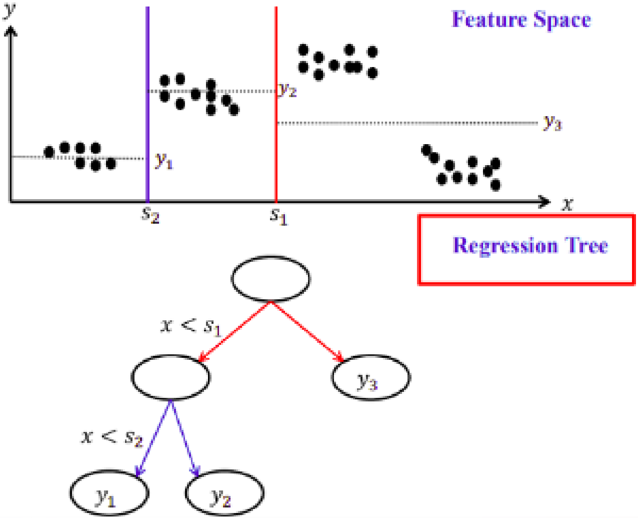
Weight	Heart Disease
220	Yes
180	Yes
225	Yes
190	No
155	No



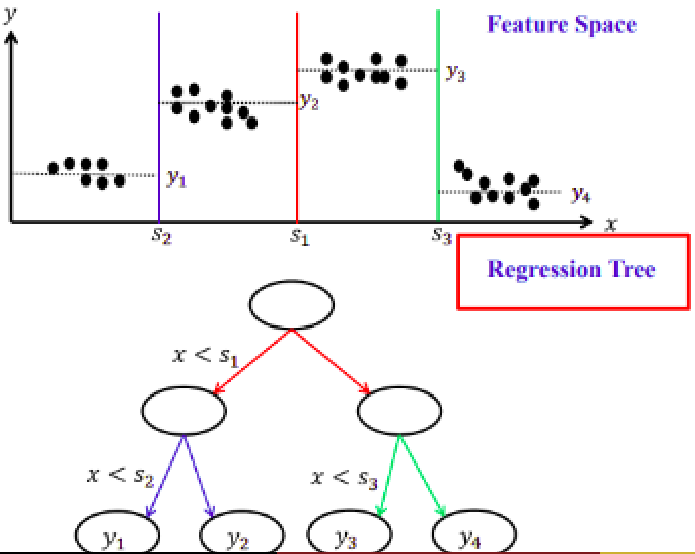
Use the lowest to divide weight

Click here for StatQuest

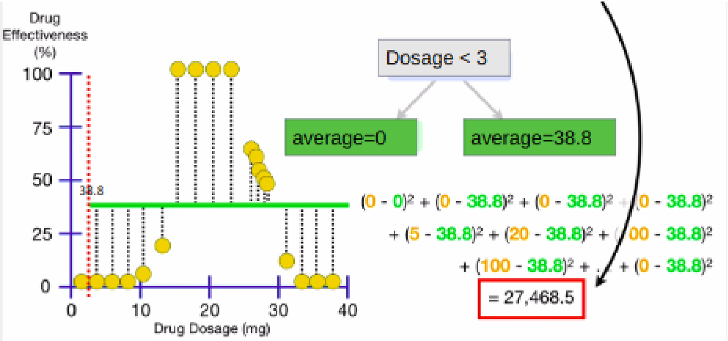
Can It Be Used for Regression?



Decision Tree for Regression

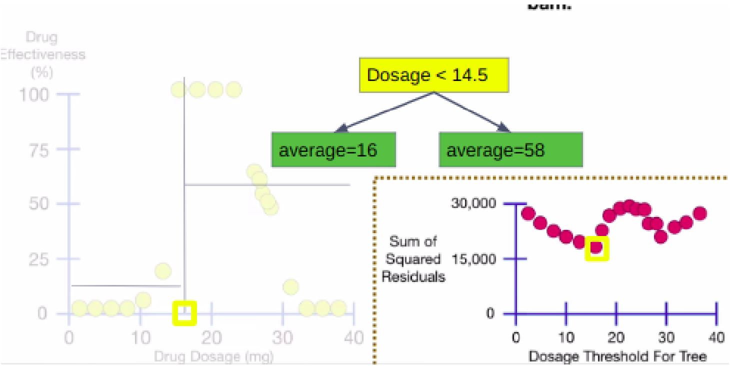


Decision Tree for Regression



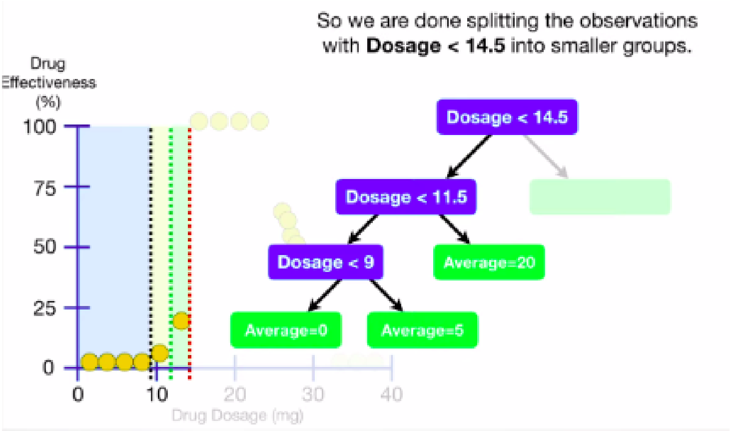
Click here for StatQuest

Decision Tree for Regression



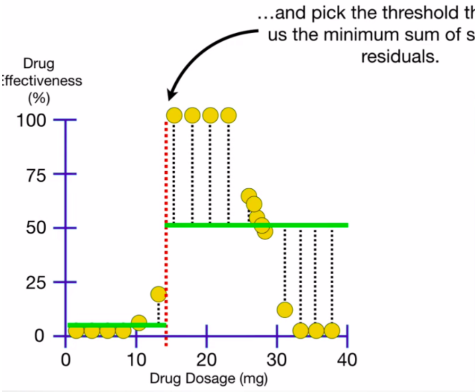
Click here for StatQuest

Decision Tree for Regression



Click here for StatQuest

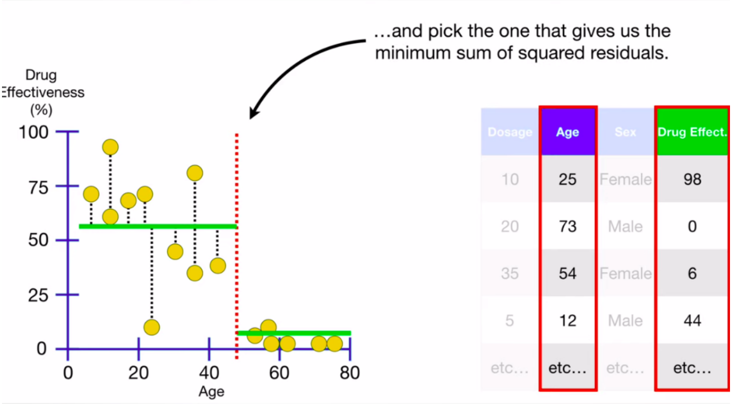
Decision Tree for Regression



Dosage	Age	Sex	Drug Effect.
10	25	Female	98
20	73	Male	0
35	54	Female	6
5	12	Male	44
etc...	etc...	etc...	etc...

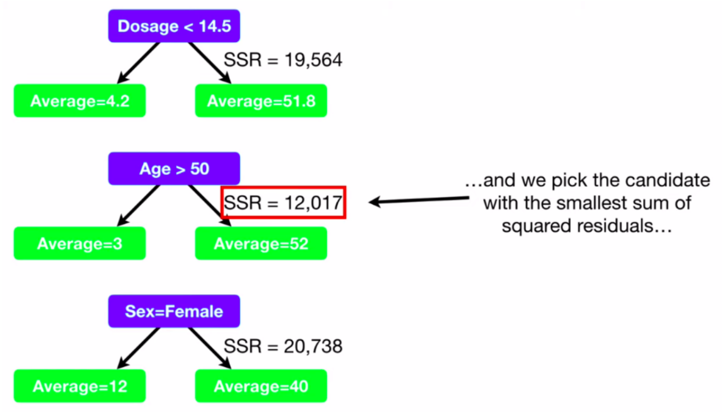
Click here for StatQuest

Decision Tree for Regression



Click here for StatQuest

Decision Tree for Regression



Click here for StatQuest

Gini vs Entropy: Which to Choose?

Gini Index

- Computationally faster
- Tends to isolate the majority class
- Simpler formula: $Gini = 1 - \sum_{i=1}^{n} p_i^2$

Entropy

- More balanced trees
- Higher computational cost
- Formula: $H = -\sum_{i=1}^{n} p_i \log_2(p_i)$

Conclusion

- Results are similar in practice
- Choose based on context:
  - Gini: better performance
  - Entropy: when balance is important

Hyperparameters in Decision Trees

Depth Control

- `max_depth`: Maximum depth, prevents overfitting
- `max_leaf_nodes`:
  - Maximum number of leaves
  - Controls final complexity

Split Criteria

- `criterion`:
  - 'gini' or 'entropy'
- `splitter`:
  - 'best': Best split
  - 'random': Random split

`min_samples_split`:

- Minimum number of samples to split a node
- Higher values: simpler tree, lower values: more complex tree

Tuning Tips

- Start with default options
- Use cross-validation for optimization
- Balance complexity and generalization

Evolution of Machine Learning Algorithms

1750	Naive Bayes
1943	Neural Network: Threshold Logic
1957	K-means & KNN
1963	Support Vector Machine
1986	Neural Networks: Backpropagation
1987	Convolutional Networks
2009	Deep Learning: ImageNet
2012	AlexNet
2016	Inception, ResNet

Core Algorithms

1984	**CART (Classification And Regression Tree) Breiman, Friedman, Olshen & Stone**
1995	Random Forest Combines multiple decision trees
2001	Gradient Boosting Decision Trees

Evolution of Decision Trees