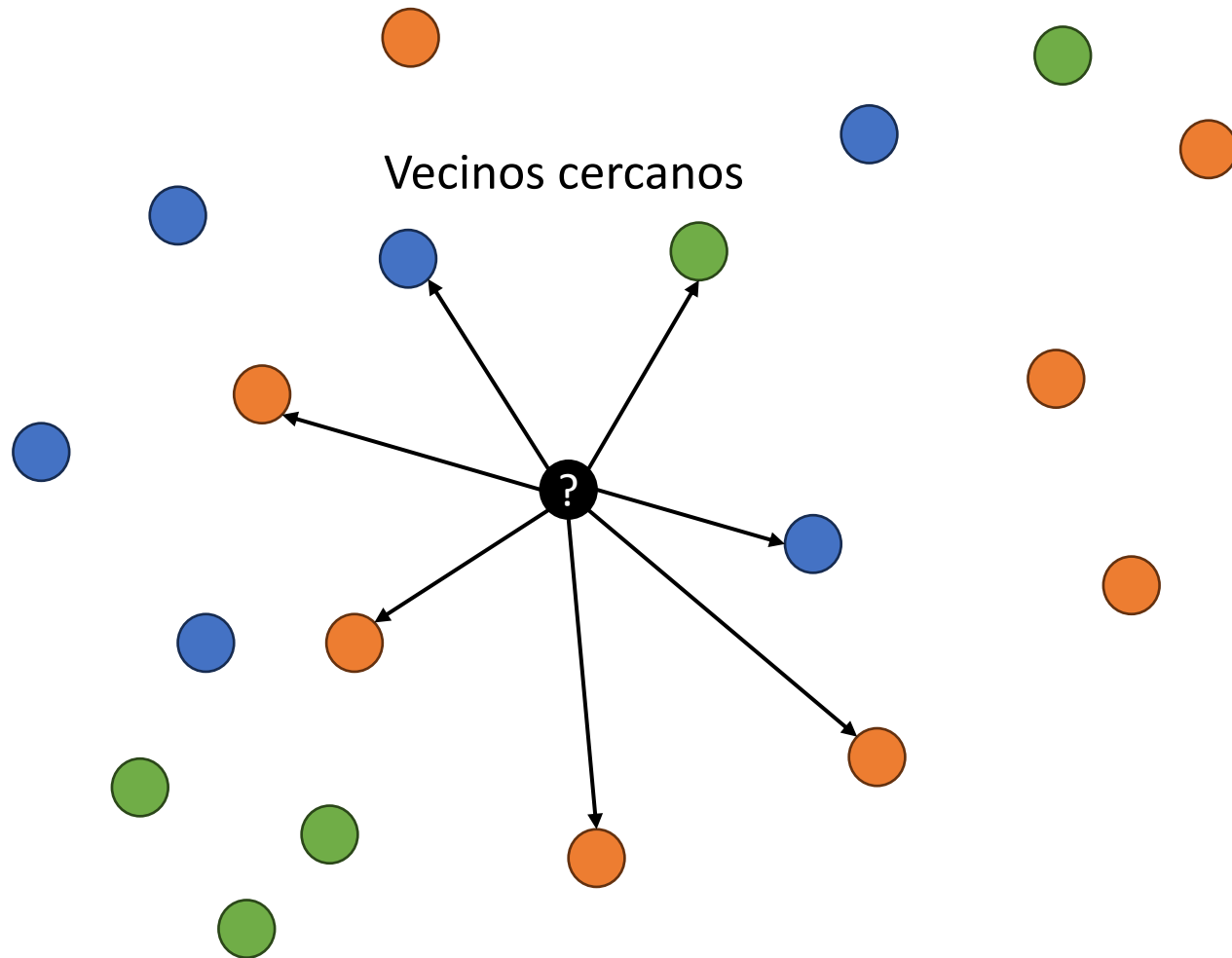




Nearest-Neighbor Methods  
(*Métodos basados en vecinos  
cercanos*)

# Vecinos más cercanos



# K-Nearest Neighbors

Es un algoritmo de aprendizaje supervisado para clasificación y regresión:

- Busca un grupo de  $k$  objetos en el conjunto de datos que son **los más cercanos** a la instancia a clasificar.
- Asigna una clase/número basado en la **dominancia de una clase/valor** en particular **en esta vecindad**.
- Basado en **instancias**: en sí, el algoritmo no aprende un modelo. En su lugar, **utiliza los datos de entrenamiento** para predecir nuevos valores.

# K-Nearest Neighbors

Puntos clave:

1. El **conjunto de datos** que se usan para evaluar la nueva instancia.
2. Una **métrica de distancia** o similitud para medir la cercanía entre instancias.
3. El valor de  $k$ , o el **número de vecinos**.
4. El **método para asignar la clase** a la nueva instancia según la mayoría de los vecinos.

# K-Nearest Neighbors

**Algoritmo** Idea básica de kNN

**Entrada** : Conjunto de datos  $D$ , nuevo punto  $z$ , posibles clases  $L$ .

**Salida** : La clase  $c_z \in L$  a la que pertenece  $z$ .

**Para cada**  $y \in D$ :

Calcular la distancia  $d(z, y)$

**Seleccionar**  $N \subseteq D$ , el conjunto de  $k$  puntos más cercanos a  $z$ , donde

$$c_z = \operatorname{argmax}_{v \in L} \sum_{y \in N} I(v = \text{clase}(c_y))$$

donde  $I(\cdot)$  es la función indicadora.

# K-Nearest Neighbors

- En caso de empate:
  - Elegir al azar.
  - Elegir la clase con mayor frecuencia en el conjunto de datos.
- Necesita almacenar todos los puntos,  $O(n)$ .
- Necesita calcular todas las distancias,  $O(n)$ .
- No necesita entrenar un modelo, por lo que tarda menos al momento de clasificar.

# K-Nearest Neighbors: Distancia

En cuanto a las distancias, se suele usar dos. Para dos puntos  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  (ambos vectores de  $n$  características):

$$d_{\text{Euclidean}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$
$$d_{\text{Manhattan}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^n |x_k - y_k|}$$

# K-Nearest Neighbors: Distancia

Estas distancias nos permiten implementar el principio «*de entre menor distancia entre dos puntos, mayor es la posibilidad de que ambos sean de la misma clase*».

$$d_{\text{Euclidean}}(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$
$$d_{\text{Manhattan}}(x, y) = \sqrt{\sum_{k=1}^n |x_k - y_k|}$$



# K-Nearest Neighbors: Distancia

- Según la definición de distancia (y como medirla), kNN se puede adaptar a distintos problemas, como clasificación de texto.
- La distancia Euclidiana no es perfecta:
  - Suele ser menos selectiva conforme incrementan el número de características de los puntos.
  - Los valores de las características deben escalarse para evitar que una domine otras.

# K-Nearest Neighbors: Regresión

## Actividad

Leer el siguiente capítulo. Realizar una exposición sobre cómo se aplica kNN para regresión:

- Realizar una presentación con lo más importante.
- Resumir el contenido, digerirlo y explicarlo.
- Tienen 30 minutos.

# Tarea

- Investigar sobre el aprendizaje Rote y su uso en el Machine Learning.
- Investigar sobre la similitud coseno, su aplicación en la clasificación de textos y dar un pequeño ejemplo sobre su uso.
- Leer el [siguiente capítulo](#) o [este capítulo](#) sobre kNN.
  - Identificar y resumir los principales problemas que sufre kNN en práctica.
  - Identificar y resumir las ventajas de kNN sobre otros modelos de aprendizaje.

**Ojo:** Buscar fuentes bibliográficas adecuadas para desarrollar su investigación. Cualquier intento de sólo traducir + copiar + pegar, en lugar de **desarrollar el tema de forma adecuada**, hará que **la tarea se considere nula**.



# Fin de la presentación

---

¡Gracias por su atención!