

DATA SCIENCE



Federico Baiocco
baioccofede@gmail.com
3512075440



Clase 21 - Agenda

CROSS VALIDATION - SKLEARN PIPELINES

¿ Dudas de la clase
pasada ?

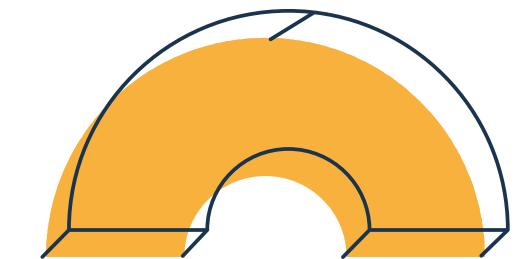
¿ Todos pudieron terminar ?



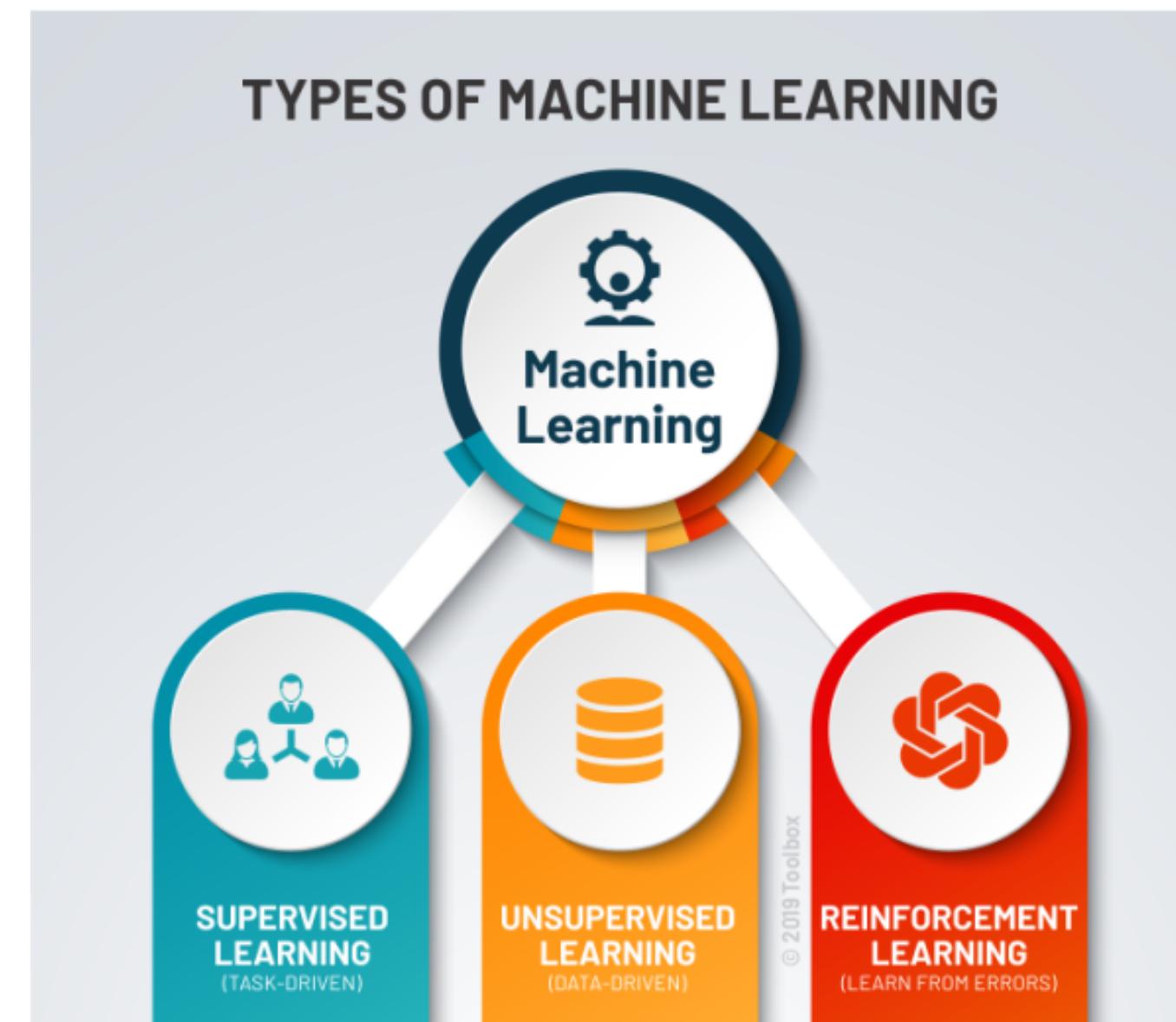


Repaso

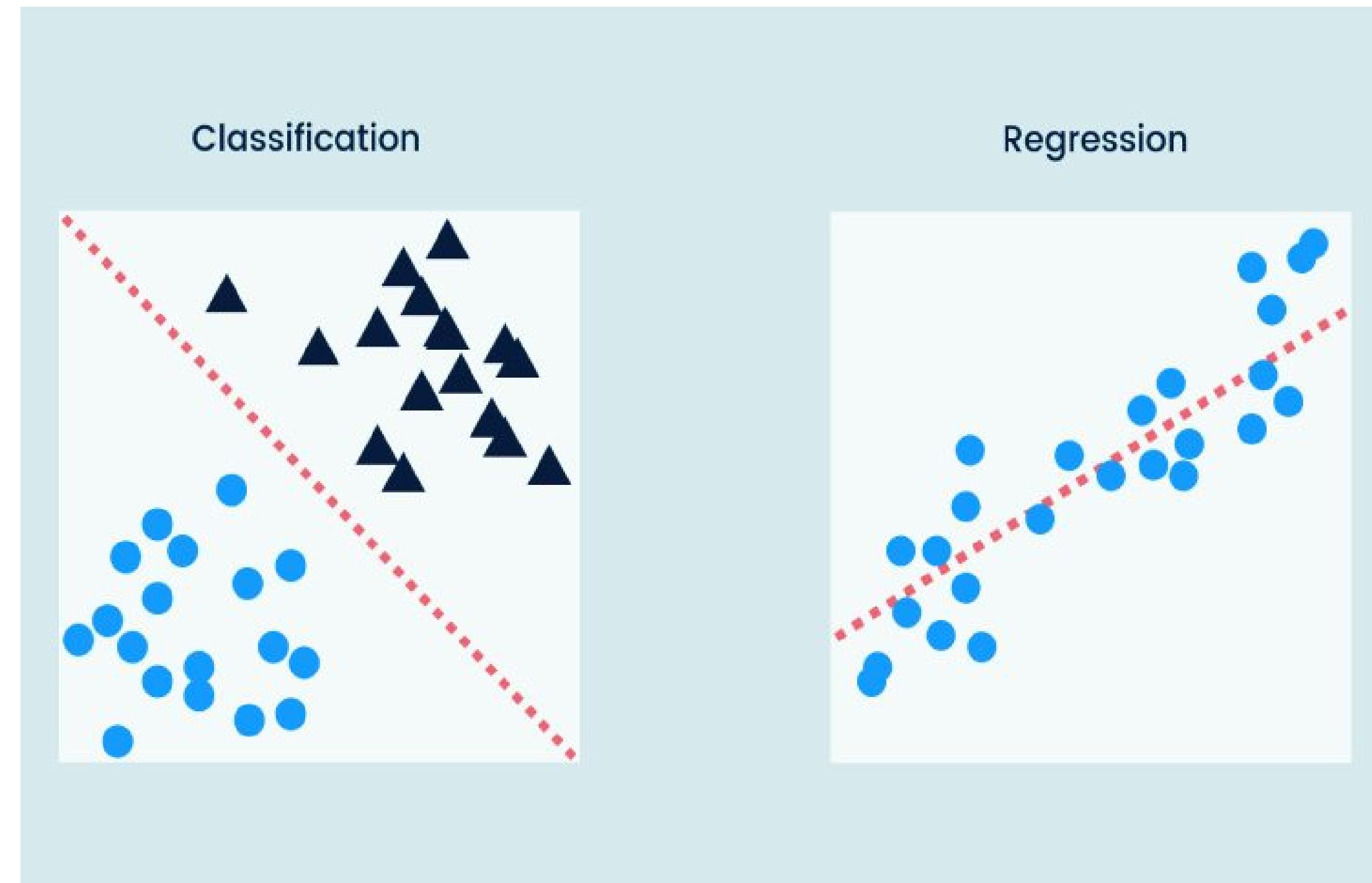
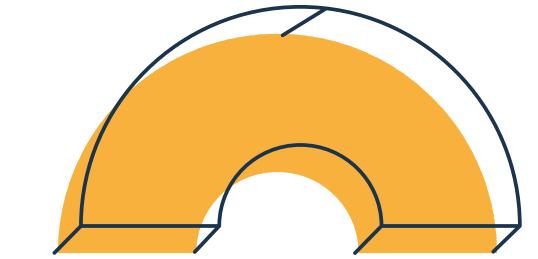
Machine learning



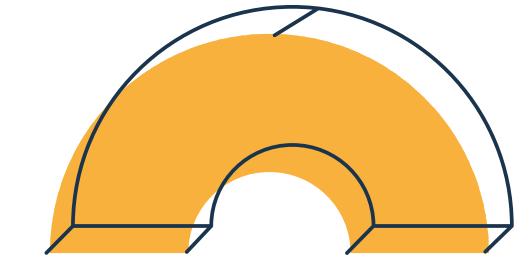
Vimos que el aprendizaje automático (Machine learning) se dedica a el estudio de programas que se encargan de aprender a realizar una tarea a partir de datos.



Aprendizaje supervisado



Parámetro - Hiperparámetro



Parámetro

Valores que nuestro algoritmo aprende automáticamente.

Por ejemplo:

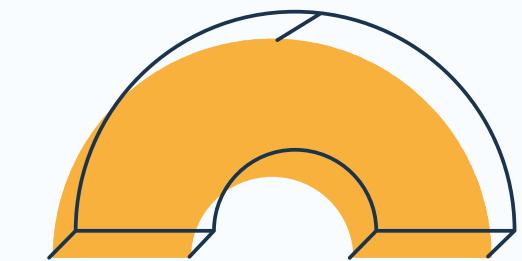
- Pendiente en una regresión lineal
- Condiciones en un árbol de decisión

Hiperparámetro

A diferencia de los parámetros, estos no se "aprenden". Son valores que definimos nosotros antes de entrenar. Por ejemplo:

- `max_depth` de un árbol
- `K` en `knn`

Clasificación

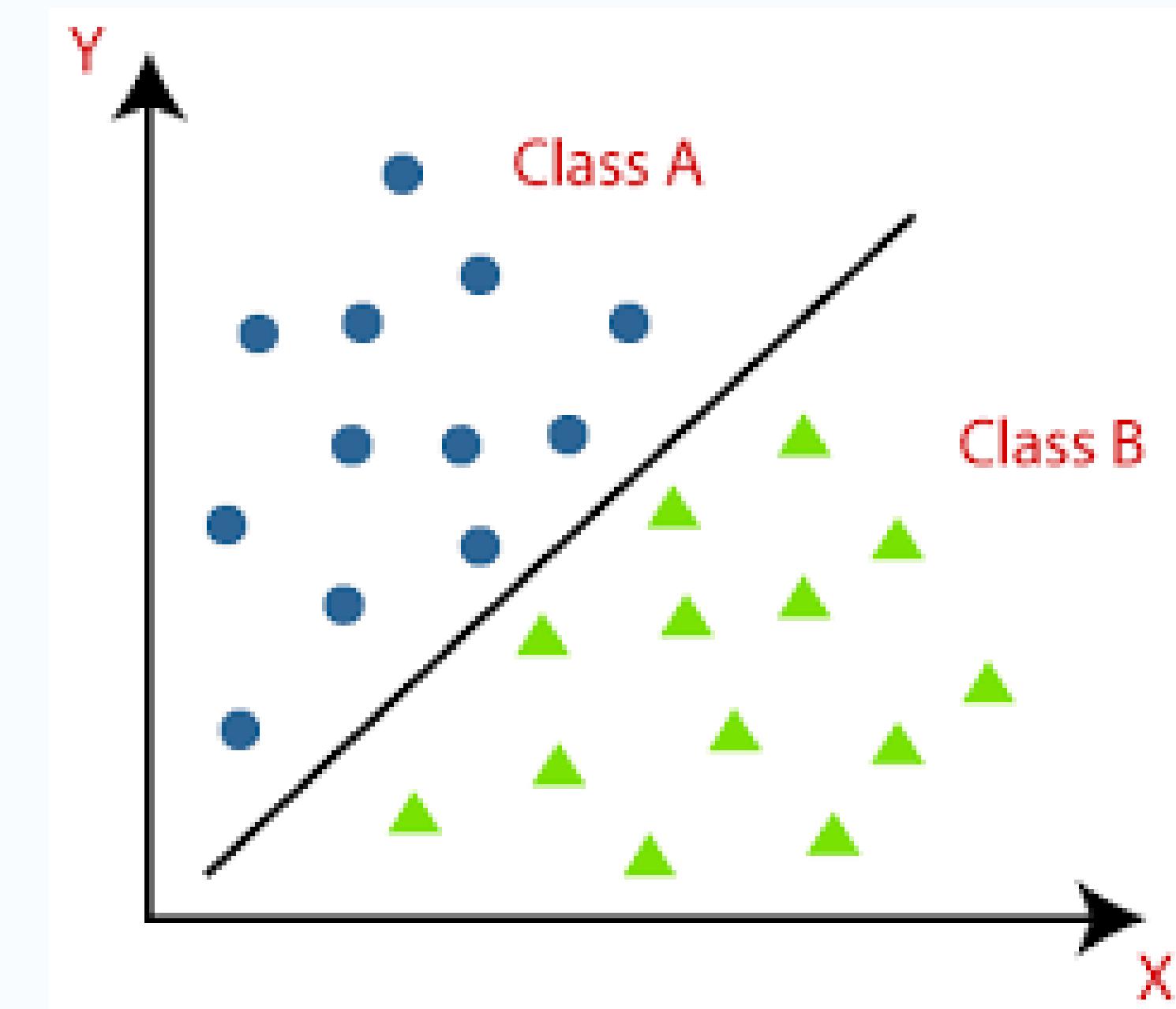


Modelos:

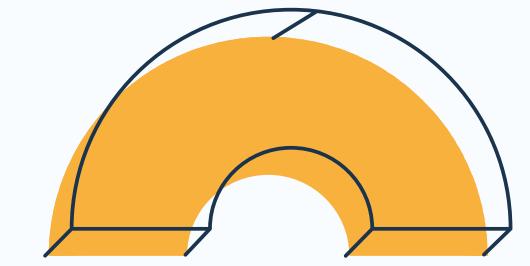
- KNN
- Decision tree

Métricas:

- Accuracy
- Precision
- Recall
- F1 Score



Regresión



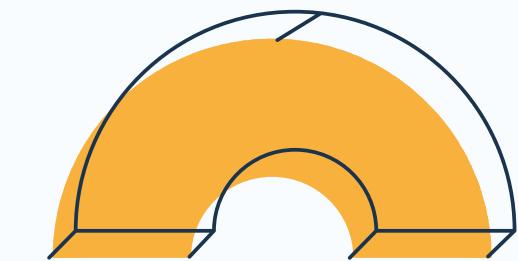
Modelos:

- Regresión lineal
- Decision tree
- KNN

Métricas:

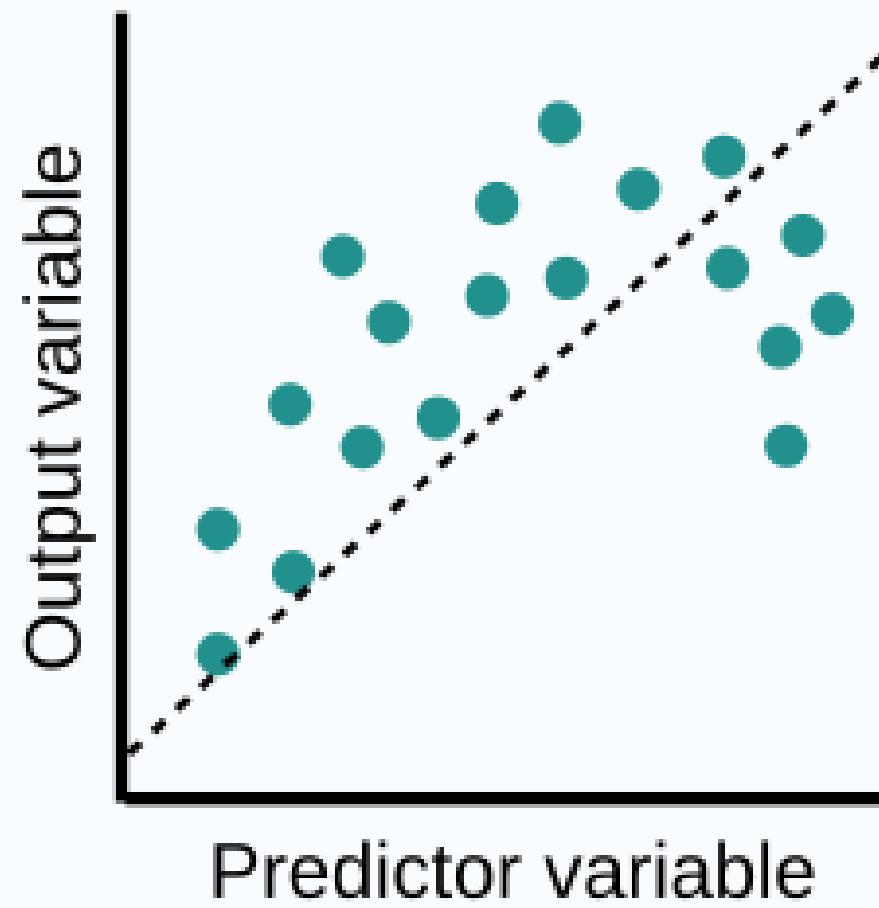
- R squared
- MAE
- MSE

Overfitting

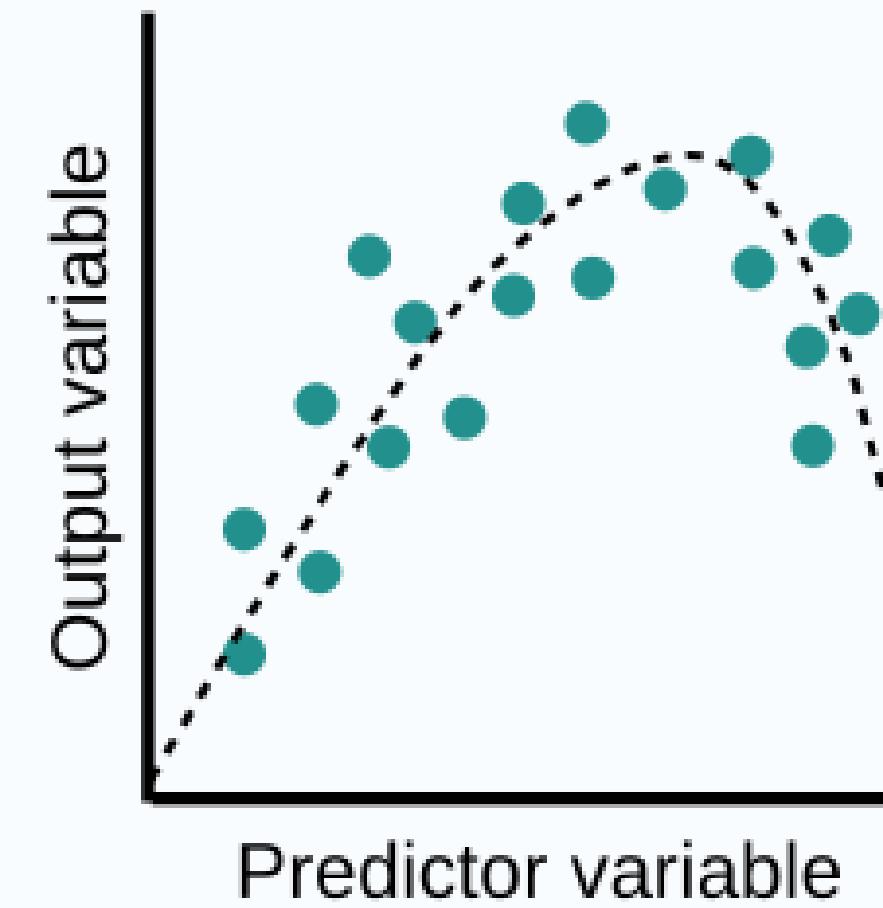


Un problema que afecta tanto a los modelos de clasificación como a los de regresión

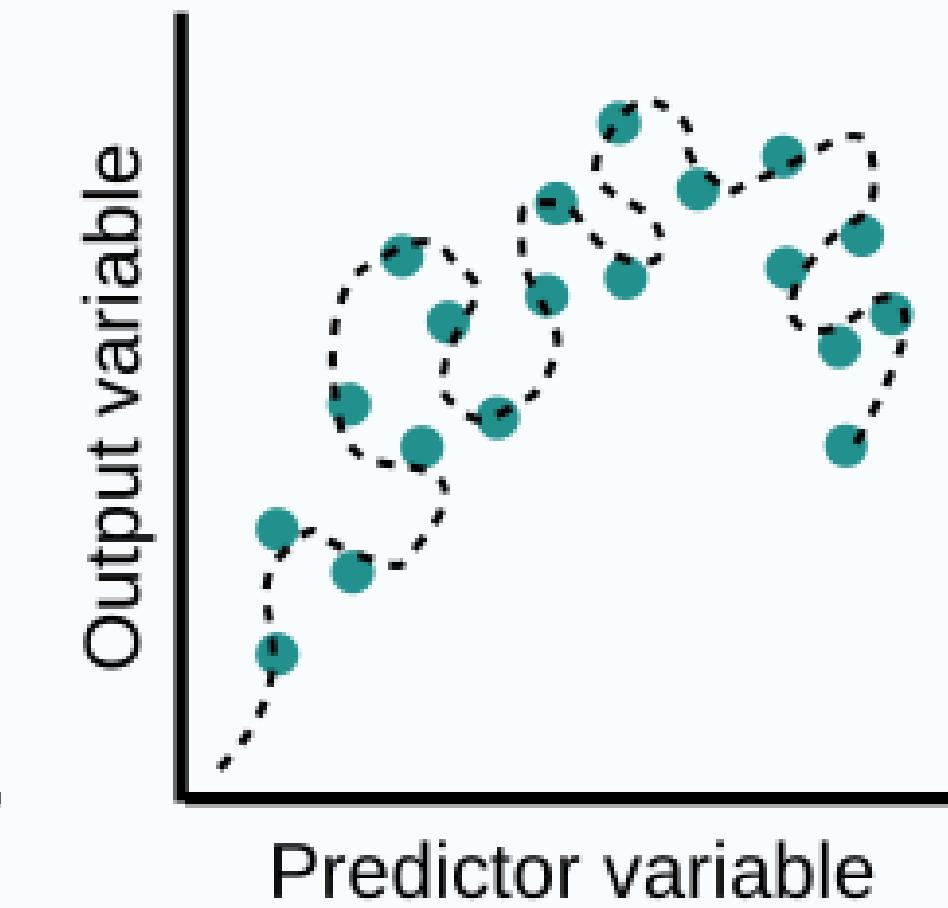
Underfit



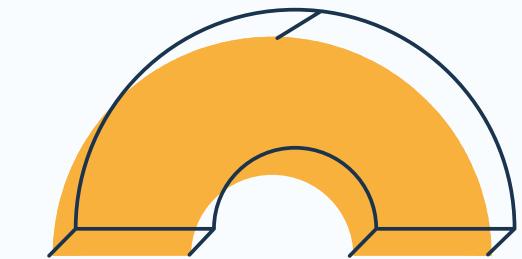
Optimal



Overfit

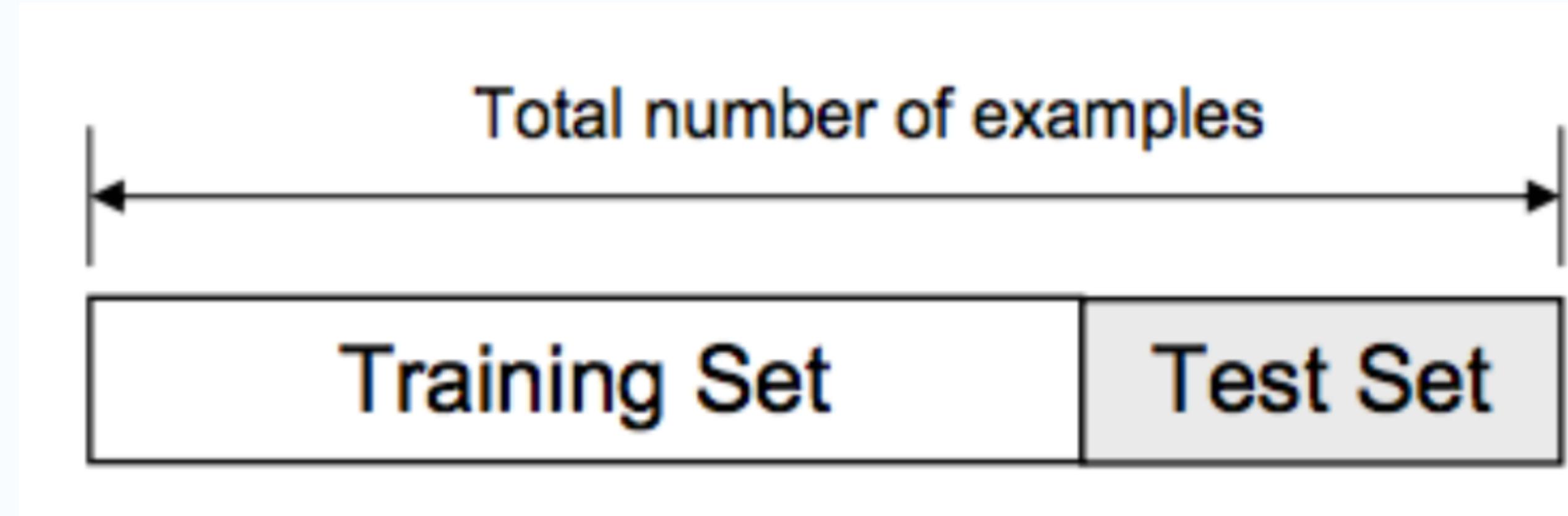


Evaluación de modelos

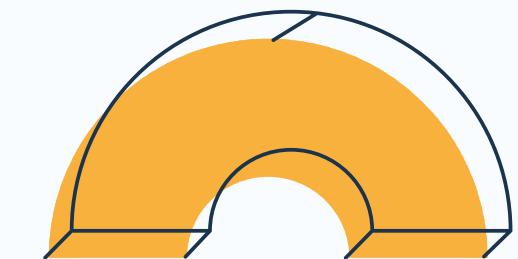


Train-Test split

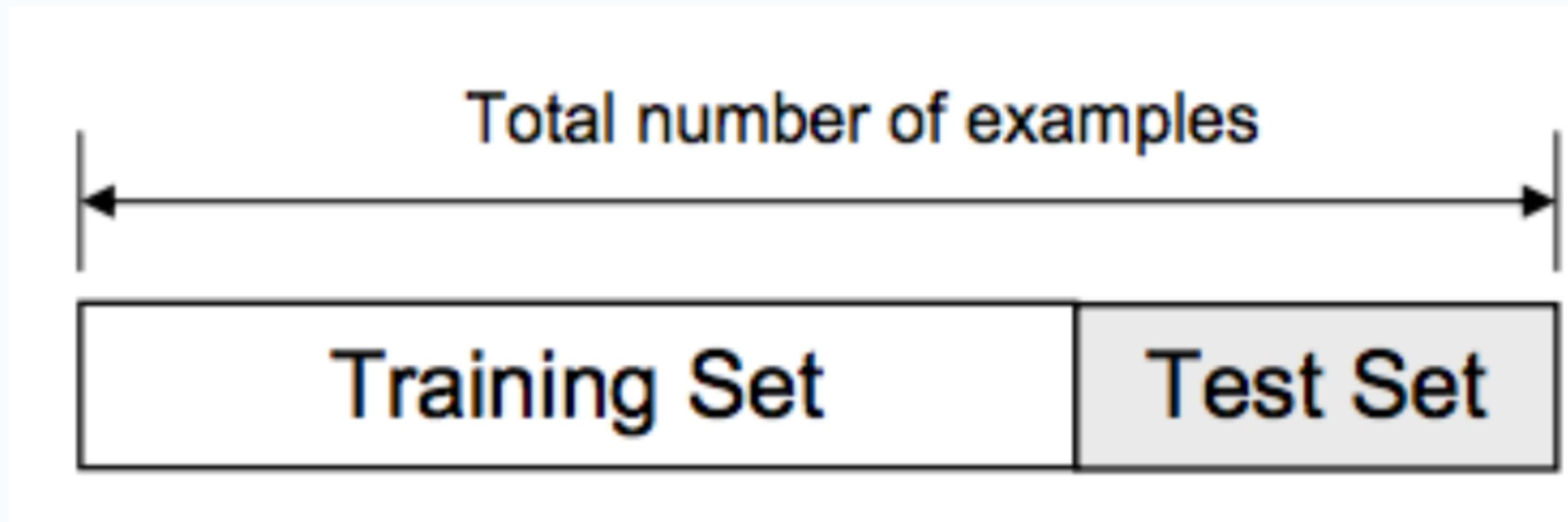
1. Separo los datos en train - test
2. Desarrollo mi modelo y entreno con el set de train
3. Evalúo la performance del modelo sobre el set de test



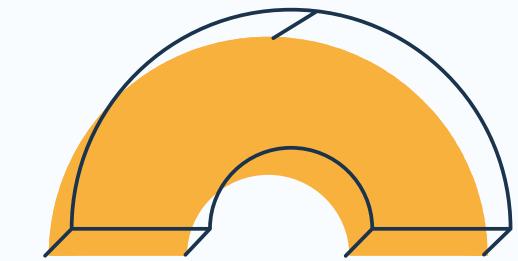
Train - test split



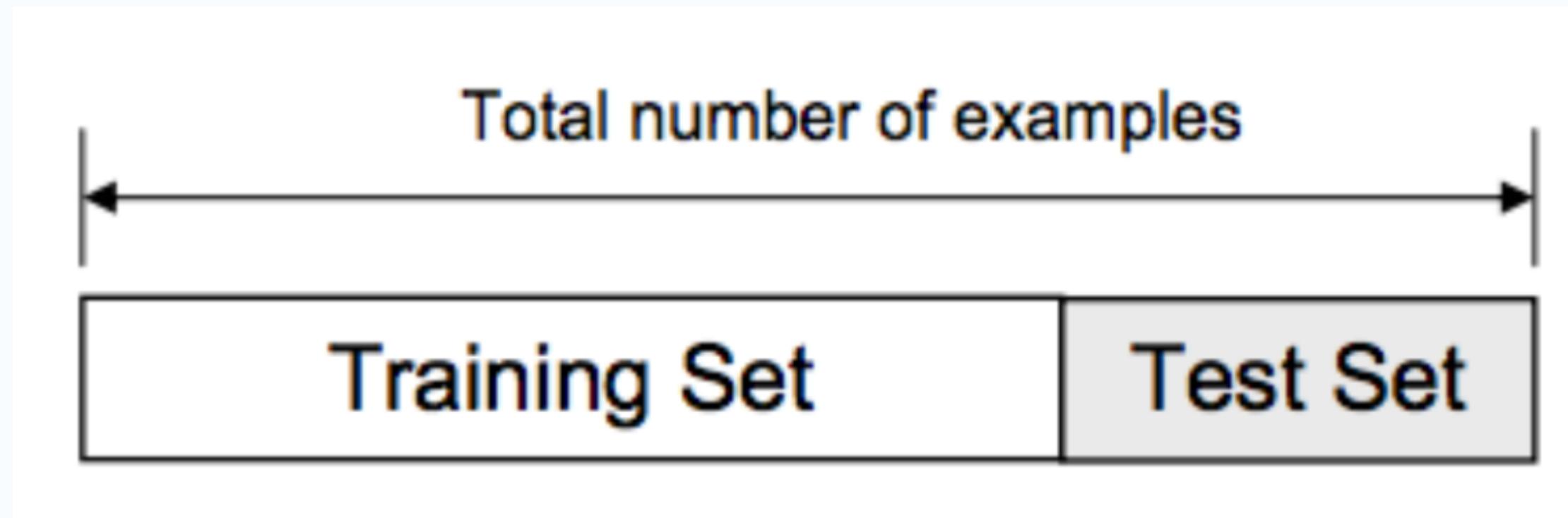
¿ Para qué lo hacemos ?



Train - test split

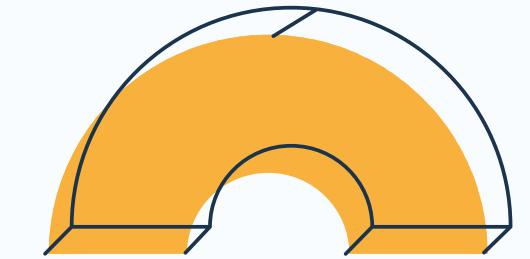


¿ Para qué lo hacemos ?



Evaluamos nuestros modelos "simulando" la realidad

Cross validation

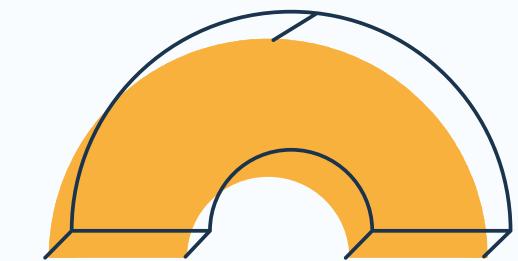


Existe un método que nos permite evaluar mejor la performance de nuestros modelos (asegurándonos de que no haya overfitting o que funcione bien sobre el set de test por casualidad)

CROSS VALIDATION

1. Particionamos el set de datos en K sub sets
2. Tomamos como set de "test" uno de los sub sets por iteración
3. En cada iteración evaluamos el modelo sobre el sub set seleccionado
4. Repetimos el proceso por cada sub set

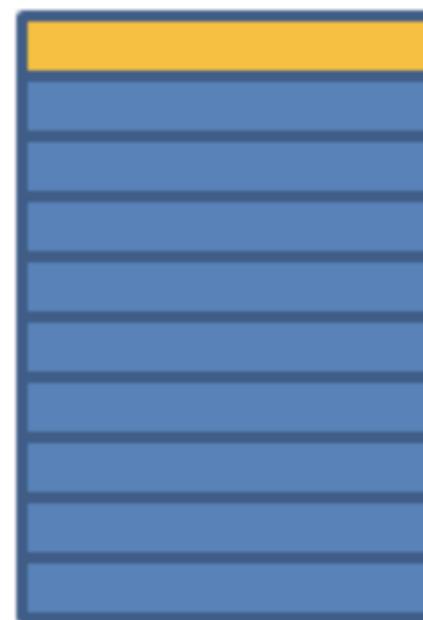
Cross validation - K fold



Este método es conocido como K-fold cross validation y está implementado en sklearn

- Validation Set
- Training Set

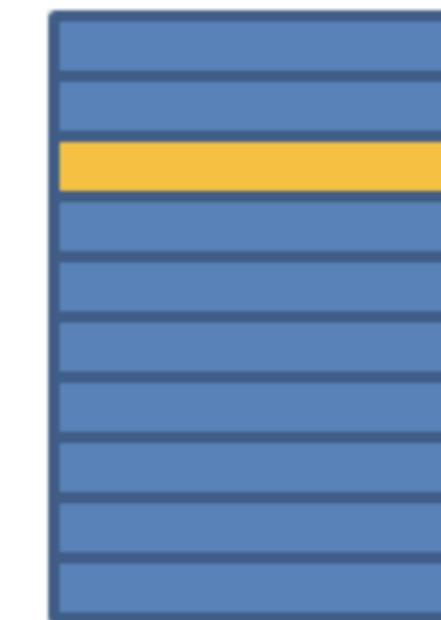
Round 1



Round 2



Round 3

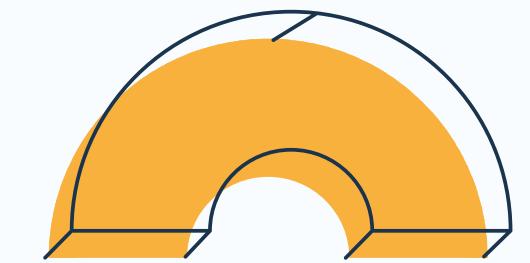


...

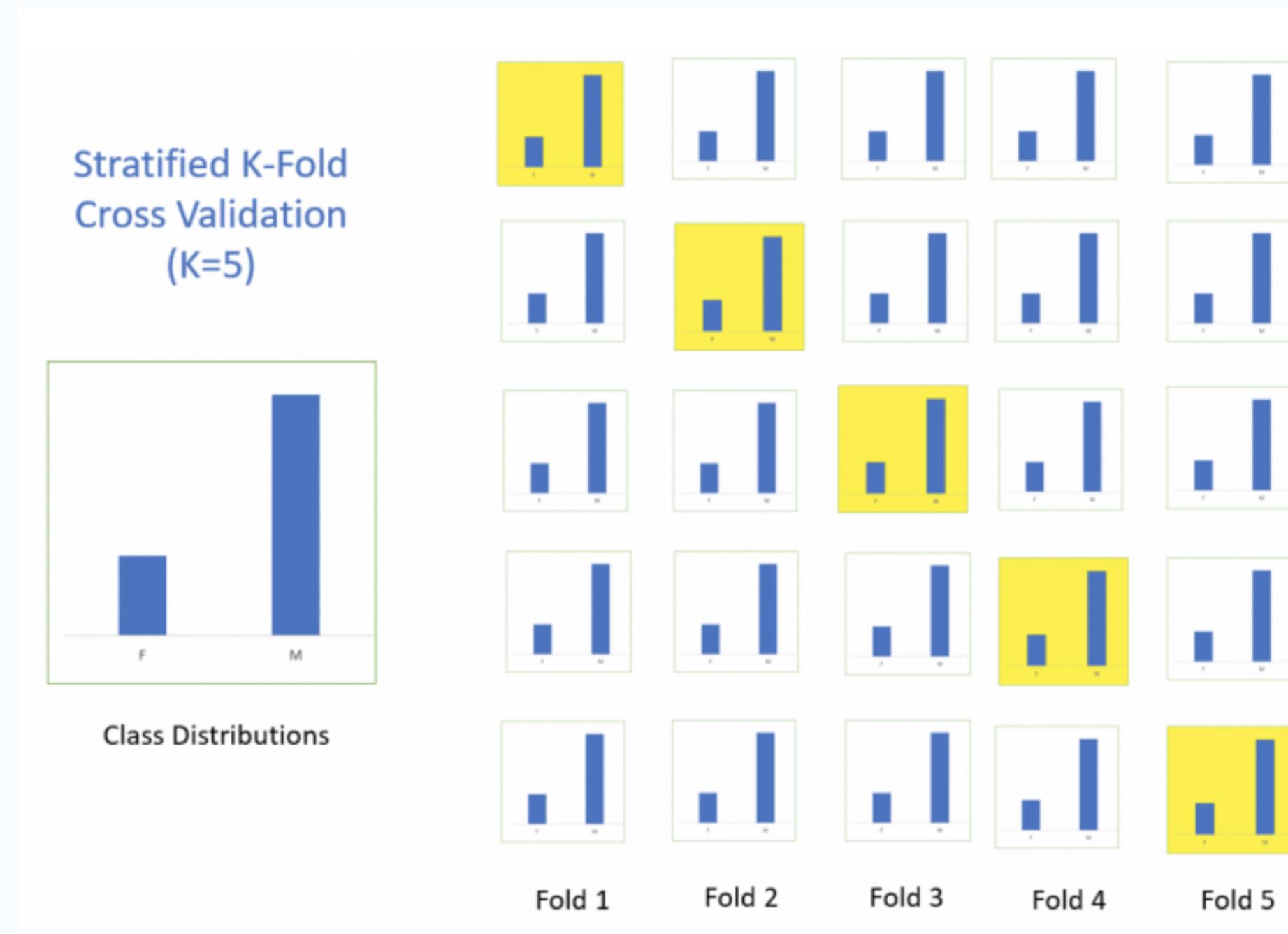
Round 10



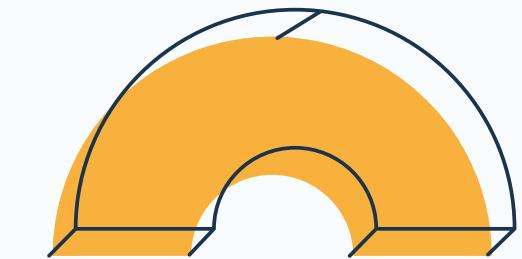
Cross validation - Stratified K fold



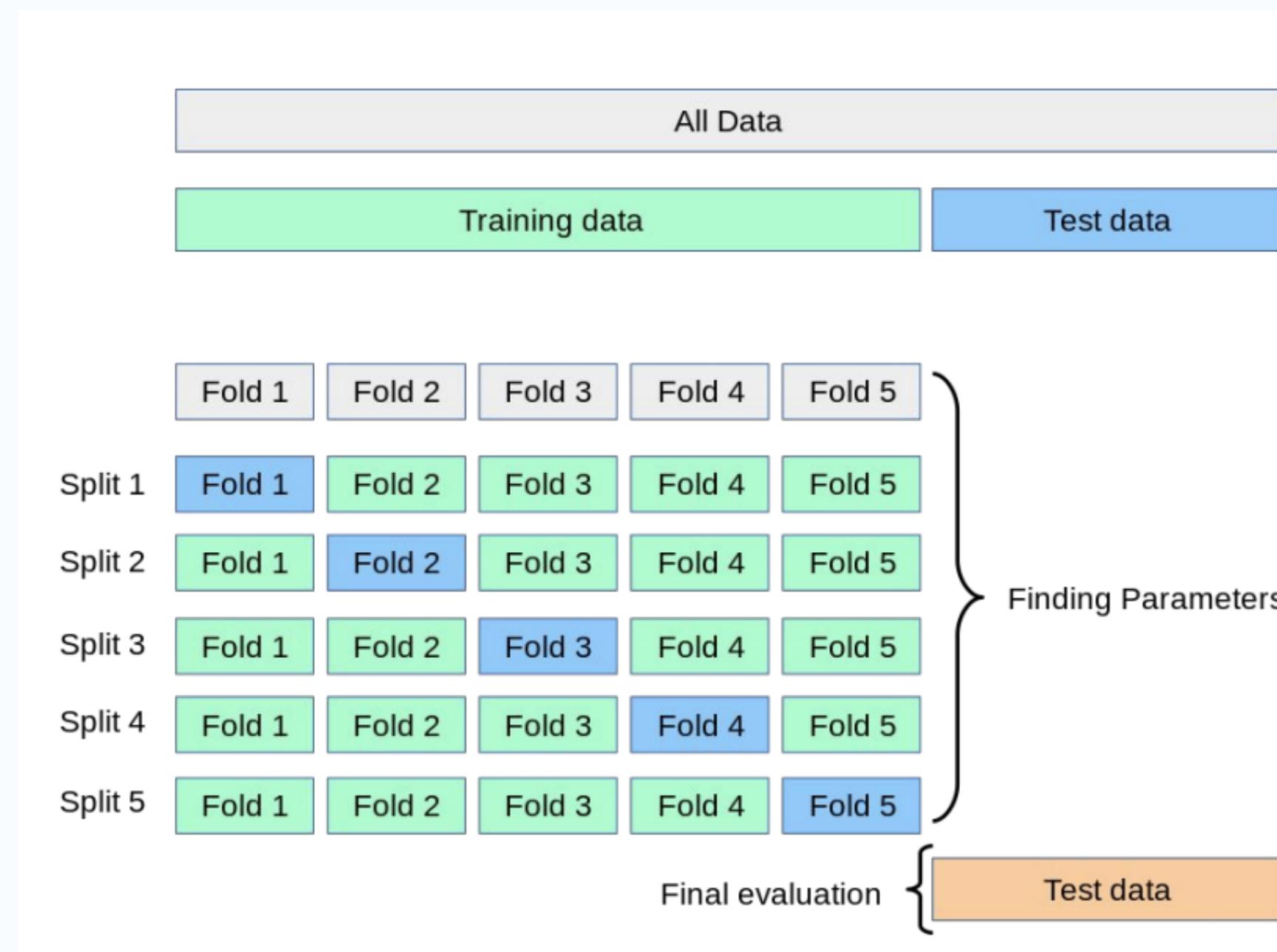
K fold no es el único método, existen otros. Por ejemplo: Stratified K fold



Cross validation - Proceso completo



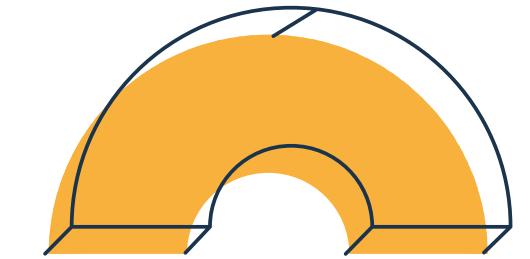
Aunque ahora evaluemos nuestro modelo con cross validation, tenemos que seguir haciendo train - test split





Scikitlearn pipelines

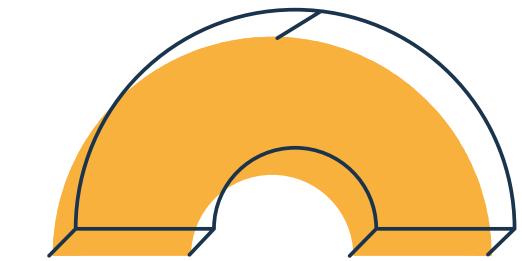
Pipelines



Vimos que cuando entrenamos un modelo, tenemos una serie de pasos que aplicar a el conjunto de train y luego al de test, entre ellos:

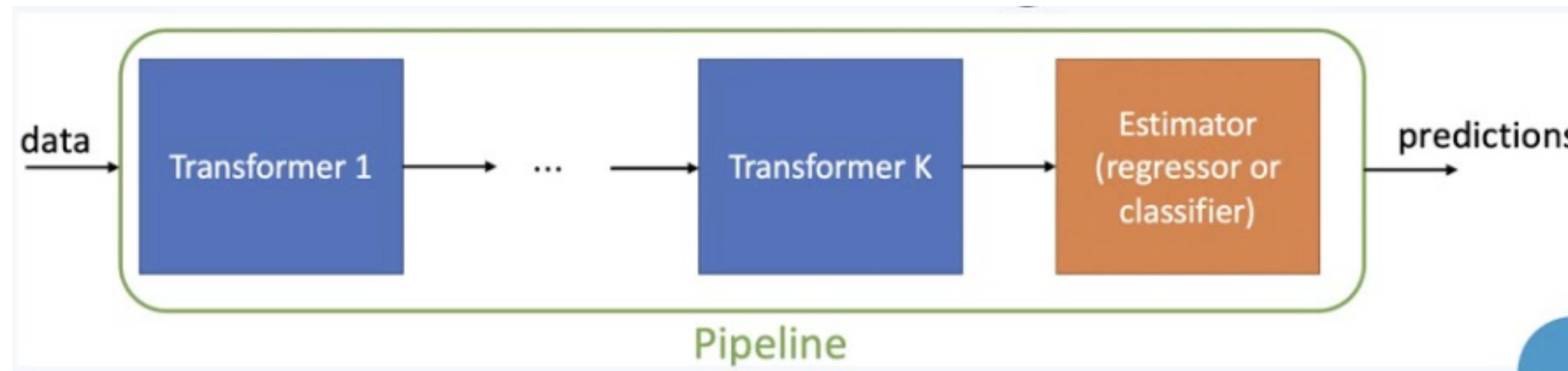
- Procesar datos nulos / erroneos
- One hot encoding
- Discretización
- Escalar los datos
- Etc

Pipelines



Sklearn nos provee una implementación de "pipeline" que nos permite armar un objeto que se encargue de hacer todo este preprocesamiento y generar las predicciones.

<https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html>

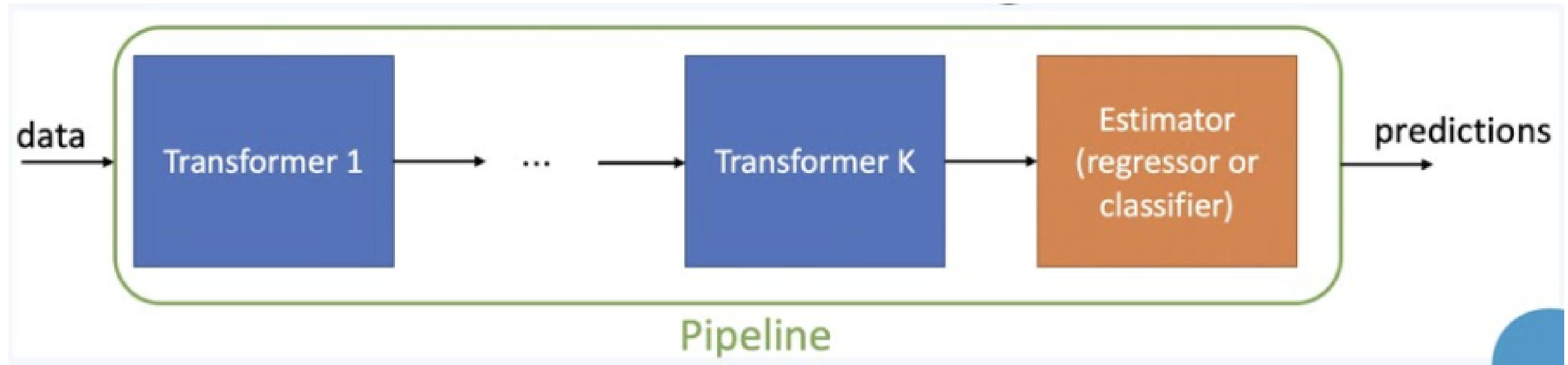


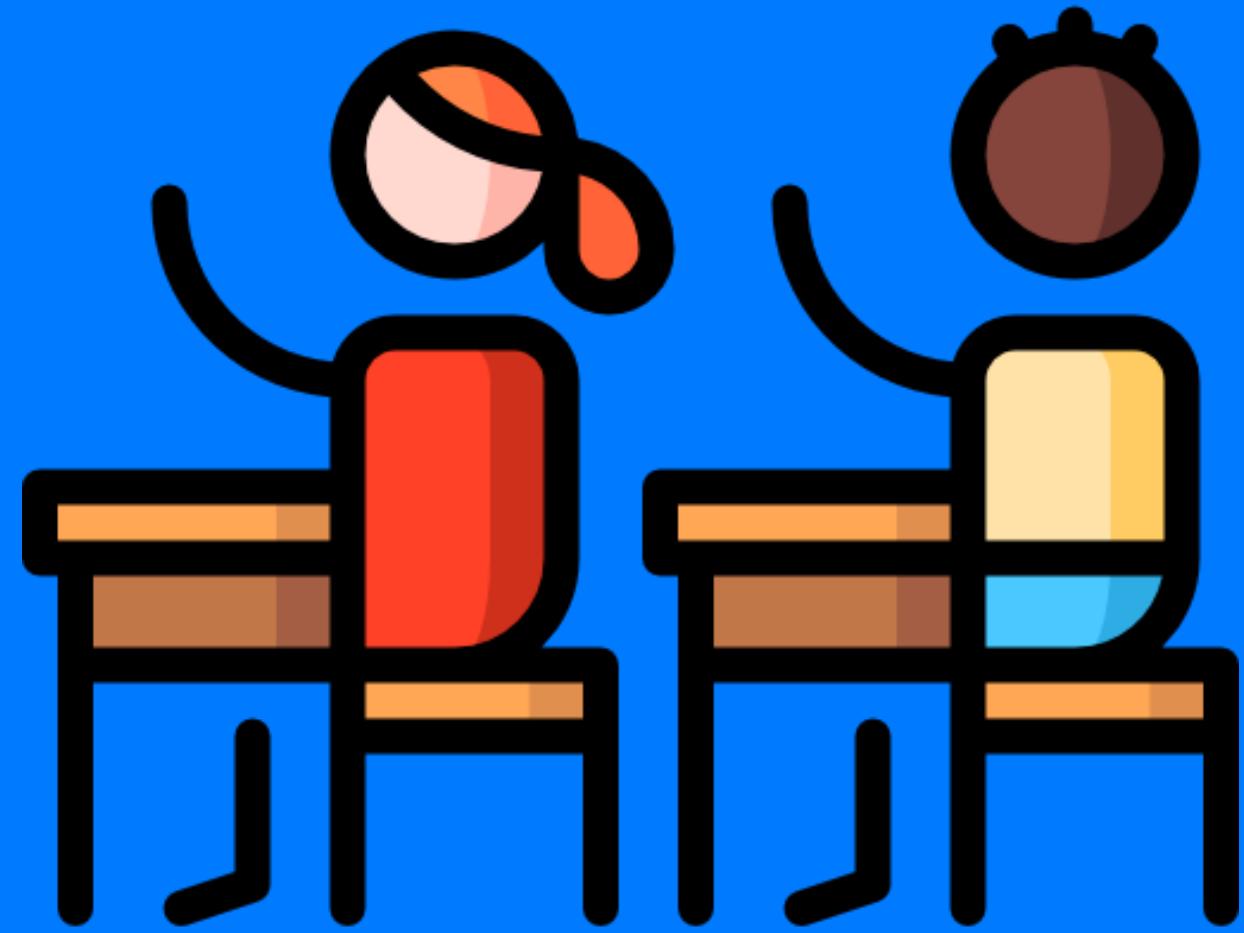
Esto nos va a permitir tener un único objeto al cual hacerle "fit" y "transform" o "predict"

Pipelines



Nuestro **estimador** pasa a ser una **cadena de estimadores** a la cual llamaremos **pipeline**





Práctica