Algoritmo K Vecinos más cercanos (K-NN)

La técnica **de K vecinos más cercanos (más conocida como K-NN por las siglas en inglés de K-Nearest Neighbors)** ha sido ampliamente utilizada en gran variedad de trabajos de investigación para dar solución a problemas de clasificación y regresión (Nguyen, Do, Le, Le, & Benjapolakul, 2019).

Esta técnica se basa en el uso de alguna **métrica de similitud/distancia** para medir la diferencia o similitud entre los registros de una instancia de datos. La selección de la métrica de similitud depende en gran medida del tipo de datos con los que se represente a los distintos registros en el problema a tratar. En la mayoría de las ocasiones existe una gran diversidad de métricas que pueden ser aptas para emplearse en el **K-NN**, siendo una de las más comunes la **distancia euclidiana**.

Dado un registro $\boldsymbol{x}$ de clase desconocida, se procede a calcular la distancia entre $\boldsymbol{x}$ y todos los registros de la instancia de datos de entrenamiento. Finalmente se asigna la clase determinada por los $\boldsymbol{K}$ registros más cercanos a $\boldsymbol{x}$ (Rajaguru & Prabhakar, 2017).

K-NN es un ejemplo típico de aprendizaje lento que almacena datos de entrenamiento en el momento del entrenamiento y retrasa su aprendizaje hasta el momento de la clasificación. A pesar de esto, **K-NN** ha sido activamente utilizado como clasificador durante décadas (L. Wang, Jiao, Shi, Lu, & Liu, 2006).

El algoritmo del **K-NN** consta de dos fases:

- **Fase de entrenamiento**. Los registros de entrenamiento son vectores (cada uno con asignado a una clase) en un espacio de características multidimensionales. En esta fase, se almacenan los vectores de características y las etiquetas de clase de las muestras de entrenamiento.

- **Fase de clasificación**. El usuario define un valor para la constante $\boldsymbol{K}$, un vector sin etiquetar se clasifica asignando una etiqueta, esta es obtenida a partir de la clase más recurrente entre los $\boldsymbol{K}$ registros de entrenamiento más cercanos al vector sin etiquetar. Esta forma de clasificar a un vector de entrada en función de su distancia a los registros de entrenamiento es una forma simple pero efectiva de clasificar nuevos registros.

Descripción detallada del Algoritmo:

1.- Cargar la instancia de datos de un archivo a alguna estructura de datos

Ejemplo de la estructura de un registro:

$Registro_1 = [C_1, C_2, C_3, \dots, C_n, Clase]$

Donde:

$C_i = Una\ de\ las\ características\ del\ registro$
$Clase = Clase\ asignada\ al\ registro$

2. Generar/Establecer/Definir a un registro "j" para ser clasificado

3.- Establecer el valor de $\boldsymbol{K}$. Donde $\boldsymbol{K}$ puede ser cualquier valor entero entre **1** y el total de registros inclusive.

4.- Para cada registro "i" en la instancia de datos:

4.1.- Calcular la distancia entre el registro "j" y el registro "i"

Nota. Recordar que la métrica de similitud usada para el cálculo de la distancia depende el problema a tratar. La distancia más común, es la distancia Euclidiana.

4.2.- Añadir a la distancia obtenida junto al registro "i" a una estructura de datos

5.- Ordenar la estructura de datos generada en el punto 4, de acuerdo con la métrica de similitud utilizada. De tal forma que se encuentre en el primer índice el registro más similar al registro "j" y en el último índice al registro menos similar.

Nota. En la mayoría de las métricas de similitud un registro es más similar a otro si cuenta con una menor distancia y viceversa.

6. Recuperar de la estructura de datos ordenada los primeros $\boldsymbol{K}$ registros

7. Si el problema es de regresión, se devolverá la media de las etiquetas de los $\boldsymbol{K}$ registros

8. Si el problema es de clasificación, se devolverá la moda de las etiquetas de los $\boldsymbol{K}$ registros