Contenido

1. Introducción al algoritmo k-NN.

El algoritmo **k-Nearest Neighbors (k-NN)** es uno de los métodos más **sencillos, intuitivos y fáciles de comprender** dentro del **aprendizaje supervisado**. Su funcionamiento se basa en una idea muy cercana al razonamiento humano:

> Un dato nuevo se clasifica observando qué ocurre con los datos más parecidos que ya conocemos.

En lugar de construir un modelo matemático complejo, k-NN utiliza directamente los **ejemplos almacenados** para realizar predicciones.

A pesar de su simplicidad, k-NN puede ofrecer **buenos resultados en determinados contextos**, especialmente cuando:

- El conjunto de datos no es muy grande.
- Las clases están bien definidas.
- Las características representan correctamente la similitud entre los datos.

Por este motivo, k-NN es muy utilizado como **algoritmo introductorio** y como **punto de comparación** con modelos más avanzados.

2. k-NN como algoritmo de aprendizaje supervisado.

k-NN es un algoritmo de **aprendizaje supervisado**, lo que implica que:

- Trabaja con **datos etiquetados**.
- Cada ejemplo del conjunto de entrenamiento está formado por:
  - Un conjunto de **características de entrada**.
  - Una **etiqueta de salida conocida**.

Además, k-NN es un algoritmo **versátil**, ya que puede aplicarse a:

- **Problemas de clasificación** → la salida es una clase.
- **Problemas de regresión** → la salida es un valor numérico.

3. Característica principal: aprendizaje basado en instancias (Lazy Learning)

A diferencia de algoritmos como **SVM** o **árboles de decisión**, k-NN pertenece a la familia de algoritmos de **aprendizaje basado en instancias**, también conocidos como **lazy learning**.

Esto significa que:

- **No construye un modelo explícito** durante el entrenamiento.
- El "entrenamiento" consiste únicamente en **almacenar los datos**.
- Todo el esfuerzo computacional se realiza en el momento de la **predicción**.

Mientras otros algoritmos "aprenden antes y predicen rápido", k-NN "aprende poco y predice lento".

4. Funcionamiento básico del algoritmo k-NN

El funcionamiento de k-NN sigue una secuencia clara de pasos:

1. Se recibe un **nuevo ejemplo** sin etiqueta.
2. Se calcula la **distancia** entre ese ejemplo y todos los ejemplos del conjunto de entrenamiento.
3. Se seleccionan los **k ejemplos más cercanos**.
4. Se toma una decisión en función de esos vecinos:
    - **Clasificación:** se asigna la clase más frecuente.
    - **Regresión:** se calcula la media de los valores.

Este proceso se repite **cada vez que se quiere realizar una predicción**.

5. Elección del parámetro k

El parámetro **k** indica el número de vecinos que se tendrán en cuenta para realizar la predicción y tiene una influencia directa en el comportamiento del modelo.

5.1. Valores pequeños de k

Ejemplo: **k = 1**

- El modelo toma la decisión basándose en un único vecino.
- Muy sensible al ruido y a valores atípicos.
- Alto riesgo de **sobreajuste (overfitting)**.

5.2. Valores grandes de k

- El modelo se vuelve más estable.
- Se reduce la influencia del ruido.
- Puede perder información local importante.
- Riesgo de **subajuste (underfitting)** si k es demasiado grande.

En la práctica, el valor óptimo de **k** se determina mediante **validación**, probando distintos valores y evaluando el rendimiento.

6. Medidas de distancia

Para determinar qué ejemplos son los más cercanos, k-NN utiliza una **métrica de distancia** que mide la similitud entre ejemplos.

Distancia Euclídea

Es la métrica más utilizada:

$$d(x, x') = \sqrt{\sum_{i=1}^{n}(x_i - x'_i)^2}$$

Representa la distancia "en línea recta" entre dos puntos en el espacio.

Otras métricas de distancia

- **Manhattan:** suma de diferencias absolutas.
- **Minkowski:** generalización de varias distancias.
- **Chebyshev:** distancia máxima en alguna dimensión.

La elección de la distancia depende del:

- Tipo de datos.
- Distribución.
- Problema a resolver.

7. Importancia del escalado de características

k-NN es **extremadamente sensible a la escala de los datos**.

Ejemplo:

- Edad: 0–100
- Ingresos: 1.000–100.000

Si no se escalan:

- La variable con valores mayores domina la distancia.
- El cálculo deja de reflejar la similitud real.
- El algoritmo produce predicciones incorrectas.

Por ello, es **obligatorio aplicar escalado o normalización** antes de usar k-NN:

- **StandardScaler** (media 0, desviación 1).
- **MinMaxScaler** (valores entre 0 y 1).

8. k-NN para clasificación

En problemas de **clasificación**:

- Cada uno de los k vecinos "vota" por su clase.
- La clase con mayor número de votos es la predicción final.

Ejemplo:

- k = 5
- 3 vecinos → clase A
- 2 vecinos → clase B
- Predicción final: **clase A**.

9. k-NN para regresión

En problemas de **regresión**:

- Se calcula la **media** de los valores asociados a los k vecinos más cercanos.

Ejemplo:

- Valores vecinos: 200, 220, 210
- Predicción final: **210**.

10. Ventajas del algoritmo k-NN

- Muy sencillo de entender e implementar. No requiere entrenamiento complejo.
- Flexible: válido para clasificación y regresión.
- Buen rendimiento en conjuntos de datos pequeños.
- Útil como algoritmo de referencia.

11. Limitaciones del algoritmo k-NN

- Muy lento en la fase de predicción.
- Requiere almacenar todo el conjunto de datos.
- Muy sensible al ruido.
- Extremadamente dependiente del escalado.
- Poco adecuado para datasets grandes.

k-NN es un algoritmo simple pero potente cuando se utiliza en el contexto adecuado. Es ideal para comprender los fundamentos del aprendizaje supervisado, aunque en problemas grandes o complejos suele ser sustituido por modelos más eficientes como SVM o árboles de decisión.