¿Qué son Los vecinos más próximos - kNN(K-Nearest Neighbor)?

Los vecinos más cercanos (kNN, por sus siglas en inglés de "k-nearest neighbors") es un algoritmo de aprendizaje supervisado utilizado en el campo de la inteligencia artificial y el machine learning.

El algoritmo kNN se basa en la idea de que los objetos que son similares están cercanos en un espacio n-dimensional. El objetivo del algoritmo kNN es clasificar nuevos puntos de datos basados en los puntos de datos existentes que están más cercanos a ellos en términos de distancia euclidiana.

En el proceso de entrenamiento del modelo kNN, el algoritmo calcula la distancia entre cada punto de datos y los demás puntos de datos en el conjunto de entrenamiento. Cuando se recibe un nuevo punto de datos, el algoritmo busca los k puntos de datos más cercanos a él y clasifica el nuevo punto de datos según la etiqueta (clase) más común de los k vecinos más cercanos.

El valor de k es un hiperparámetro del algoritmo y se selecciona de acuerdo con la complejidad del problema y el tamaño del conjunto de datos. El algoritmo kNN es simple y fácil de implementar, pero su eficacia puede verse afectada por la elección del valor de k y la dimensión de los datos.

¿Dónde se aplica k-Nearest Neighbor?

Aunque sencillo, se utiliza en la resolución de multitud de problemas, como en **sistemas de recomendación, búsqueda semántica y detección de anomalías**.

Pros y contras

Como **pros** tiene sobre todo que es sencillo de aprender e implementar. Tiene como **contras** que *utiliza todo el dataset* para entrenar “cada punto” y por eso requiere de uso de mucha memoria y recursos de procesamiento (CPU). Por estas razones kNN tiende a funcionar mejor en datasets pequeños y sin una cantidad enorme de features (las columnas).

¿Cómo funciona kNN?

1. Calcular la distancia entre el item a clasificar y el resto de items del dataset de entrenamiento.
2. Seleccionar los "k" elementos más cercanos (con menor distancia, según la función que se use)
3. Realizar una "votación de mayoría" entre los k puntos: los de una clase/etiqueta que <<dominen>> decidirán su clasificación final.

Teniendo en cuenta el punto 3, veremos que para decidir la clase de un punto **es muy importante el valor de k**, pues este terminará casi por definir a qué grupo pertenecerán los puntos, sobre todo en las "fronteras" entre grupos. Por ejemplo -y a priori- yo elegiría valores impares de k para desempatar (si las features que utilizamos son pares). No será lo mismo tomar para decidir 3 valores que 13. Esto no quiere decir que necesariamente tomar más puntos implique mejorar la precisión. Lo que es seguro es que *cuantos más "puntos k", más tardará nuestro algoritmo en procesar* y darnos respuesta 😉

Las formas más populares de "medir la cercanía" entre puntos son la **distancia Euclidiana** (la "de siempre") o la **Cosine Similarity** (mide el ángulo de los vectores, cuanto menores, serán similares). Recordemos que este algoritmo -y prácticamente todos en ML- funcionan mejor con varias características de las que tomemos datos (las columnas de nuestro dataset). Lo que entendemos como "distancia" en la vida real, quedará abstracto a muchas dimensiones que no podemos "visualizar" fácilmente (como por ejemplo en un mapa).