**Ejercicios Aprendizaje Automático**

**Ejercicio.**

**Explica la diferencia entre la aproximación tradicional y la aproximación del aprendizaje automático para resolver problemas.**

La diferencia es que por una parte la aproximación tradicional se programa a al programa para que resuelva ciertas tareas, por otra parte la aproximación pro aprendizaje automatic se programa sin saber todos las tareas a la que se enfrentara, de forma que pueda resolver tareas para las que no habia sido programada inicialmente sin la necesidad de reprogramarlo.

**Ejercicio.**

**En aprendizaje automático, ¿qué es la generalización?**

En aprendizaje automático, la generalización es la capacidad de un modelo para aplicar correctamente lo que ha aprendido a nuevos datos no vistos durante el entrenamiento. Es decir, un modelo bien generalizado puede hacer predicciones precisas no solo sobre los datos de entrenamiento, sino también sobre datos desconocidos.

**Ejercicio.**

**Explica la diferencia entre aprendizaje supervisado y aprendizaje no supervisado.**

La diferencia es que el aprendizaje supervisado es el que se hace por medio de un entrenamiento teniendo un “professor” que nos da un conjunto de ejemplos y sus caracteristicas para entrenarnos, el no supervidado se nos dan las caracteristicas sin clasificarnoslas y la maquina generaliza sola y une los grupos, sin que nadie le haya dicho nada.

**Ejercicio.**

**Queremos desarrollar una aplicación capaz de distinguir imágenes de perros y de gatos. Para construir dicha aplicación construimos un banco de datos que solo contiene imágenes de perros de razas grandes y gatos de razas pequeñas ¿estamos construyendo de manera correcta nuestro banco de datos? ¿por qué?**

No, porque estas sesgando el banco de datos a que una caracteristica de los perros es que es grande y que la de los gatos es pequeño, entonces si luego tras entrenar la IA le enseñas un gato grande, es possible que lo clasifique como un perro, porque el banco de entrenamiento debe ser lo más uniforme possible.

**Ejercicio.**

**Para desarrollar la aplicación de perros y gatos descrita en el apartado anterior pensamos inicialmente en utilizar los siguientes descriptores para cada instancia de nuestro dataset: número de patas, número de ojos, ¿es mamífero?, ¿vuela?, y color de pelo. ¿Hemos hecho una buena elección de descriptores? ¿Por qué?**

No, se deben elegir caractristicas diferenciatorias entre los dos tipos que queremos clasificar. En este caso por ejemplo podria ser si sus uñas son plegables o caracteristicas de su capacidad visual.

**Ejercicio.**

**¿Por qué es habitual trabajar con descriptores numéricos?**

Porque estos son más faciles de medir y tartar, incluso se transforman los no numericos en numericos, clasifciando por ejemplo el sexo en 1 a los hombres y 0 a las mujeres o vicervers.

**Ejercicio.**

**¿Qué son los embeddings?**

Son una tecnica de procesamiento de lenguaje que permite transformar el lenguaje humano en vectores matematicos para que sean procesados de major manera.

**Ejercicio.**

**Disponemos de dos datasets de viviendas, en el primero de ellos cada vivienda se ha descrito mediante los siguientes descriptores: nº de habitaciones, nº de baños, localización geográfica y precio; en el segundo de los datasets, cada vivienda es descrita mediante nº de habitaciones, metros cuadrados, y años desde su construcción. ¿Podríamos juntar ambos datasets para tener un único dataset más grande?**

No, porque su único descriptor común es el nº de habitaciones, lo que generaría una cantidad enorme de nulos a tratar que generaría muchos problemas.

**Ejercicio.**

**Disponemos de un dataset de imágenes de bicicletas, motos y coches; y suponed que podemos obtener cualquier propiedad de las imágenes (e.g. color del vehículo, si tiene tubo de escape, etc). Dar tres descriptores que resultarían útiles en este dataset.**

Numero de ruedas, motor, necesita carnet para conducirlo (bool)

**Ejercicio.**

**Dar ejemplos de descriptores binarios, numéricos, cadenas de caracteres y categóricos para describir a una persona.**

Bimarios: calvo

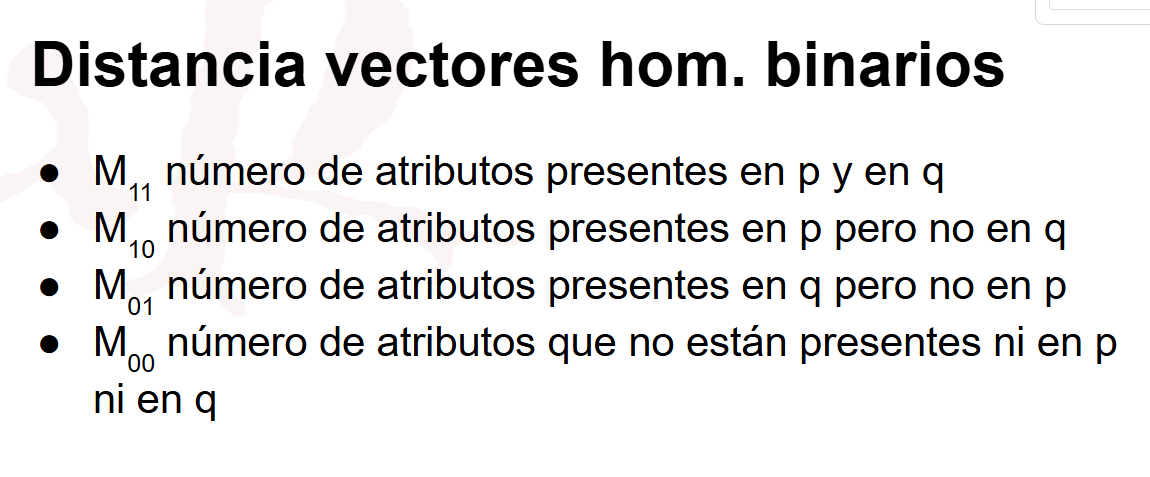
Numéricos edad

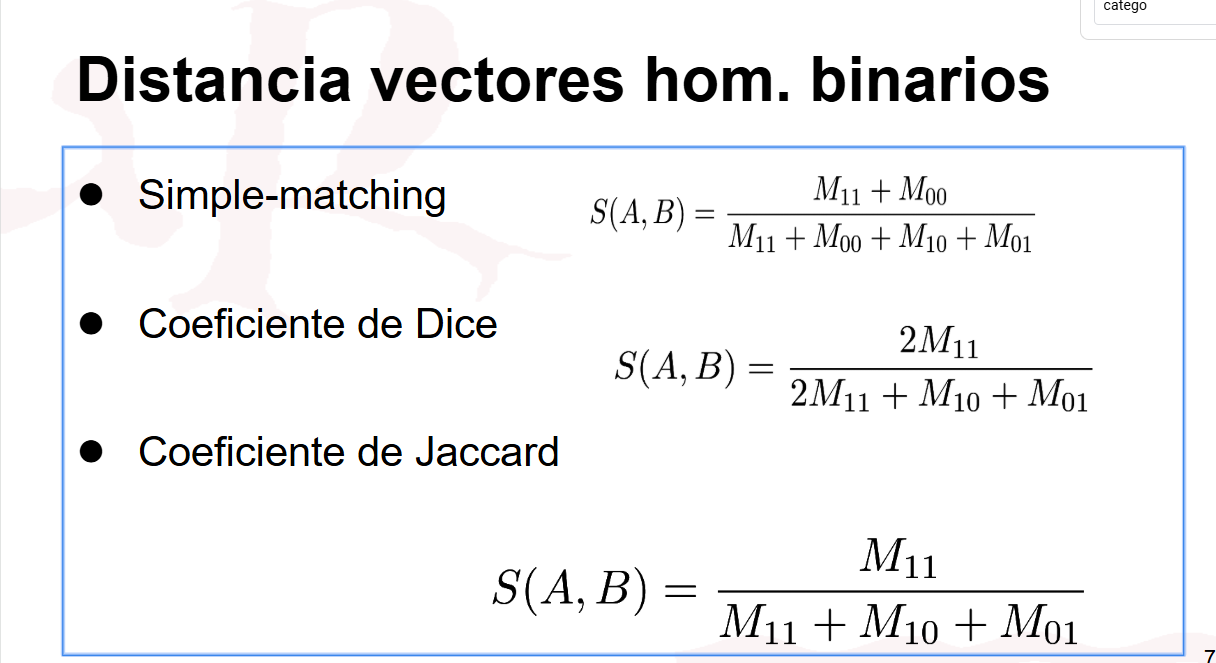
Cad: nombre

Categórico: genero

**Ejercicio.**

**Dadas los siguientes vectores de descriptores binarios A=(1,0,0,1,0) y B=(0,1,1,1,0) calcular los coeficientes de similaridad: simple-matching, coeficiente de Dice y coeficiente de Jaccard.**





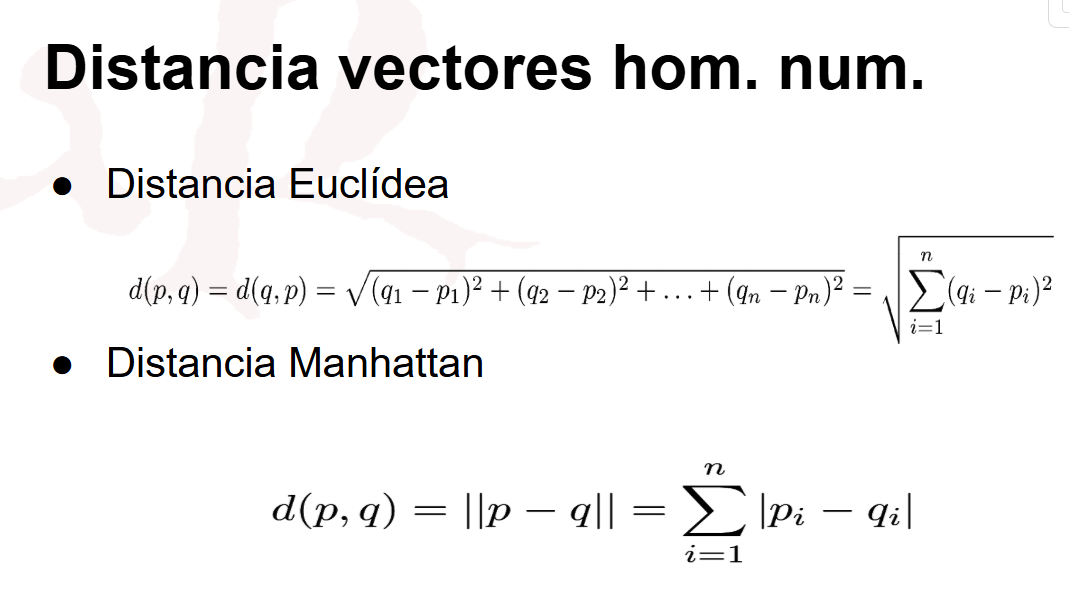
**simple-matching**  =

coeficiente de Dice = = =

cof de Jaccard = = =

**Ejercicio.**

**Dados los puntos (1,2,3) y (-2,4,5), calcular la distancia Euclídea y Manhattan entre ellos.**



Distancia euclídea= Sqrt (3+-2+-2)^2)=1

Distancia manhatan 3+2+2=7

**Ejercicio.**

**Dados los vectores de descriptores A=(verde,azul,amarillo,rojo) y B=(verde,morado,lila,rojo), calcular la similaridad entre ellos utilizando la medida de simple-matching.**

**2/6=1/3**

**Ejercicio.**

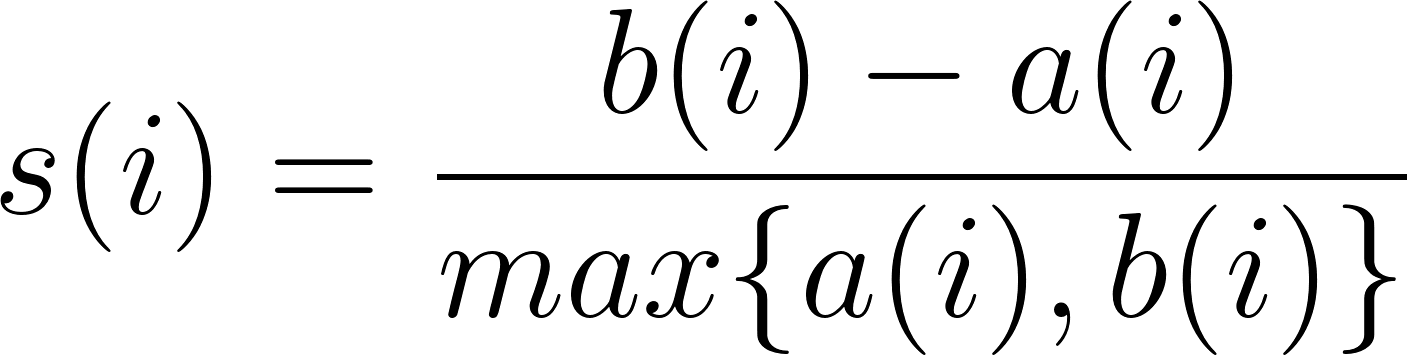
**Considerar el siguiente dataset: (2,2), (0,5), (3,3), (-1,6), (4,7). Aplicar el algoritmo K-means utilizando como centroides iniciales (0,0) y (4,0); en lugar de iterar el proceso hasta converger, repetir el proceso descrito en K-means 4 veces.**

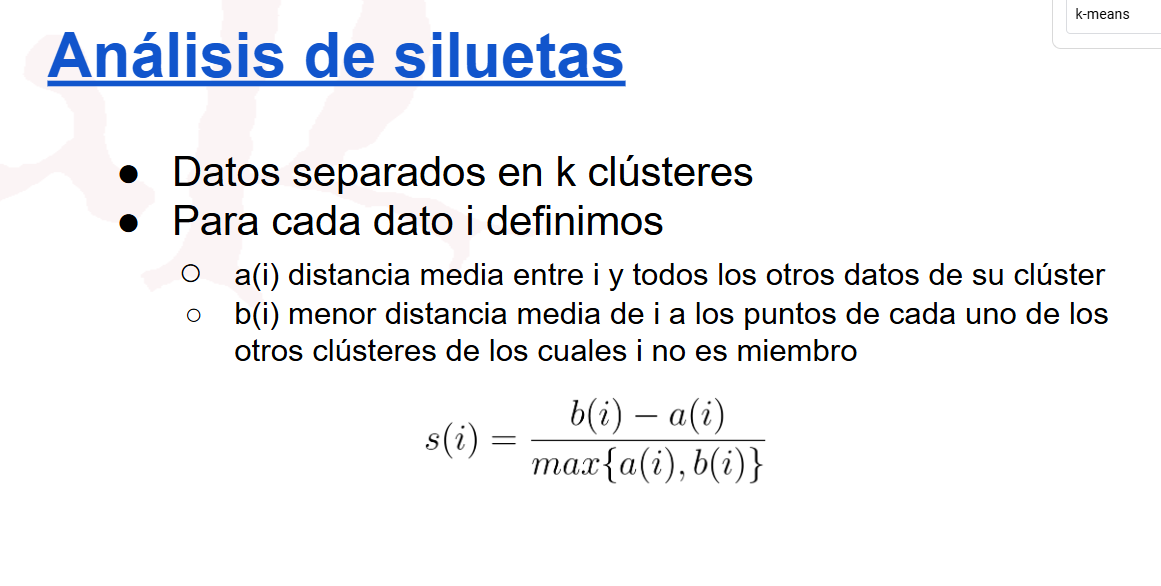
**Ejercicio.**

**¿Cuál es el mayor inconveniente del algoritmo K-means? ¿Qué técnicas podemos aplicar para abordarlo? ¿En qué consisten dichas técnicas?**

Calcular el numero de k y la solución son técnicas de análisis de siluetas que calcula por medio de operaciones el numero adecuado de k

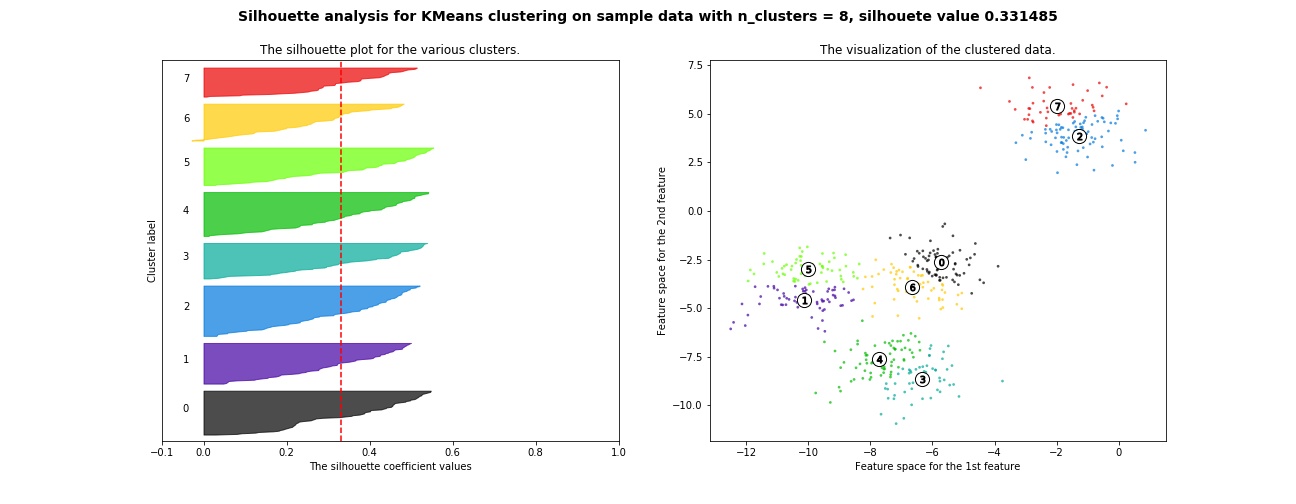
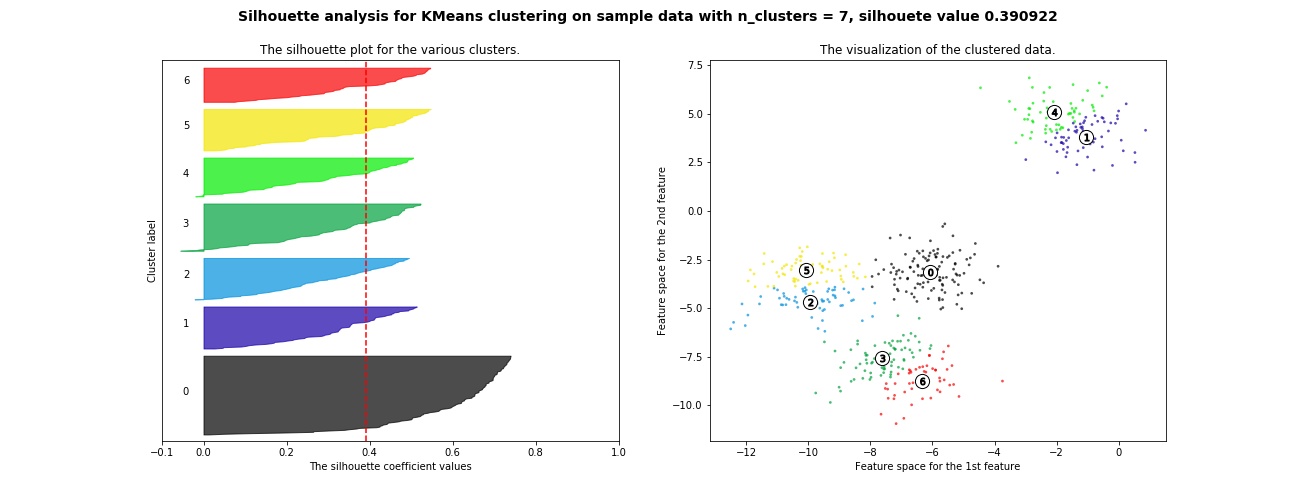
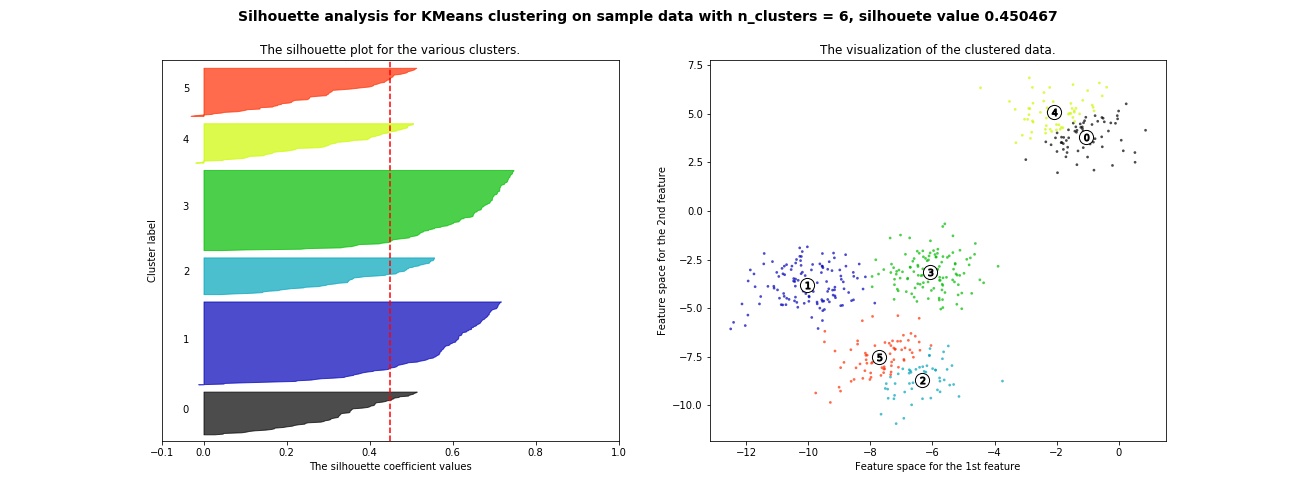
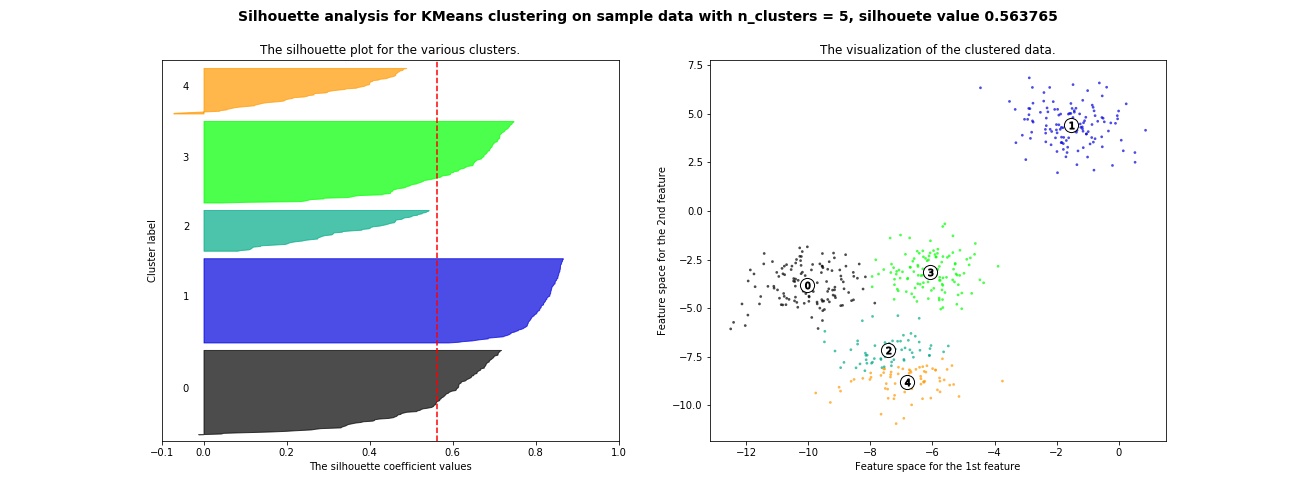
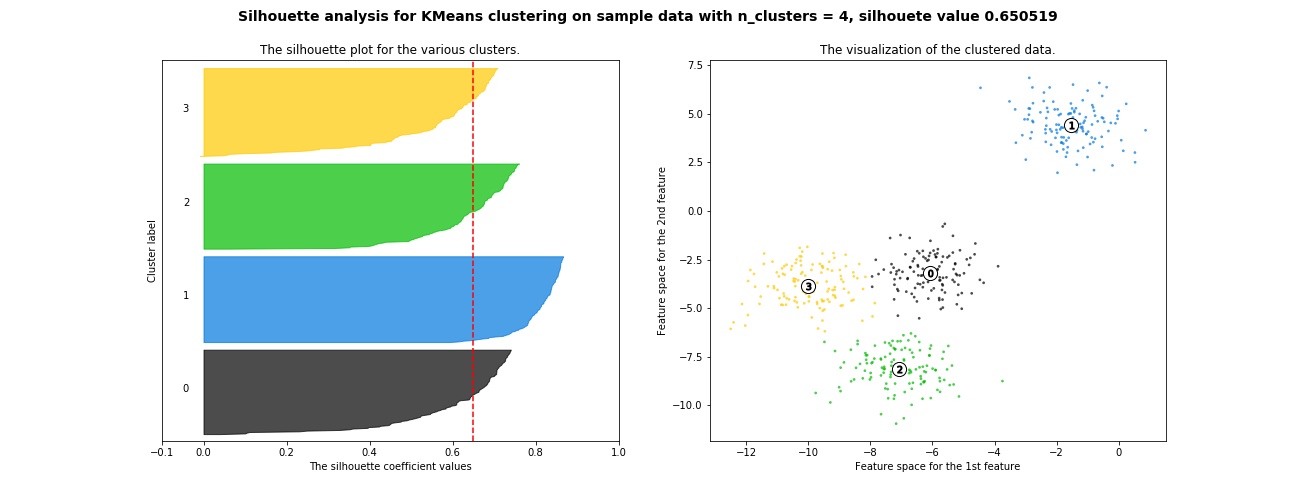
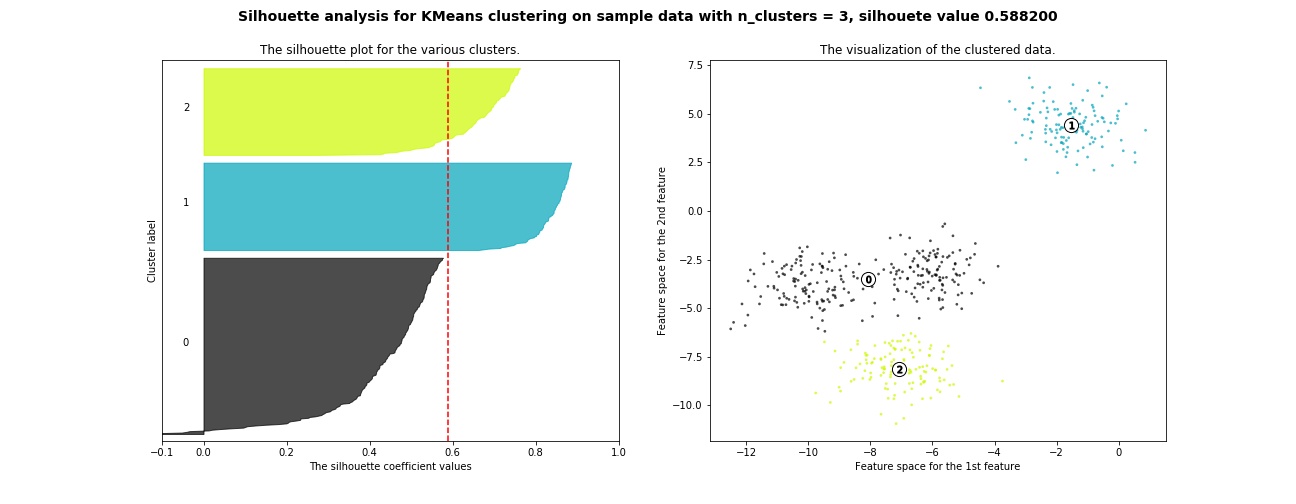
**Ejercicio.**

**En la fórmula para calcular el valor de la silueta (**[****](https://www.codecogs.com/eqnedit.php?latex=s(i)%20%3D%20%5Cfrac%7Bb(i)-a(i)%7D%7Bmax%5C%7Ba(i)%2Cb(i)%5C%7D%7D%250)**), ¿que miden a(i) y b(i)?**



**Ejercicio.**

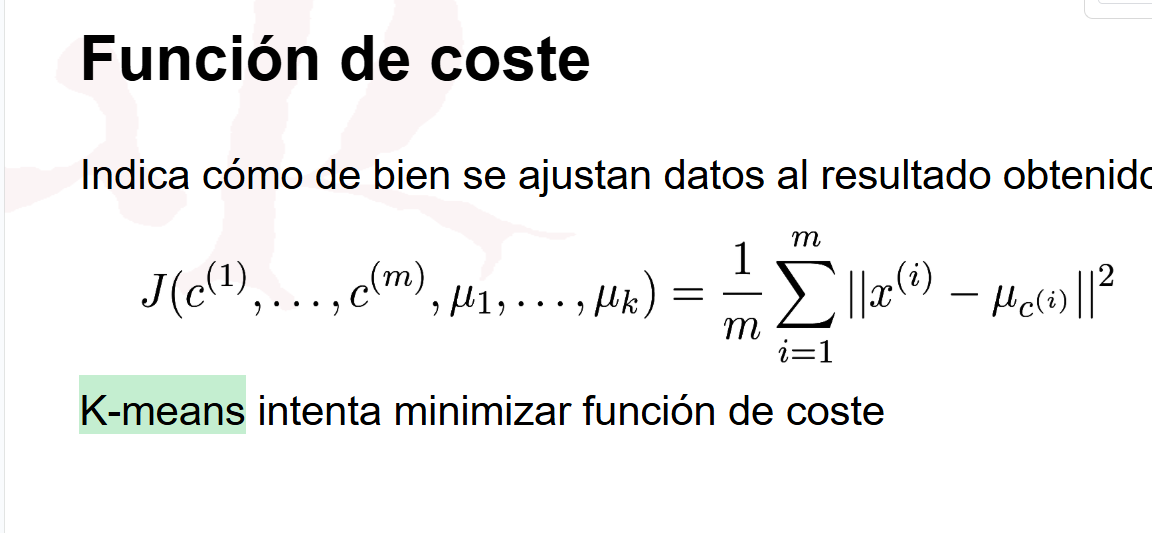
**Ante los siguientes diagramas de siluetas ¿qué número de clústeres recomendarías para el algoritmo de k-means?**

****

**4 klusters porq son el masximo de figuras que pasan la linea, siendo esta la más lejana**

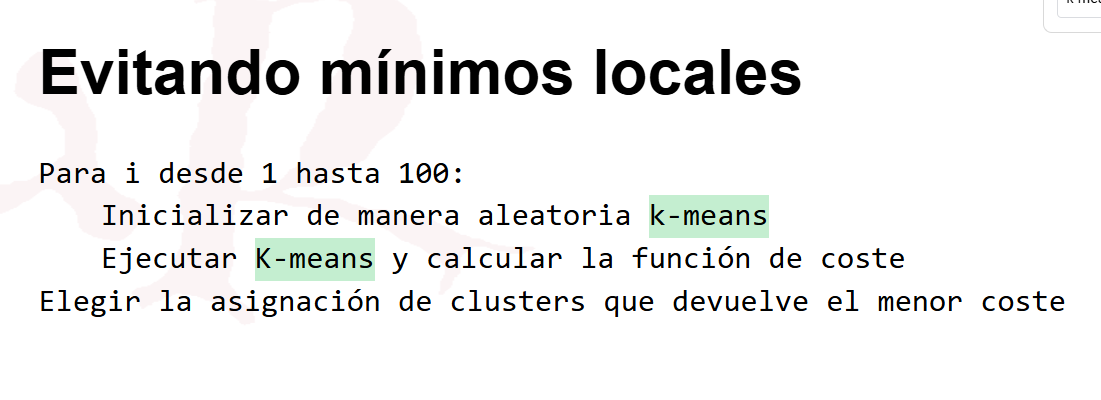
**Ejercicio.**

**Explica que mide la función de coste en el algoritmo de K-means.**



**Ejercicio.**

**¿Qué técnica se utiliza para evitar caer en mínimos locales en el algoritmo de K-means?**



**Ejercicio.**

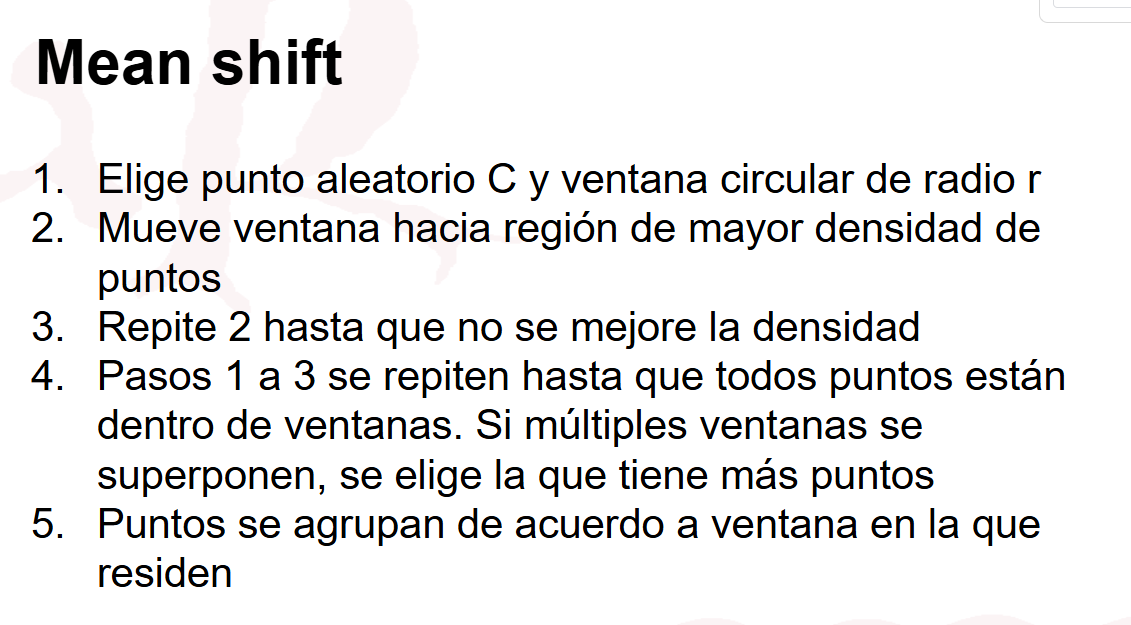
**¿Qué ventajas tienen los algoritmos de clustering DBSCAN y Mean-shift con respecto al algoritmo k-means? ¿Qué problemas tienen estos algoritmos?**

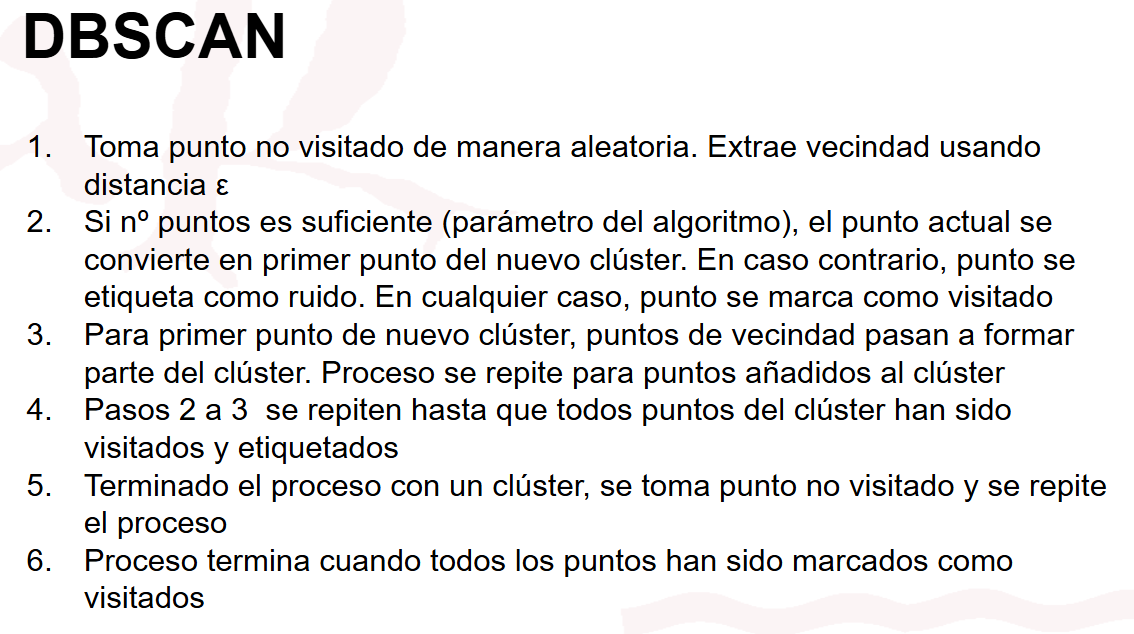
Que son mejores y no necesitan especicar el numero de kluster s el problema que tardan mas (son computacionalmente costosos mimimi)

**Ejercicio.**

**Explica los pasos de los algoritmos de clustering DBSCAN, Mean-shift y K-means.**

Kmins pone puntos aleatorios que luego en cada iteración recalcula en función de los puntos que tienen en su grupo.



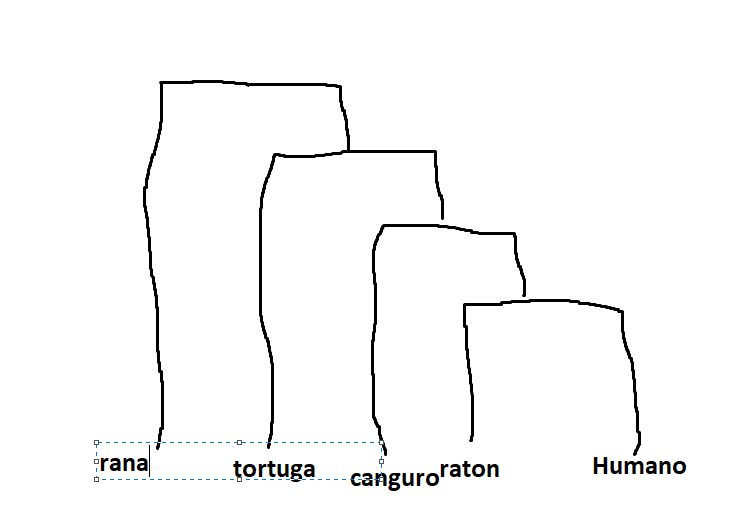


**Ejercicio.**

**Considerando los siguientes datos y utilizando la distancia Euclídea, construir los clústeres que se producen respectivamente al utilizar clustering jerárquico con enlace completo y con enlace de la media. Construye también los dendrogramas que se generan a partir de ellos.**

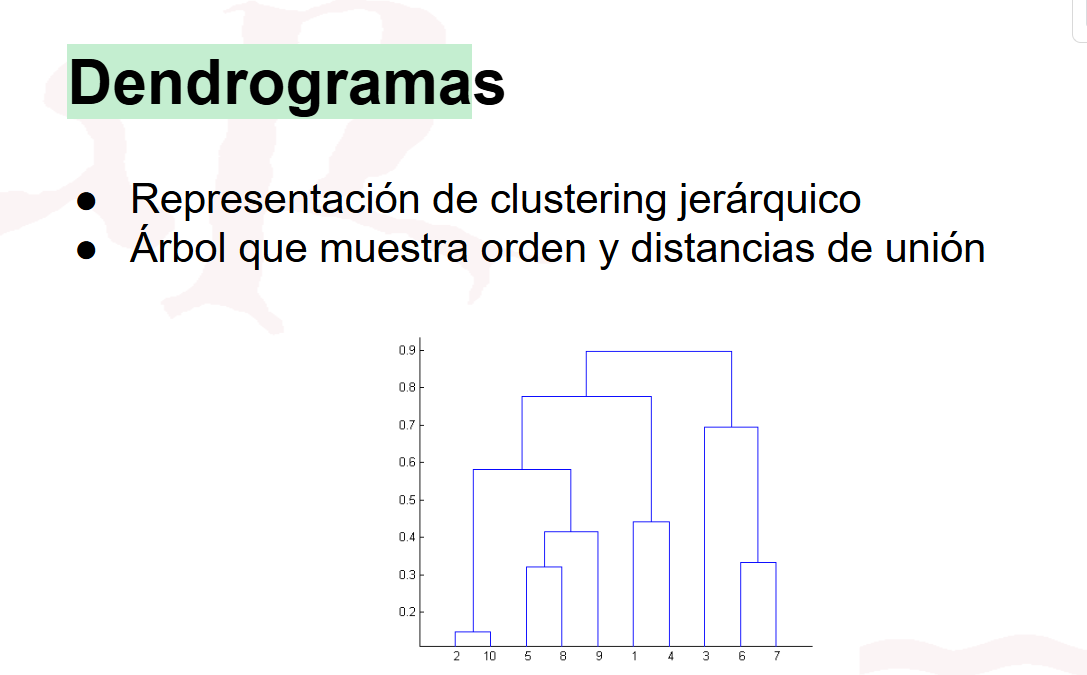
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **4 piernas** | **Huevo amniótico** | **Pelo** | **Desarrollo en placenta** | **Gran cerebro** |
| **Rana** | **1** | **0** | **0** | **0** | **0** |
| **Tortuga** | **1** | **1** | **0** | **0** | **0** |
| **Canguro** | **1** | **1** | **1** | **0** | **0** |
| **Ratón** | **1** | **1** | **1** | **1** | **0** |
| **Humano** | **1** | **1** | **1** | **1** | **1** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Rana** | **Tortuga** | **Canguro** | **Raston** | **Humano** |
| **Rana** | **0** | **1** | **1.4** | **1.7** | **2** |
| **Tortuga** | **1** | **0** | **1** | **1.4** | **1.7** |
| **Canguro** | **1.4** | **1** | **0** | **1** | **1.4** |
| **Ratón** | **1.7** | **1.4** | **1** | **0** | **1** |
| **Humano** | **2** | **1.7** | **1.4** | **1** | **0** |



**Ejercicio.**

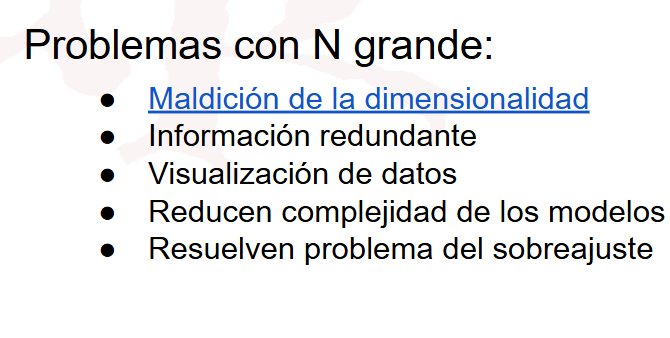
**Explica qué es un dendrograma y para qué se utiliza.**



**Ejercicio.**

**¿Cuáles son las razones para reducir la dimensionalidad de nuestros datos?**

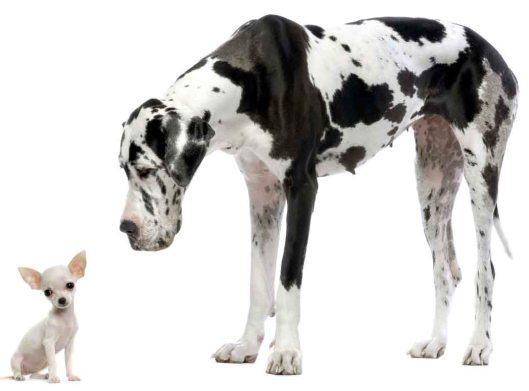
Para mostrarse gráficamente



**Ejercicio.**

**¿A qué nos referimos cuando hablamos de la maldición de la dimensionalidad?**

**Que cuando tenemos demasiados descriptores muchas veces**

La **maldición de la dimensionalidad** es un fenómeno que surge en el análisis de datos y el aprendizaje automático cuando se trabaja con espacios de alta dimensionalidad, es decir, cuando los datos tienen un gran número de características o variables. A medida que aumenta la dimensionalidad, el volumen del espacio de datos crece exponencialmente, lo que genera varios desafíos:

1. **Esparcidad de los datos**: En un espacio de alta dimensionalidad, los datos tienden a estar muy dispersos. Esto dificulta la identificación de patrones o relaciones, ya que la mayoría de las regiones del espacio están vacías o contienen muy pocos puntos.
2. **Dificultad en la distancia entre puntos**: En dimensiones altas, la distancia entre cualquier par de puntos tiende a ser similar. Esto afecta negativamente a algoritmos que dependen de medidas de distancia, como los métodos de clustering o los vecinos más cercanos (k-NN).
3. **Sobrecarga computacional**: El procesamiento y almacenamiento de datos en espacios de alta dimensionalidad requiere más recursos computacionales y tiempo.
4. **Sobreajuste en modelos de aprendizaje automático**: Con muchas variables, los modelos pueden ajustarse demasiado a los datos de entrenamiento, perdiendo generalización y desempeño en datos nuevos.
5. **Dificultad en la visualización**: Visualizar datos en más de tres dimensiones es complicado, lo que dificulta la interpretación y el análisis exploratorio.

**Ejercicio.**

**Explica la diferencia entre la selección de descriptores y la extracción de descriptores como métodos para reducir la dimensionalidad de los datos.**

### **Selección de descriptores**

* **Definición**: Consiste en seleccionar un subconjunto de las características originales del conjunto de datos, descartando aquellas que son irrelevantes o redundantes.

### **Extracción de descriptores**

* **Definición**: Consiste en transformar las características originales en un nuevo conjunto de características de menor dimensionalidad, generalmente mediante una combinación lineal o no lineal de las características originales.

**Ejercicio.**

**Explica la diferencia entre la selección secuencial de descriptores hacia adelante y la selección secuencial de descriptores hacia atrás.**

### **Selección secuencial hacia adelante (Forward Selection)**

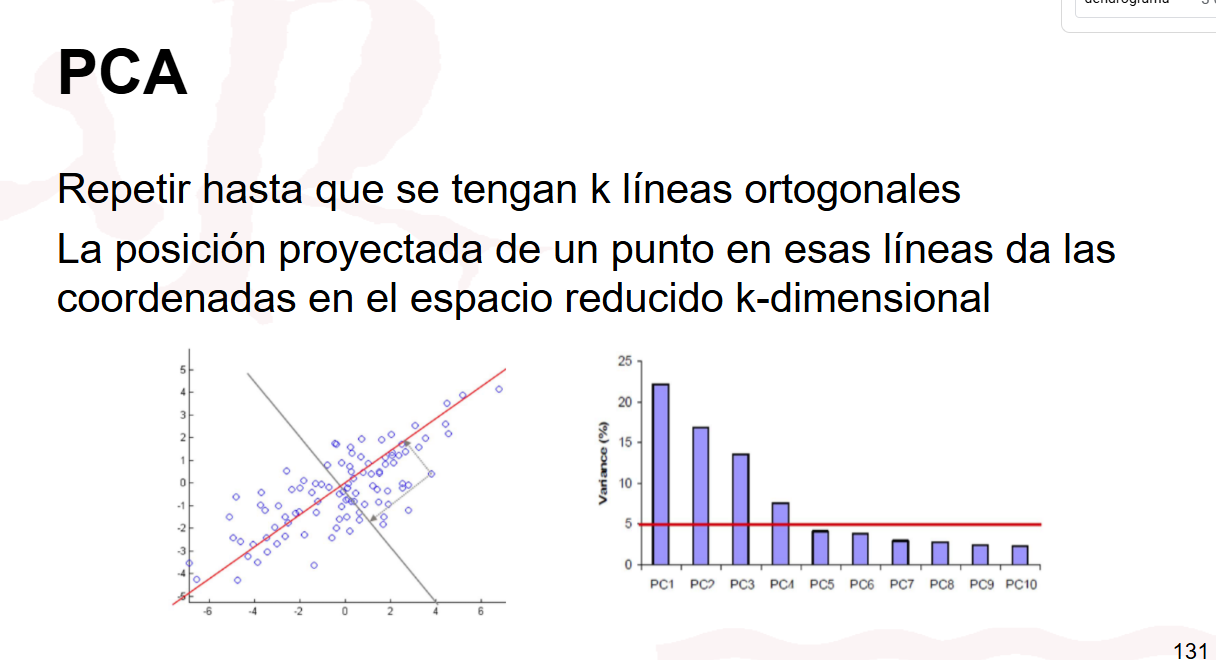
* **Enfoque**: Comienza con un conjunto vacío de características y, en cada paso, añade la característica que más mejora el rendimiento del modelo.

### **Selección secuencial hacia atrás (Backward Selection)**

* **Enfoque**: Comienza con todas las características y, en cada paso, elimina la característica que menos contribuye al rendimiento del modelo.

**Ejercicio.**

**¿En qué nos basamos para elegir el número de componentes principales en el algoritmo de PCA?**



**Ejercicio.**

**¿Por qué decimos que el método t-SNE tiene una componente estocástica?**

La componente estocástica de t-SNE surge principalmente de la inicialización aleatoria y el proceso de optimización no convexo. Esto hace que el método sea flexible y poderoso para visualizar estructuras complejas en datos de alta dimensionalidad, pero también introduce variabilidad en los resultados. Por ello, es importante entender y manejar esta aleatoriedad para obtener interpretaciones confiables.

**Ejercicio.**

**Suponiendo que los datos no son linealmente separables, ¿qué técnicas de reducción de la dimensionalidad sería conveniente usar?**

Técnicas de **Extracción de descriptores ya que estas sacan descriptores nuevos de los iniciales, sin eliminar ninguno.**

**Ejercicio.**

**Si queremos desarrollar un modelo de clasificación para predecir si un email es “spam” o “no es spam”. ¿Cuál de las siguientes afirmaciones es cierta?**

* **Las palabras en el asunto serán buenas etiquetas.**
* **Usaremos ejemplos no etiquetados para entrenar el modelo.**
* **Algunas etiquetas pueden ser engañosas.**
* **Los emails que no son marcados como “spam” o “no spam” son ejemplos no etiquetados.**

**Ejercicio.**

**En los problemas de aprendizaje supervisado, ¿por qué es importante que todas las categorías cuenten con un número uniforme de instancias?**

Si hay un numero menor de una categoría, podrá precedir esa categoría de forma mucho menos eficiente y eficaz. Esto se debe a que en el entrenamiento no ha tenido suficientes ejemplos para poder generalizar.

**Ejercicio.**

**Explica en qué tres conjuntos se suele partir un dataset en el aprendizaje supervisado y para qué se utiliza cada uno de esos conjuntos.**

* Entrenamiento:
  + Entrenamiento puro
  + Verificación
* testeo

**Ejercicio.**

**Al partir un dataset en conjunto de entrenamiento y de test, ¿por qué es importante que estos conjuntos sean independientes?**

Porque sino no estará generalizado si no que dira lo que ya sabe porque se lo han dicho, estará memorizando y no razonando.

**Ejercicio.**

**¿Por qué no podemos utilizar el conjunto de test para optimizar los hiperparámetros de los modelos de aprendizaje supervisado? ¿Cuál es el modo correcto de optimizar dichos hiperparámetros?**

No podemos utilizar el **conjunto de test** para optimizar los hiperparámetros de los modelos de aprendizaje supervisado porque esto introduciría un **sesgo de evaluación** y comprometería la capacidad del modelo para generalizar a datos no vistos.

**Ejercicio.**

**¿Cuáles son los hiperparámetros del algoritmo Knn?**

El numero de klusters y distancia

**Ejercicio.**

**¿Por qué el algoritmo Knn es tan rápido de entrenar?**

Porque no necesita tiempo de entrenamiento

**Ejercicio.**

**Dado un dataset con 10 instancias, 8 de la clase A y 2 de la clase B y trabajando con el algoritmo de Knn, ¿sería razonable utilizar un k con valor 3? ¿y con valor 5?**

No seria razonable

**Ejercicio.**

**En el algoritmo de Knn, ¿por qué no usamos valores pares para k?**

Debido al riego de empate en las predicciones, especialmente en claisificaciones binarias

**Ejercicio.**

**En el algoritmo de Knn, ¿qué representa el valor de k?**

El numero de vecinos que se tiene en cuenta para decidir a que clase pernece el punto

**Ejercicio.**

**¿Por qué no es adecuado utilizar el algoritmo de Knn cuando la dimensión de los vectores de descriptores crece?**

k-NN no es adecuado en alta dimensionalidad debido a:

1. La **maldición de la dimensionalidad**, que hace que las distancias sean poco informativas y los datos estén muy dispersos.
2. La **ineficiencia computacional** en el cálculo de distancias y el almacenamiento de datos.
3. El **sobreajuste** causado por características irrelevantes o ruidosas.
4. La **dificultad para interpretar** los resultados.

Para estos casos, es recomendable aplicar técnicas de reducción de dimensionalidad, selección de características o utilizar algoritmos más robustos para alta dimensionalidad.

**Ejercicio.**

**Dadas las siguientes instancias (separadas por punto y coma):**

* **Clase A: (1,5); (2,4); (3,2); (4,3)**
* **Clase B: (0,3); (-2,3); (1,2); (-1,1)**

**Clasificar con el algoritmo Knn usando la distancia Euclídea y valor de k=3 las instancias (2,2); (0,0) y (1,3).**

Lo hacemos para (2,2)

A

(1,5)-> sqrt(10)

(2,4)->sqrt(4)

(3,2)->sqrt(1)

(4,3)-> sqrt(5)

B

(0,3)->sqrt(5)

(-2,3)->sqrt(17)

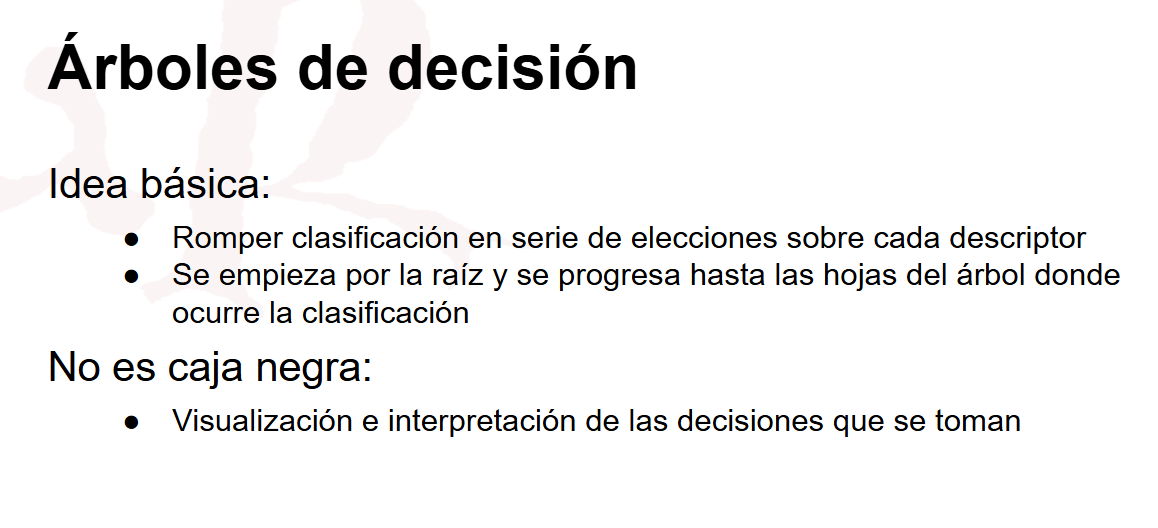
(1,2)->sqr(1)

(-1,1) ->sqrt(10)

Seria de clase A

**Ejercicio.**

**Explica cuál es la idea intuitiva de los árboles de decisión.**



**Ejercicio.**

**¿Por qué se dice que los árboles de decisión son un algoritmo de caja blanca?**

Visualización e interpretación de las decisiones que se toman

**Ejercicio.**

**Considerando el siguiente árbol de decisión**

**Completa la siguiente tabla donde la columna Examen indica la decisión que se toma basándonos en el árbol de decisión anterior:**

Aprovechamiento

Extraordinaria

Deficiente

Asiste

Regular

Extraordinaria

No

Sí

Final

Asiste

Bueno

Final

No

Sí

Exento

Exento

Excelente

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aprovechamiento** | **Asiste** | **Examen** |
| **Instancia 1** | **Deficiente** | **Sí** | **extraordinarias** |
| **Instancia 2** | **Bueno** | **No** | **Final** |
| **Instancia 3** | **Excelente** | **No** | **exento** |

**Ejercicio.**

**Explica el papel que juega el ratio de aprendizaje en el método del descenso de gradiente.**

El **ratio de aprendizaje** en el descenso de gradiente controla el tamaño de los pasos que se toman para actualizar los parámetros del modelo. Un valor adecuado es crucial para:

* **Converger rápidamente** al mínimo de la función de costo.
* **Evitar divergencias** u oscilaciones.
* **Lograr un equilibrio** entre velocidad y precisión.

Elegir el ratio de aprendizaje correcto es un paso clave en la optimización de modelos de aprendizaje automático.

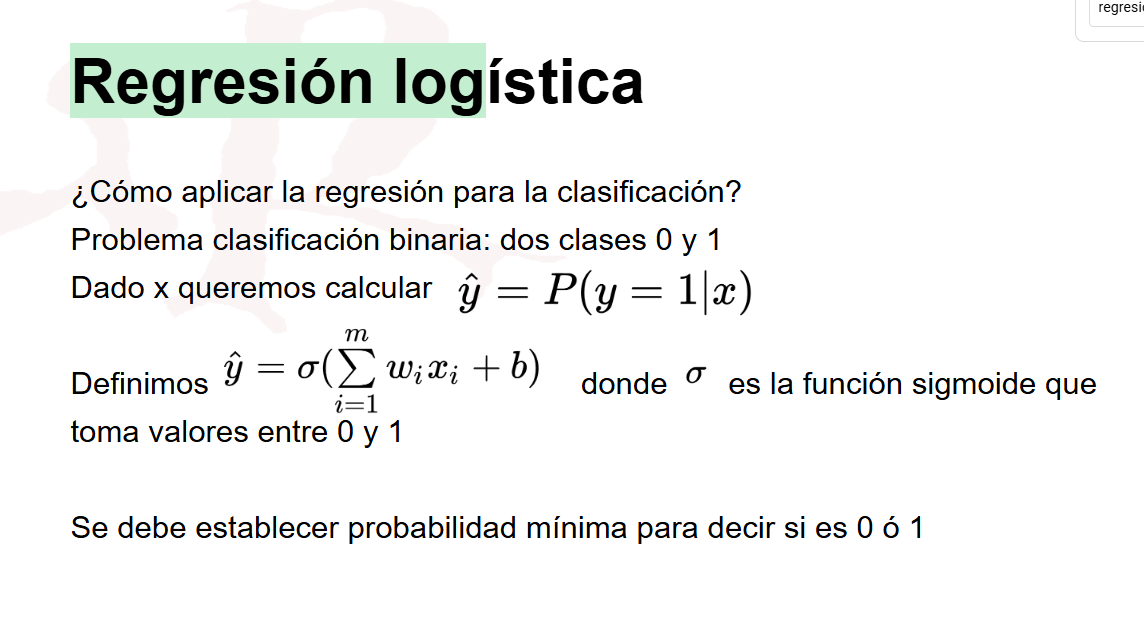
**Ejercicio.**

**Explica en qué consiste el método de descenso de gradiente.**

El descenso de gradiente es un algoritmo de optimización utilizado para minimizar una función de costo (o pérdida) ajustando los parámetros del modelo. La idea es iterativamente mover los parámetros en la dirección opuesta al gradiente de la función de costo, ya que el gradiente indica la dirección de mayor crecimiento de la función.

**Ejercicio.**

**En regresión logística, ¿qué característica debe tener una función para reemplazar a la función sigmoide?**



Para reemplazar la función sigmoide en regresión logística, la función alternativa debe:

1. Mapear valores reales a un rango [0,1][0,1].
2. Ser monótona creciente.
3. Ser diferenciable.
4. Ser simétrica alrededor de 0.
5. Tener comportamiento asintótico correcto.

Aunque existen alternativas como la función **tanh**, **probit** o **arctan**, la sigmoide sigue siendo la opción más común debido a su simplicidad y propiedades matemáticas favorables.

**Ejercicio.**

**En regresión logística, si la función sigmoide nos devuelve el valor 0.6 para una instancia, ¿qué etiqueta asignaremos a dicha instancia?**

Clase positiva

**Ejercicio.**

**Explica cómo la regresión logística puede ser aplicada para abordar problemas de clasificación múltiple.**

### **Resumen**

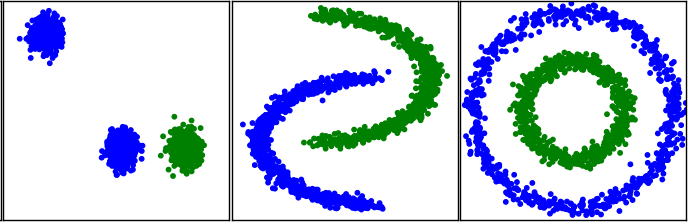
La regresión logística puede aplicarse a problemas de clasificación múltiple utilizando dos estrategias principales:

1. **One-vs-Rest (OvR)**: Entrena un modelo por cada clase y selecciona la clase con la probabilidad más alta.
2. **Multinomial Logistic Regression**: Entrena un único modelo que predice probabilidades para todas las clases usando la función softmax.

La elección entre estas estrategias depende del problema específico, el tamaño del dataset y los recursos computacionales disponibles.

**Ejercicio.**

**Dados los siguientes gráficos donde los puntos verdes representan una clase y los puntos azules otra, ¿cuáles son linealmente separables?**

****

El primero

**Ejercicio.**

**¿Qué son los support vectors?**

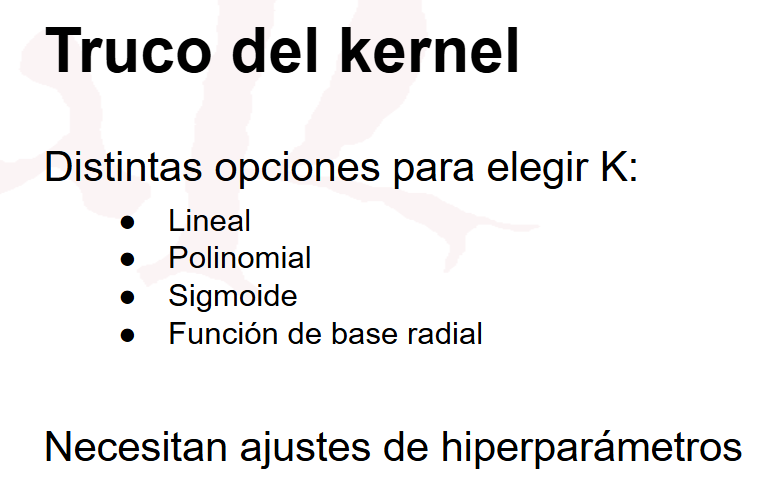
Los **support vectors** son los puntos de datos más cercanos al hiperplano de separación en SVM. Son cruciales porque:

1. Definen la posición y orientación del hiperplano.
2. Maximizan el margen entre las clases, lo que mejora la generalización del modelo.
3. Hacen que SVM sea eficiente, ya que solo se necesitan estos puntos para hacer predicciones.

En resumen, los support vectors son el "corazón" del algoritmo SVM, y entender su papel es clave para comprender cómo funciona este poderoso método de clasificación

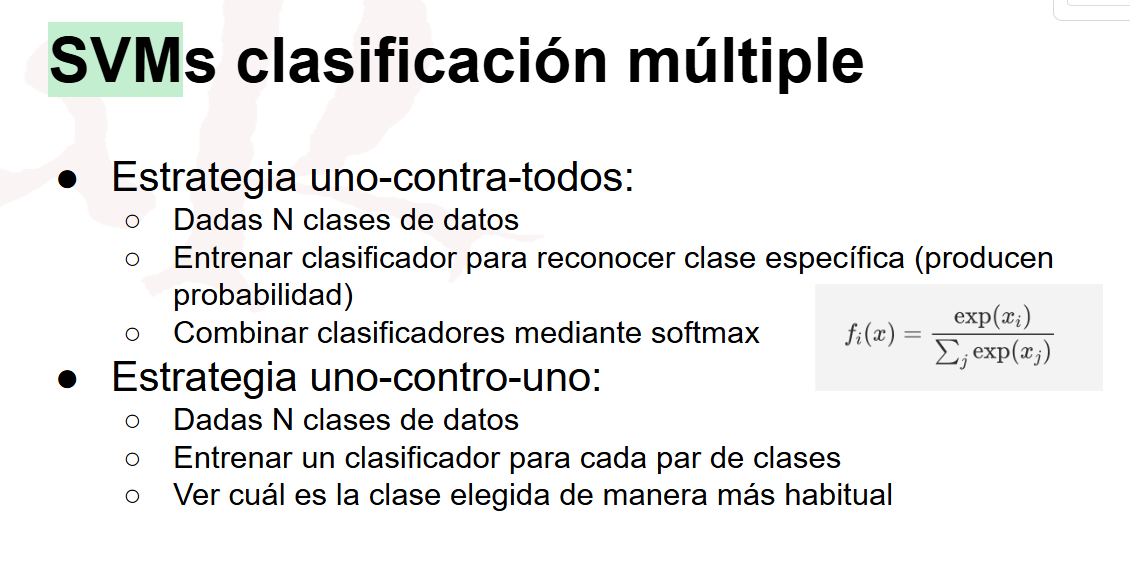
**Ejercicio.**

**Intuitivamente, ¿en qué consiste el truco del kernel en el algoritmo SVM?**



**Ejercicio.**

**Explica la diferencia entre las estrategias uno-contra-todos y uno-contra-uno aplicadas en SVMs para clasificación múltiple.**



**Ejercicio.**

**¿Es posible utilizar el perceptrón para clasificar de manera correcta conjuntos de datos que no son linealmente separables? ¿Por qué?**

El **perceptrón** no puede clasificar correctamente conjuntos de datos **no linealmente separables** porque es un modelo lineal y solo puede aprender fronteras de decisión lineales. Para manejar datos no linealmente separables, es necesario utilizar alternativas como:

* **Perceptrón multicapa (MLP)**.
* **Máquinas de vectores de soporte (SVM) con kernels**.
* **Transformación de características**.
* **Árboles de decisión o métodos de ensemble**.

Estas alternativas permiten aprender fronteras de decisión no lineales y manejar problemas de clasificación más complejos.

**Ejercicio.**

**El perceptrón en su formulación original, ¿sirve para resolver problemas de clasificación binarios o multi-clase?**

Binaria

**Ejercicio.**

**En el aprendizaje de las redes neuronales, ¿qué es una época?**

### **Resumen**

* Una **época** es un paso completo del entrenamiento en el que la red neuronal procesa todas las muestras del conjunto de entrenamiento.
* El número de épocas es un hiperparámetro clave que controla cuántas veces la red "ve" los datos de entrenamiento.
* Demasiadas épocas pueden causar sobreajuste, mientras que muy pocas pueden resultar en un modelo subentrenado.
* El entrenamiento suele monitorearse utilizando un conjunto de validación para determinar el número óptimo de épocas.

Nº iteraciones bucle determinado por nº veces que se muestra conjunto entrenamiento a la red, a este nº de iteraciones se le conoce como épocas

En resumen, las épocas son una parte esencial del entrenamiento de redes neuronales, ya que definen cuántas veces el modelo ha aprendido de los datos completos y ayudan a controlar la convergencia y el sobreajuste.

**Ejercicio.**

**¿Cuándo está asegurada la terminación en el algoritmo de aprendizaje del perceptrón?**

La terminación en el algoritmo de aprendizaje del perceptrón está asegurada **si y solo si las clases son linealmente separables.**

**Ejercicio.**

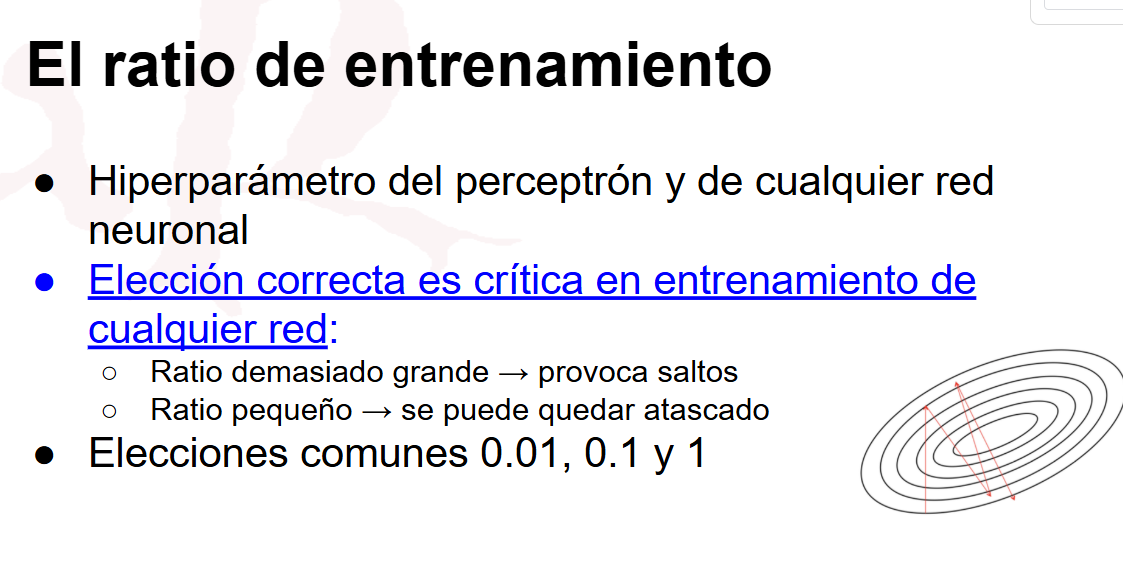
**En caso de que el conjunto de entrenamiento no sea linealmente separable, ¿cuándo termina el proceso de aprendizaje del perceptrón?**

No garantiza que se termine

**Ejercicio.**

**Explica por qué es fundamental elegir un ratio de entrenamiento que no sea excesivamente grande o excesivamente pequeño en el entrenamiento de una red neuronal.**

Porque si es muy pequeño será malo y si es muy grande será poco eficiente



**Ejercicio.**

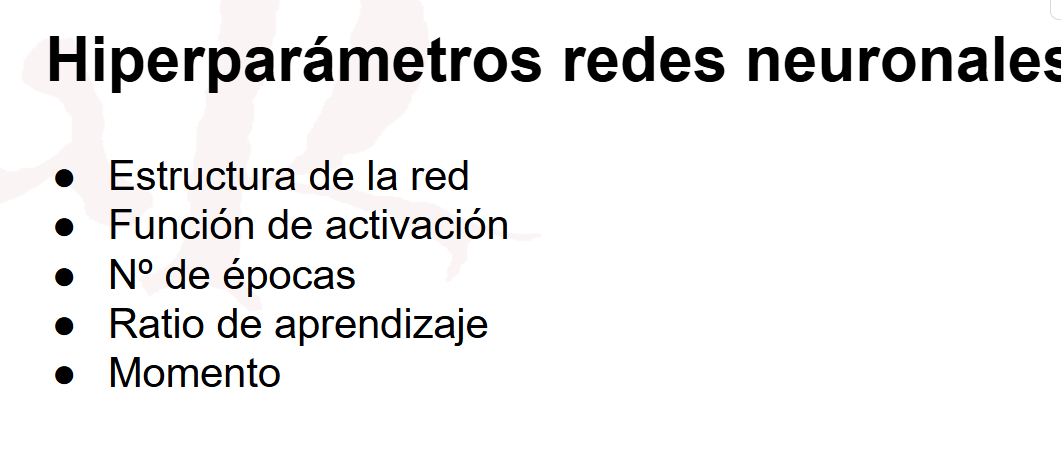
**¿Qué técnica utiliza el método de backpropagation para encontrar los pesos que minimizan el error durante el proceso de entrenamiento?**

 En resumen, backpropagation proporciona los gradientes necesarios, y el descenso de gradiente (o sus variantes) utiliza esos gradientes para ajustar los pesos y mejorar el rendimiento de la red.

**Ejercicio.**

**¿Cuáles son los hiperparámetros de las redes neuronales?**

Nº de capas ocultas y numero de neuronas por capas



**Ejercicio.**

**Explica por qué son importantes las funciones de activación en las redes neuronales.**

Las funciones de activación son importantes en las redes neuronales porque:

1. Introducen **no linealidad**, permitiendo que el modelo aprenda patrones complejos.
2. Hacen posible la **aproximación universal** de funciones.
3. Afectan la **propagación del gradiente** durante el entrenamiento.
4. Permiten a las redes profundas aprender **representaciones jerárquicas** de los datos.

En resumen, sin funciones de activación, las redes neuronales serían incapaces de modelar la complejidad de la mayoría de los problemas del mundo real.

**Ejercicio.**

**¿Qué dice el “no free lunch theorem”?**

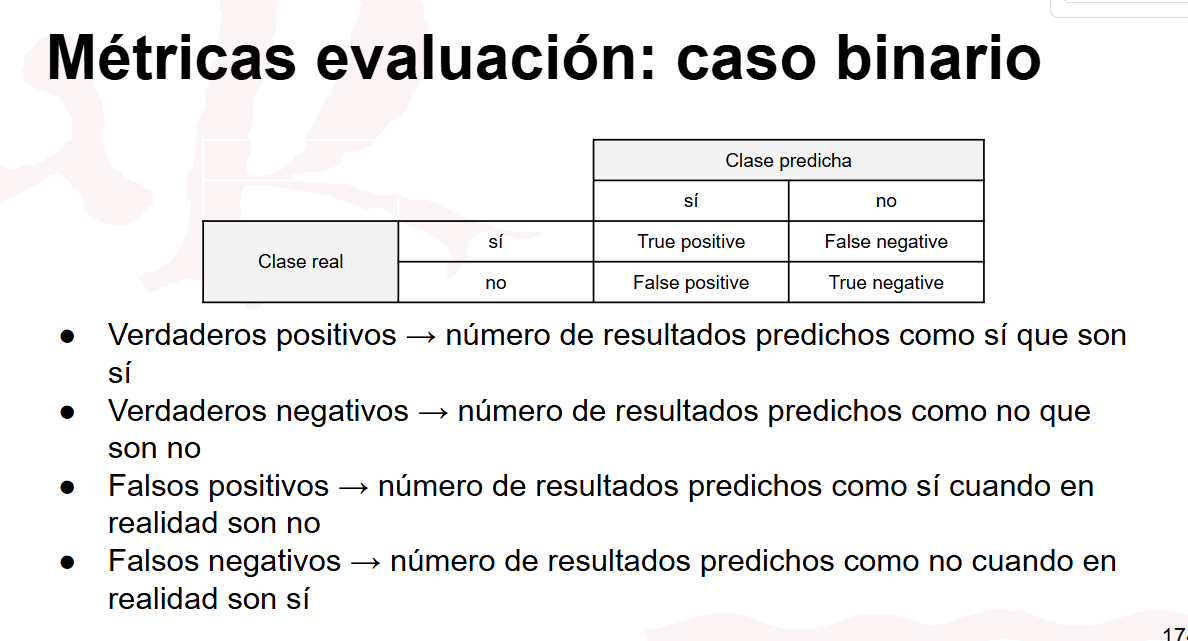
El **No Free Lunch Theorem** nos recuerda que:

* No existe un algoritmo de optimización o aprendizaje que sea universalmente superior.
* La elección del algoritmo debe basarse en el conocimiento del problema y los datos.
* Es importante probar y comparar múltiples algoritmos para encontrar el más adecuado.

En resumen, el NFLT es una advertencia contra la búsqueda de soluciones universales y un recordatorio de la importancia de adaptar las herramientas al problema específico que se está abordando.

**Ejercicio.**

**Describe qué son los verdaderos positivos, los verdaderos negativos, los falsos positivos y los falsos negativos.**



**Ejercicio.**

**Tenemos un clasificador capaz de determinar si una imagen contiene un gato o no. De las 100 imágenes que disponemos, 50 de ellas contienen un gato y las otras 50 no. De las 50 imágenes con gato, el clasificador dice que hay gato en 30 de ellas, y en las otras 20 dice que no. De las 50 imágenes sin gato, el clasificador dice que hay gato en 10 de ellas, y en las otras 40 dice que no. A partir de esta información dar los valores de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos; y calcular los siguientes ratios: sensitivity (definido como TP/(TP+FN)), specificity (definido como TN/(FP+TN)) y accuracy (definido como (TP+TN)/(TP+FP+FN+TN)).**

**Ejercicio.**

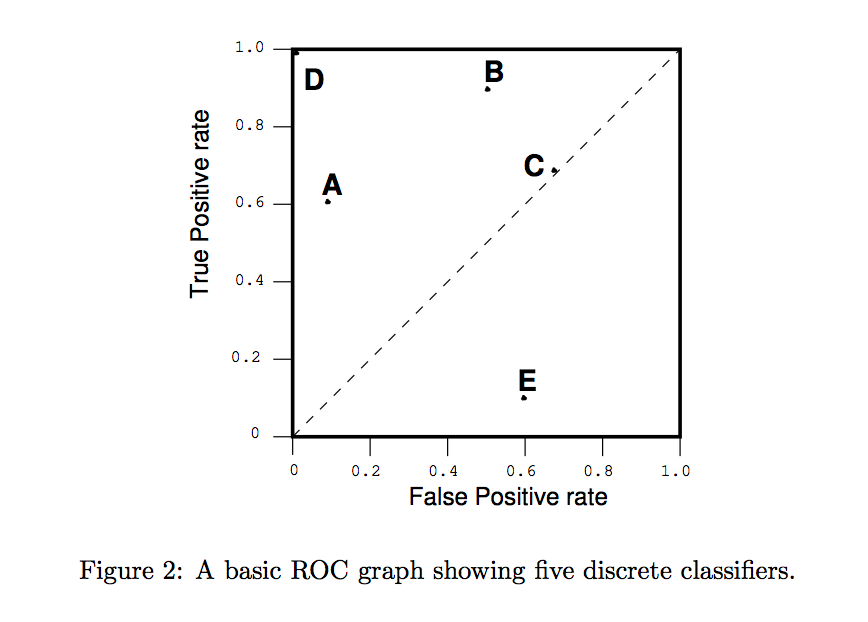
**¿Por qué nos interesa que una matriz de confusión sea diagonal?**

### **Resumen**

Una matriz de confusión diagonal es deseable porque indica que el modelo está realizando **predicciones perfectas**, sin cometer errores de clasificación. Sin embargo, en la práctica, lograr una matriz diagonal es poco común y puede ser un indicador de sobreajuste o de un problema demasiado simple. Por lo tanto, es importante evaluar el modelo en conjuntos de datos de prueba y validación para asegurar su generalización.

**Ejercicio.**

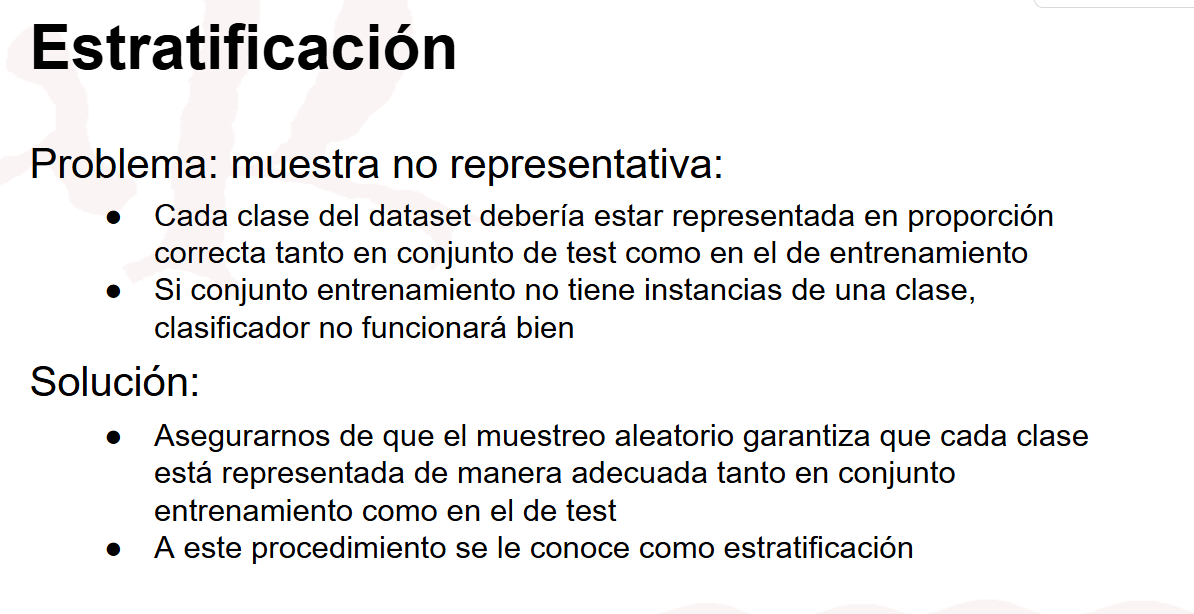
**En la siguiente imagen de un espacio ROC ¿cuál es el mejor clasificador? ¿y el peor? ¿cuál es el equivalente a predecir de manera aleatoria?**

****

**D es el major y E el peor y la aleatoria el C**

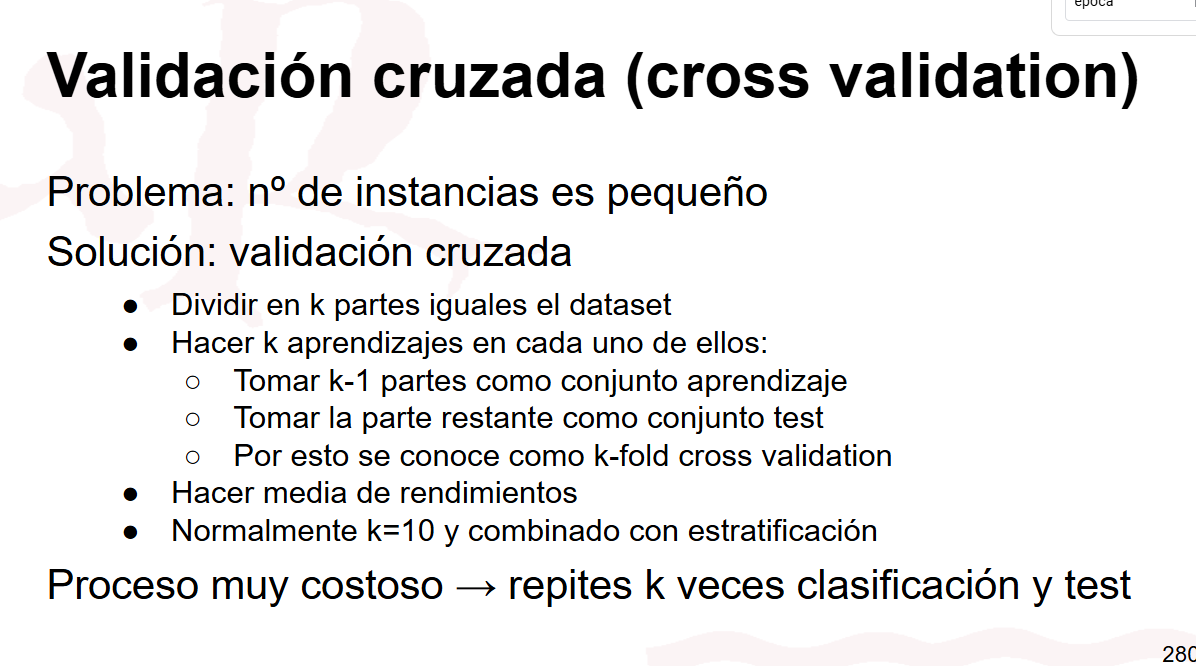
**Ejercicio.**

**¿Qué problema resuelve la estratificación? y ¿cómo lo resuelve?**



**Ejercicio.**

**¿Qué problema resuelve la validación cruzada? y ¿cómo lo resuelve?**



**Ejercicio.**

**¿Por qué el proceso de validación cruzada es un proceso costoso?**

Proceso muy costoso → repites k veces clasificación y test

**Ejercicio.**

**Explica cuándo ocurre el subajuste y cuándo el sobreajuste.**

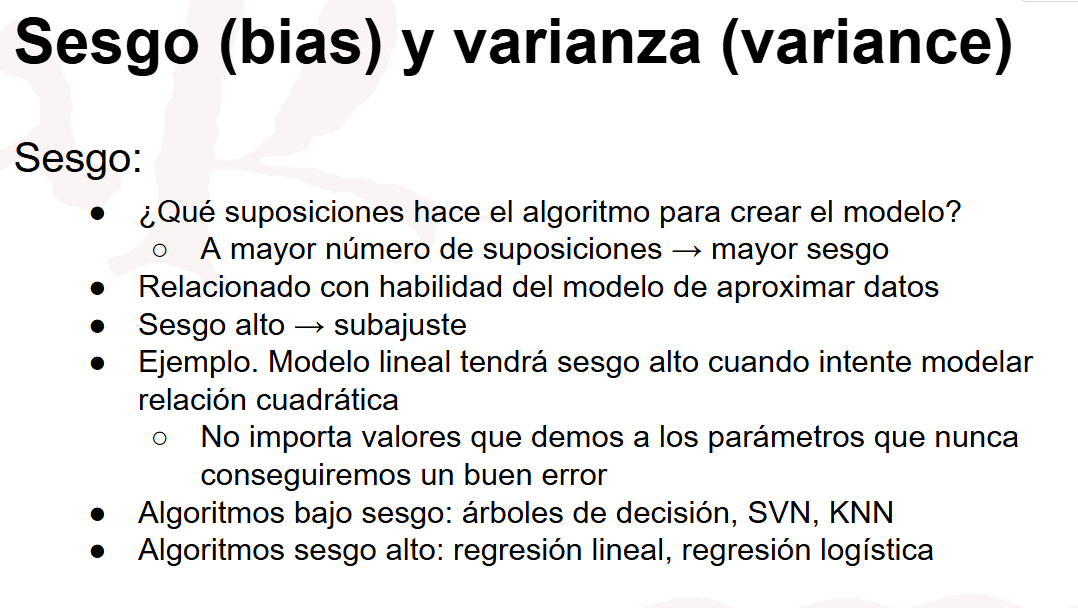
### **Resumen**

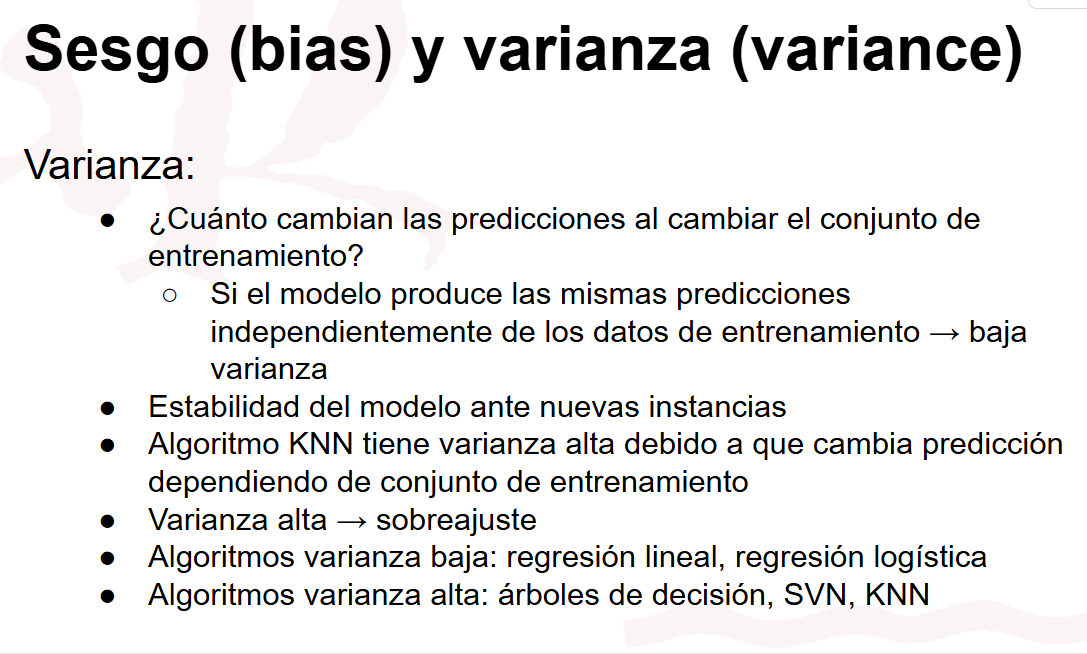
* **Subajuste** ocurre cuando el modelo es demasiado simple y no captura la complejidad de los datos.
* **Sobreajuste** ocurre cuando el modelo es demasiado complejo y captura ruido en los datos.
* Ambos problemas se pueden identificar comparando el rendimiento del modelo en los conjuntos de entrenamiento y prueba.
* Las soluciones incluyen ajustar la complejidad del modelo, usar regularización y mejorar el preprocesamiento de los datos.

En resumen, encontrar el equilibrio entre subajuste y sobreajuste es clave para construir modelos que generalicen bien a datos no vistos

**Ejercicio.**

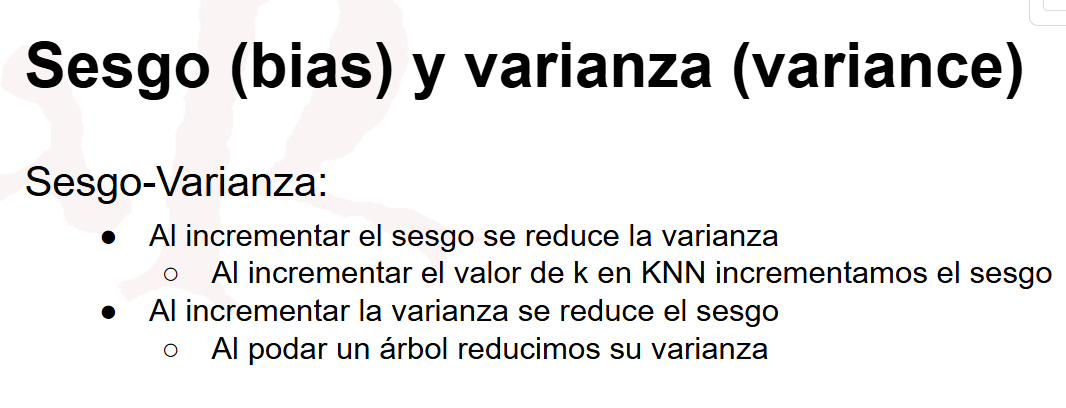
**Explica qué es el sesgo y qué es la varianza.**





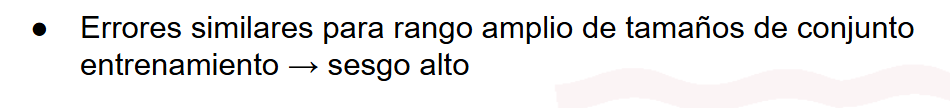
**Ejercicio.**

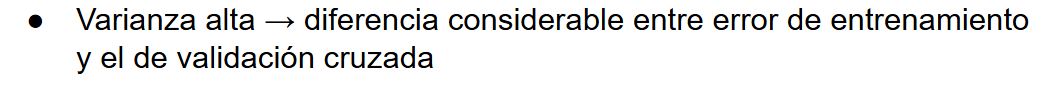
**¿Qué indica un sesgo alto? ¿y una varianza alta? Indica algoritmos que tengan un sesgo bajo y una varianza alta.**



**Ejercicio.**

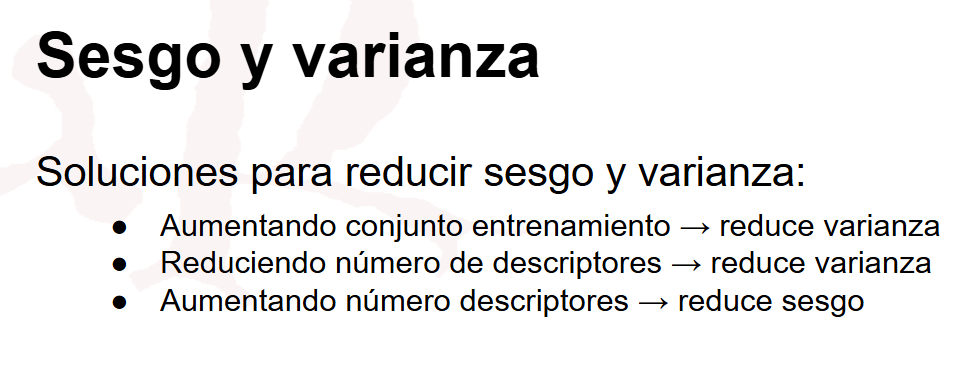
**Utilizando las curvas de validación (o aprendizaje) ¿qué indica que se obtengan errores similares para un rango amplio de tamaños de conjunto de entrenamiento? ¿y que haya una diferencia considerable entre el error de entrenamiento el de validación cruzada?**





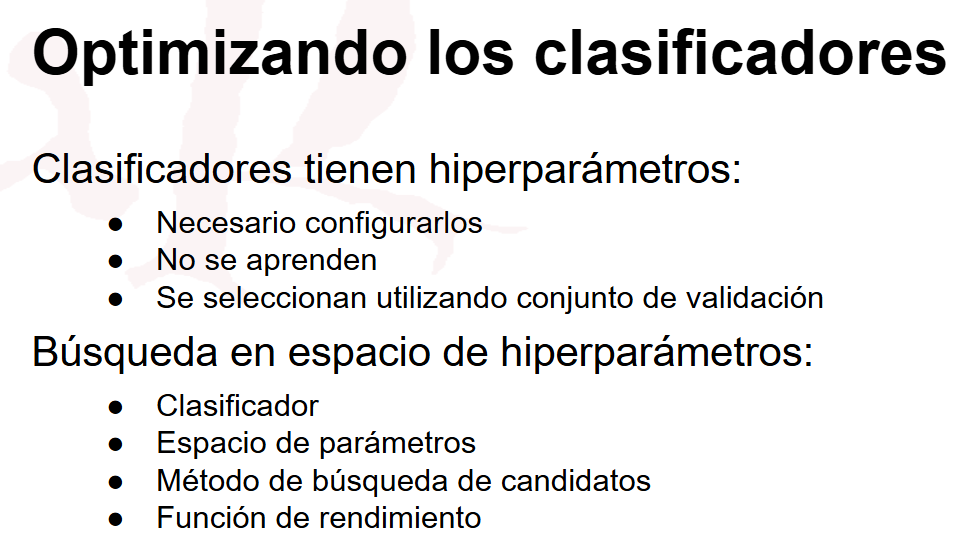
**Ejercicio.**

**¿Cuáles son las soluciones para reducir la varianza y el sesgo?**



**Ejercicio.**

**¿De qué partes consta la búsqueda en el espacio de hiperparámetros?**



**Ejercicio.**

**Explica la diferencias entre buscar hiperparámetros utilizando la técnica de GridSearch y la de RandomSearch.**

**Ejercicio.**

**¿Para qué sirven la normalización y la estandarización?**

### **Resumen**

* **Grid Search** es exhaustivo pero costoso computacionalmente, ideal para espacios de búsqueda pequeños.
* **Random Search** es más eficiente en espacios grandes y puede encontrar buenas combinaciones con menos evaluaciones, aunque no garantiza optimalidad.

En la práctica, muchas veces se combinan ambas técnicas: se usa **Random Search** para explorar el espacio de búsqueda rápidamente y luego se refina con **Grid Search** en áreas prometedoras.

**Ejercicio.**

**Explica la metodología CRISP-DM.**

**Ejercicio.**

**Explica la metodología OSEMN.**