Ciencia de Datos

2025

Introducción a la Ciencia de Datos

OPTATIVA - LICENCIATURA EN INFORMÁTICA
FACET-UNT

Ciencia de Datos: Fundamentos y Herramientas

CURSO DE POSTGRADO - FACET-UNT

CD2023

Adquisición

Pre-procesamiento

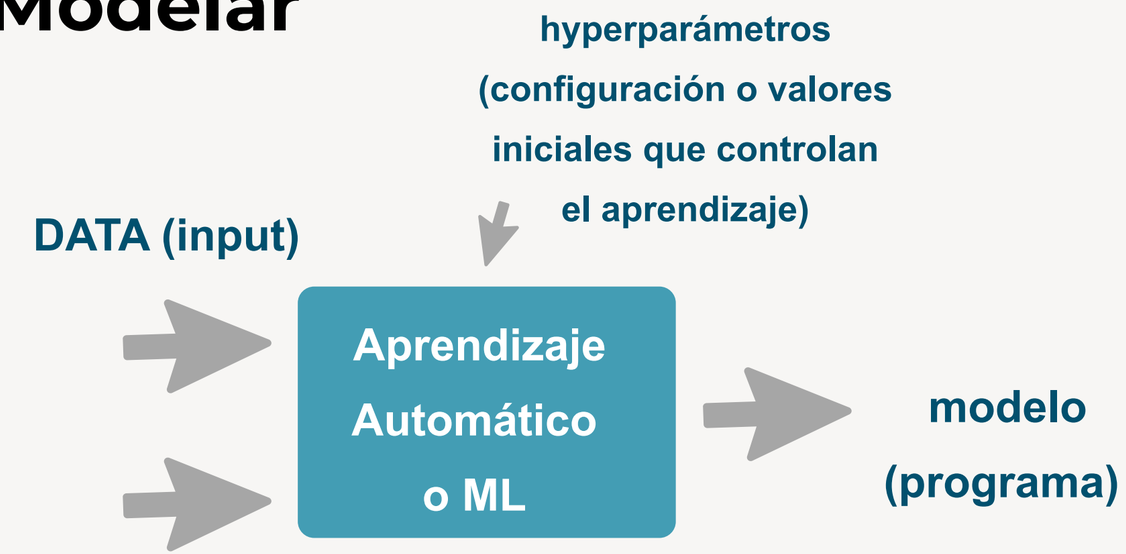
Almacenamiento

Procesamiento (modelado)

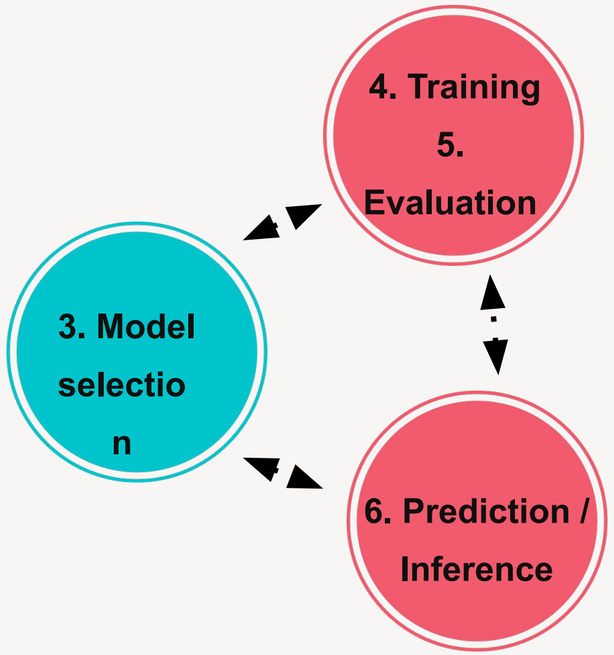
- **Preparación**
- Selección
- Entrenamiento
- Evaluación
- Predicción

Deployment

Modelar



Output



Para pensar

- La cantidad y calidad de los datos es importante!
- Tenemos datos numéricos o categóricos? se pueden etiquetar, o están etiquetados?
- Que transparente necesitamos que sea el modelo?
- Que tan preciso queremos que sea el modelo?
- Hay que hacer las preguntas adecuadas (que pueda responder el ML)
- Explorar la bibliografía relacionada al campo de aplicación que se trate
- Comparar más de una técnica
- No quedarse con un modelo (optimizar hiperparámetros)

Choose your weapon

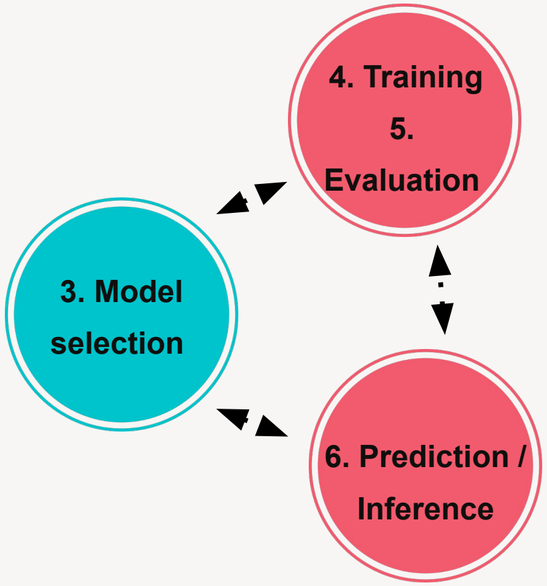
El problema

El algoritmo

Elegimos un algoritmo ...

cómo sigue la cosa?

Entrenamiento



WHILE NOT CONVERGE
train

WHILE NOT MIN(loss function)
train

- Entrenamiento -> proceso iterativo -> en cada iteración (epoch) pasa por todo el conjunto de entrenamiento para "ajustar" a los datos y encontrar los parámetros del modelo
- Se inicia con valores "aleatorios" de los parámetros (para un set de hiperparámetros) y estima el costo. El modelo con menor costo es el elegido.
- Función de costo o de pérdida (loss function)
  - Mide el costo que tienen las predicciones incorrectas
  - Se mide mediante la función de pérdida también denominada función de costo o función objetivo
  - **Queremos optimizar (minimizar) la función de pérdida durante el entrenamiento.**

Entrenamiento

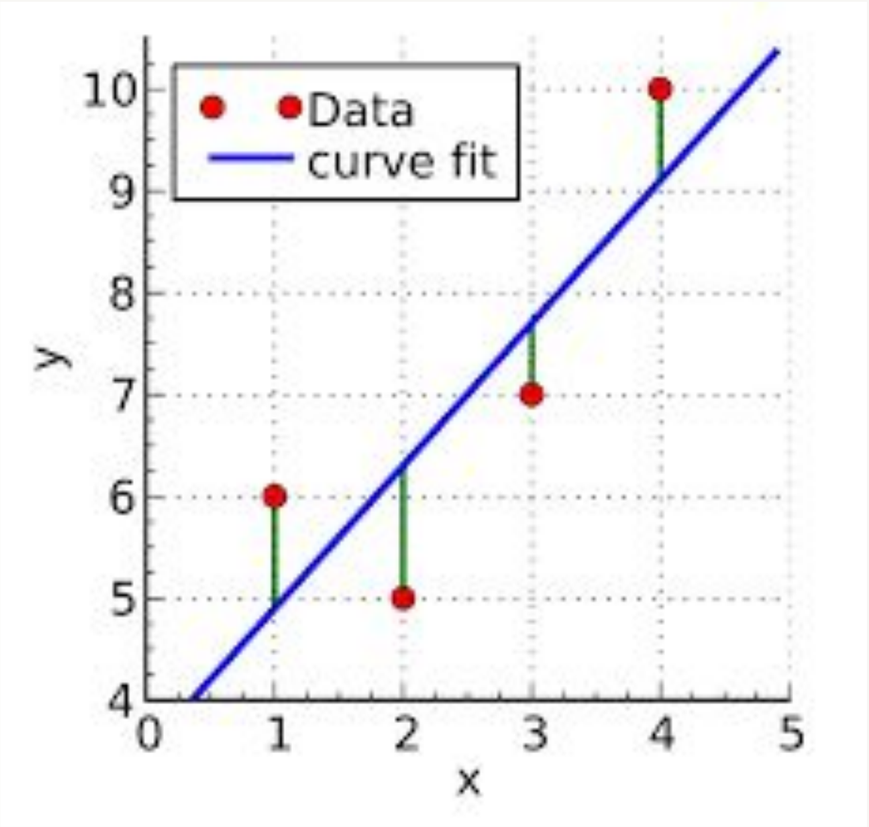
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Mean · Error · Squared

WHILE NOT MIN(MSE)

TRAIN

	Loss Function	☺
Regression	MSE (mean squared Error)	ideal > cercano a 0
Classification	Cross entropy Binary cross entropy	ideal > cercano a 0



Entrenamiento

	Loss Function	
Regression	MSE (mean squared Error)	ideal > cercano a 0
Classification	Cross entropy Binary cross entropy	ideal > cercano a 0

- Que tan mal/bien clasifica el modelo?
- Entropía alta > necesitamos más información para representar un evento
- Cross-entropy o entropia cruzada (teoría de la información):- Diferencia entre dos distribuciones de probabilidad
- Cross-entropy loss > aumenta a medida que la probabilidad de las predicciones divergen del valor verdadero de las etiquetas.

- Cross Entropy

$$H(p,q) = -\sum_{i=1}^{n} p(x_i) log_2(q(x_i))$$

- Binary Cross Entropy

$$L = -\sum_{i=1}^{2} t_i \log(p_i)$$
$$= -\left[t \log(p) + (1-t)\log(1-p)\right]$$

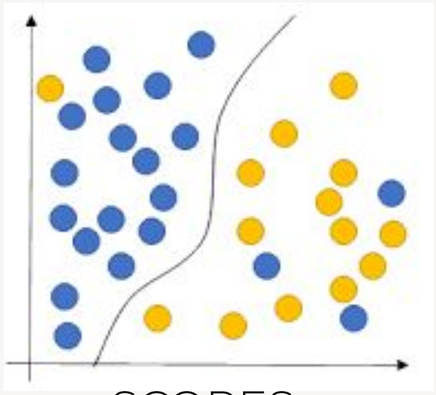
where t_i is the truth value taking a value 0 or 1 and p_i is the Softmax probability for the i^{th} class.

- Se calcula como:

$$L = -\frac{1}{N}\left[\sum_{j=1}^{N} \left[t_j \log(p_j) + (1-t_j)\log(1-p_j)\right]\right]$$

for N data points where t_i is the truth value taking a value 0 or 1 and p_i is the Softmax probability for the i^{th} data point.

Entrenamiento



SCORES (salidas del clasificador)

logits = datos no normalizados de las predicciones (o salidas) del modelo

Softmax

$$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Probabilidades

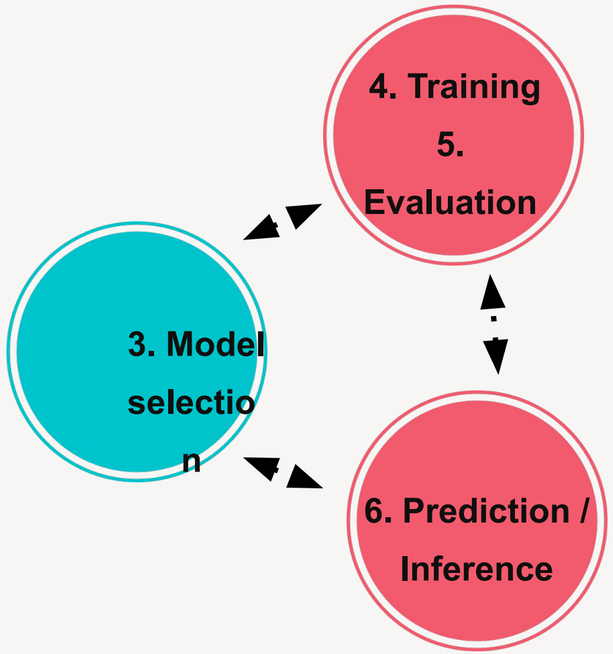
S
0.775
0.126
0.039
0.070

L

T
1
0
0
0

$$\begin{aligned} L_{CE} &= -\sum_{i=1} T_i \log(S_i) \\ &= -\left[1 \log_2(0.775) + 0 \log_2(0.126) + 0 \log_2(0.039) + 0 \log_2(0.070)\right] \\ &= -\log_2(0.775) \\ &= 0.3677 \end{aligned}$$

Ya tenemos un modelo!

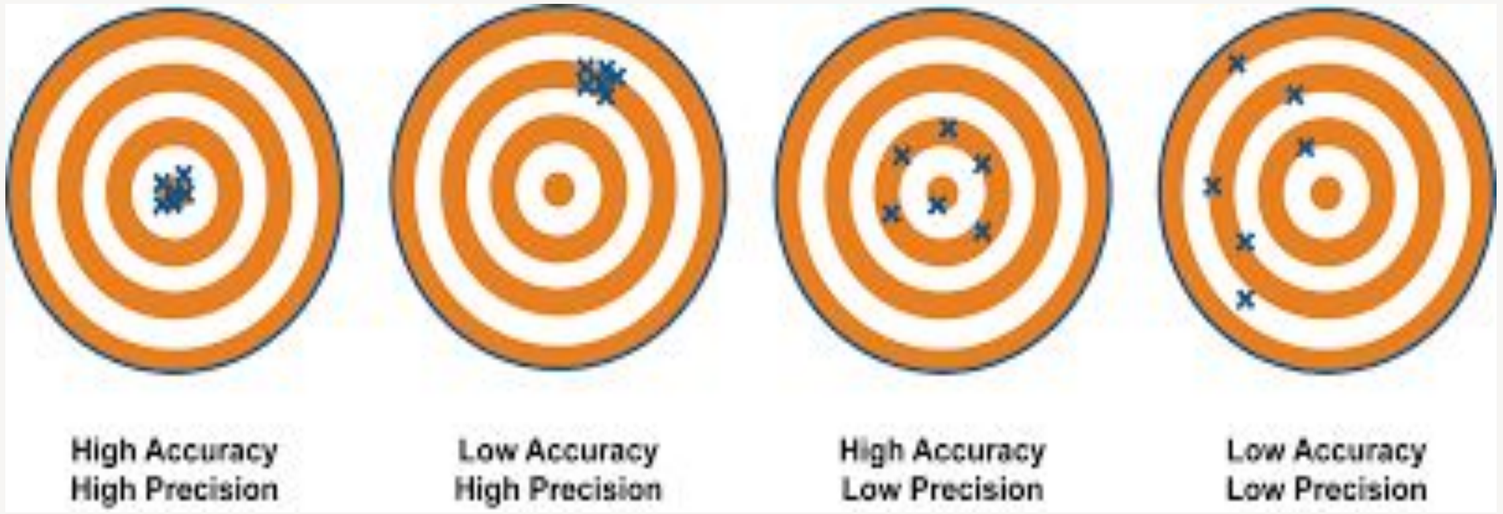


Que tan bueno es?

- Cuántas iteraciones necesité?
- Cómo se comporta ante datos nuevos?
- Cómo medimos que tan bien anda para el test set?
- Y que pasa si cambio los valores de los hiperparámetros?

Evaluación - métricas

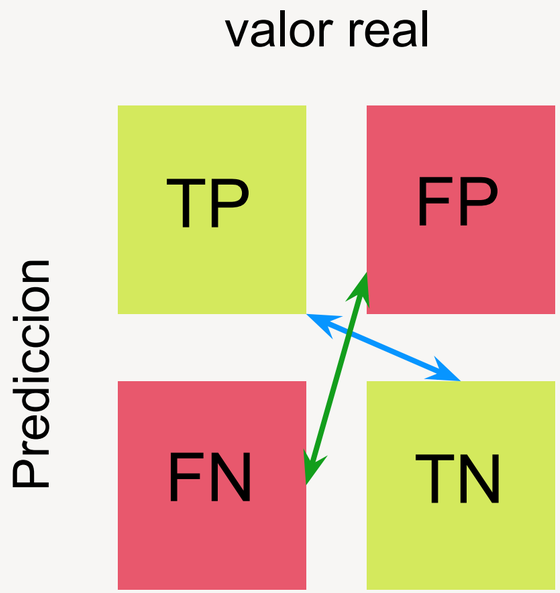
- Se utiliza el **testset** para comparar la performance del modelo ante datos nuevos
- Dos conceptos importantes:
  - Precision: se refiere a la dispersión del conjunto de valores obtenidos de mediciones repetidas de una magnitud. Cuanto menor es la dispersión mayor la precisión.
  - Exactitud: se refiere a cuán cerca del valor real se encuentra el valor medido (error absoluto). Está relacionada con el sesgo de una estimación. Cuanto menor es el sesgo más exacta es una estimación.



Evaluación - métricas

Metricas:

- Clasificación: **matriz de confusión**



$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$specificity = \frac{TN}{TN + FP}$$

TP: true positive FN: False negative

FP: false positive TN: True negative

- Caso "ideal" -> FN=0 y FP=0
- Que es preferible? Minimizar falsos negativos o falsos positivos?
  - Ejemplo 1: detectar cancer/no cancer. Si de 100 personas solo 5 tienen cáncer -> queremos detectar todos los casos que tienen cáncer -> buscamos << FN ~ 0 -> recall >>
  - Ejemplo 2: detectar spam en el correo y estamos esperando un correo importante -> evitar clasificarlo como spam -> buscamos FP <<

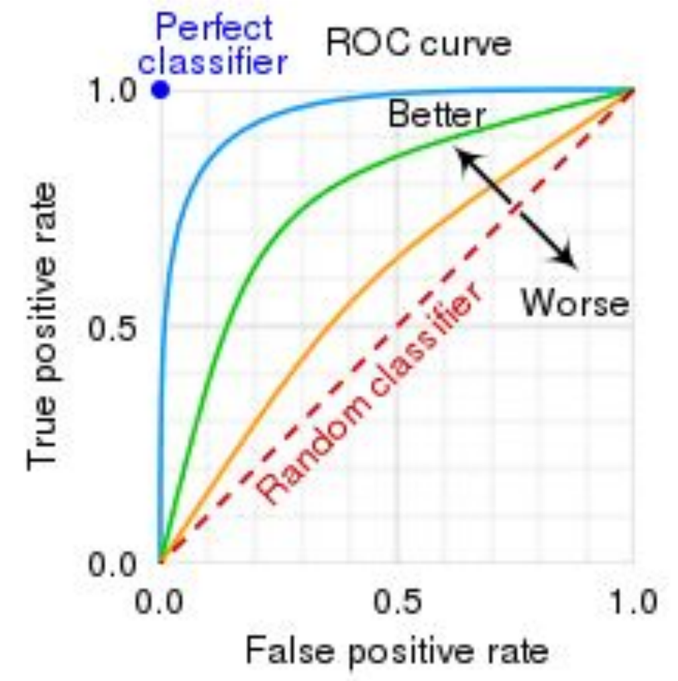
Evaluación - métricas

Metricas:

- Clasificación: **AU & ROC**

AUC =Area Under The Curve
ROC = Receiver Operating Characteristics curve.

- mejor modelo, >> AUC
- AUC = me dice que tan capaz es un modelo de distinguir entre clases

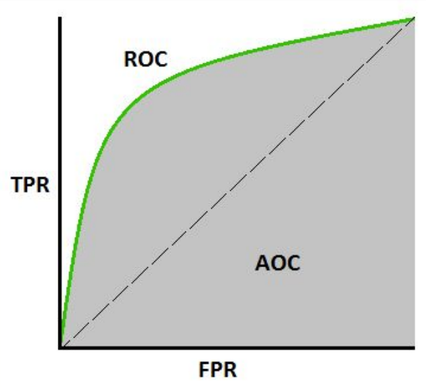


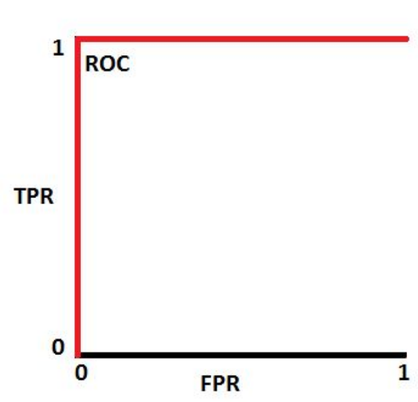
$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

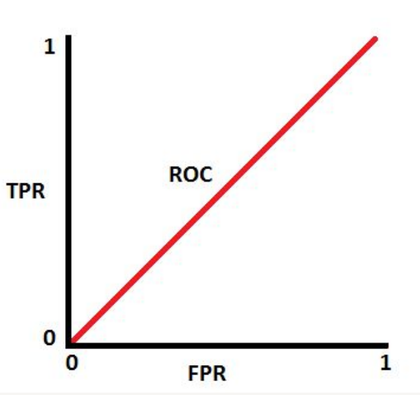
$$\text{Specificity} = \frac{TN}{TN + FP}$$

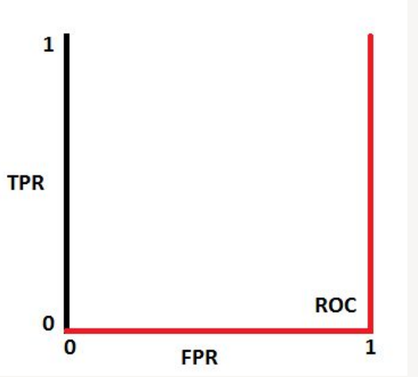
$$\text{FPR} = 1 - \text{Specificity} = \frac{FP}{TN + FP}$$

AUC representa el grado de separabilidad entre las clases









Evaluación - métricas

Metricas:

- Regresión: MAE,MSE,RMSE

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

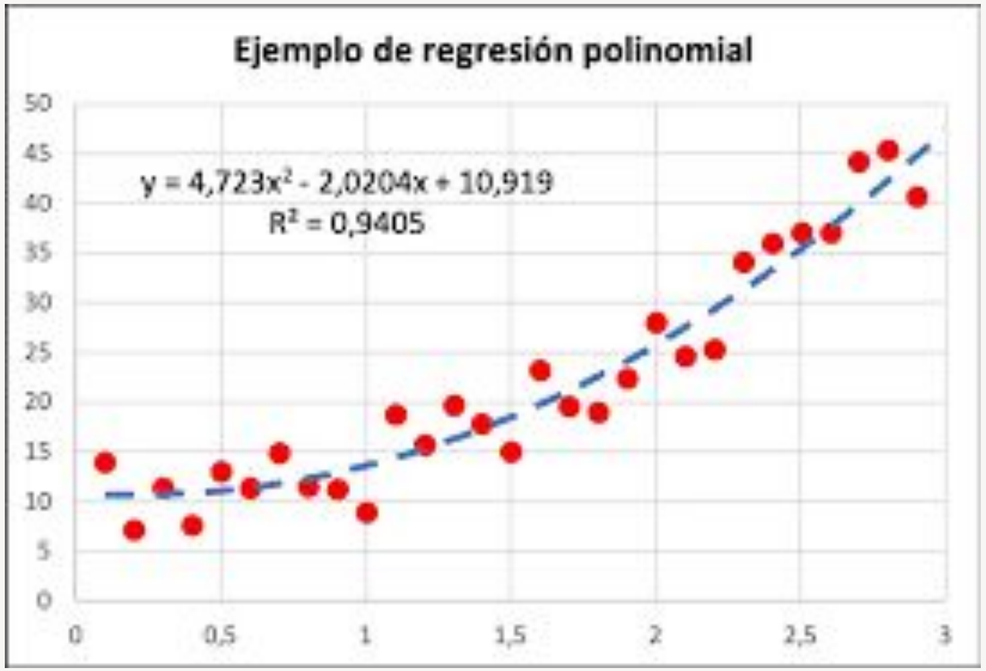
$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Where,

$\hat{y}$ – *predicted value of* y

$\bar{y}$ – *mean value of* y



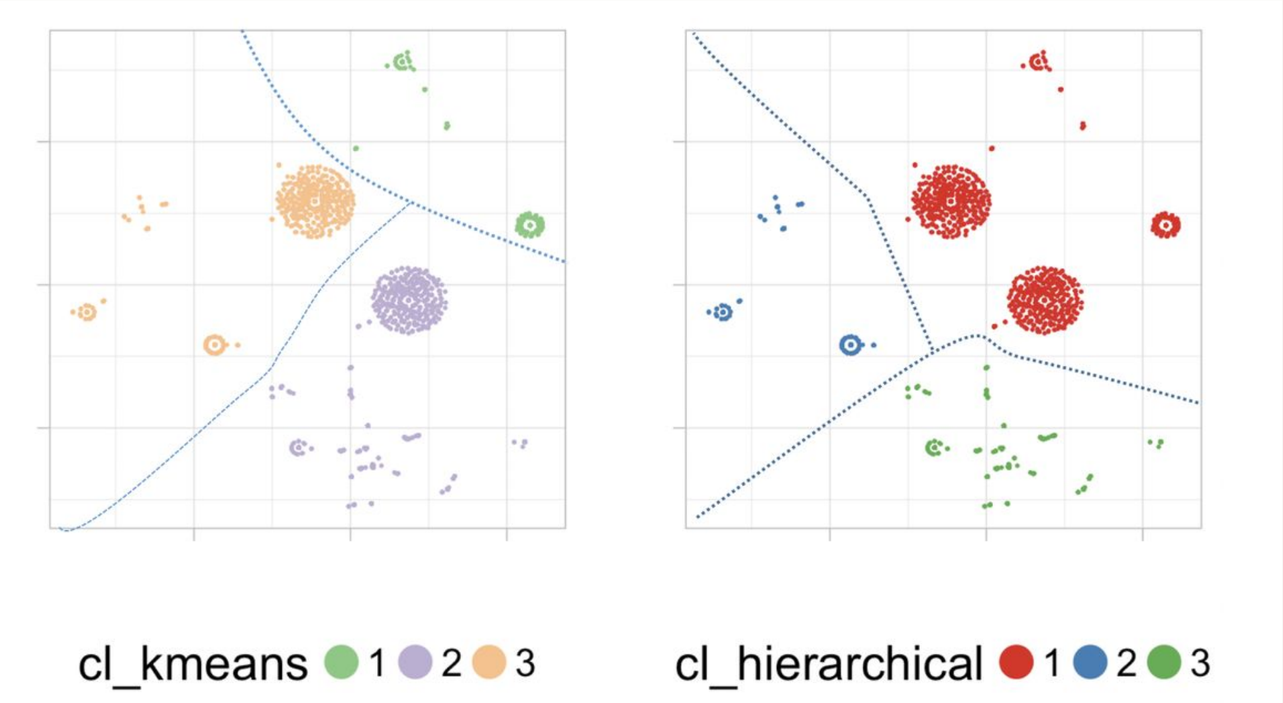
- MAE (Mean absolute error)
- MSE (Mean Squared Error)
- RMSE (Root Mean Squared Error)
- R-cuadrado (Coefficient of determination) (>>)

Analizaremos en los TPs

Evaluación - métricas

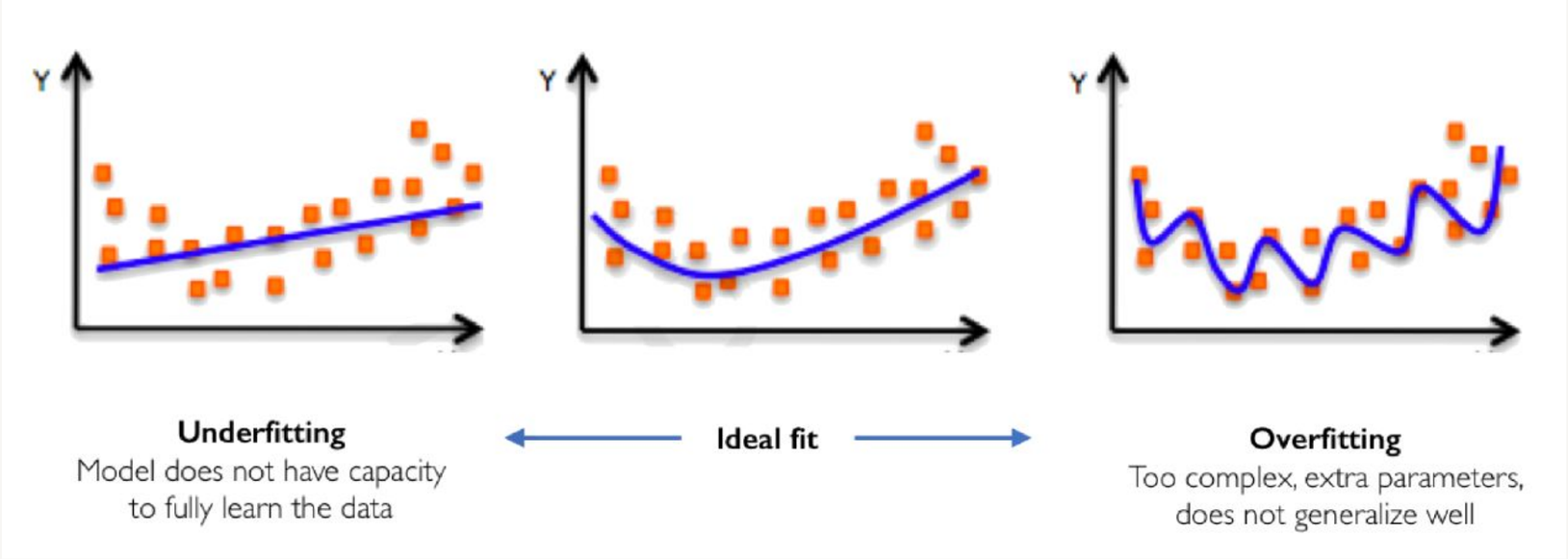
Metricas:

- Clustering: ? ? ? ?
  - no conocemos los valores verdaderos
  - muchas veces es subjetivo, requiere de expertos
  - Métodos intrínsecos: examinan que tan separados están los clusters o que tan compactos.
  - Elbow method (entre otros): basado en heurísticas (veremos en el tp) - ayuda a elegir los clusters

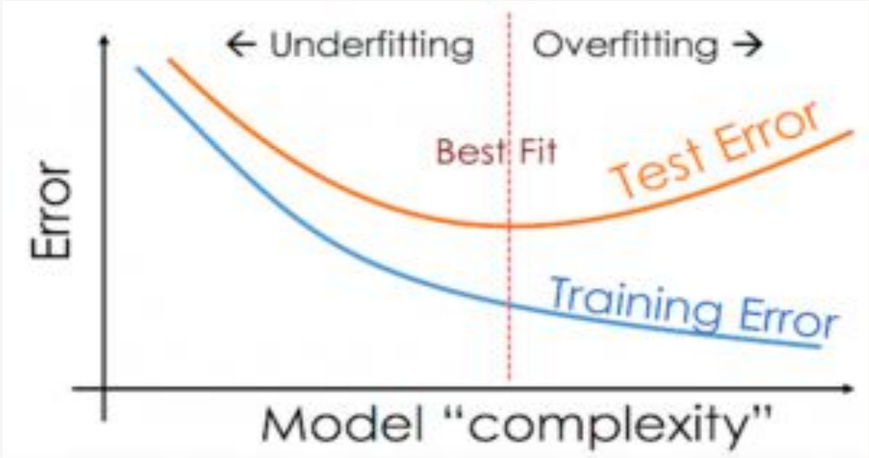


Evaluación del modelo

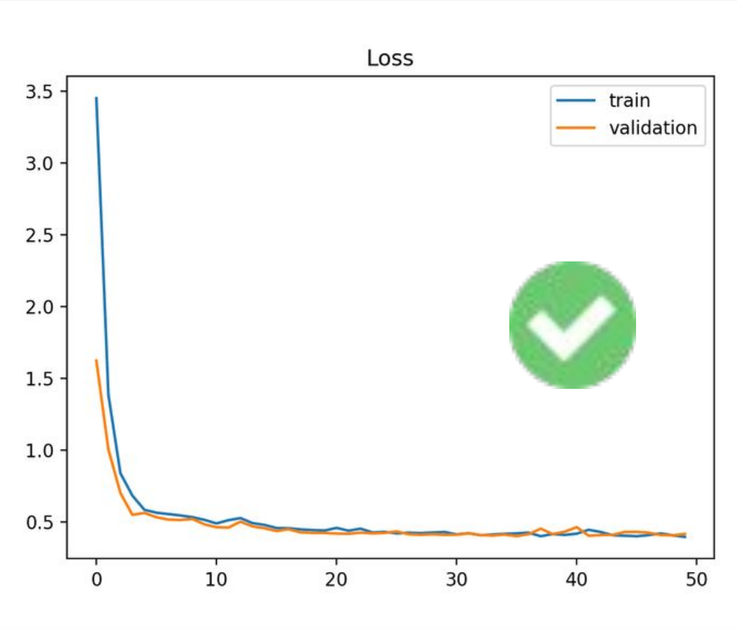
- Cómo sabemos que capacidad de generalización tiene el modelo?
- Cómo sabemos cuánto iterar? cuanto sigue aprendiendo?



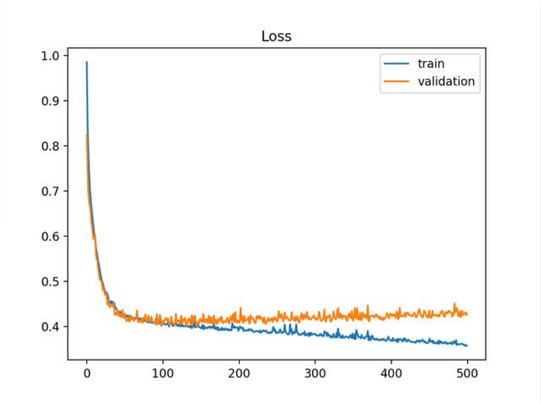
Hay técnicas para evitar el overfitting



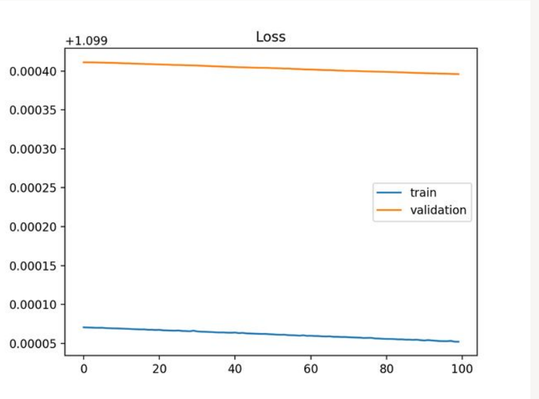
Diagnóstico del entrenamiento

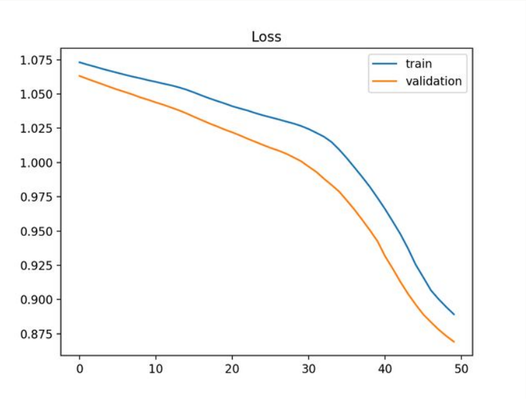


overfitting



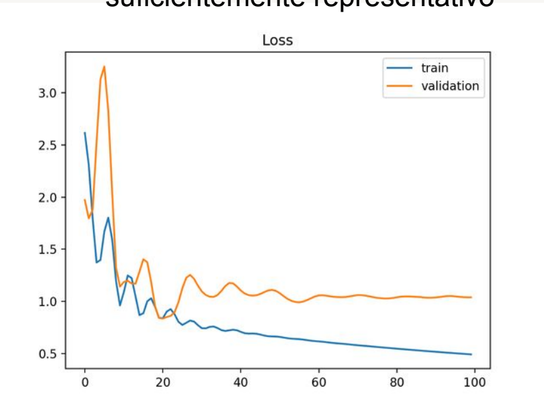
Modelo muy simple para la complejidad en los datos (underfit)





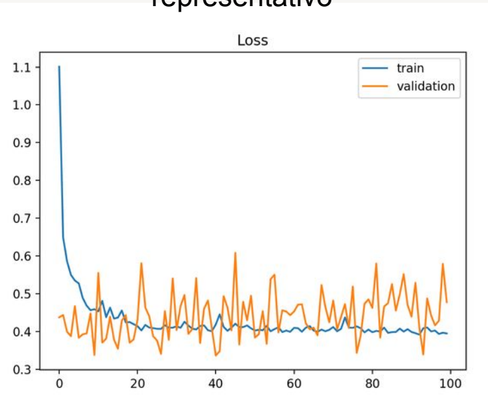
El model necesita mas training (underfit)

set de datos no es suficientemente representativo

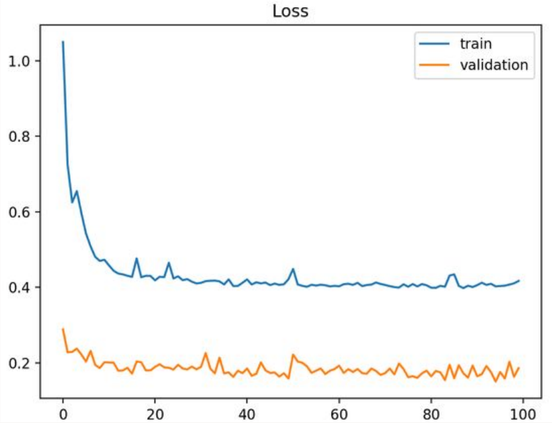


El training dataset es relativamente pequeño en comparación al de validacion

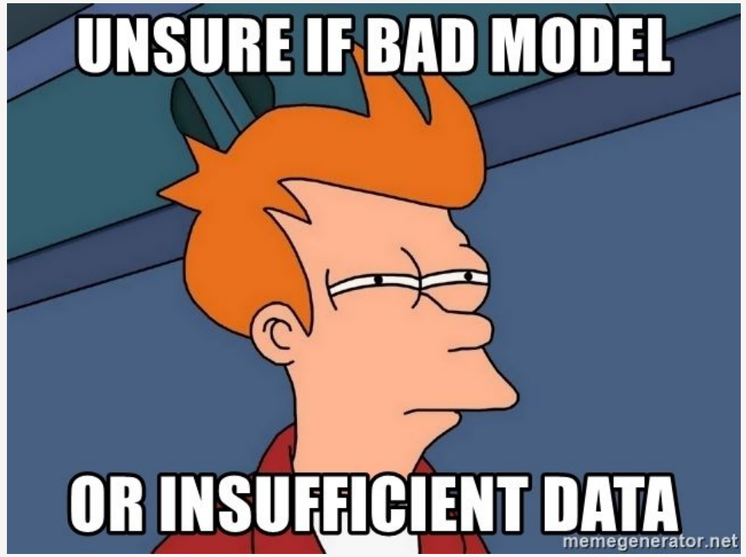
ser de validación no representativo



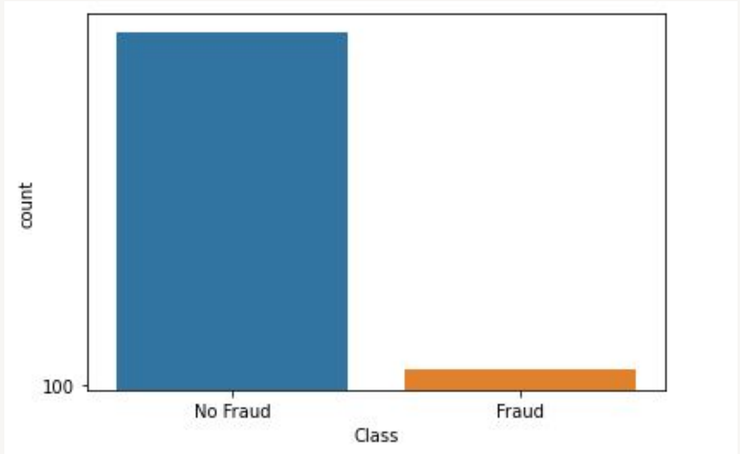
el set de validacion no provee suficiente información para evaluar la habilidad del modelo para generalizar



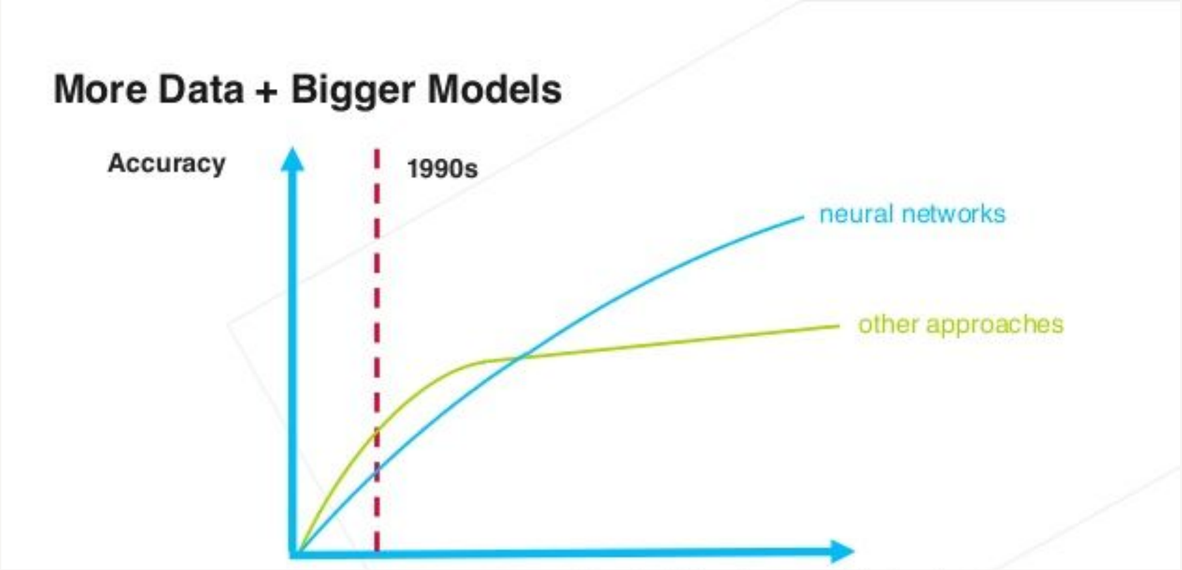
Que pasa si no mejora?



- Representatividad de casos en el set de datos



- Suficiente cantidad de datos para la complejidad del problema/modelo

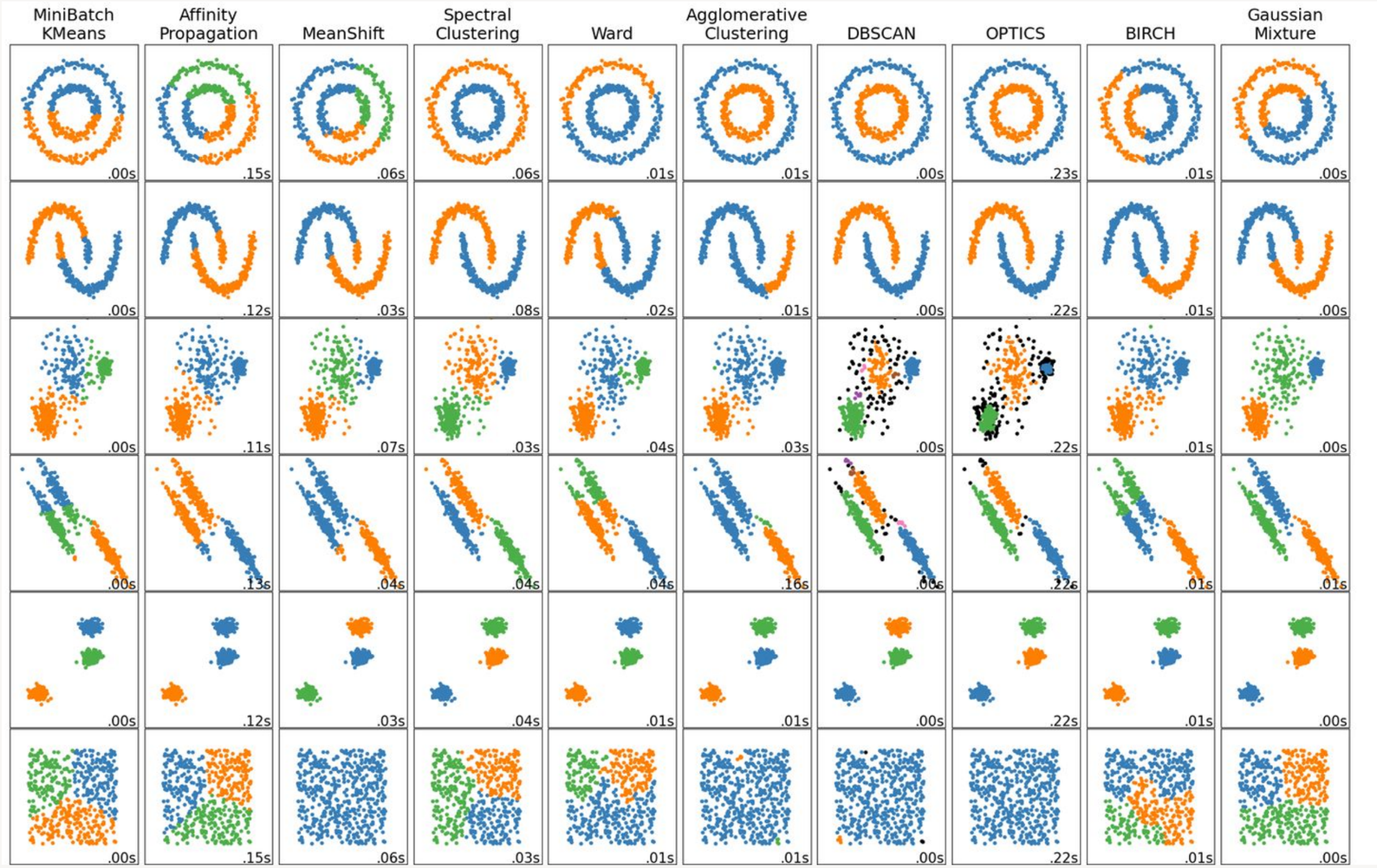


- Elegí bien los hiperparámetros?
- Debería cambiar la arquitectura?


KEEP
CALM
AND
TRAIN
HARD

Clustering

- Aprendizaje no supervisado: se utilizan datos no etiquetados para aprender patrones o anomalías en los datos de entrada
- **Clustering**: el objetivo es encontrar grupos de datos que tienen un comportamiento espacial o temporal similar.
- No existe la idea de minimizar el error"minimizing error" que guíe el proceso de aprendizaje.
- Los algoritmos tienen que ‘ descubrir’ similitudes entre los puntos. Si dos puntos son similars, pertenecen al mismo grupo/cluster.
- Si dos puntos estan ‘ cerca’ entre ellos, pertenecen al mismo cluster -> idea de distancia

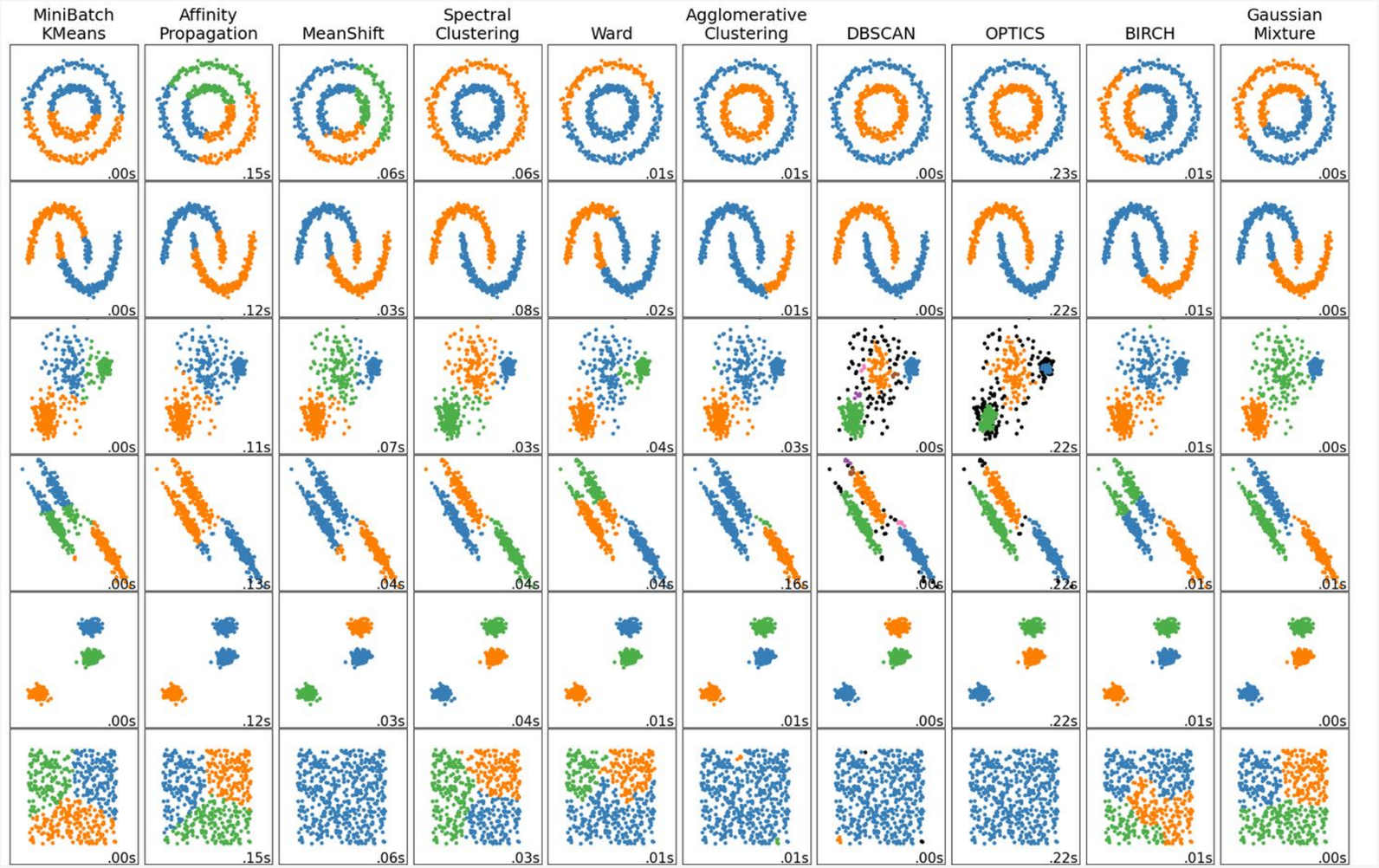


https://scikit-learn.org/stable/modules/clustering.html

Clustering

Tipos:

- Métodos por particionamiento: A partir de un set de datos de tamaño n, el algoritmo obtiene k clusters (particion del dataset)
- Métodos Jerárquicos: realiza una descomposición jerárquica del dataset.
- Métodos basados en densidad: Se tiene en cuenta la densidad de los valores en vez de la distancia para determinar los diferentes clusters.
- Grid-based methods, etc



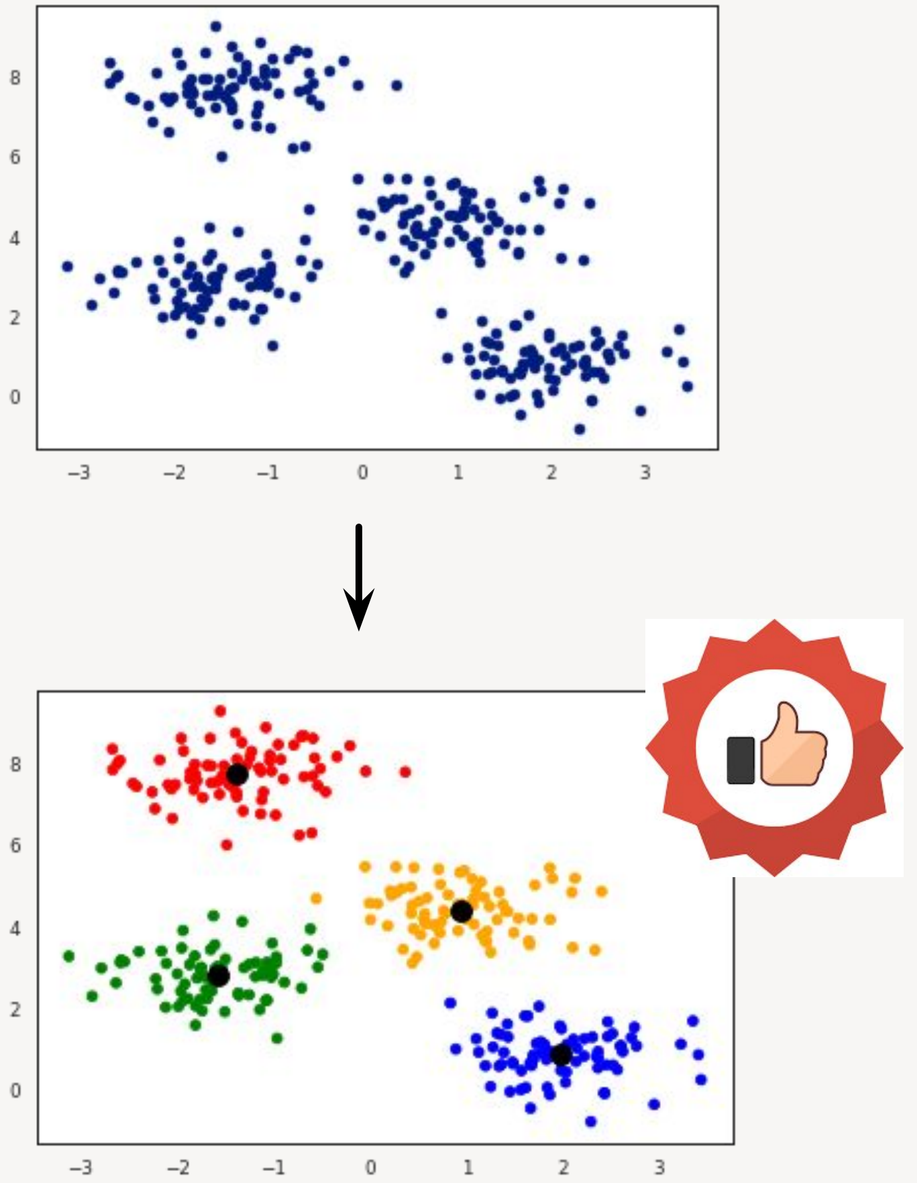
https://scikit-learn.org/stable/modules/clustering.html

K-means

El algoritmo busca un pre-determinado número k de clusters en un dataset no etiquetado multidimensional. Se basa en dos conceptos:

- El centro del cluster (grupo) es el promedio aritmético de todos los puntos que componen ese grupo.

- Un punto pertenece a un cluster si este se encuentra cerca del centro de ese cluster, más que la distancia a otros clusters.



k = 4 groups

K-means

Como se encuentran los centros en K-Means(centroids)?

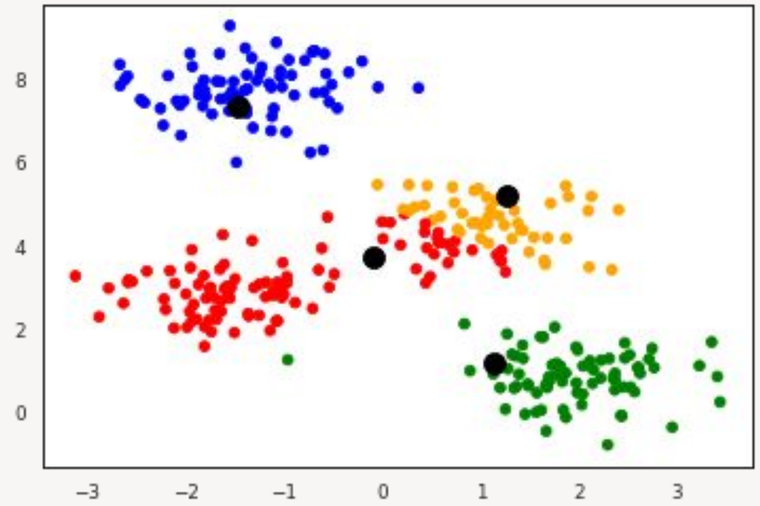
algoritmo **Expectation-Maximization** (E-M):

- Elige aleatoriamente un centro para un cluster
- Repetir hasta que converja:

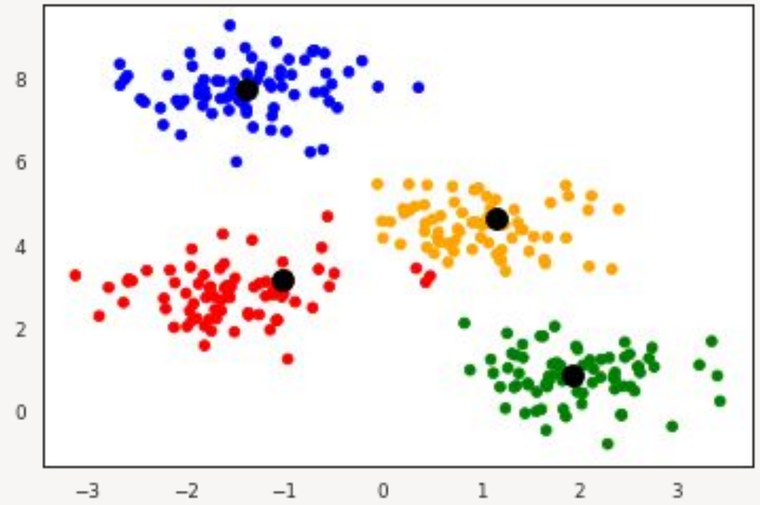
  E-Step: asignar un punto al centroide más próximo

  M-Step: elegir el centro del cluster como el promedio de los puntos que tiene ese cluster

iteration i



iteration i+n

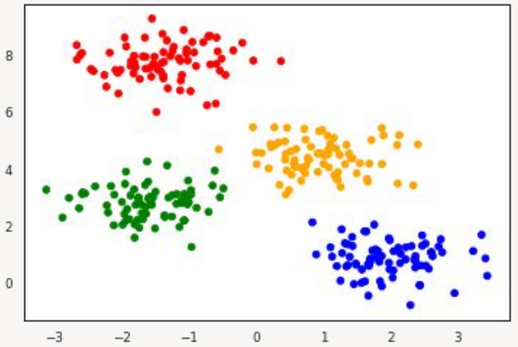


K-means

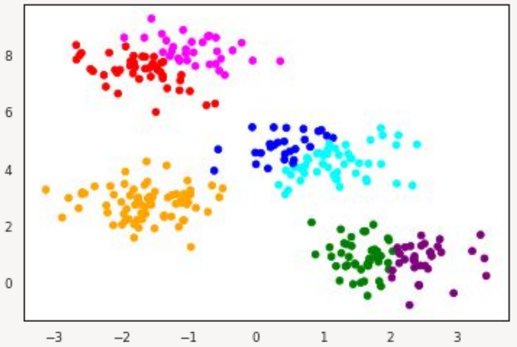
- El proceso E-M garantiza que el resultado mejore con cada iteración (mejor solución local), pero no garantiza que alcancemos la mejor solución global.
- Algunas veces nos alcanza con el subóptimo
- SKlearn agrega un parámetro de inicialización (n_init) que se usa para los n valores iniciales de los centroides elegidos de manera aleatoria.
- Solo es recomendable para problemas con clusters linealmente separados
- > sensibilidad a valores alejados
- Para datos numericos (sino Kmodes)
- ‘hard clustering’: cada observación es enviada a un único cluster sin mucha informacion sobre que tanta confianza hay de que sea el cluster adecuado (fuzzy C- means).

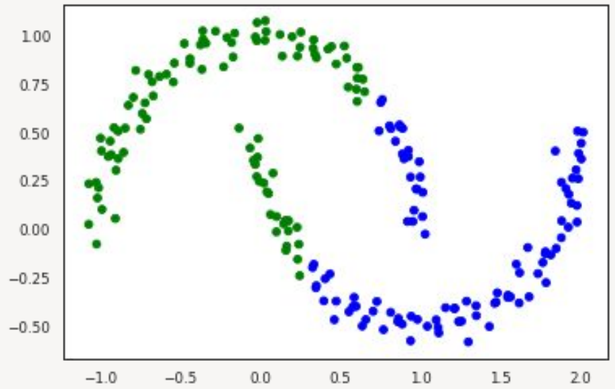
- Como elegir k?. **Elbow Method**

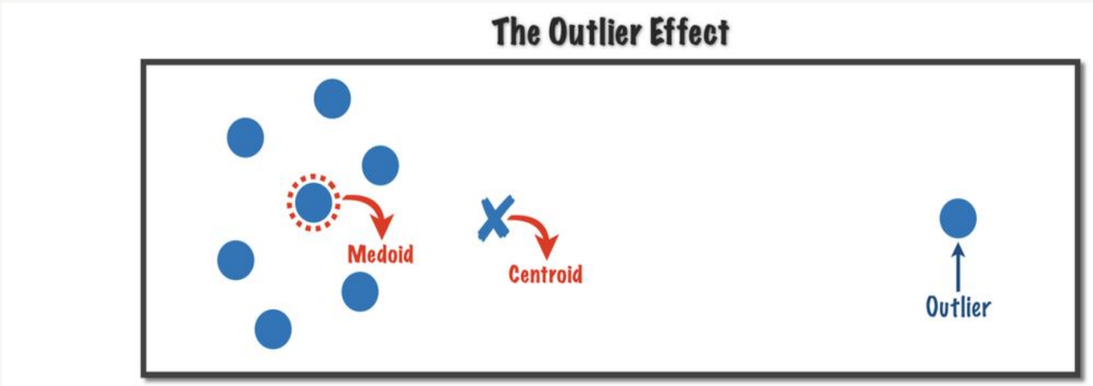
k = 4



k = 7







K-means

Elbow Method

- Usa el concepto de ‘sum of squared distance (SSE)’ o suma de las distancias cuadradas [tambien se llama ‘Within-Cluster Sum of Square (WCSS)’ o inercia] para elegir un valor ideal para k basándose en la distancia al cuadrado entre cada punto y el centroide.

- Se elige el valor de k donde SSE comienza a amesetarse y se puede observar un punto de inflexión (the elbow).

```python
from sklearn.cluster import KMeans

WCSS = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i)
    kmeans.fit(data.iloc[:, 0:4])
    WCSS.append(kmeans.inertia_)

plt.plot(range(1, 11), WCSS)
plt.show()
```