Métricas de distancia en Aprendizaje No Supervisado

1. ¿Qué es el *Clustering*?

El **clustering** es una técnica de aprendizaje no supervisado cuyo objetivo es **encontrar patrones o agrupamientos naturales dentro de los datos**. Es decir, busca agrupar datos que son similares entre sí según cierta medida de "cercanía" o "distancia".

2. ¿Por qué es importante la métrica de distancia?

Para saber si dos datos están cerca o lejos, los algoritmos de clustering usan una **métrica de distancia**.

Pero **no existe una única forma de medir la distancia**: diferentes métricas pueden producir agrupamientos muy distintos.

Principales métricas de distancia

a) Distancia Euclidiana

- **¿Qué es?**

  Es la distancia "recta" entre dos puntos, la más intuitiva.

- **Fórmula (en 2D):**

  Si los puntos son (x_1, y_1) y (x_2, y_2), la distancia es:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- **Ejemplo:**

  Para los puntos (2, 2) y (4, 5):

$$d = \sqrt{(4-2)^2 + (5-2)^2} = \sqrt{4+9} = \sqrt{13} \approx 3.6$$

- **¿Dónde se usa?**

  Es la más usada en clustering tradicional.

---

b) Distancia Chebyshev

- **¿Qué es?**

  También llamada "distancia del tablero de ajedrez".

  Es el **máximo** de las diferencias absolutas entre cada dimensión.

- **Fórmula (en 2D):**

$$d = \max(|x_2 - x_1|, |y_2 - y_1|)$$

- **Ejemplo:**

  Para los puntos (2, 2) y (4, 5):

  - Diferencia en X: $|4 - 2| = 2$
  - Diferencia en Y: $|5 - 2| = 3$
  - Distancia Chebyshev: $\max(2, 3) = 3$

- **¿Por qué tablero de ajedrez?**

  Porque la **pieza Rey** en ajedrez puede moverse en cualquier dirección pero de a un cuadro por vez, y la cantidad mínima de movimientos para llegar de un punto a otro es igual a esta distancia.

---

c) Distancia Manhattan

- **¿Qué es?**

  Es la suma de las diferencias absolutas entre cada dimensión.

  Llamada así por el diseño cuadriculado de Manhattan, donde para llegar de un punto a otro, hay que avanzar por las calles (en línea recta por bloques).

- **Fórmula (en 2D):**

$$d = |x_2 - x_1| + |y_2 - y_1|$$

- **Ejemplo:**

  Para los puntos (2, 2) y (4, 5):

$$|4 - 2| + |5 - 2| = 2 + 3 = 5$$

¿Por qué importa la elección de métrica?

- **El tipo de métrica afecta directamente los clusters** que el algoritmo va a detectar.
- Por ejemplo, usando el algoritmo DBSCAN en datos de ventas inmobiliarias (precio y antigüedad de la propiedad), **cada métrica puede generar diferente número de clusters**, diferente composición, incluso diferentes puntos detectados como ruido.
- No existe una regla universal: **la mejor métrica depende de los datos y el problema**.

Resumen visual

Métrica	Fórmula	Ejemplo (2,2)-(4,5)	Resultado
Euclidiana	$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$	$\sqrt{13}$	3.6
Chebyshev	$\max(\lvert x_2 - x_1\rvert, \lvert y_2 - y_1\rvert)$	$\max(2, 3)$	3
Manhattan	$\lvert x_2 - x_1\rvert + \lvert y_2 - y_1\rvert$	$\lvert 2\rvert + \lvert 3\rvert$	5

¿Cómo elegir la mejor métrica?

- **Experimentación:** No hay fórmula mágica, hay que probar varias métricas y ver cuál funciona mejor para tus datos y tu problema.
- **Reglas generales:**
  - Euclidiana: Para datos numéricos y escalas comparables.
  - Manhattan: Suele funcionar bien con datos categóricos o cuando hay muchas dimensiones.
  - Chebyshev: Casos muy específicos, por ejemplo, movimientos tipo “rey” en ajedrez o problemas de logística.
- **Conclusión:** ¡Experimenta y analiza! La métrica correcta es la que mejor representa la “cercanía” en tu contexto.

¿Por qué cambiar la métrica de distancia?

- Porque **diferentes métricas de distancia pueden encontrar patrones diferentes** en los datos.

- Elegir la métrica adecuada puede mejorar la calidad de los clusters o agrupaciones.

¿Cómo se cambia la distancia?

1. Identifica el algoritmo

No todos los algoritmos permiten cambiar la métrica.

Algunos que sí lo permiten son:

- **KMeans** (en su versión clásica usa siempre Euclidiana, pero existen variantes)
- **DBSCAN**
- **Agglomerative Clustering** (clustering jerárquico)
- **KNeighbors** (para búsquedas de vecinos más cercanos)
- **Mean Shift, Spectral Clustering, etc.** (depende)

---

2. Usa el parámetro adecuado

Casi siempre el parámetro es `metric`, `affinity` o `distance` al crear el modelo.

Ejemplo en **Scikit-learn (Python):**

a) DBSCAN

```
from sklearn.cluster import DBSCAN

# Cambiar la métrica a 'manhattan'
db = DBSCAN(eps=0.5, min_samples=5, metric='manhattan')
db.fit(X)
```

b) Agglomerative Clustering

```
from sklearn.cluster import AgglomerativeClustering

# Cambiar la afinidad (métrica) a 'chebyshev'
cluster = AgglomerativeClustering(n_clusters=3, affinity='chebyshev', linka
```

```
ge='complete')
cluster.fit(X)
```

En versiones recientes, `affinity` *a veces se llama* `metric`.

c) KNeighborsClassifier o KNeighborsRegressor

```
from sklearn.neighbors import KNeighborsClassifier

# Usar distancia 'cosine'
knn = KNeighborsClassifier(n_neighbors=3, metric='cosine')
knn.fit(X_train, y_train)
```

d) KMeans

- El KMeans clásico **solo usa Euclidiana**.
- Si quieres cambiar la métrica, debes usar variantes como **KMedoids** (`scikit-learn-extra`) o programar tu propio método.

---

3. Métricas disponibles

Algunos nombres de métricas que puedes usar:

- `'euclidean'`
- `'manhattan'`
- `'chebyshev'`
- `'minkowski'`
- `'cosine'`
- `'hamming'`
- `'precomputed'` (si tú mismo calculas las distancias y le pasas la matriz)

Consulta la documentación de Scikit-learn sobre métricas para más detalles.

Resumen paso a paso

1. **Elige el algoritmo** que permita personalizar la distancia.

2. **Pasa el nombre de la métrica** que quieres usar al parámetro correspondiente (`metric`, `affinity`, etc.).
3. **Ajusta los hiperparámetros** y entrena normalmente.