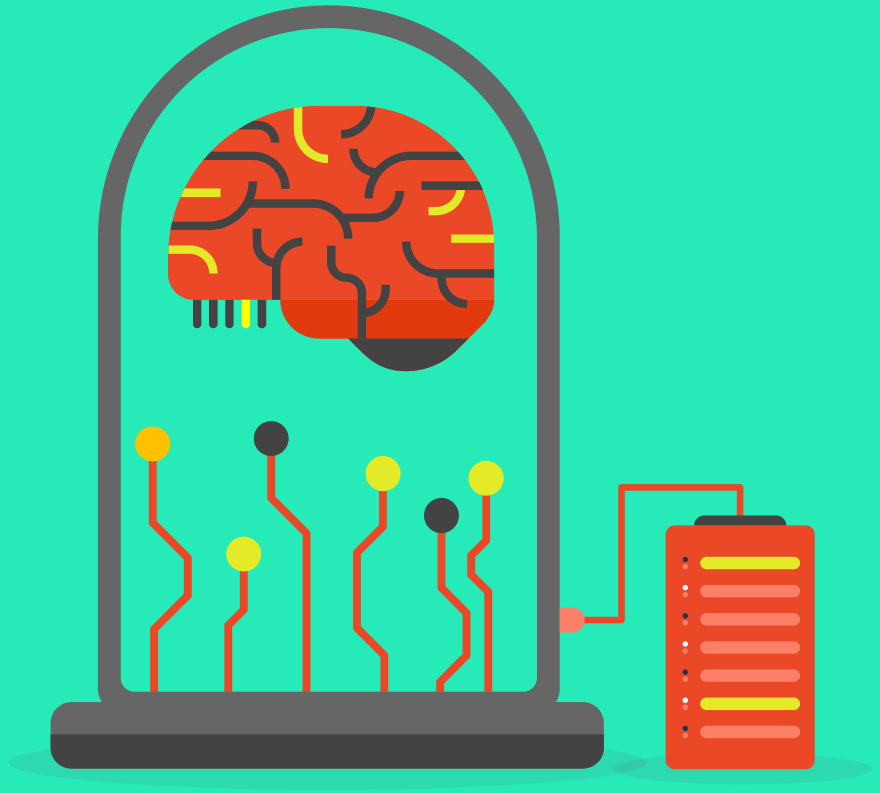
12. Clasificación por K Vecinos Cercanos



Algoritmo de los K Vecinos Cercanos

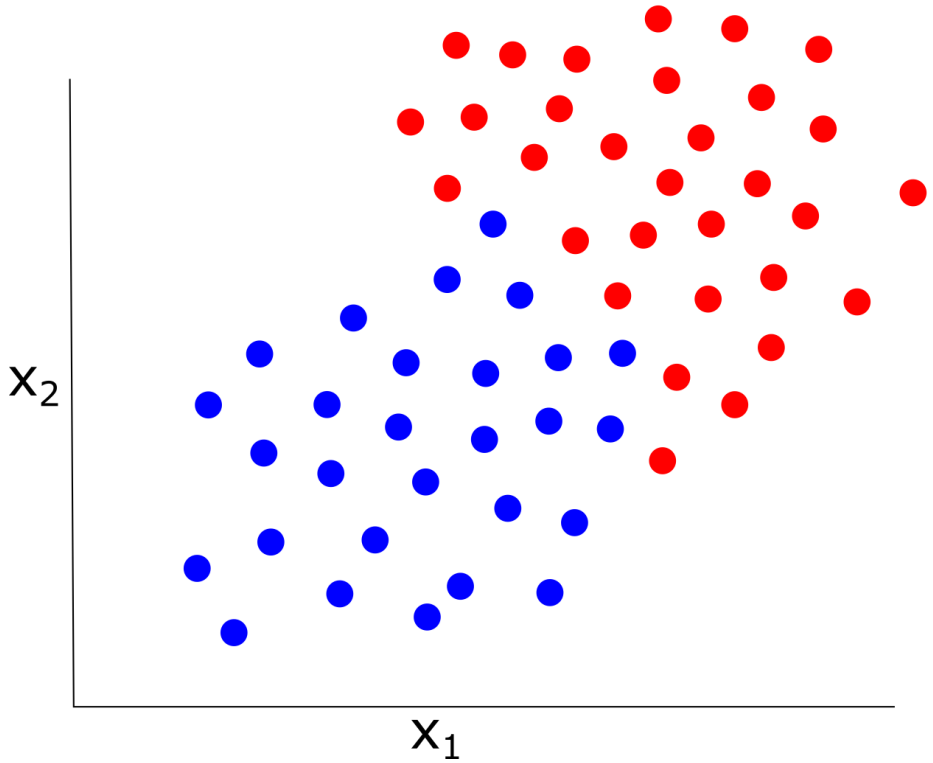
El algoritmo de los **K Vecinos Cercanos** (K Nearest Neighbors, o KNN por sus siglas en inglés) es uno de los algoritmos supervisados más conocidos dentro del Machine Learning, y que puede emplearse tanto para clasificación como para regresión.

Se trata de un tipo de algoritmo que suele referirse también como “vago” (lazy) pues en realidad el mismo no ejecuta un aprendizaje (por ejemplo, basado en la minimización de una función de costo) sino que, simplemente, el algoritmo asigna al conjunto de prueba una clase en función de la mayoría de datos “vecinos” presentes en el conjunto de entrenamiento.

El único parámetro que posee este algoritmo es el valor de K, que hace referencia a la cantidad de vecinos que seleccionaremos para tomar una decisión de clasificación.

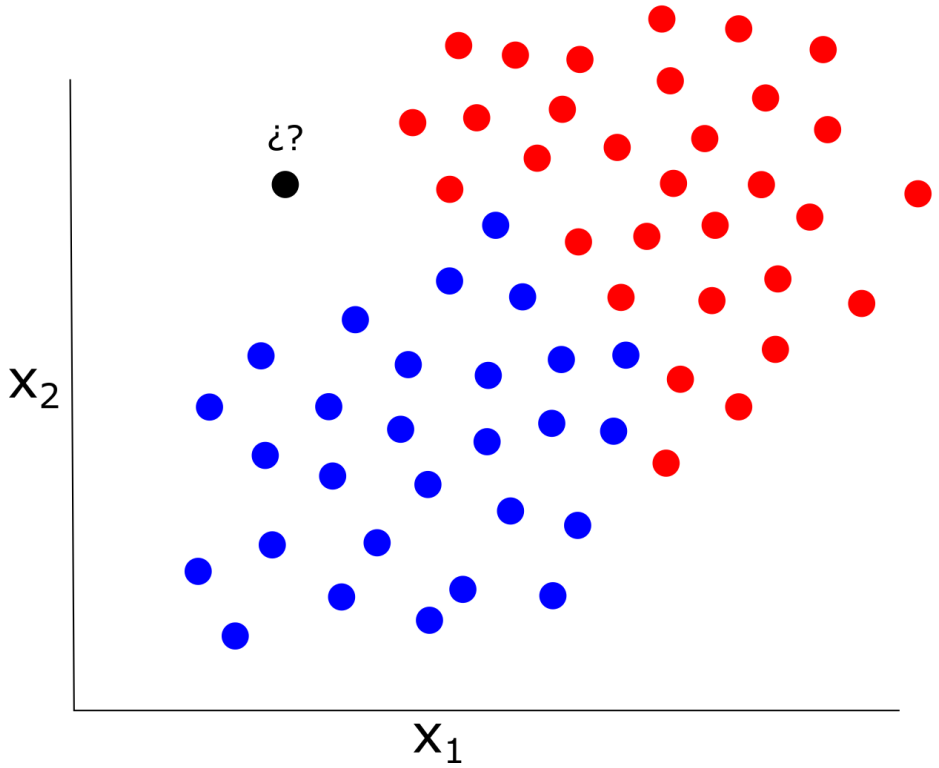
Algoritmo de los K Vecinos Cercanos

Veamos un ejemplo gráfico:



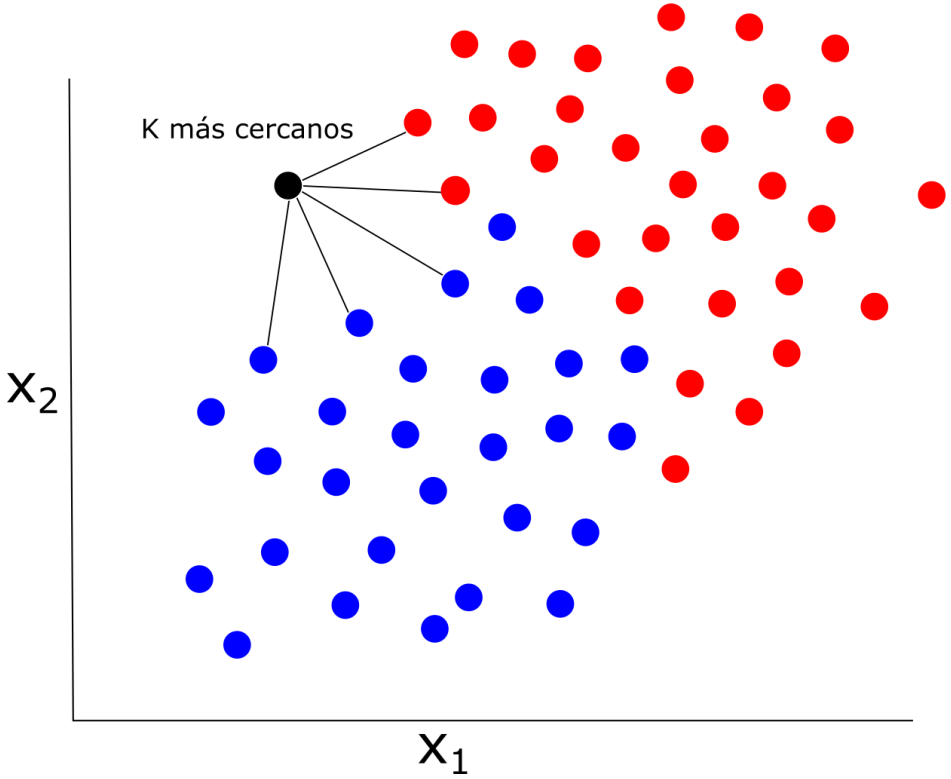
Algoritmo de los K Vecinos Cercanos

Dado un nuevo dato como: ¿a qué clase correspondería?



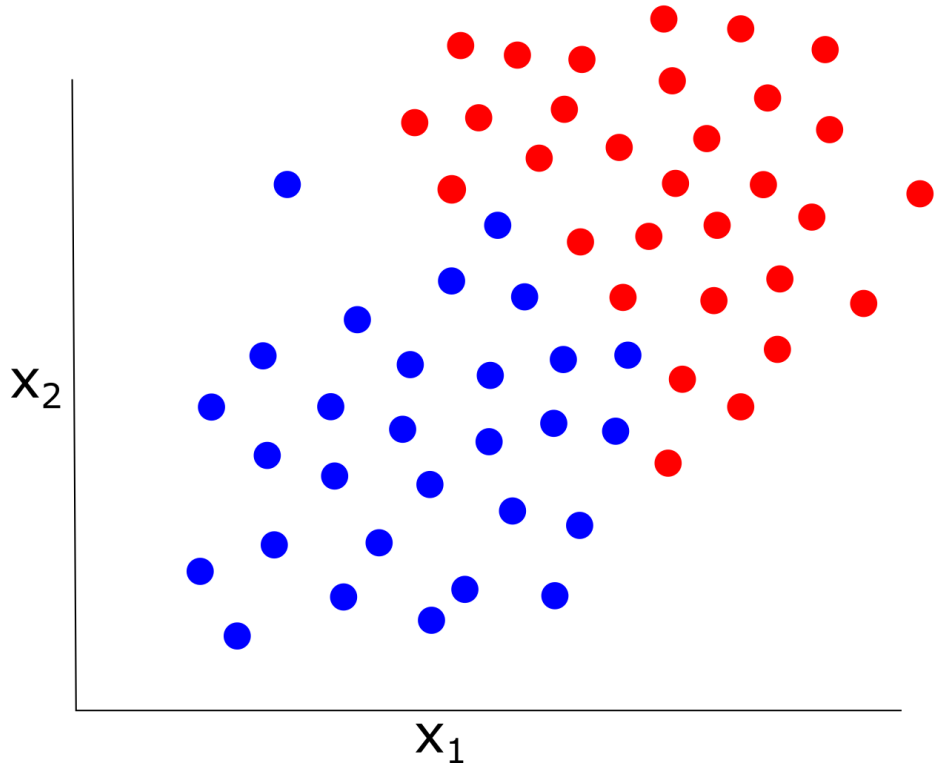
Algoritmo de los K Vecinos Cercanos

Determina cuáles son los K vecinos más cercanos (distancia):



Algoritmo de los K Vecinos Cercanos

Asigna la nueva clase al voto de la mayoría (moda):



Pseudo código de los K Vecinos Cercanos

1. Cargar todos los datos de entrenamiento.
2. Seleccionar un valor de K.
3. Para un nuevo punto a predecir (que no forma parte del conjunto de entrenamiento), calcular la **distancia** entre dicho punto y todos los demás puntos del conjunto de entrenamiento. Dicha distancia puede ser, por ejemplo, la distancia euclídea (pero pueden usarse otras métricas de distancia).
4. Ordenar las distancias (y puntos correspondientes) de menor a mayor.
5. Entre todas las distancias calculadas y ordenadas, seleccionar las primeras K.
6. Determinar las clases del conjunto de entrenamiento a la que pertenecen estos K puntos más cercanos.
7. Finalmente, la clase del punto a predecir, será la mayoría (moda) de las clases obtenidas en el paso anterior.

Ventajas del algoritmo de los K Vecinos Cercanos

La principal ventaja de este algoritmo es su sencillez y facilidad para comprender.

A su vez, el mismo es muy versátil y se puede usar con datos tanto lineales como no lineales.

Es posible usarlo también para Regresión, considerando la predicción como el promedio numérico (en vez de la moda) de los K vecinos obtenidos.

Puede llegar a ofrecer mejores resultados que algoritmos más complejos.

No hace falta afinar parámetros ni llevar a cabo optimización de funciones de costo (usando Descenso del Gradiente, por ejemplo).

Desventajas del algoritmo de los K Vecinos Cercanos

Como requiere cargar todo el conjunto de entrenamiento completo, puede tener problemas computacionales de almacenamiento.

La velocidad de cómputo dependerá de la cantidad de registros del conjunto de entrenamiento, por lo que se hará más lento a medida que este crezca.

Ya que con cada nuevo dato a predecir es necesario recalcular todas las distancias, los tiempos de predicción (inferencia) son altos.

Su propia sencillez puede quitarle robustez dependiendo de la naturaleza de los datos.

Importancia del valor de K

Debido a que la decisión final del algoritmo depende del valor de K seleccionado, dicha cantidad puede afectar el funcionamiento del algoritmo:

- Valores altos de K aumentan la confianza en la predicción, pero si este valor es demasiado alto se rompería el propio principio del algoritmo.

- Valores bajos de K harán que el ruido de los datos prevalezca, lo que tiende a generar sobre entrenamiento.

- La selección óptima de K resulta entonces un ejercicio de ensayo y error, aunque se recomienda aplicar **Validación Cruzada** a fin de decidir el valor final.

Décimo Segundo Notebook Práctico

Ya que conocemos la teoría detrás del algoritmo de los K Vecinos Cercanos, veamos cómo se implementa en Python haciendo uso de la librería *Scikit-Learn*.

Décimo Segundo Notebook Práctico