

Árboles de decisión



Qué es un clasificador de árbol de decisión

- Árbol de decisión
- Ejemplo
- Ciclo de un árbol de decisión

Algoritmos

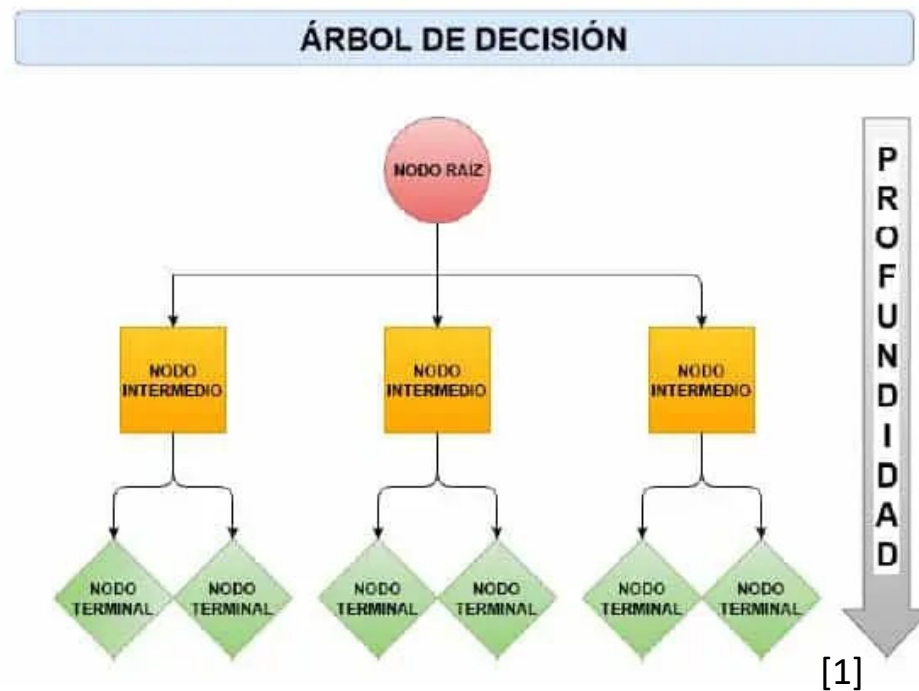
Construcción de un árbol de decisión

Que es?



Árboles de desición

Es un conjunto de
nodos conectado
entre sí



Que es?



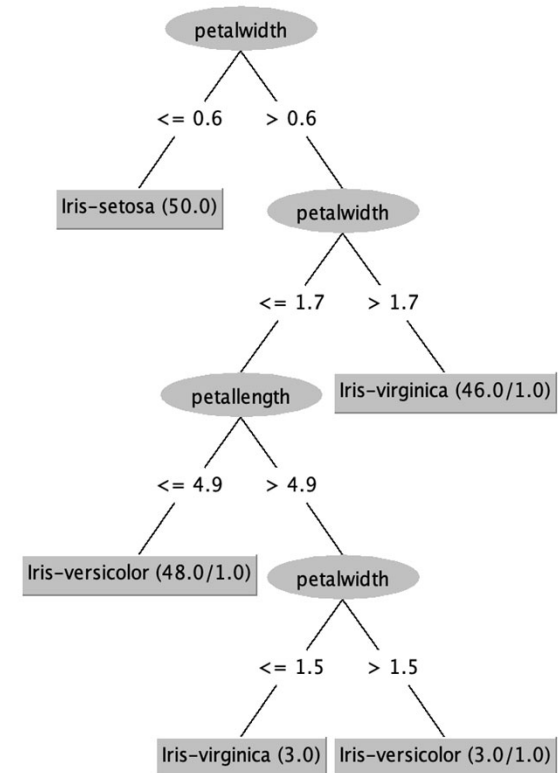
Árboles de desición

En teoría de grafos, el término árbol se utiliza para referirse a un grafo para el que se cumple que, dos vértices cualesquiera, están conectados por exactamente un camino

Nodo raíz

Nodo interno

Nodo hoja o terminal



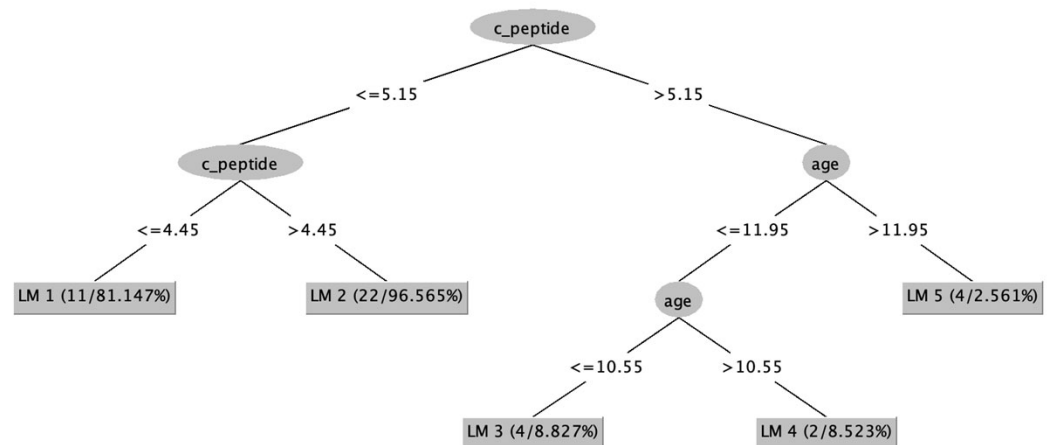
Ejemplo



Árboles de decisión

En minería de datos, los árboles se utilizan como herramienta de clasificación. Para ello, se tilda el valor de los atributos conocidos del objeto para ir descendiendo por el árbol.

Cada nodo del árbol tiene una condición sobre dichos atributos conocidos, que determina la rama por la que descender, hasta llegar a un nodo hoja, que indica la clase dentro de la cual ha sido clasificado el objeto



Ciclo de un árbol

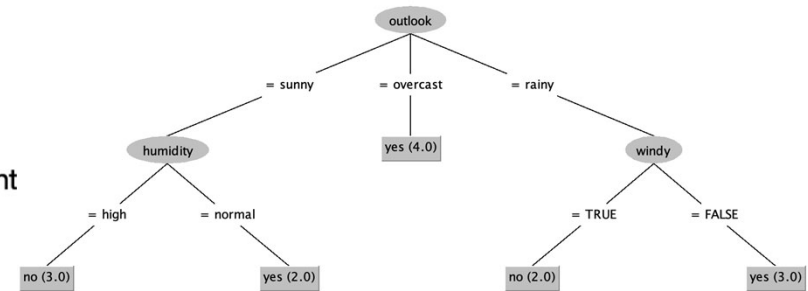


Árboles de decisión

El ciclo de un árbol está integrado por dos etapas. La primera es el aprendizaje
El proceso de convertir el conjunto de datos en un modelo predictivo se llama inducción



- 1 ☐ checking_status
- 2 ☐ duration
- 3 ☐ credit_history
- 4 ☐ purpose
- 5 ☐ credit_amount
- 6 ☐ savings_status
- 7 ☐ employment
- 8 ☐ installment_commitment
- 9 ☐ personal_status
- 10 ☐ other_parties
- 11 ☐ residence_since
- 12 ☐ property_magnitude
- 13 ☐ age
- 14 ☐ other_payment_plans
- 15 ☐ housing
- 16 ☐ existing_credits



Ciclo de un árbol



Árboles de decisión

La segunda es la clasificación: en este caso se va a predecir la clase, y es un un modelo deductivo



Algunos Algoritmos son:

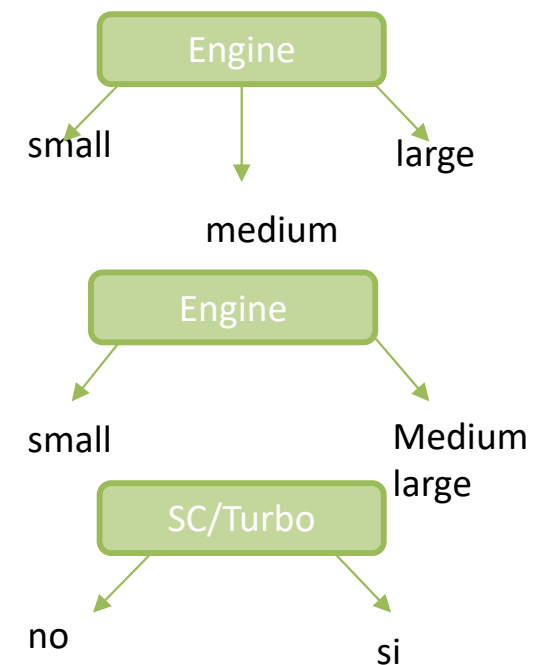
- Hunt's
- Cart
- ID3, C4.5
- SLIQ, SPRINT

Construcción de un árbol de decisión



Árboles de desición

Engine	SC/Turbo	Weiht	Fuel Eco	Fast
small	no	average	good	no
small	no	light	average	no
small	yes	average	bad	yes
medium	no	heavy	bad	yes
large	no	average	bad	yes
medium	no	light	bad	no
large	yes	heavy	bad	no
large	no	heavy	bad	no
medium	yes	light	bad	yes
large	no	average	bad	yes
small	no	light	bad	no
small	no	average	bad	no
medium	no	heavy	good	no
small	yes	average	average	no
medium	no	heavy	bad	no





¿Cuál es el
atributo raíz?

$$I(p,n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

Ec. 1

$$E(A) = \sum_{i=1}^v \frac{p_i + n}{p + n} I(p_i, n_i)$$

Ec. 2



Pasos para construir un árbol

- 1.- Se asignan todos los elementos del conjunto de entrenamiento a la raíz del árbol
- 2.- Se realizan divisiones del árbol de clasificación, atendiendo a una determinada heurística
- 3.- Se repite el paso 2 hasta llegar a los nodos hoja
- 4.- Por último, se puede realizar una poda del árbol para eliminar ramas que representan ruido



Normalmente, la heurística utilizada a la hora de construir el árbol (paso 2), consiste en seleccionar en cada nodo, el atributo que proporciona la mayor **ganancia de información**.

Para explicar dicho concepto, supongas que se tiene dos clases, **P** y **N**, y un conjunto de ejemplos **S** que contiene **p** elementos de la clase **P** y **n** elementos de la clase **N**. En este caso, la cantidad de información que se necesita para decidir si un objeto cualquiera de **S** pertenece a **P** o a **N** se define según indica la ecuación (1)

$$I(p,n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$



Supóngase también que, en un determinado nodo del árbol, se utiliza un atributo **A**, y que el conjunto **S** se divide en subconjunto

s $\{S_1, S_2, \dots, S_3\}$. En este caso si S_i contiene P_i ejemplo de P y n_i ejemplo de N la **entropía**, o la información necesaria para clasificar objetos en cada uno de los subárboles S_i se calcula utilizando la ecuación (2)

$$E(A) = \sum \frac{p_i + n_i}{p + n} I(p_i, n_i)$$



Finalmente, la ganancia de información en el caso de utilizar el atributo A viene dada por la ecuación (3)

Este valor mide la capacidad discriminatoria dl atributo en cuestión considerando las diferentes clases del problema

$$Ganancia(A) = I(p, n) - E(A) \quad \text{Ec. 3}$$

Construcción de un árbol de decisión



Árboles de decisión

Identificador	Edad	Estudia	Ratio de crédito	Compra
1	alta	no	aceptable	no
2	alta	no	excelente	no
3	alta	no	aceptable	si
4	media	no	aceptable	si
5	baja	si	aceptable	si
6	baja	si	excelente	no
7	baja	si	excelente	si
8	baja	no	aceptable	no
9	alta	si	aceptable	si
10	media	si	aceptable	si
11	media	si	excelente	si
12	media	no	excelente	si
13	alta	si	aceptable	si
14	baja	no	excelente	no



Se desea clasificar nuevos individuos en función del atributo *Compra*, que determina si un individuo va a comprar o no un producto

Una vez construido el árbol de decisión se tendrá un modelo que permitirá clasificar nuevos individuos para lo que el valor del atributo *Compra* es desconocido. dicho valor se puede predecir gracias al árbol de decisión, a partir del valor del resto de atributos .
(*Edad, Estudia y Ratio de Crédito*)



1.- En este problema existen, por tanto, dos clases. La clase $P(Compra=Si)$ y la clase $N(Compra = no)$

En un primer momento, al crear el árbol de decisión, todos los elementos se asignan al nodo raíz del árbol. A continuación, hay que dividir ese nodo en función del atributo que proporcione mayor **ganancia de información**

Si se calcula la ganancia para cada uno de los atributos, se puede ver que el atributo *Edad* proporciona la mayor ganancia de información y, por lo tanto, es el que se utiliza para dividir el nodo raíz



1.- La clase $P(Compra=Si) = 9$ y la clase $N(Compra = no) = 5$
Utiliza la ecuación 1 para obtener $I(p, n)$

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$



Solución

$$I(p,n)$$

$$= I(9,5)$$

$$= -9/(9+5) \log_2 5/(9+5) - 5/(9+5) \log_2 9/(9+5)$$

$$= -(9/14) \log_2 (9/14) - (5/14) \log_2 (5/14) = 0,94$$



2.- A continuación, se calcula la entropía para el atributo *Edad*

En este caso, dicho atributo toma tres posibles valores (Baja, Media y Alta), por lo que $v=3$

De los individuos con edad Baja, 2 sí compran el producto ($p_1=2$), mientras que 3 no lo compran ($n_1=3$)

De los individuos con edad Media, 4 sí compran el producto ($p_2=4$), mientras que 0 no lo compran ($n_2=0$)

Por último, de los individuos con edad Alta, 3 sí compran el producto ($p_3=3$), mientras que 2 no lo compran ($n_3=2$)



Utilizando la ecuación 2, calcular la entropía

$$E(A) = \sum \frac{p_i + n_i}{p + n} I(p_i, n_i)$$



Solución

$$\begin{aligned} &= 5/14 I(2,3) + 4/14 I(4,0) + 5/14 I(3,2) \\ &= 5/14 (0,971) + 4/14 (0) + 5/14 (0,971) \\ &= 69 \end{aligned}$$

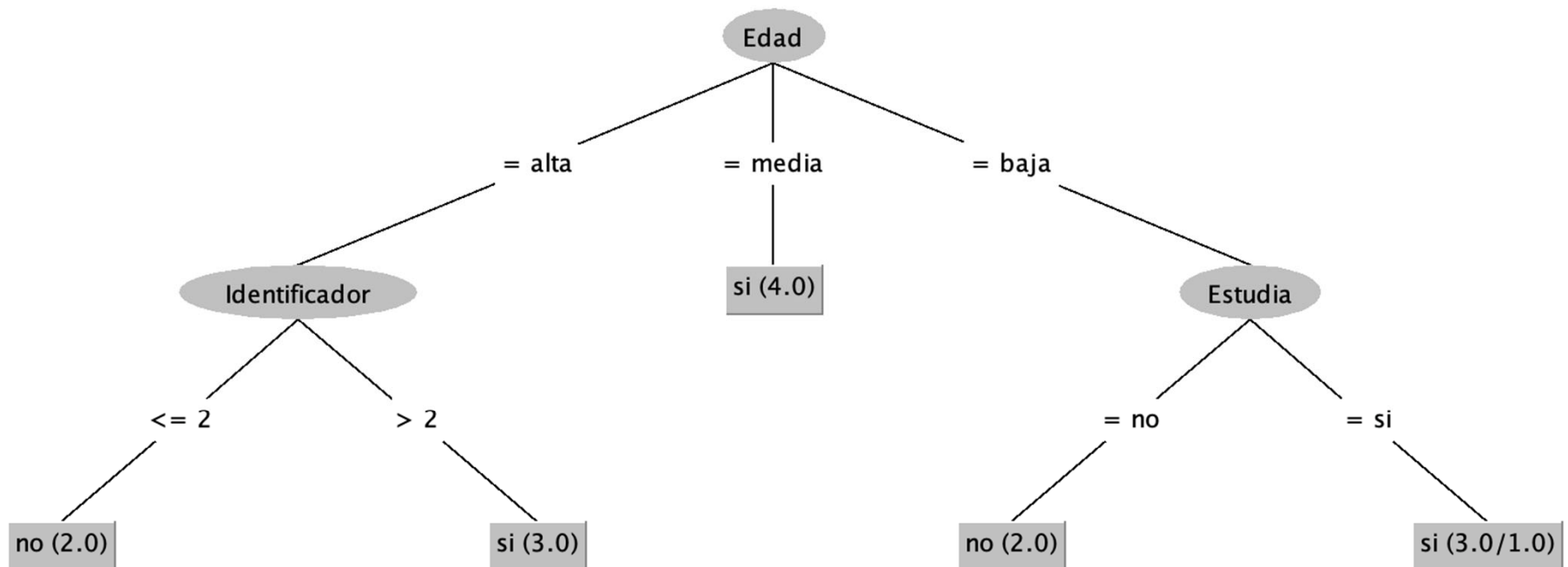
$$E(\text{Edad}) = \frac{p_1 + n_1}{p + n} I(p_1, n_1) + \frac{p_2 + n_2}{p + n} I(p_2, n_2) + \frac{p_3 + n_3}{p + n} I(p_3, n_3)$$



3.- Finalmente, la ganancia de información para el atributo *Edad* será:

$$Ganancia(A) = I(p, n) - E(A)$$

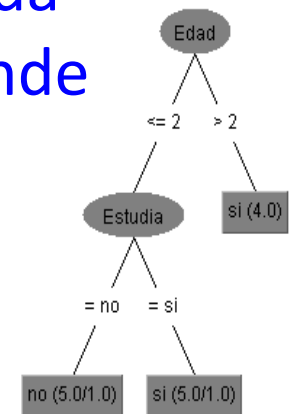
$$Ganancia (Edad) = I(9,5) - E(Edad) = 0,94 - 0,69 = 0,25$$





Si se calcula la ganancia de información para el resto de los atributos, se confirma que, para todos ellos, es menor que para el atributo Edad, por lo que dicho atributo es el que marca la primera división del árbol

Esto se refleja en la figura 1, donde se aprecia que el atributo Edad está presente en la raíz del árbol. Cada una de las ramas que parten de la raíz se corresponde con cada posible valor de dicho atributo





Los árboles de decisión también se pueden representar como un conjunto de reglas de decisión, las reglas de decisión determinada la clase final del objeto a clasificar en función del valor del resto de atributos de dicho objeto

J48 pruned tree

```
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   |   petalwidth > 1.7: Iris-virginica (46.0/1.0)
```

Number of Leaves : 5

Size of the tree : 9



Instalar los paquetes necesarios si no están instalados

```
if (!require(rpart)) install.packages("rpart")
```

```
if (!require(rpart.plot)) install.packages("rpart.plot")
```

Cargar los paquetes

```
library(rpart)
```

```
library(rpart.plot)
```

```
library(caret)
```

```
library(Momocs)
```

Descargar y cargar el conjunto de datos wine

```
wine_url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data"
```

```
wine <- read.csv(wine_url, header = FALSE)
```



Asignar nombres a las columnas

```
colnames(wine) <- c("Clase", "Alcohol",  
"Ácido_málico", "Cenizas",  
"Alcalinidad_de_las_cenizas", "Magnesio",  
"Fenoles_totales", "Flavanoides",  
"Fenoles_no_flavonoides", "Proantocianinas",  
"Intensidad_del_color", "Tonalidad",  
"OD280_OD315", "Prolina")
```



Resumen del modelo

```
summary(modelo_arbol)
```

Visualizar el árbol de decisión

```
rpart.plot(modelo_arbol, main = "Árbol de Clasificación para el Conjunto de Datos Wine")
```

Realizar predicciones

```
#prediccion_1 <- predict(modelo_arbol, newdata = wine_prueba, type = "class")  
predicciones <- predict(modelo_arbol, wine_prueba, type = "class")
```

Crear la matriz de confusión

```
matriz_confusion <- table(Predicted = predicciones, Actual = wine$Clase)  
#confusionMatrix(prediccion_1, wine_prueba[["Clase"]])
```



Mostrar la matriz de confusión

```
print(matriz_confusion)
```

Calcular la precisión del modelo

```
precision <- sum(diag(matriz_confusion)) / sum(matriz_confusion)  
print(paste("Precisión del modelo:", round(precision, 4)))
```

Evaluación del modelo

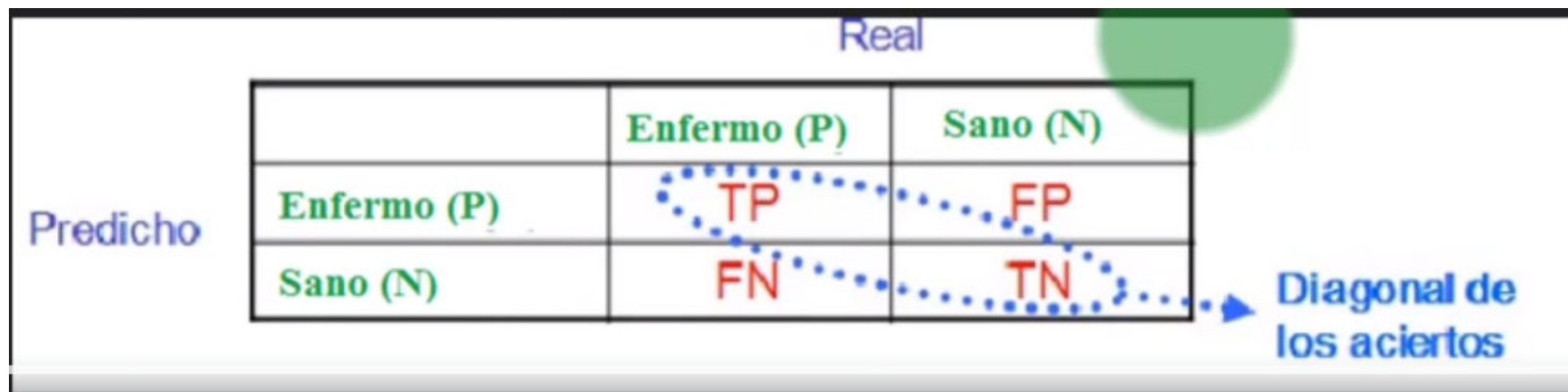


Matriz de confusión: Es una tabla que permite visualizar la ejecución de un algoritmo de clasificación (aprendizaje supervisado)

Es una matriz de un clasificador de dos o más clases

La matriz es $n \times n$, en donde n es el número de clases

Cada columna representa los casos que el algoritmo predijo, mientras que las filas representan los casos en una clase real



		Real	
		Enfermo (P)	Sano (N)
Predicho	Enfermo (P)	TP	FP
	Sano (N)	FN	TN

Diagonal de los aciertos

Evaluación del modelo



True Positive: son los casos que pertenecen a la clase y el clasificador los definió en esa clase

Falsos negativos: son los casos que sí pertenecen a la clase, pero el clasificador no los definió en esa clase

Falsos positivos: son los casos que no pertenecen a la clase, pero el clasificador los definió en esa clase

Verdaderos negativos: son los casos que no pertenecen a la clase y el clasificador definió que no pertenecen a esa clase



Accuracy (Exactitud): Es la proporción del número total de predicciones que son correctas, se determina utilizando la siguiente ecuación

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Error= 100 - Exactitud

Cálculo de la exactitud

Predicha	Real (Actual)			
	Cancer = Si	Cancer = No	Total	
	Cancer = Si	90 TP	210 FP	300
	Cancer = No	140 FN	9560 TN	9700
	Total	230	9770	10000

Exactitud = $(90 + 9560) / 10000 = 96.5\%$

Error = $100 - 96.5 = 3.5$

Evaluación del modelo



Sensibilidad: indica la capacidad del clasificador para dar como casos positivos los que realmente es positivo, en este ejemplo la proporción de enfermos correctamente clasificados

La sensibilidad caracteriza la capacidad de la prueba para detectar la enfermedad en sujetos enfermos

Sensibilidad = $\frac{TP}{TP + FN}$ - $\frac{\text{Número de resultados de pruebas positivos verdaderos}}{\text{todos los pacientes con enfermedad}}$		
Predicho	Real	
	Enfermo (P)	Sano (N)
	Enfermo (P)	FP
	Sano (N)	TN

Evaluación del modelo



Especificidad: Indica la capacidad del clasificador para dar como negativos los casos realmente negativos (casos sanos), para este ejemplo, proporción de sanos que realmente están sanos

La especificidad caracteriza la capacidad de la prueba para detectar la ausencia de la enfermedad en sujetos sanos

$$\text{Especificidad} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{Número de resultados de pruebas negativos verdaderos}}{\text{todos los pacientes sin enfermedad}}$$

		Real	
Predicho		Enfermo (P)	Sano (N)
	Enfermo (P)	TP	FP
	Sano (N)	FN	TN

Cálculo de la sensibilidad y especificidad

Real (Actual)				
Predicha	Cancer = Si	Cancer = No	Total	
	Cancer = Si	90 TP	210 FP	300
	Cancer = No	140 FN	9560 TN	9700
	Total	230	9770	10000

Sensibilidad = $90 / (90 + 140) = 39.13\%$

Especificidad = $9560 / (210 + 9560) = 97.84\%$



Cálculos cuando la matriz no es cuadrada

C0	C1	C2
68	0	0
1	63	1
0	1	76

C0) Maduros

C1) Pintón

C2) Verde



Cálculos cuando la matriz no es cuadrada

Clases	TP	TN	FP	FN
Maduro	68	141	1	0
Pintón	63	144	1	2
Verde	76	132	1	1

TP: True Positives Es la diagonal

TN: True Negatives Se ubica en el verdadero positivo de la clase, se omite toda la fila y toda la columna de la clase y se suman los valores que están afuera

FP: False Positives: De la clase maduro, se suma la columna excepto el verdadero positivo

FN: False Negatives: Inverso a como se obtiene los fp, se suma la fila excepto el verdadero positivo

Evaluación del modelo



Clases	TP	TN	FP	FN
Maduro	68	141	1	0
Pintón	63	144	1	2
Verde	76	132	1	1

$$TP = 68 + 63 + 76$$

$$TN = 63 + 1 + 1 + 76; 68 + 0 + 0 + 76; 68 + 0 + 1 + 63$$

$$FP = 1 + 0; 0 + 1; 0 + 1$$

$$FN = 0 + 0; 1 + 1; 0 + 1$$



Calcular indicadores

Sensibilidad: $(TP/TP+FN) * 100$

Especificidad: $(TN/TN+FP) * 100$

Exactitud: $(TP+TN)/(TP+FN+FP+TN)*100$

Precisión: $TP/(TP+FP) * 100$

Precisión negativos: $TN/TN+FN * 100$

Precisión General: $\sum TP/TotalMuestreo * 100$



Construir un árbol con el data(iris)

Cargas el conjunto de datos
`data(iris)`

Observar el conjunto de datos
`view(iris)`

Determinar la variable a clasificar
`iris$Species`



[1] <https://www.maximaformacion.es/blog-dat/que-son-los-arboles-de-decision-y-para-que-sirven/>

[2] <https://www.tecnologias-informacion.com/arboles.html>

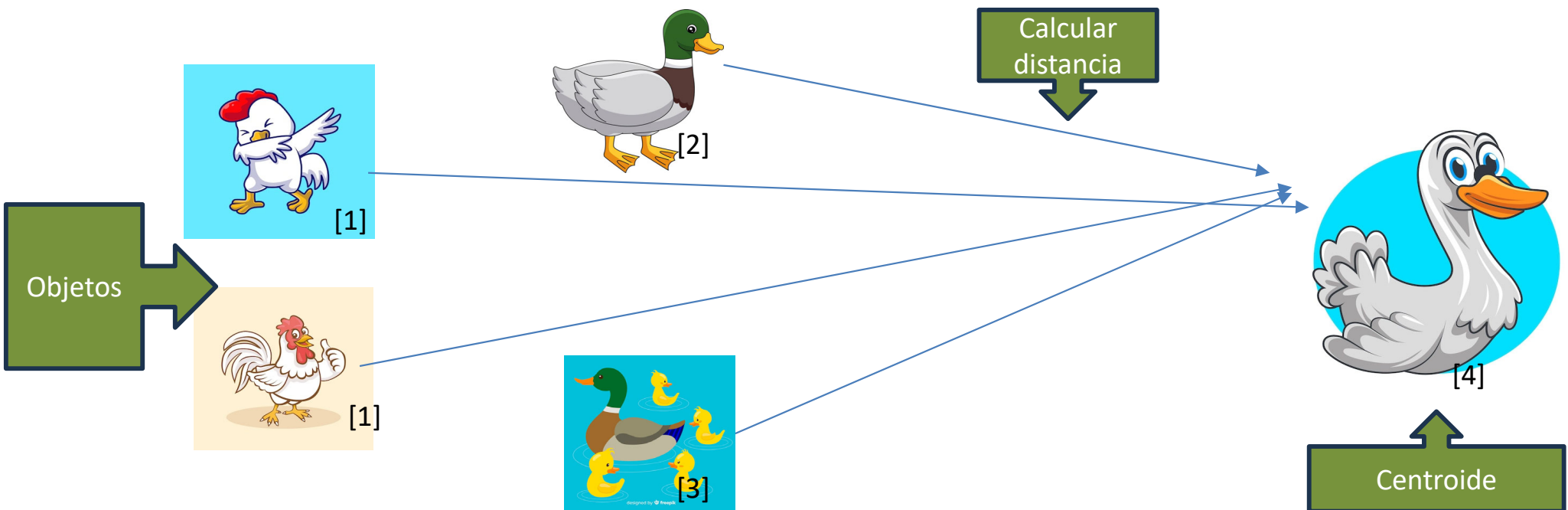
[7]

Clousters

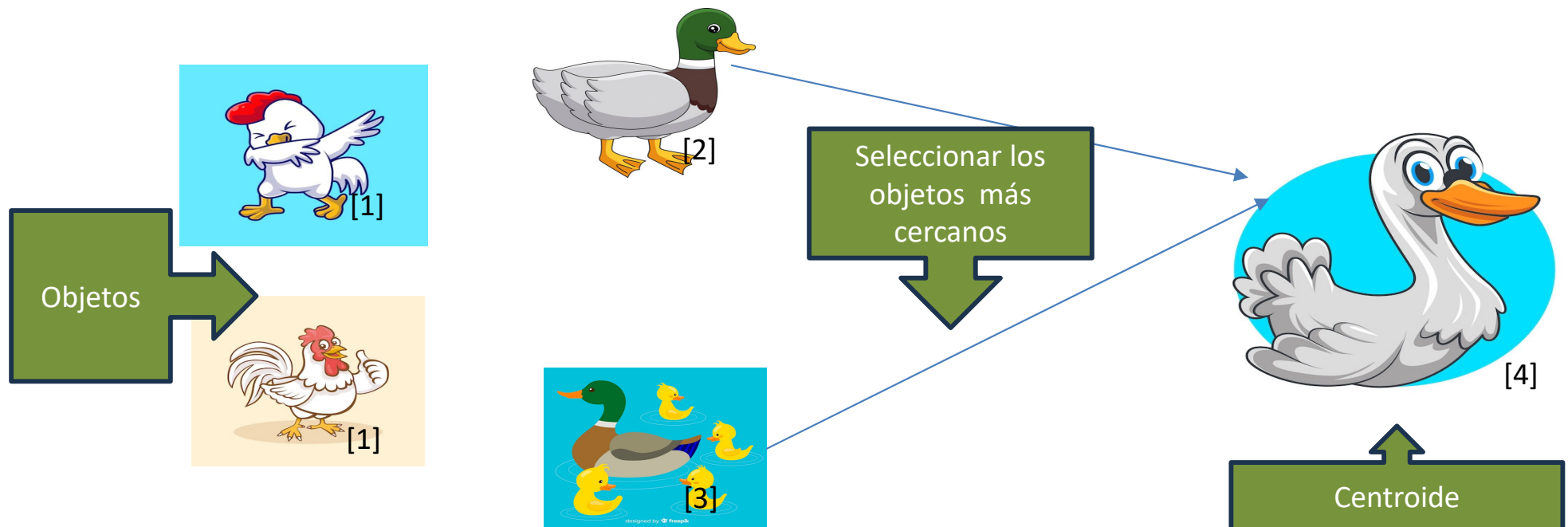


- Qué es kmean?
- Ejemplo

KMEANS



KMEANS





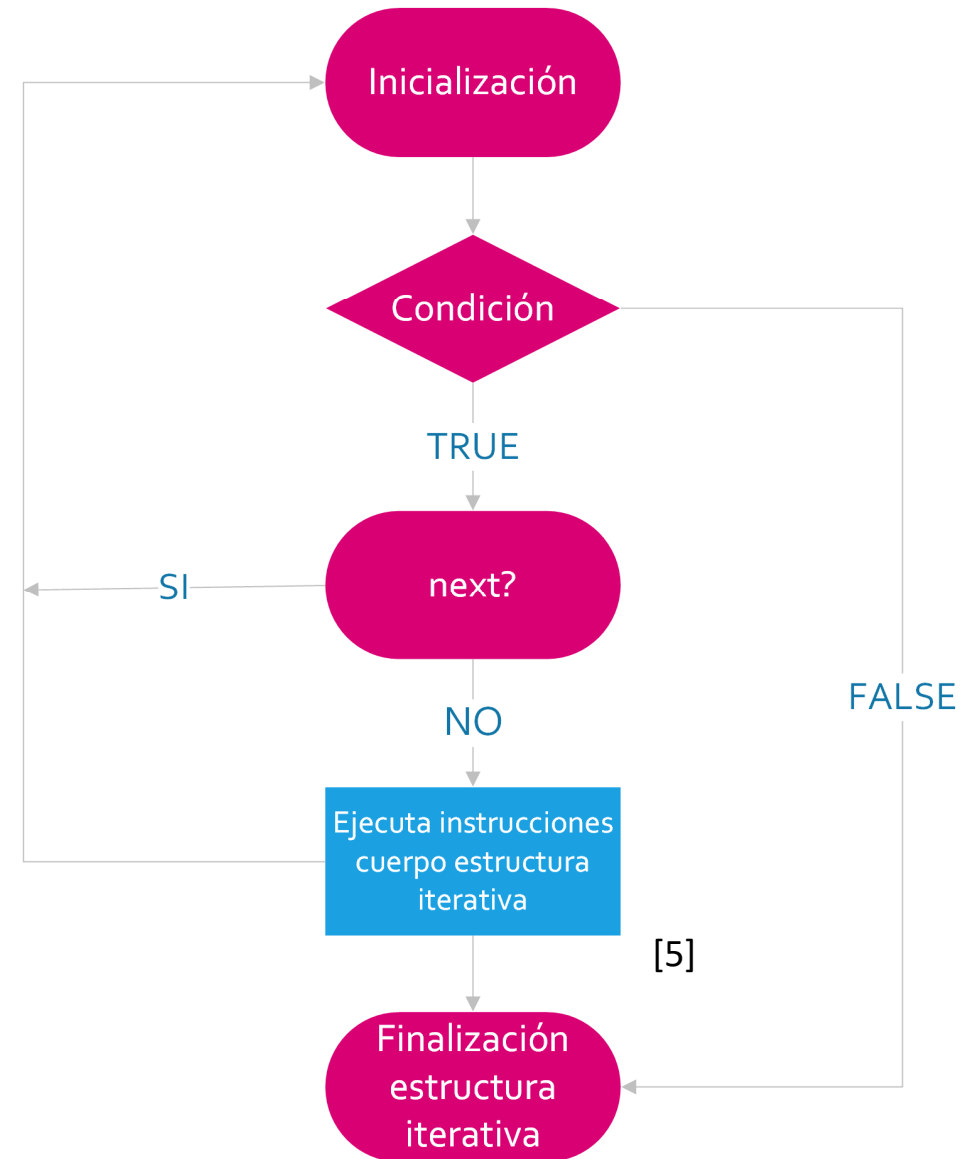
K-means es un algoritmo de agrupamiento no supervisado que se utiliza para dividir un conjunto de datos en k grupos (o clusters) distintos, donde cada observación pertenece al grupo con el valor medio más cercano

La idea es que este algoritmo trata de encontrar la mejor partición posible del conjunto de objetos para un número dado de clases k

KMEANS



Kmeans no es un algoritmo de entrenamiento, sino iterativo, en donde cada iteración, reduce la varianza interna de la clase





Proceso de K-means

El proceso de ajuste del algoritmo K-means puede ser descrito en los siguientes pasos:

1.Inicialización: Seleccionar k centroides iniciales, que pueden ser seleccionados aleatoriamente o usando algún método de inicialización específica como K-means++

2.Asignación de Clusters: Asignar cada punto de datos al centroide más cercano

3.Actualización de Centroides: Calcular nuevos centroides como el promedio de los puntos asignados a cada cluster

4.Repetición: Repetir los pasos de asignación y actualización hasta que los centroides no cambien significativamente o se alcance un número máximo de iteraciones



Ejemplo:

Supóngase un sistema reconocedor de formas geométricas de dos dimensiones en un entorno médico (tumores presentes en una radiografía)

Es posible que uno de los requisitos de dicho sistema sea la identificación de objetos similares entre sí, con el objetivo de conocer las diferentes tipologías de objetos que se pueden dar en el entorno indicado. Para ello, se cuenta con información de los objetos identificados, tal y como se muestra en la tabla 1

Como se aprecia en dicha tabla, cada objeto está caracterizado por la Longitud y Altura (por simplicidad, se han obviado los detalles acerca de las unidades de medida)

Identificador	Longitud	Altura
1	1	2
2	1	1
3	2	1
4	2	2
5	100	74
6	101	75
7	98	76
8	95	78
9	27	29
10	26	28

Tabla 1: Tamaño de tumores



Se aprecian tres clusters o grupos de objetos bien diferenciados:

Objetos de pequeña longitud y pequeña altura

Objetos de gran longitud y gran altura

Objetos de una altura y longitud media (entre 25 y 30)

Dada una base de datos $D = \{t_1, t_2, \dots, t_n\}$ compuesto por n registros y un valor entero k (número de cluster) la tarea de clustering consiste en establecer una correspondencia:

$$f: D \rightarrow \{C_1, \dots, C_k\}$$

En donde cada t_i se asigna a un cluster C_j

$$1 \leq j \leq k$$

Cálculo de clustering



Para poder llevar a cabo dicha tarea es necesario contar con un mecanismo que permita determinar qué tan parecidos son dos objetos cualesquiera. Una forma de determinar dicho parecido es utilizando una medida de distancia

Existe un gran número de medidas de distancia, aunque las más empleadas, probablemente, son la distancia City-Block o Manhattan, la distancia Euclídea y la distancia Minkowski

Cálculo de clustering



Sean t_i y t_j dos registros con p atributos cada uno y sea W_m el peso asignado al atributo m -ésimo. La distancia D entre los registros T_i y T_j se calcula como sigue:

City-Bloc o Manhattan:

$$d(t_i, t_j) = \sum_{m=1}^p W_m |t_{i_m} - t_{j_m}|$$

Euclídea:

$$(t_i, t_j) = \sqrt{\sum_{m=1}^p W_m (t_{i_m} - t_{j_m})^2}$$



Minkowski:

$$d(t_i, t_j) = \sqrt[\lambda]{\sum_{m=1}^p W_m (t_{i_m} - t_{j_m})^\lambda}$$

En donde $\lambda >$

En las ecuaciones anteriores, t_{i_m} representa el atributo m-ésimo del objeto t_i , y t_{j_m} representa el atributo m-ésimo del objeto t_j . Analizando las medidas de distancia anteriores, se aprecia que, en realidad la distancia Euclídea es un caso particular de la distancia Minkowski para $\lambda = 2$

Cálculo de clustering



Ejemplo:

Cálculo de distancias entre los objetos con identificador 4 y 5 de la tabla 1, utilizando las medidas de distancia City-Block y Euclídea

Considere el conjunto de datos de la tabla 1 Se desea calcular la distancia entre los objetos con Identificador 4 y 5. Se supondrá que todos los pesos W_m valen 1

El primero de estos objetos toma los valores $\langle 2, 2 \rangle$ para sus atributos Longitud y Altura ($p=2$), mientras que el segundo toma los valores $\langle 100, 74 \rangle$

Las distancia entre ellos utilizando las medidas de distancia City-Block y Euclídea se calcula según se indica a continuación



City-Block o Manhattan:

$$d(t_i, t_j) = \sum_{m=1}^p W_m |t_{4_m} - t_{5_m}| = W_1 |t_{4_1} - t_{5_1}| + W_2 |t_{4_2} - t_{5_2}| =$$
$$1 |2-100| + 1 |2-74| = 98 + 72 = 170$$



Euclides:

$$d(t_4, t_5) = \sqrt{\sum_{m=1}^p W_m (t_{4_m} - t_{5_m})^2} = \sqrt{W_1(t_{4_1} - t_{5_1})^2 + W_2(t_{4_2} - t_{5_2})^2} =$$

$$\sqrt{1(2 - 100)^2 + 1(2 - 74)^2} = \sqrt{1(-98)^2 + 1(-72)^2} = \sqrt{9604 + 5184} = 121,6$$



```
w<-c(1,1)
t1<-c(2,2)
t2<-c(100,74)
euclides <- function(w,t1,t2){
  sqrt( w[1]*(t1[1]-t2[1])^2+w[2]*(t1[2]-t2[2])^2)
}
```

```
euclides(w,t1,52)
```

Ejercicio



El conjunto de datos USArrests es un conjunto de datos integrado en R que contiene estadísticas sobre arrestos en los 50 estados de EE. UU. en 1973

Cada fila del conjunto de datos representa un estado y cada columna contiene información sobre el número de arrestos por diferentes crímenes y características demográficas del estado. Aquí está la descripción de las variables en el conjunto de datos USArrests:

Murder: Número de arrestos por asesinato y homicidio no negligente por cada 100,000 habitantes

Assault: Número de arrestos por asalto por cada 100,000 habitantes

UrbanPop: Porcentaje de la población que vive en áreas urbanas

Rape: Número de arrestos por violación por cada 100,000 habitantes



Cargar el conjunto de datos USArrests

```
data(USArrests)
```

Visualizar las primeras filas del conjunto de datos

```
head(USArrests)
```

Verificar la estructura del conjunto de datos

```
str(USArrests)
```

Ejercicio



```
# Escalar las variables para el clustering
datos <- scale(USArrests)
```

```
# Instalar y cargar el paquete factoextra si no está instalado
if (!require(factoextra)) install.packages("factoextra")
library(factoextra)
```

```
# Calcular el método del codo para determinar el número
óptimo de clusters
fviz_nbclust(datos, kmeans, method = "wss")
```

```
# Establecer el número de clusters
set.seed(123) # Fijar la semilla para reproducibilidad
k <- 4 # Número de clusters determinado previamente
```


Ejercicio



Aplicar el algoritmo K-means

```
modelo_kmeans <- kmeans(datos, centers = k, nstart = 25)
```

Ver los resultados del clustering

```
print(modelo_kmeans)
```

Visualizar los clusters en un gráfico

```
fviz_cluster(modelo_kmeans, data = datos,  
              ellipse.type = "convex",  
              palette = "jco",  
              ggtheme = theme_minimal())
```

Significado del resultado



cluster: Un vector de enteros que indica la asignación de cada observación a los clusters. Cada valor en el vector representa el número del cluster al que pertenece la observación correspondiente

centers: Una matriz que contiene las coordenadas de los centroides de los clusters. Cada fila representa un cluster y cada columna representa una de las variables del conjunto de datos

totss: La suma total de cuadrados, que es una medida de la variabilidad total en los datos. Es la suma de las distancias al cuadrado desde cada punto de datos hasta la media global de todos los puntos

withinss: Un vector que contiene la suma de cuadrados dentro del cluster para cada cluster. Representa la variabilidad dentro de cada cluster. Es la suma de las distancias al cuadrado desde cada punto de datos hasta el centroide de su cluster



tot.withinss: La suma total de cuadrados dentro del cluster. Es la suma de todas las componentes de withinss. Representa la variabilidad total dentro de los clusters

betweenss: La suma de cuadrados entre clusters. Representa la variabilidad entre los clusters. Es la diferencia entre totss y tot.withinss

size: Un vector que contiene el número de observaciones en cada cluster

iter: El número de iteraciones que realizó el algoritmo antes de converger. Indica cuántas veces el algoritmo recalculó los centroides y reasignó los puntos de datos

ifault: Un código de error que indica el estado de la ejecución del algoritmo. Un valor de 0 indica que el algoritmo se ejecutó correctamente. Otros valores indican diferentes tipos de errores

Referencias



- [1] <https://www.freepik.es/vectores/pollo>
- [2] <https://es.vecteezy.com/arte-vectorial/7528060-pato-dibujos-animados-color-clipart-ilustracion>
- [3] <https://www.freepik.es/vectores/par-patos-silbadores>
- [4] <https://static.vecteezy.com/system/resources/previews/005/836/459/original/isolated-cute-swan-cartoon-illustration-free-vector.jpg>
- [5] https://rsanchezs.gitbooks.io/ciencia-de-datos-con-r/content/estructuras_control/iterativas/estructuras_iterativas.html



Minería de Datos con R

FIN

jcgcastolo@cucea.udg.mx

silviarc@cuvalles.udg.mx

Referencias

[1] https://comein.uoc.edu/divulgacio/comein/_recursos/imatges/articles/Data-Science-Roles.jpg