Non-parametric models

KNN and Decision Trees

Tech Lead Data Science

Master en Data Science
2022-2023

Assembler
Institute of Technology

ÍNDICE

PARAMETRIC VS NON-PARAMETRIC

PARAMETRIC VS NON-PARAMETRIC

PARAMETRIC

The amount of parameters does **not depend on the size** of the sample.

NON-PARAMETRIC

The amount of parameters is not fixed and **might depend on the size** of the sample.

Assembler
Institute of Technology

PARAMETRIC VS NON-PARAMETRIC

PARAMETRIC

The amount of parameters does **not depend on the size** of the sample.

Linear regression

Logistic regression

NON-PARAMETRIC

The amount of parameters is not fixed and **might depend on the size** of the sample.

Assembler
Institute of Technology

PARAMETRIC VS NON-PARAMETRIC

PARAMETRIC	NON-PARAMETRIC
The amount of parameters does **not depend on the size** of the sample.	The amount of parameters is not fixed and **might depend on the size** of the sample.
Linear regression Logistic regression	KNN (K-Nearest Neighbours) Decision Trees

Assembler
Institute of Technology

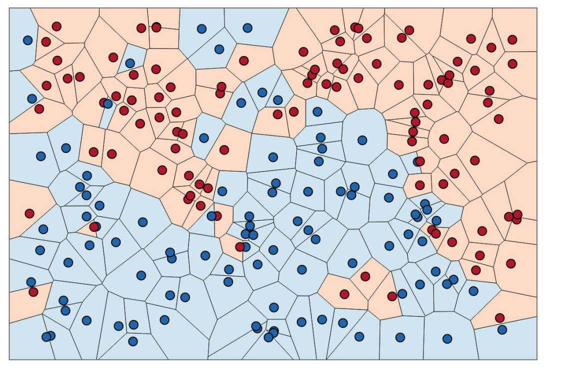
K-NEAREST NEIGHBOURS

INTRODUCTION

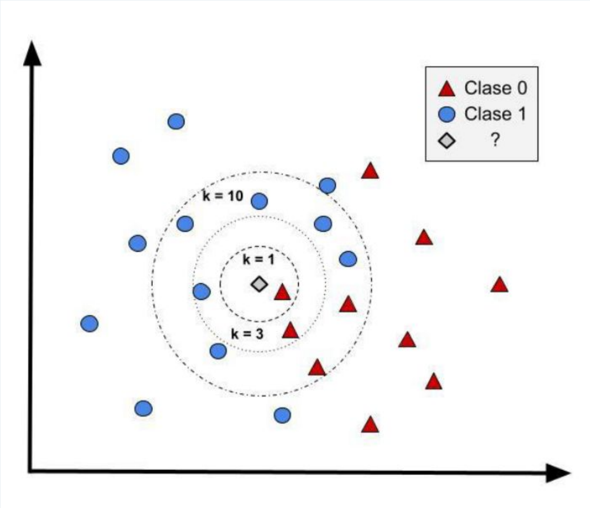
- K-Nearest Neighbours (KNN) are **non-parametric models** that can be used for both **classification and regression** problems.

- The main idea of this models is to **use the k nearest data to predict new data**:
  - For classification, they will search for the class with highest frequency
  - For regression, they will take the mean of the k nearest data

- k will be set by the user.



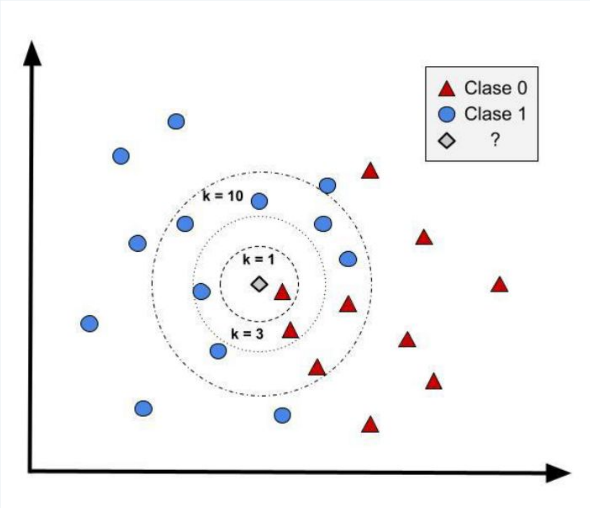
HOW DOES IT WORK?



HOW DOES IT WORK?

KNN follows this structure to predict Y from X:

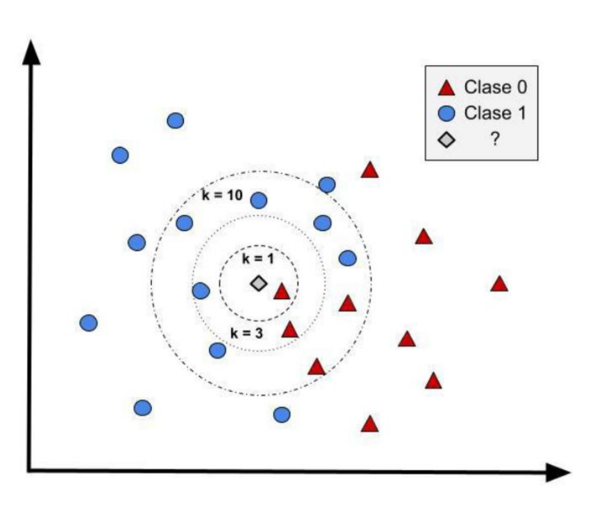
1. Calculates the distance from every data to Y.
2. Choose the nearest k values to Y.
3. Labeled Y using a criteria:
    - Mode for classification
    - Mean for regression



HOW DOES IT WORK?

KNN follows this structure to predict Y from X:

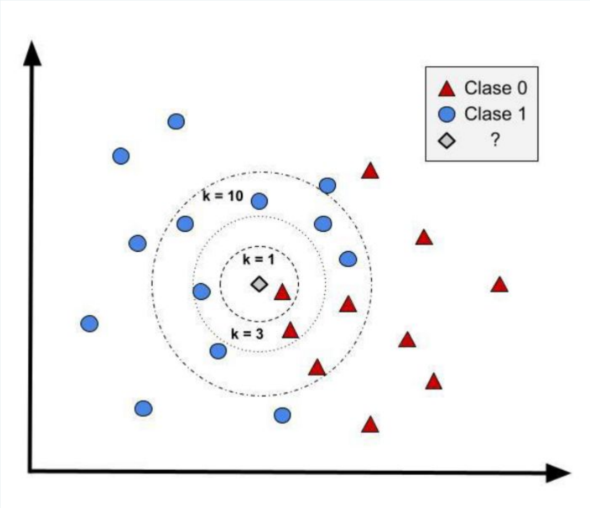
1. Calculates the distance from every data to Y.
2. Choose the nearest k values to Y.
3. Labeled Y using a criteria:
    - **Mode for classification**
    - Mean for regression



HOW DOES IT WORK?

KNN follows this structure to predict Y from X:

1. Calculates the distance from every data to Y.
2. Choose the nearest k values to Y.
3. Labeled Y using a criteria:
    - **Mode for classification**
    - Mean for regression

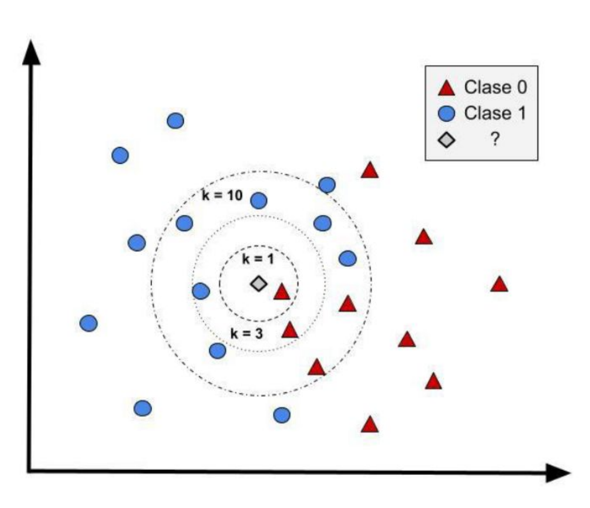


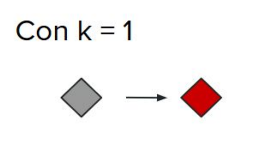
Con k = 1

HOW DOES IT WORK?

KNN follows this structure to predict Y from X:

1. Calculates the distance from every data to Y.
2. Choose the nearest k values to Y.
3. Labeled Y using a criteria:
    - **Mode for classification**
    - Mean for regression



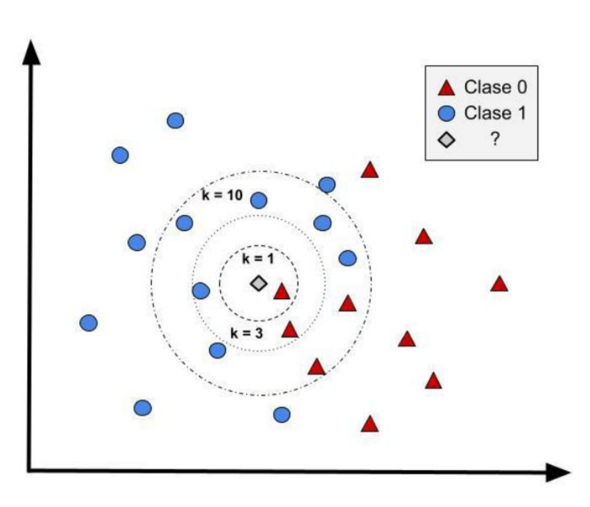


KNN MODELS

HOW DOES IT WORK?

KNN follows this structure to predict Y from X:

1. Calculates the distance from every data to Y.
2. Choose the nearest k values to Y.
3. Labeled Y using a criteria:
    - **Mode for classification**
    - Mean for regression



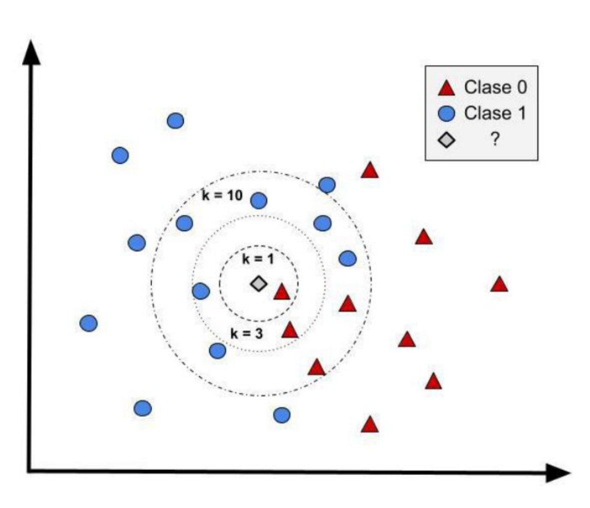
Con k = 1

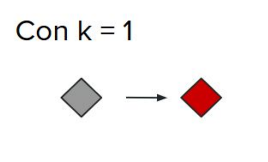
Con k = 3

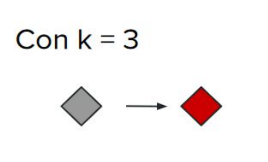
HOW DOES IT WORK?

KNN follows this structure to predict Y from X:

1. Calculates the distance from every data to Y.
2. Choose the nearest k values to Y.
3. Labeled Y using a criteria:
   - **Mode for classification**
   - Mean for regression





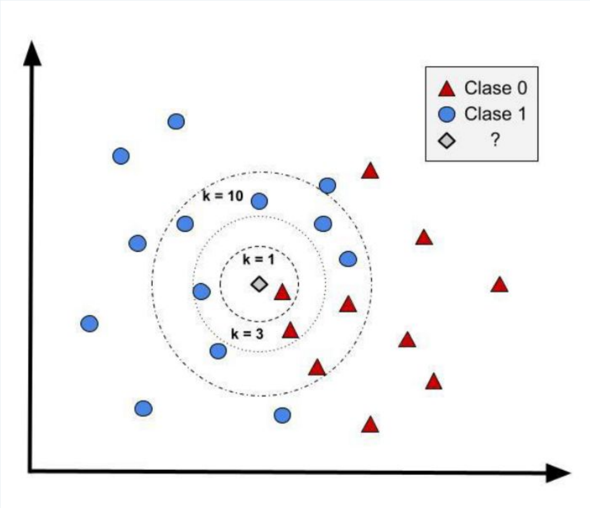


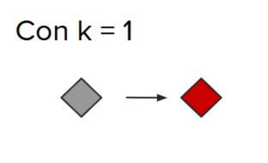
Assembler
Institute of Technology

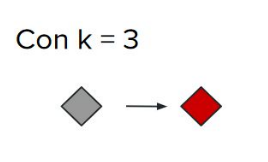
HOW DOES IT WORK?

KNN follows this structure to predict Y from X:

1. Calculates the distance from every data to Y.
2. Choose the nearest k values to Y.
3. Labeled Y using a criteria:
   - **Mode for classification**
   - Mean for regression





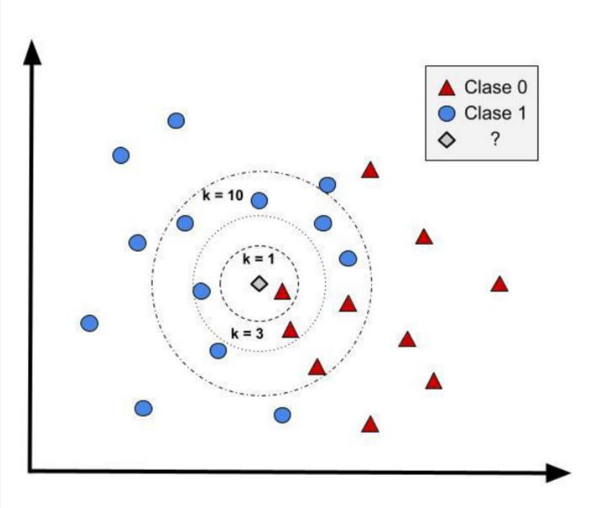


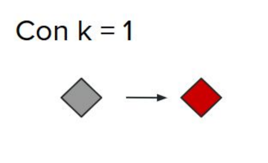
Con k = 10

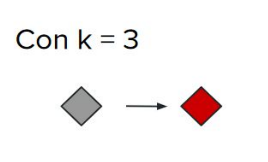
HOW DOES IT WORK?

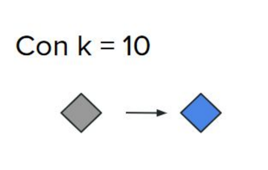
KNN follows this structure to predict Y from X:

1. Calculates the distance from every data to Y.
2. Choose the nearest k values to Y.
3. Labeled Y using a criteria:
    - **Mode for classification**
    - Mean for regression



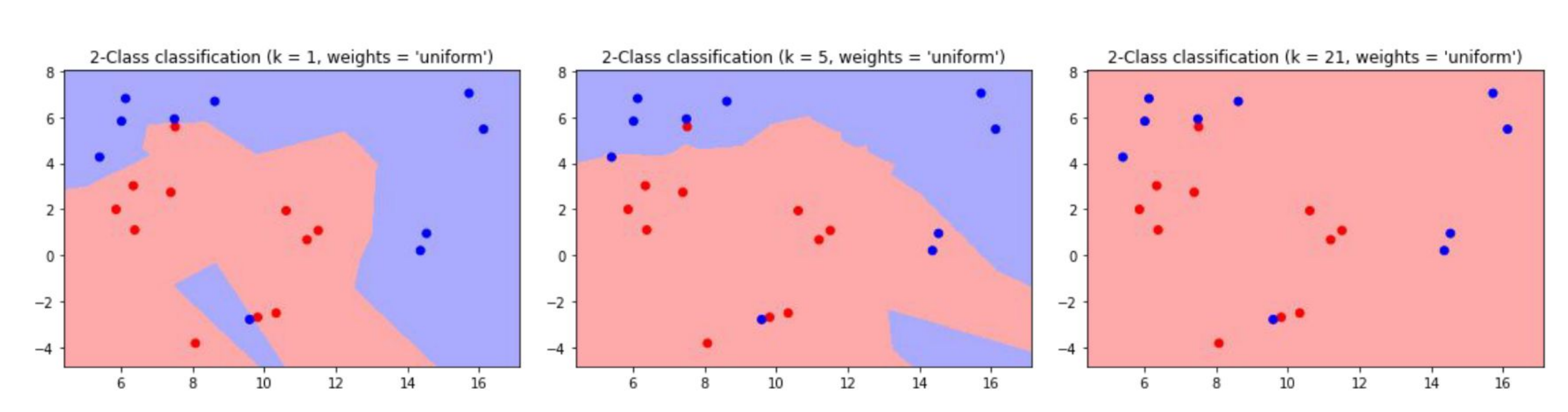






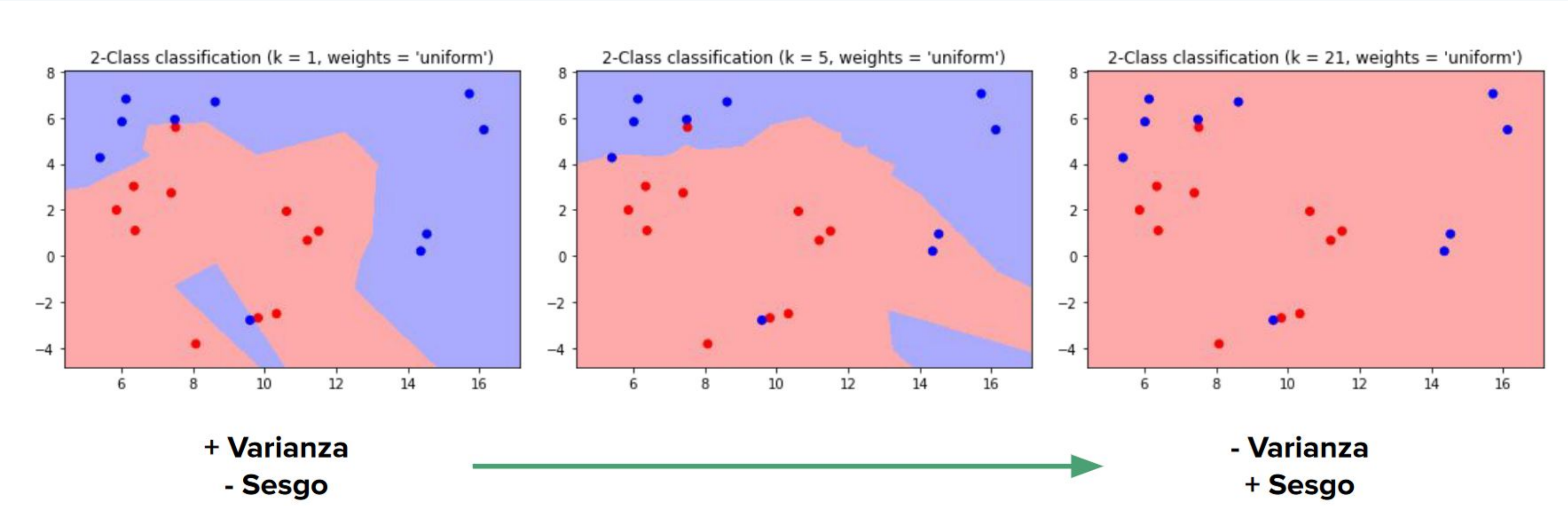
KNN MODELS

K'S IMPACT



Assembler
Institute of Technology

K'S IMPACT


2-Class classification (k = 1, weights = 'uniform')
2-Class classification (k = 5, weights = 'uniform')
2-Class classification (k = 21, weights = 'uniform')
+ Varianza
- Sesgo
- Varianza
+ Sesgo

PREDICTIVE STRATEGIES

CLASSIFICATION	REGRESSION

PREDICTIVE STRATEGIES

CLASSIFICATION

- To use the modal class: **uniform weights**
- To weight the nearest classes considering the distances: **distance weights**

$$\frac{1}{d(x, x_i)}$$

REGRESSION

PREDICTIVE STRATEGIES

CLASSIFICATION

- To use the modal class: **uniform weights**
- To weight the nearest classes considering the distances: **distance weights**

$$\frac{1}{d(x, x_i)}$$

REGRESSION

- **uniform weights**

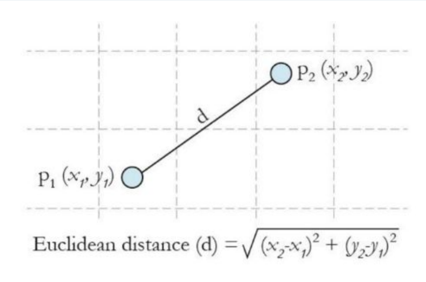
$$\hat{y} = \frac{1}{k} \sum_{i \in Vec(x)} y_i$$

- **distance weights**

$$\hat{y} = \frac{1}{k} \sum_{i \in Vec(x)} \frac{y_i}{d(x, x_i)}$$

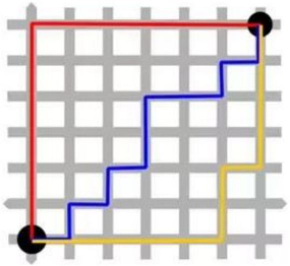
DISTANCES

The most commonly used distance is the **Euclidean distance**:



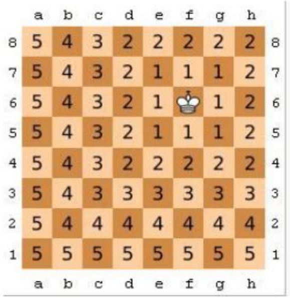
Manhattan o distancia City-Block

$$\sum_{i=1}^{m} |x_i - y_i|$$



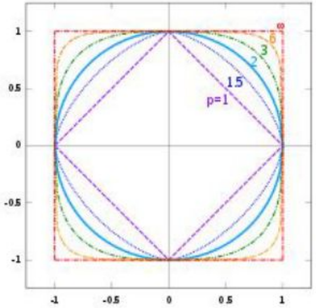
Chebyshev

$$\max_{i=1,\ldots,m} \{|x_i - y_i|\}$$



Minkowski *p*

$$\sqrt[p]{\sum_{i=1}^{m} |x_i - y_i|^p}$$



Ver más en: [sklearn.neighbors.DistanceMetric — scikit-learn 0.24.2 documentation]()

KNN MODELS

PROS

- It has almost no assumptions about the data.
- It's easy to update once it is deployed to production.

CONS

Assembler
Institute of Technology

KNN MODELS

PROS

- It has almost no assumptions about the data.
- It's easy to update once it is deployed to production.

CONS

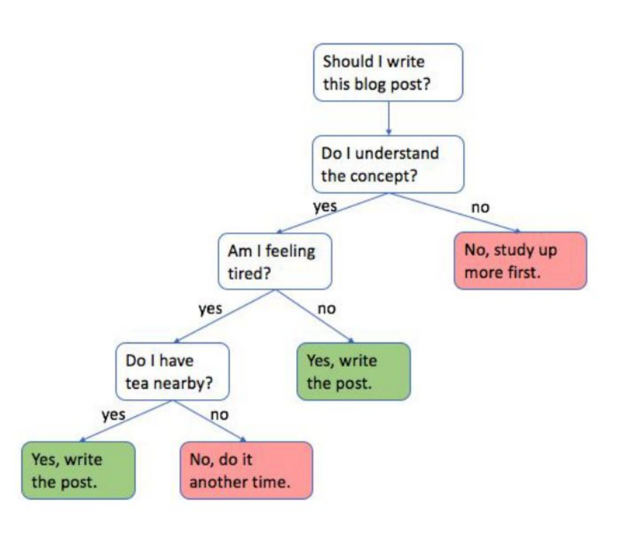
- Very sensible to outliers.
- We need to handle nas.
- It has a high computational cost:
  - Space: We need to store all the training data.
  - Time: It needs to calculate all the distances between the different data points.

Assembler
Institute of Technology

DECISION TREES

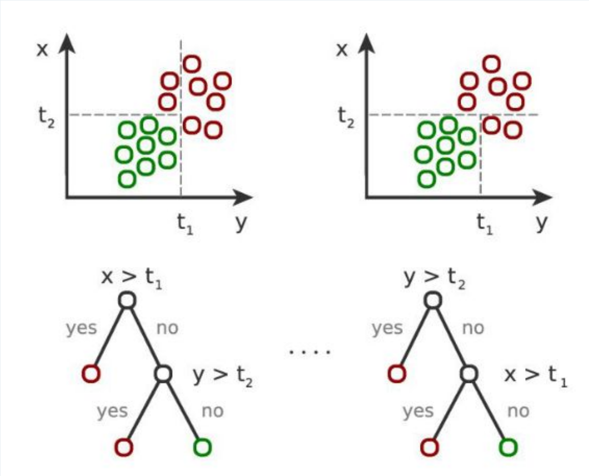
INTRODUCTION

- Decision trees are **non-parametric models** that can be used for both **classification and regression** problems.
- They are highly interpretable and flexible.

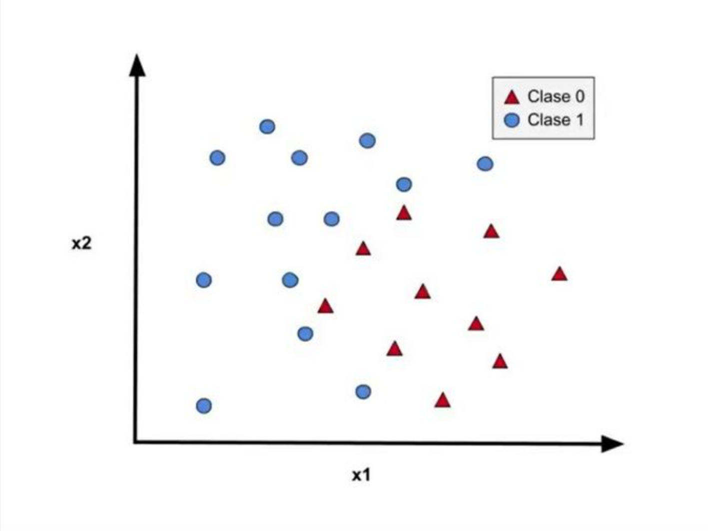


INTRODUCTION

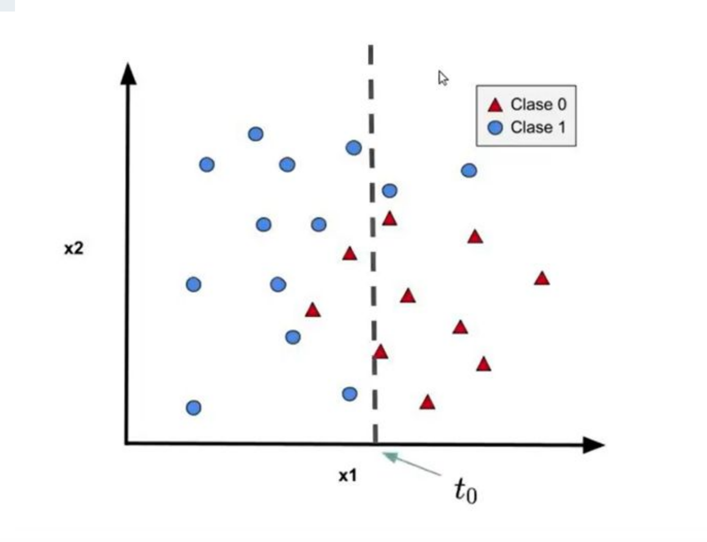
- Decision trees are **non-parametric models** that can be used for both **classification and regression** problems.
- They are highly interpretable and flexible.



INTRODUCTION

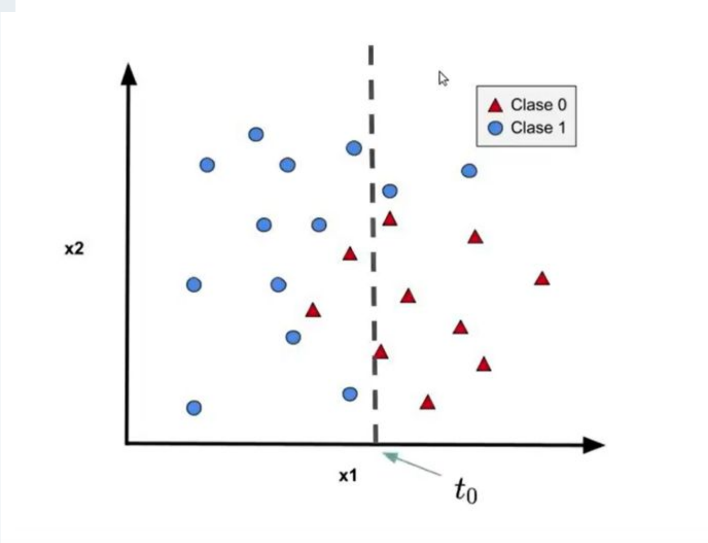


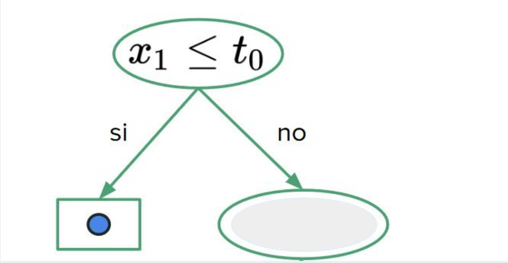
INTRODUCTION



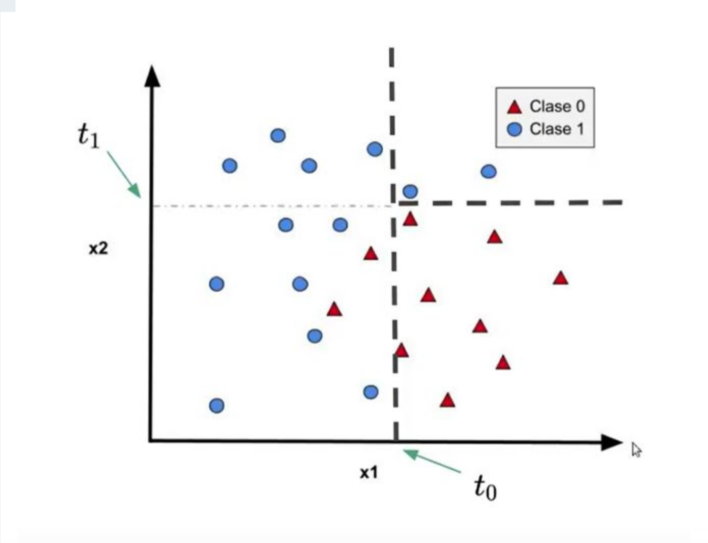
$$x_1 \leq t_0$$

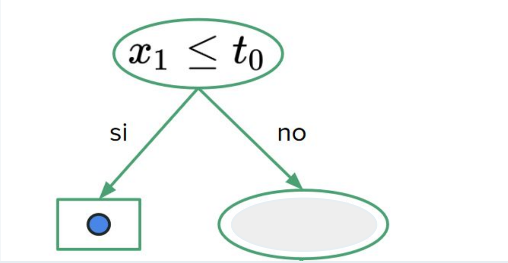
INTRODUCTION



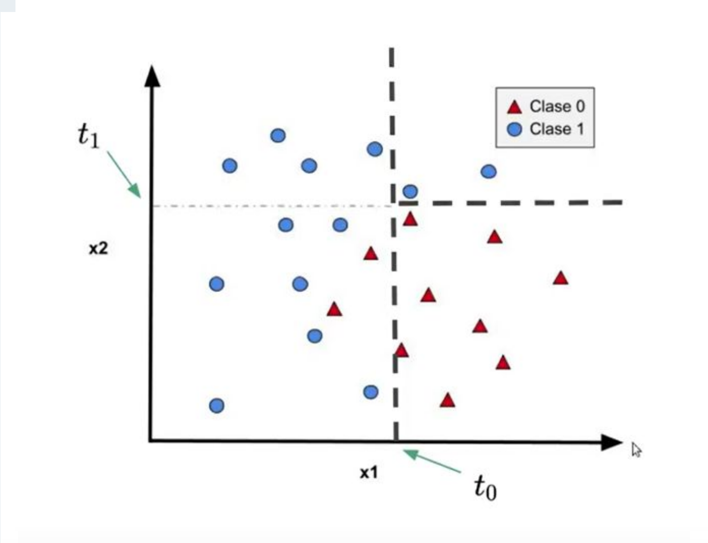


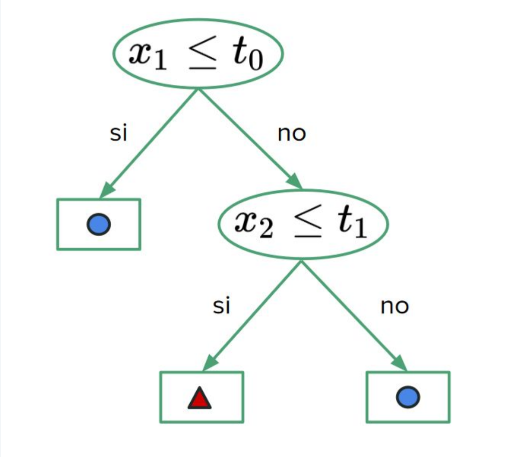
INTRODUCTION





INTRODUCTION





Assembler
Institute of Technology

GOOD OR BAD SPLIT?

- In classification problems, we are going to measure the **impurity** to decide if a split was good or bad.
- The most common impurity measures are:

Entropía

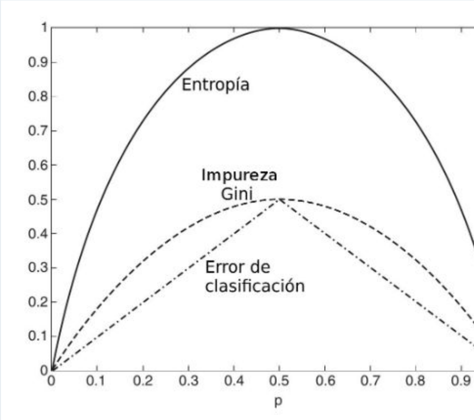
$$Entropia(S) = \sum_{c \in Clases(S)} -p_c \cdot \log_2(p_c)$$

$$\text{donde } p_c = \frac{|\{x \in S, clase(x) = c\}|}{|S|}$$

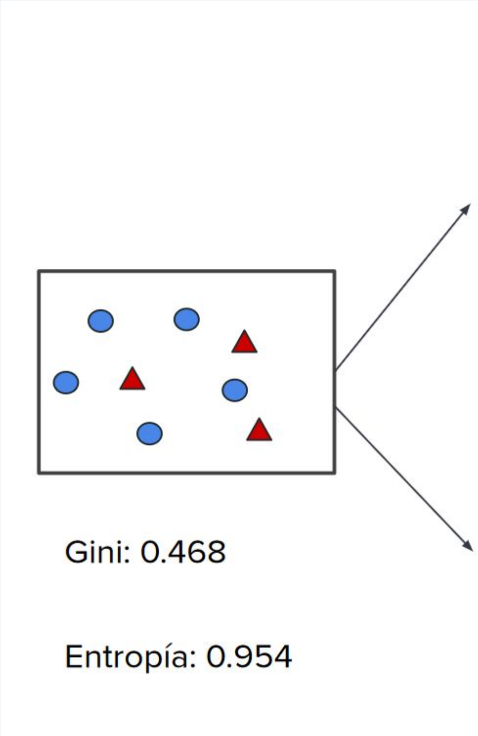
Índice de Gini

$$Gini(S) = \sum_{c \in Clases(S)} p_c(1 - p_c)$$

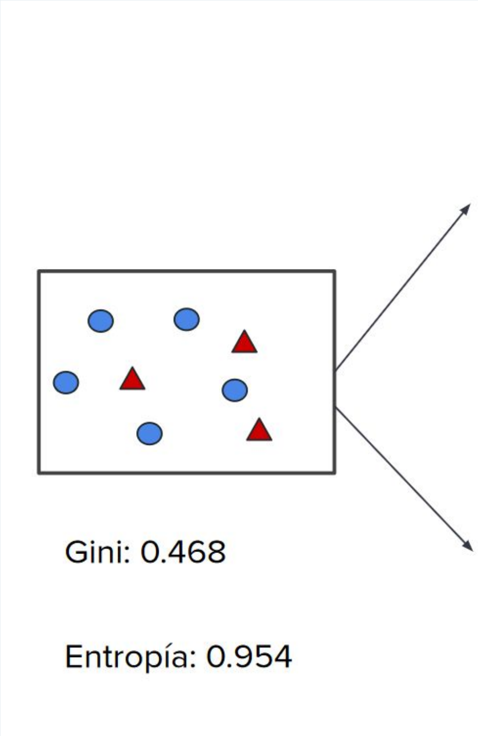
$$= 1 - \sum_{c \in Clases(S)} p_c^2$$

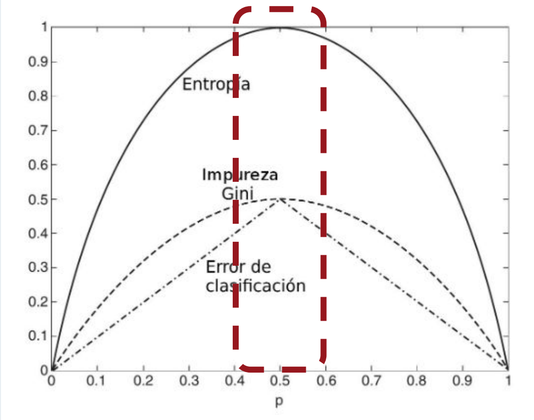


GOOD OR BAD SPLIT?

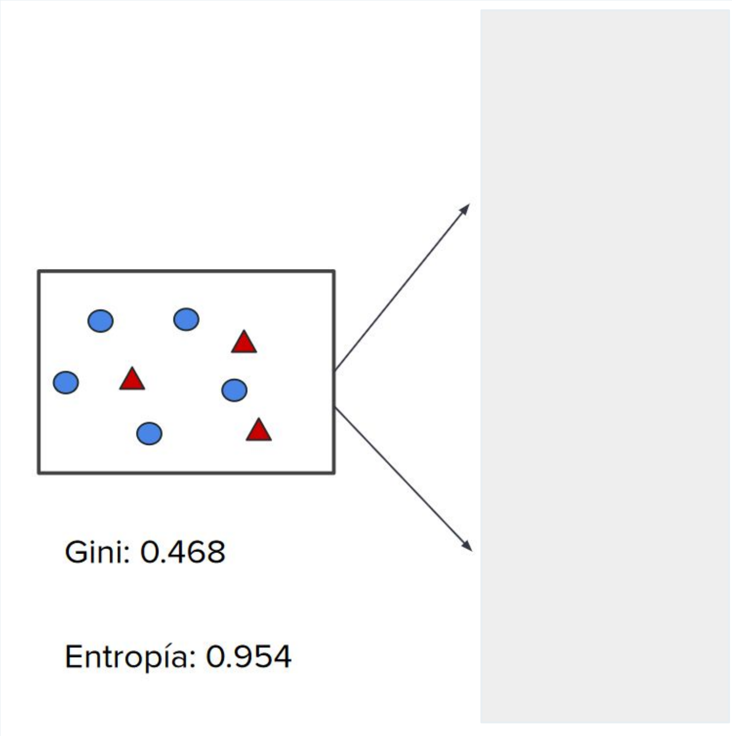


GOOD OR BAD SPLIT?

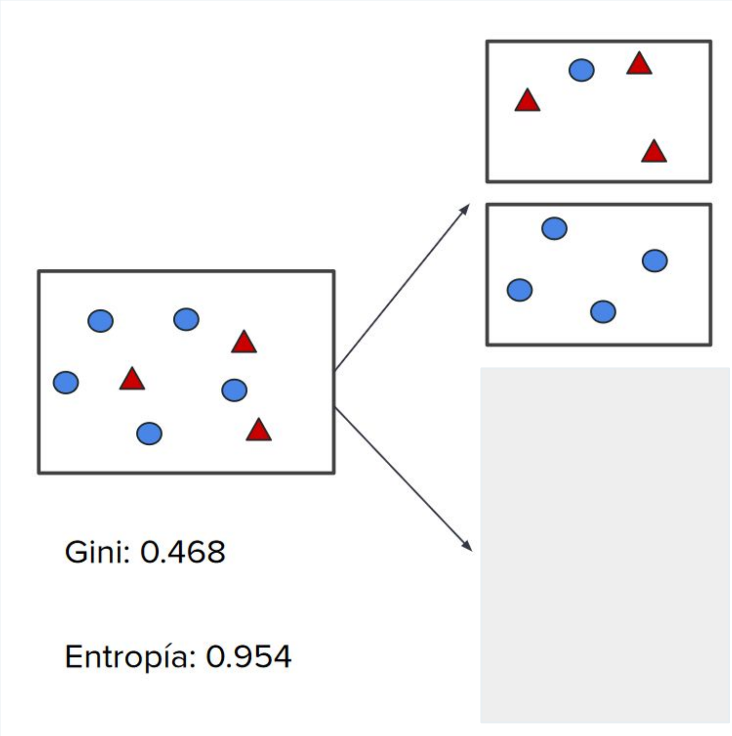




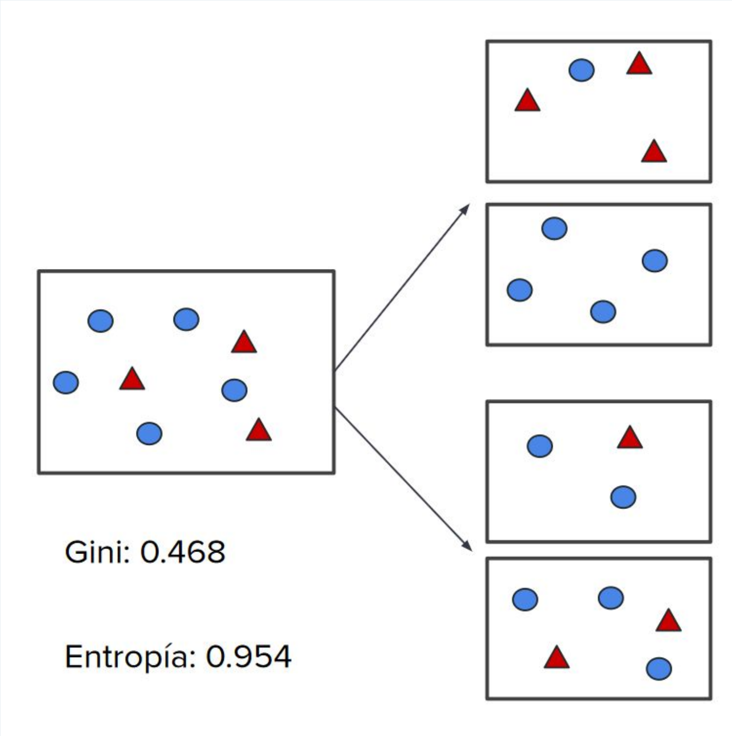
GOOD OR BAD SPLIT?



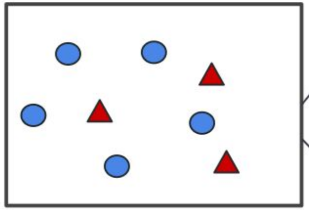
GOOD OR BAD SPLIT?



GOOD OR BAD SPLIT?

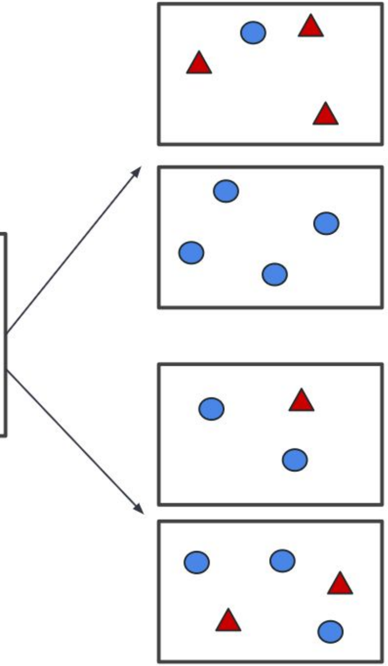


GOOD OR BAD SPLIT?



Gini: 0.468

Entropía: 0.954



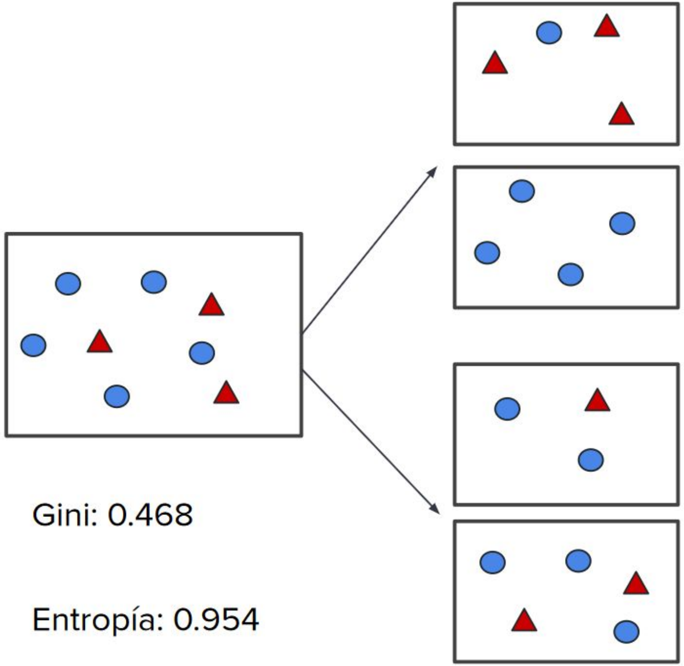
Gini:
$1-(\frac{1}{4})^2-(\frac{3}{4})^2= \frac{3}{8}$

Entropía:
$-(\frac{1}{4}).\log(\frac{1}{4})-(\frac{3}{4}).\log(\frac{3}{4})=$
$= 0.811$

Gini:
$1-(4/4)^2-(0/4)^2=0$

Entropía:
$-(4/4).\log(4/4)-(0/4).\log(0/4)=$
$= 0$

GOOD OR BAD SPLIT?



Gini:
$1-(1/4)^2-(3/4)^2 = 3/8$

Entropía:
$-(1/4).\log(1/4)-(3/4).\log(3/4)=$
$= 0.811$

Gini:
$1-(4/4)^2-(0/4)^2=0$

Entropía:
$-(4/4).\log(4/4)-(0/4).\log(0/4)=$
$= 0$

Gini:
$1-(2/3)^2-(1/3)^2= 4/9$

Entropía:
$-(2/3).\log(2/3)-(1/3).\log(1/3)=$
$= 0.918$

Gini:
$1-(3/5)^2-(2/5)^2=12/25$

Entropía:
$-(3/5).\log(3/5)-(2/5).\log(2/5)=$
$= 0.971$

GOOD OR BAD SPLIT?

- To decide if a split is better than another one, we want to **minimize the impurity**:

Entropía(nodo padre) - Promedio pesado(Entropía (nodos hijos))

GOOD OR BAD SPLIT?

- To decide if a split is better than another one, we want to **minimize the impurity**:

Entropía(nodo padre) - Promedio pesado(Entropía (nodos hijos))

- **Information gain:**
Let V be the parent node with N data, X be the feature and s the split using that feature (X<s)

$$IG(X_j, s) = Entropy(V) - \sum_{V_i} \frac{N_i}{N} Entropy(V_i)$$

$$V_i = \{X \in V | X_j \leq s\} \text{ o } \{X \in V | X_j > s\}$$

GOOD OR BAD SPLIT?

- To decide if a split is better than another one, we want to **minimize the impurity**:

  Entropía(nodo padre) - Promedio pesado(Entropía (nodos hijos))

- **Information gain:**
  Let V be the parent node with N data, X be the feature and s the split using that feature (X<s)

$$IG(X_j, s) = Entropy(V) - \sum_{V_i} \frac{N_i}{N} Entropy(V_i)$$

$$V_i = \{X \in V | X_j \leq s\} \text{ o } \{X \in V | X_j > s\}$$

THE HIGHER THE INFORMATION GAIN, THE BETTER THE ENTROPY REDUCTION

GOOD OR BAD SPLIT?

- To decide if a split is better than another one, we want to **minimize the impurity**:

  Entropía(nodo padre) - Promedio pesado(Entropía (nodos hijos))

- **Information gain:**
  Let V be the parent node with N data, X be the feature and s the split using that feature (X<s)

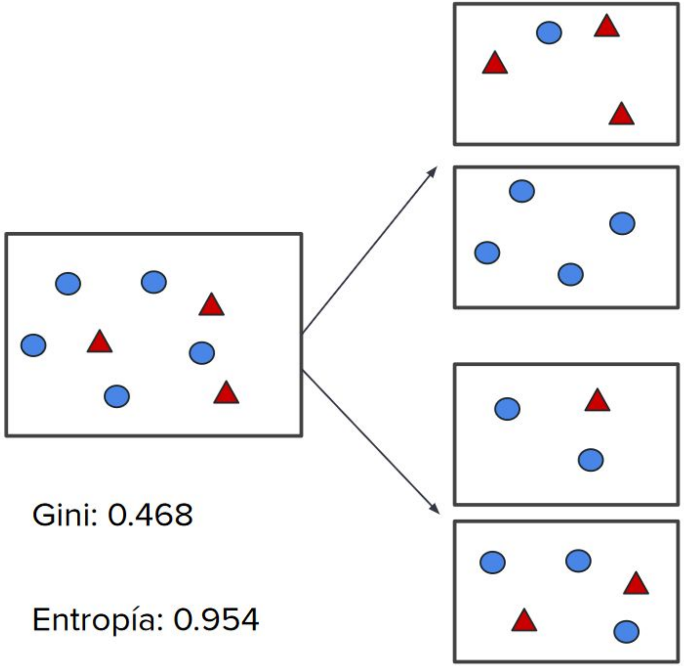
$$IG(X_j, s) = Entropy(V) - \sum_{V_i} \frac{N_i}{N} Entropy(V_i)$$

$$V_i = \{X \in V | X_j \leq s\} \text{ o } \{X \in V | X_j > s\}$$

THE HIGHER THE INFORMATION GAIN, THE BETTER THE ENTROPY REDUCTION

If we use Gini index instead of entropy, it is called Gini Gain.

GOOD OR BAD SPLIT?



Gini: 0.468

Entropía: 0.954

Gini:
$1-(\frac{1}{4})^2-(\frac{3}{4})^2= \frac{3}{8}$

Entropía:
$-(\frac{1}{4}).\log(\frac{1}{4})-(\frac{3}{4}).\log(\frac{3}{4})=$
$= 0.811$

Gini:
$1-(4/4)^2-(0/4)^2=0$

Entropía:
$-(4/4).\log(4/4)-(0/4).\log(0/4)=$
$= 0$

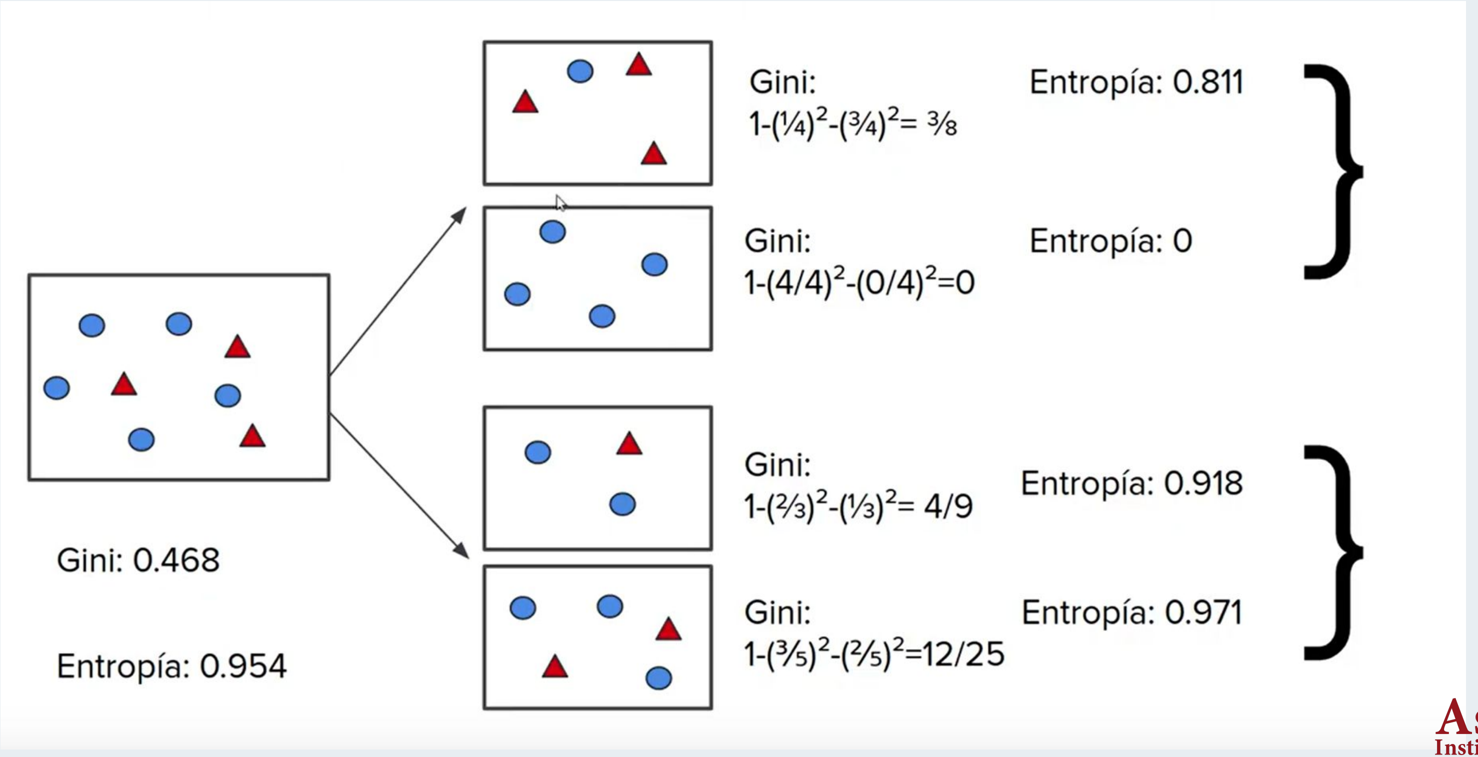
Gini:
$1-(\frac{2}{3})^2-(\frac{1}{3})^2= 4/9$

Entropía:
$-(\frac{2}{3}).\log(\frac{2}{3})-(\frac{1}{3}).\log(\frac{1}{3})=$
$= 0.918$

Gini:
$1-(\frac{3}{5})^2-(\frac{2}{5})^2=12/25$

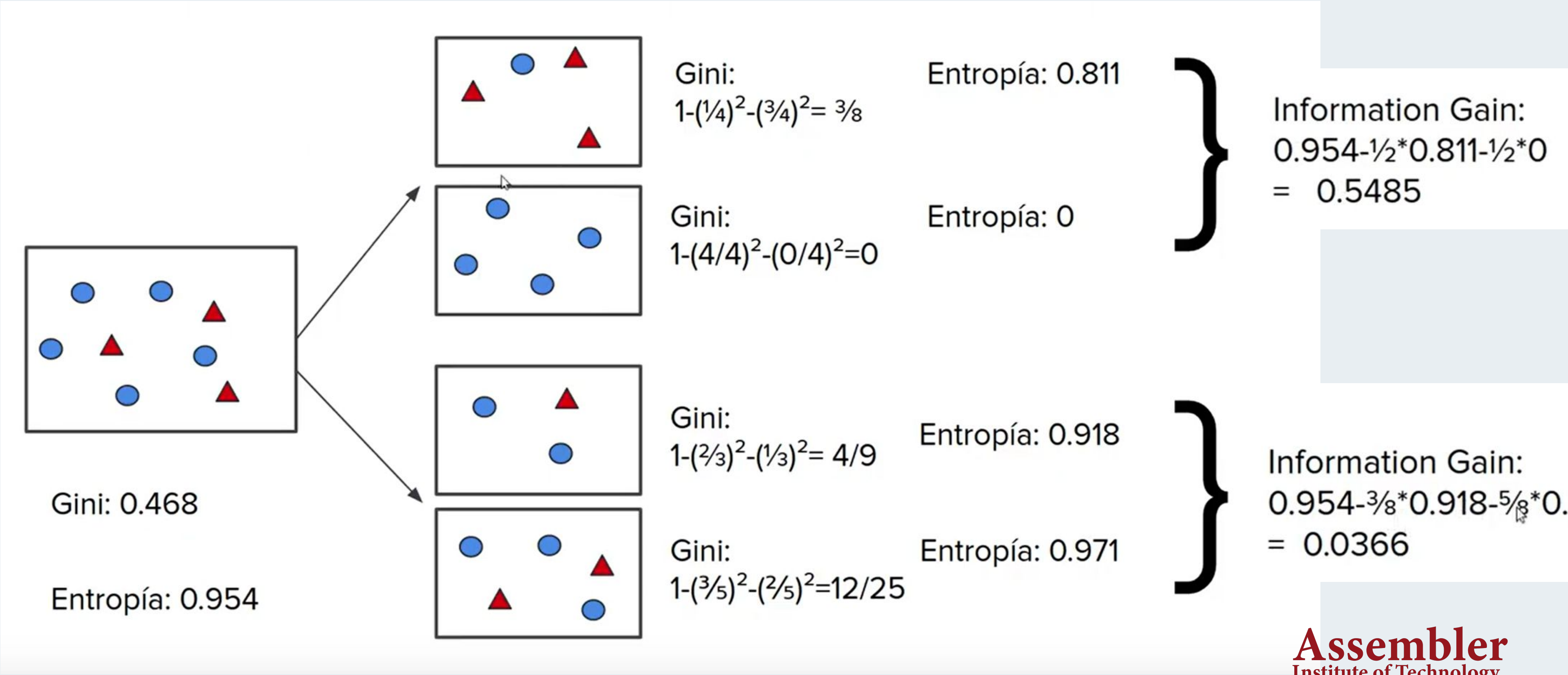
Entropía:
$-(\frac{3}{5}).\log(\frac{3}{5})-(\frac{2}{5}).\log(\frac{2}{5})=$
$= 0.971$

GOOD OR BAD SPLIT?



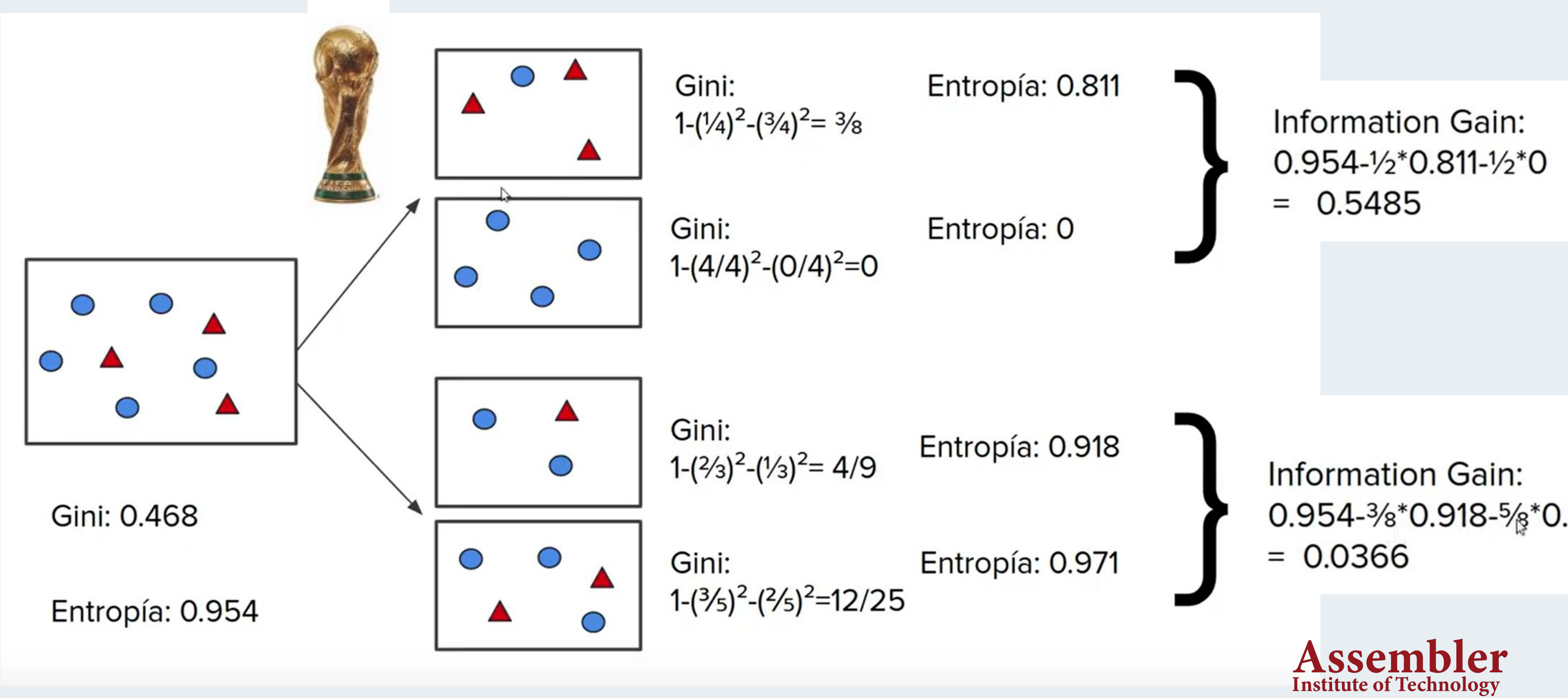
Assembler
Institute of Technology

GOOD OR BAD SPLIT?



DECISION TREES

GOOD OR BAD SPLIT?



TRAINING

- For each feature, the model will try different splits.
- The best split is selected according to the information gain value.
- It checks the stop constraints:
  - Maximum depth
  - Minimum amount of data to split
  - Minimum amount of data in a leaf to consider a split

TRAINING

- For each feature, the model will try different splits.
- The best split is selected according to the information gain value.
- It checks the stop constraints:
    - Maximum depth
    - Minimum amount of data to split
    - Minimum amount of data in a leaf to consider a split

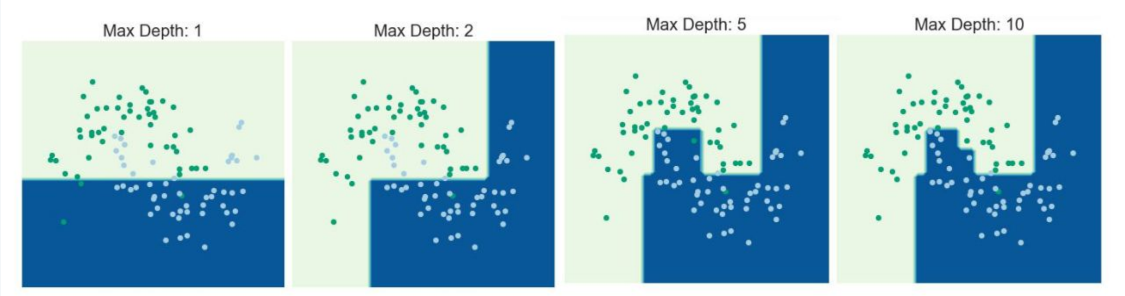
Without them, the tree keeps growi

until it finds a "perfect" model

TRAINING

- For each feature, the model will try different splits.
- The best split is selected according to the information gain value.
- It checks the stop constraints:
    - Maximum depth
    - Minimum amount of data to split
    - Minimum amount of data in a leaf to consider a split

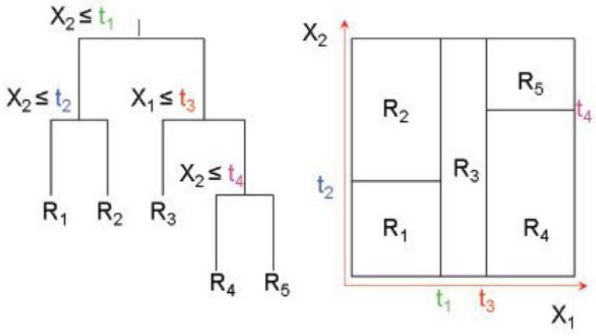
Without them, the tree keeps growi
until it finds a “perfect” model



REGRESSION

- We'll try to minimize the Mean Squared Error

$$ECM = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

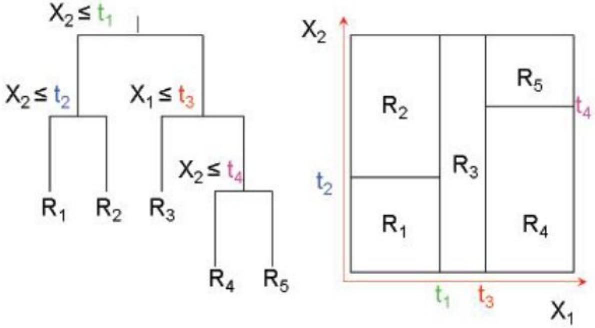


DECISION TREES

REGRESSION

- We'll try to minimize the Mean Squared Error

$$ECM = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$



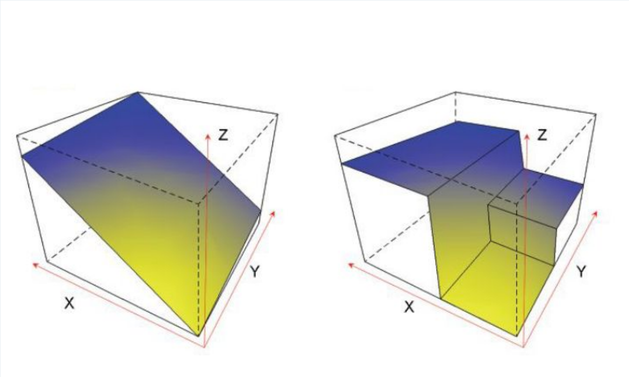
$$\sum_{i:\, x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\, x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

$$R_1(j, s) = \{X | X_j < s\} \text{ and } R_2(j, s) = \{X | X_j \geq s\}$$

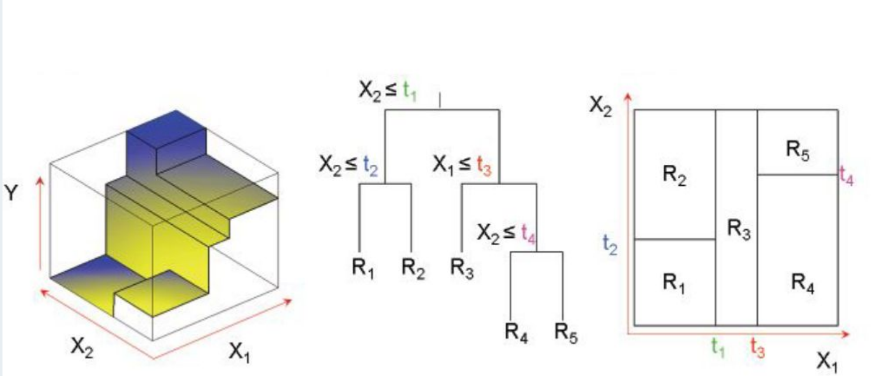
$$\Delta = ECM(\text{padre}) - \sum_{j \in \text{hijos}} \frac{N_j}{N} ECM(\text{hijo}_j)$$

Assembler
Institute of Technology

REGRESSION



REGRESSION


Y
X_2
X_1
$X_2 \leq t_1$
$X_2 \leq t_2$
$X_1 \leq t_3$
$X_2 \leq t_4$
R_1
R_2
R_3
R_4
R_5
X_2
R_2
R_5
t_4
R_3
t_2
R_1
R_4
t_1
t_3
X_1

DECISION TREES

PROS

- Easy to interpret
- Fast training
- They can handle continuous and discrete data and missing values

CONS

- **They tend to overfit**
- Their performance is usually worse than the one for other classical models