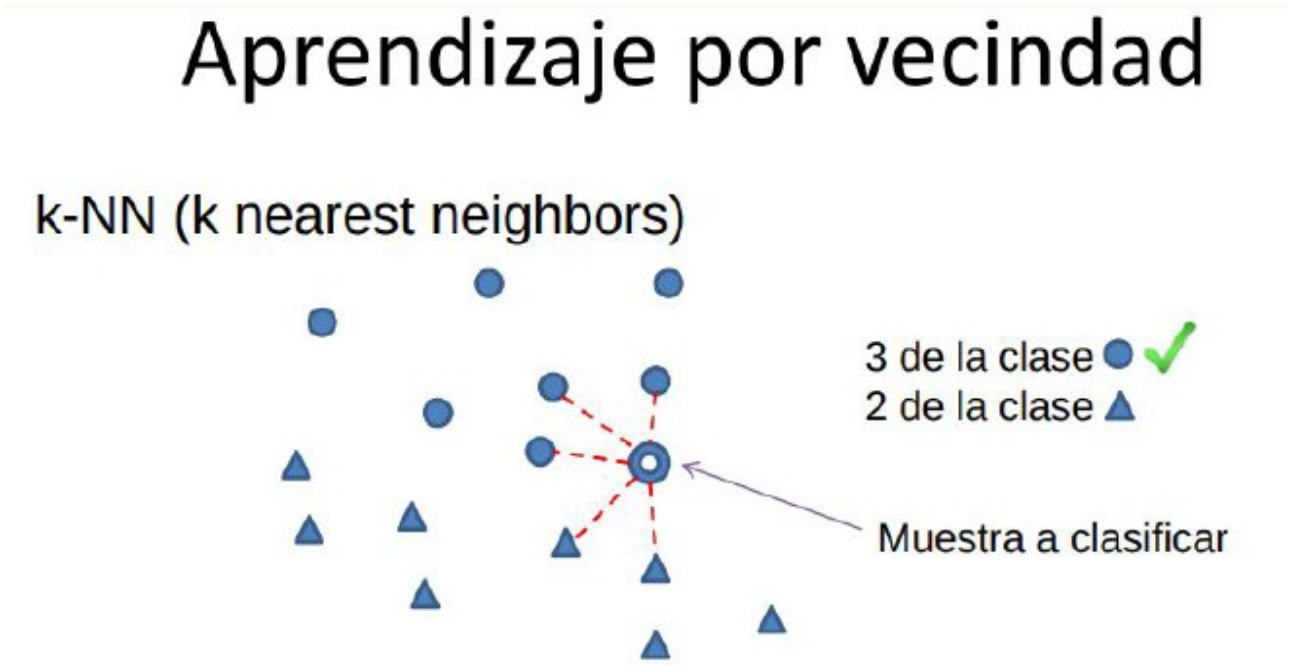
K-NN (K Nearest Neighbor) como algoritmo de clasificación supervisada

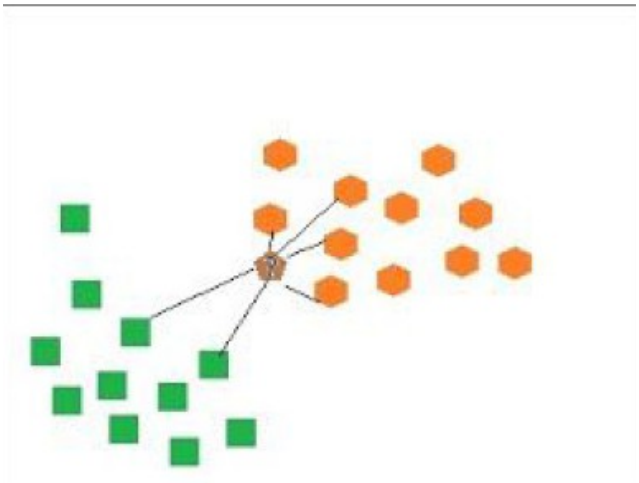
Significa K-Nearest-Neighbor por sus siglas en ingles, también llamado **algoritmo del vecino más cercano**. Este método sirve para clasificar nuestra información y obtener predicciones

También llamado **aprendizaje por vecindad.** Las reglas de clasificación por vecindad están basadas en la búsqueda en un conjunto de prototipos de los k prototipos más cercanos al patrón a clasificar. Las predicciones se realizan basándose en los ejemplos más parecidos al que hay que predecir.



Básicamente se basa en tomar la información que deseamos clasificar, calcular las distancias de todos los miembros de la base de datos (comúnmente un conjunto de entrenamiento) y se toma un número particular de vecinos (K) que son los que tuvieron distancias menores, luego se visualiza la clase de los mismos, de allí se determina que tipo es el dato que estamos buscando.

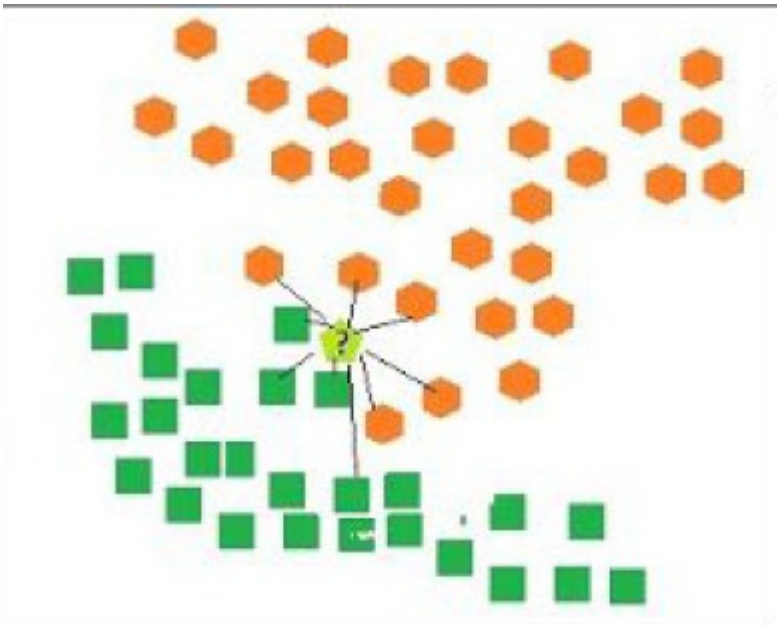
La siguiente imagen muestra un ejemplo muy claro de la idea de este algoritmo:



En la imagen tenemos dos figuras, los cuadrados y los hexágonos, después intentamos clasificar un tercer elemento (un pentágono), calculamos las distancias de todos sus vecinos y bueno , nos damos cuenta que tiene más cercanía con un hexágono, entonces KNN diría la nueva figura es un hexágono, o al menos es probable que lo sea.

Que distancia es más precisa para KNN, bueno, eso depende de nuestros datos, si tenemos atributos como ingresos mensuales, gastos mensuales, deudas, otros ingresos, entonces posiblemente una **distancia euclídea** sería la respuesta correcta, pero si tenemos estatura, peso, edad, número de hijos, ingreso mensual, entonces tenemos datos que por su número opacarían a los demás (a decir ingreso mensual estaría en miles) entonces una **distancia de Mahalanobis** sería una mejor aproximación.

Bien hasta este punto todo bien, pero no siempre ocurre así, qué pasa si tenemos las mismas figuras, pero con la siguiente disposición:

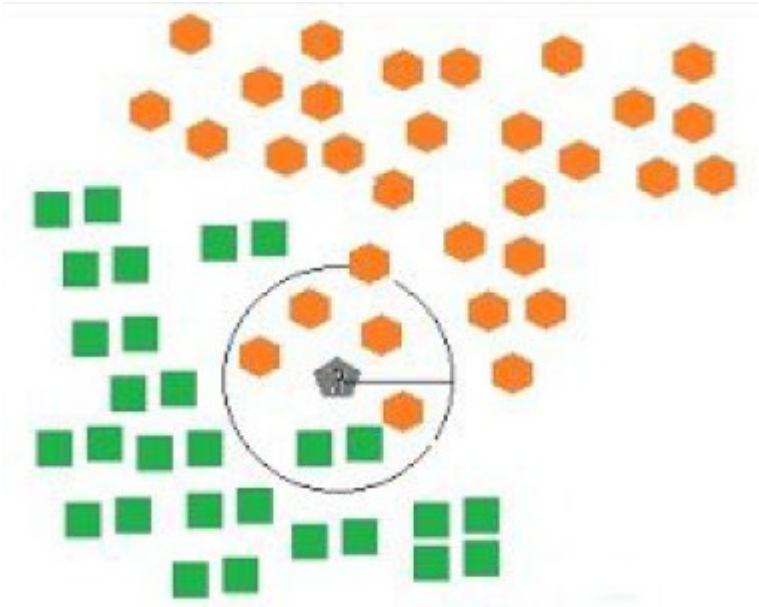


Si K = 3 , es decir, buscamos los tres vecinos más cercanos, clasificaríamos a nuestra figura como un cuadrado, lo mismo ocurrirìa si K = 4 o K = 5,pero cuando K = 6 entonces tendríamos un empate (cuando esto ocurre he visto que lo que se suele realizar es sumar las distancias y tomar la suma menor) y si K = 8, entonces KNN nos diría que es un hexágono.

Es por ello que con este tipo de algoritmos es exageradamente necesario conocer bien la información y como parte medular decidir el tipo de cálculo para las distancias . Y el número de vecinos a considerar.

Otro punto importante es el factor de búsqueda, ya que, sería muy costoso (de hecho KNN es un algoritmo costoso computacionalmente hablando) el calcular la distancia de toda nuestra base, más si tenemos millones de registros, es por ello que suelen utilizarse ciertas heurísticas para determinar con qué elementos realizar la comparación, el más utilizado es el de determinar elementos clave (conjuntos de entrenamiento), es decir, si tenemos 1 millón de registros de tipos de clientes y dos clases a decir, “se acepta préstamo ” y “se rechaza préstamo”, entonces cuando llega un posible cliente y le pedimos los datos,no sería práctico calcular la distancia con el millón de clientes, en vez de ello, podemos tener unos mil clientes que representan bien a su clase y con ellos calcular las distancias.

Otra **heurística** común es establecer un rango de búsqueda, aquí usualmente se propone una distancia máxima, es decir, supongamos que la distancia que queremos calcular es una distancia euclídea, entonces, queremos que nuestros vecinos no pasen de cierta distancia:



Entre las principales **características** de este clasificador podemos destacar:

- No requiere la construcción de un modelo.
- Trabaja con los datos originales (no requiere representaciones especiales de los mismos).
- Realiza su predicción en base a información local (esto cuando se aplican heurísticas de búsqueda).
- Puede producir resultados equivocados si el cálculo de distancia no es el adecuado.
- En general, mientras K sea mayor (sin ser esto una regla), es decir, mientras más vecinos sean tomados en cuenta, la probabilidad de una predicción correcta aumenta.

Algoritmo

Tenemos tres conjuntos de datos, el conjunto de entrenamiento que utilizaremos para el aprendizaje del clasificador y los conjuntos de validación y de test con los que comprobaremos si el clasificador es capaz de **generalizar** , es decir **si presenta buenos resultados al introducir datos no empleados durante el entrenamiento.**

A continuación describiremos los pasos seguidos para la implementación de este algoritmo:

El proceso de aprendizaje de este clasificador consiste en almacenar en un vector el conjunto de entrenamiento, junto a la clase asociada a cada muestra de este conjunto.

En primer lugar, y con motivo del aprendizaje del algoritmo,calcularemos la distancia euclídea de cada muestra de entrenamiento, a todas las demás que tenemos almacenadas en el vector del punto anterior y de las que conocemos la clase a la que corresponden, quedándonos con las K muestras más cercanas y clasificando la nueva muestra de entrenamiento en la clase más frecuente a la que pertenecen los K vecinos obtenidos anteriormente.

La segunda tarea para diseñar el clasificador, es realizar el mismo proceso con los datos de validación Se calcula el porcentaje de clasificación sobre los ejemplos de este conjunto(desconocidos en la tarea de aprendizaje) para conocer su poder de generalización.