DATA SCIENCE

Martin Jaureguy
martin.jaureguy.95@gmail.com

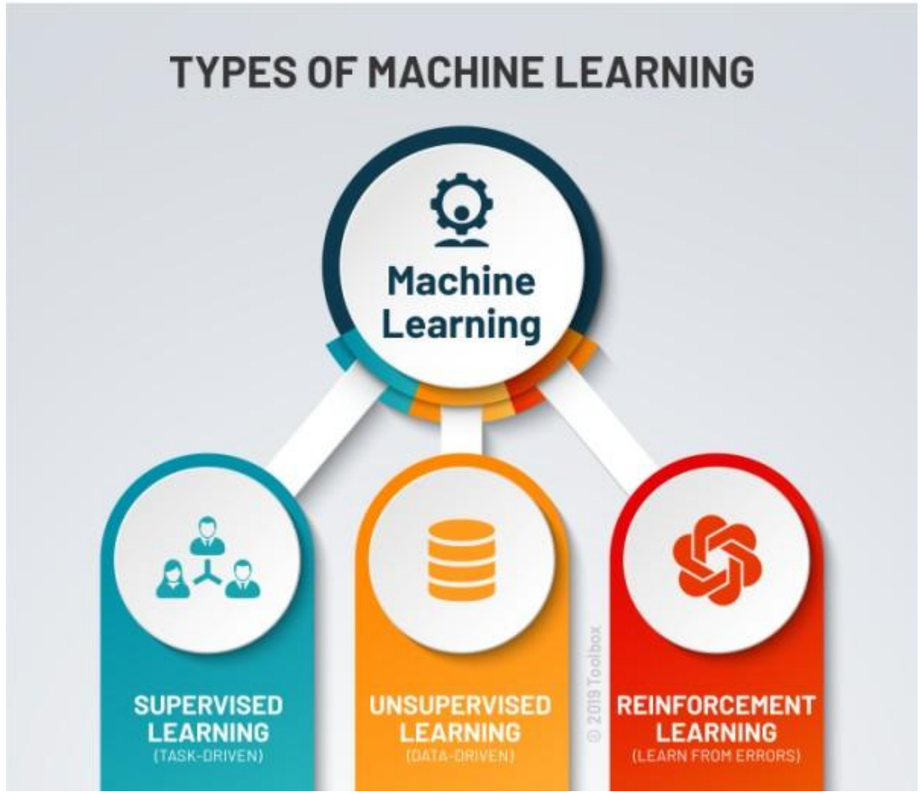
Clase 21 - Agenda

CROSS VALIDATION - SKLEARN PIPELINES

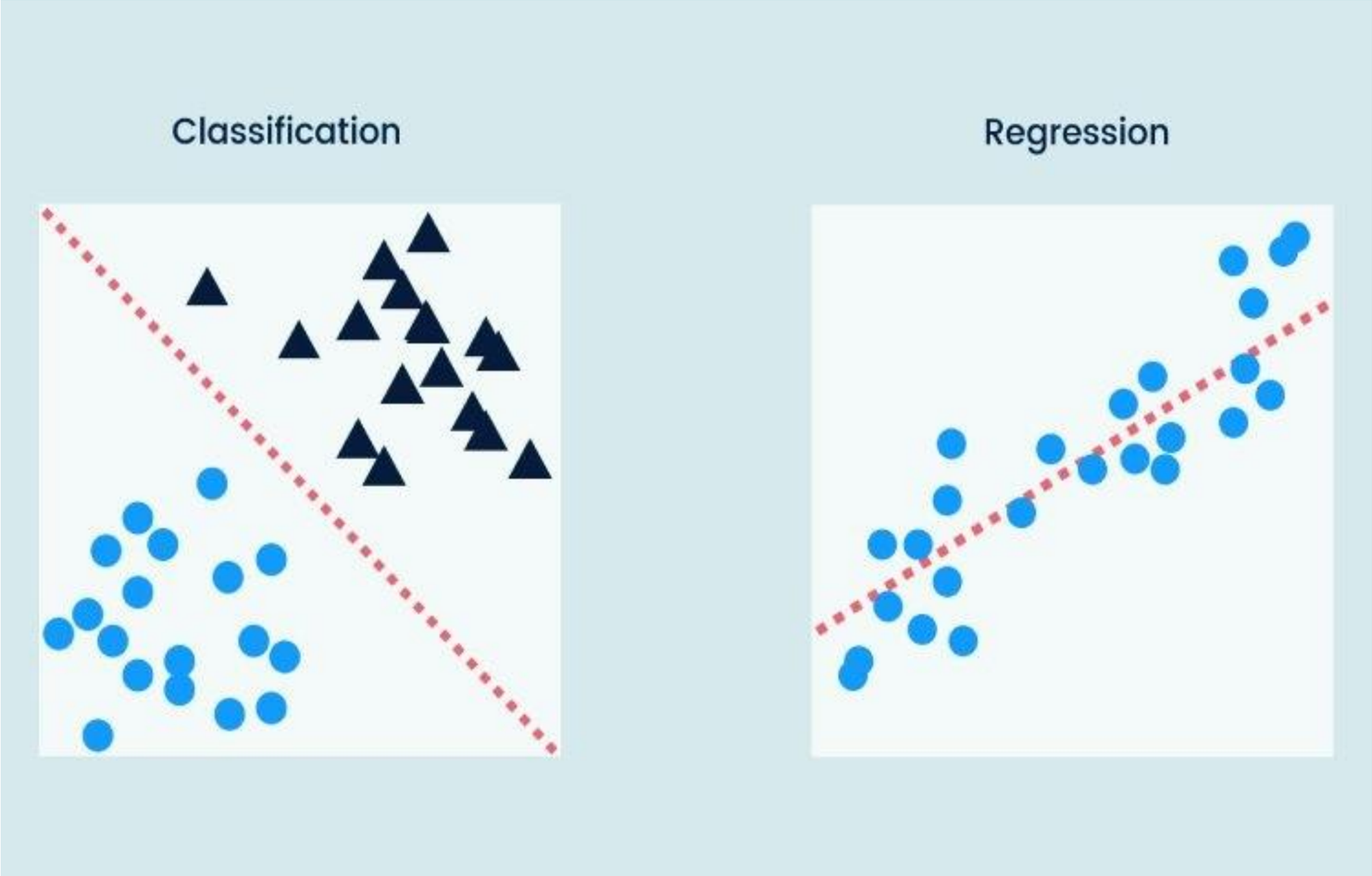
Repaso

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

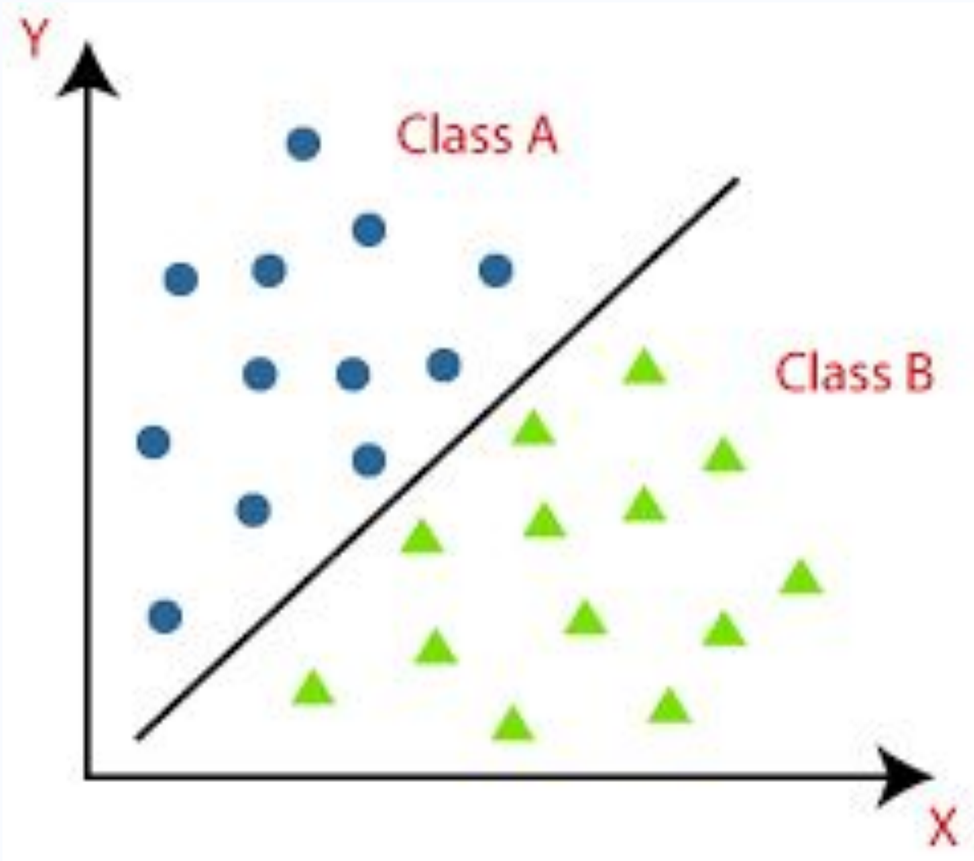
A diferencia de los parámetros, estos no se "aprenden". Son valores que definimos nosotros antes de entrenar. Por ejemplo:

-max_depth de un

Clasificación

Modelos:
- KNN
- Decision tree

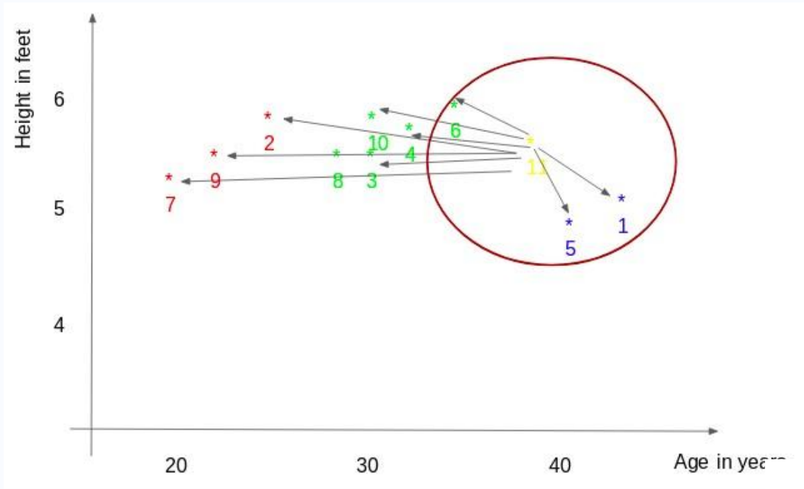
Métricas:
- Accuracy
- Precision
- Recall
- F1 Score

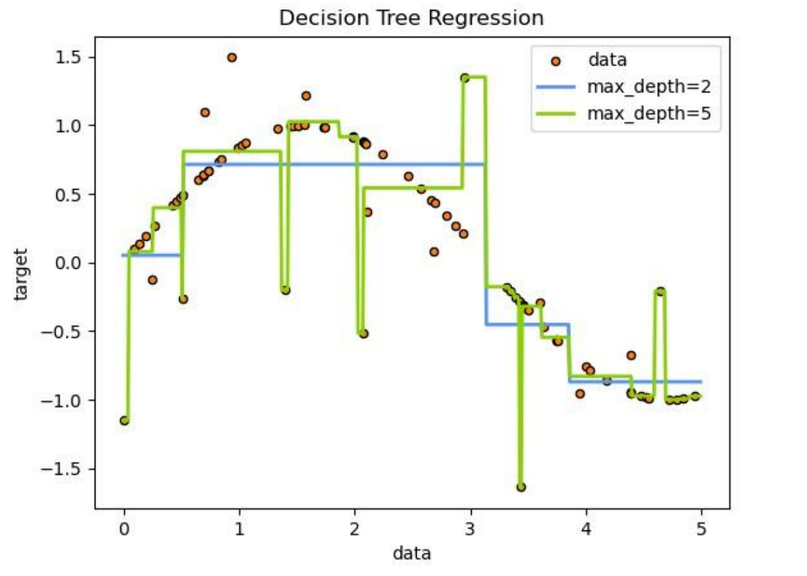


Regresión

Modelos:
- Regresión lineal
- Decision tree
- KNN

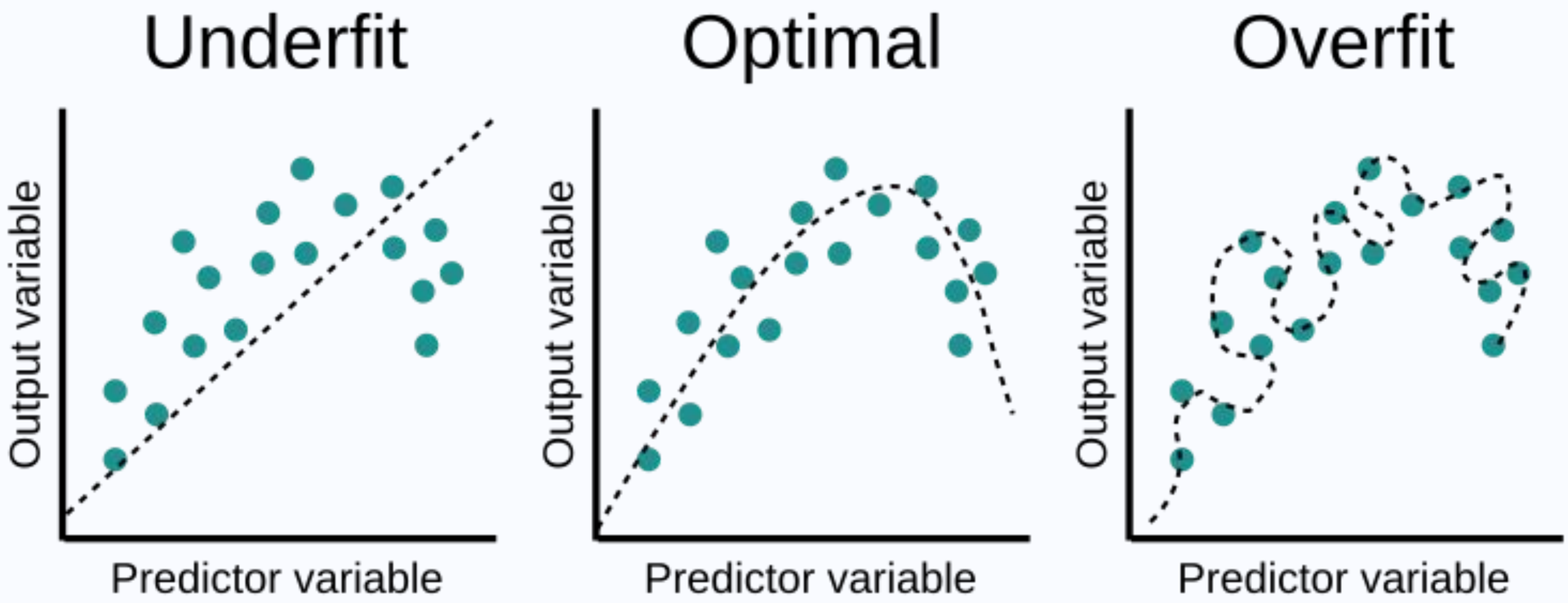
Métricas:
- R squared
- MAE
- MSE





Overfitting

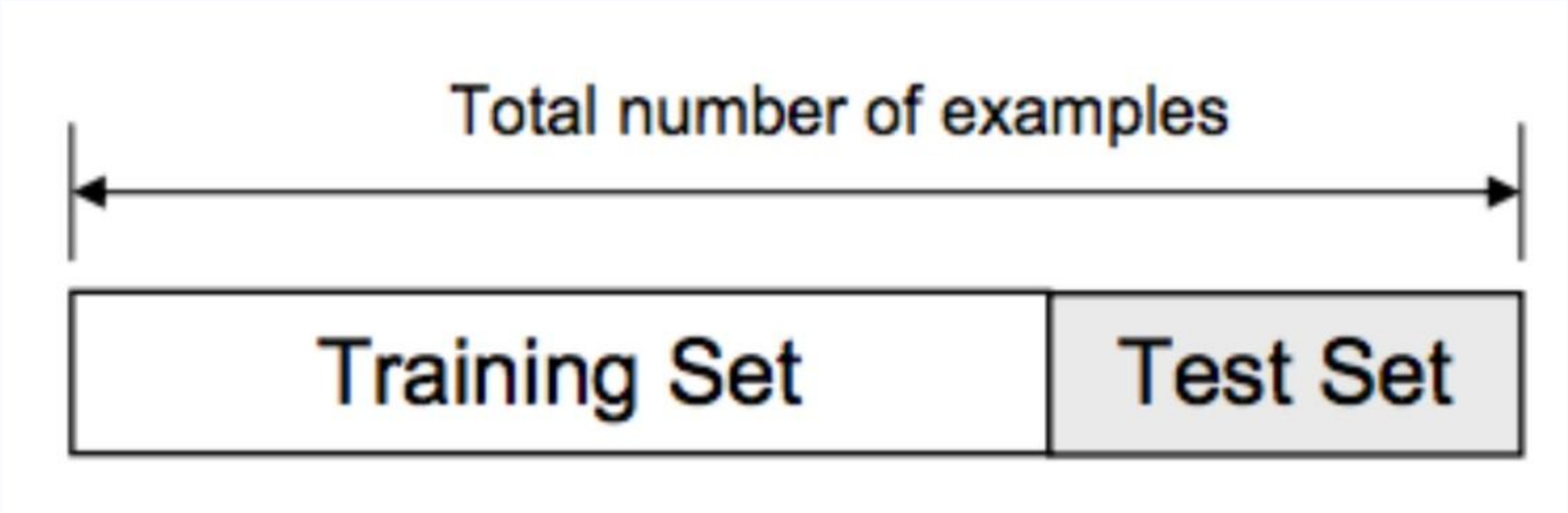
Un problema que afecta tanto a los modelos de clasificación como a los de regresión



Evaluación de modelos

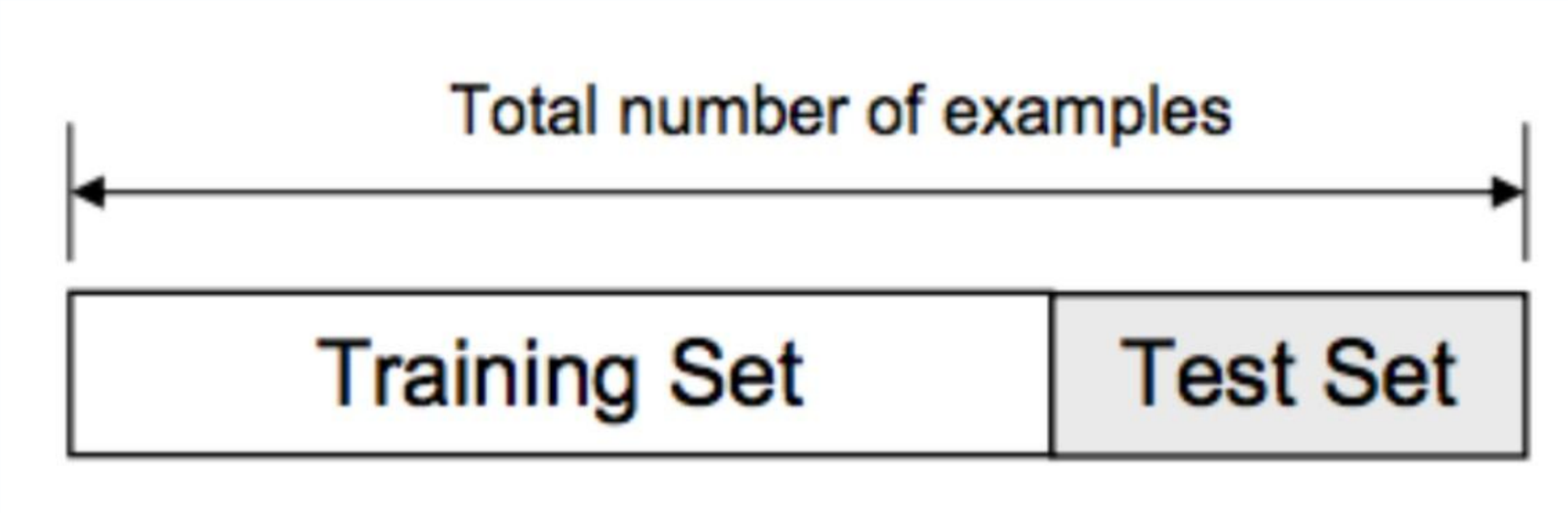
Train-Test split

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



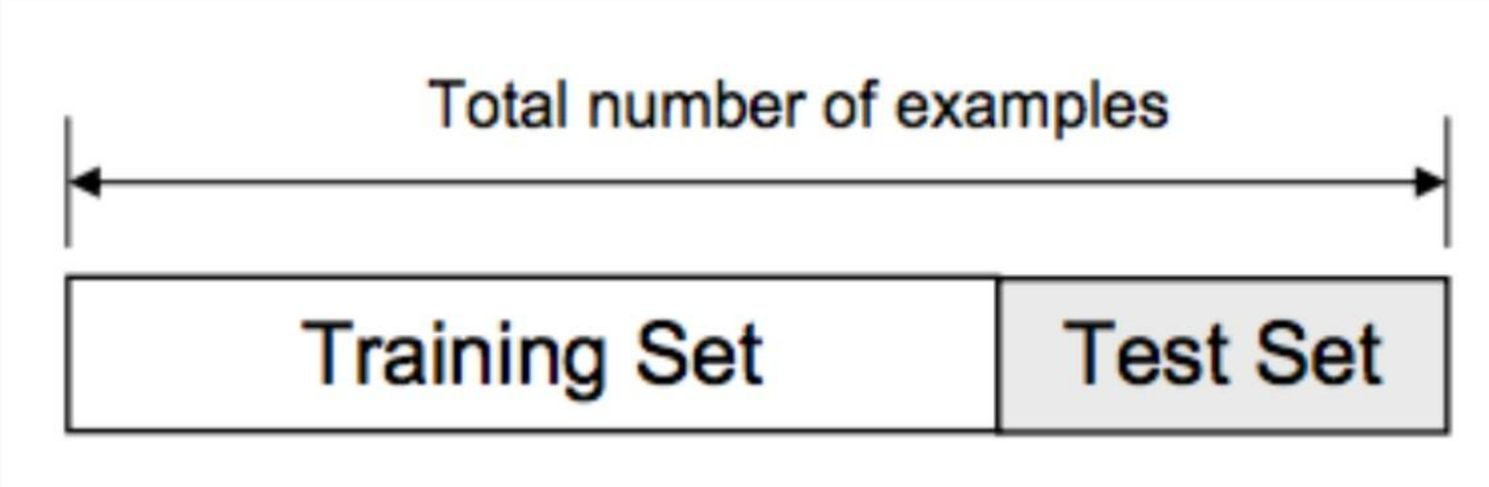
Train - test split

¿ Para qué lo hacemos ?



Train - test split

¿ Para qué lo hacemos ?



Evaluamos nuestros modelos "simulando" la realidad

Cross validation

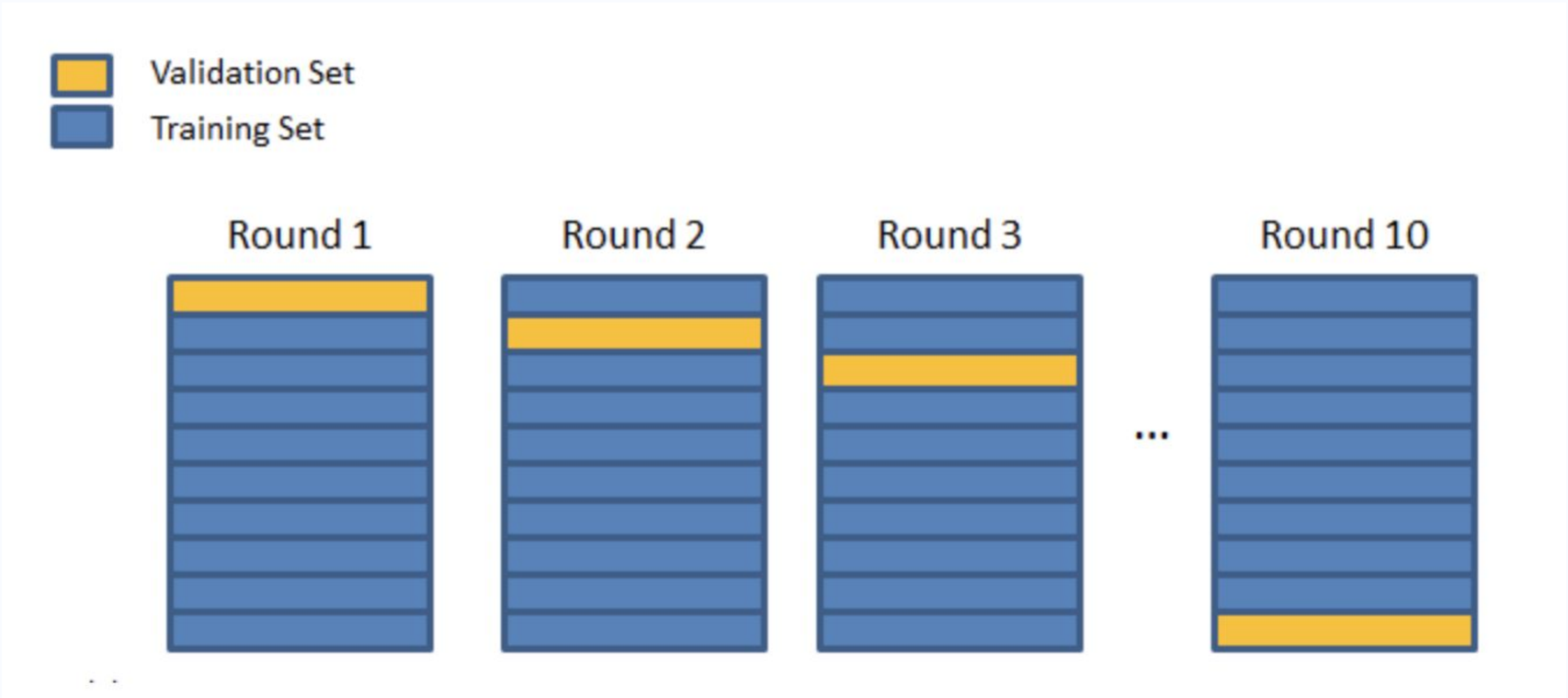
Existe un método que nos permite evaluar mejor la performance de nuestros modelos (asegurándonos de que no haya overfitting o que funcione bien sobre el set de test por casualidad)

PROCESO DE 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

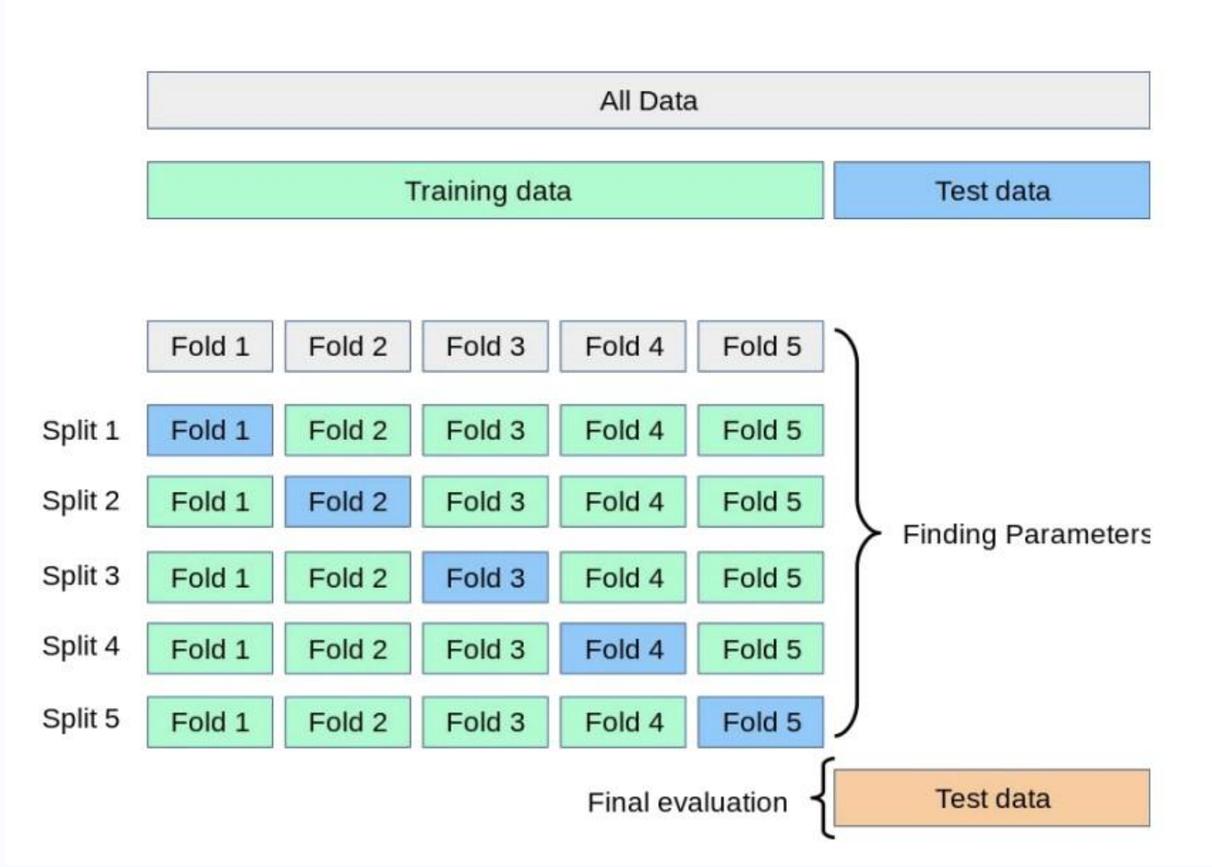
Cross validation - K fold

Este método es conocido como K-fold cross validation y está implementado en sklearn
El set de validación está dentro del train. Es la porción de datos que va a servir para testear cada iteración



Cross validation - Proceso completo

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



Cross validation - Otros métodos

Stratified K-Fold: cómo su palabra lo dice, busca que en cada iteración (k) se mantengan equilibrada la clase al hacer la división dentro del set de train y validación.

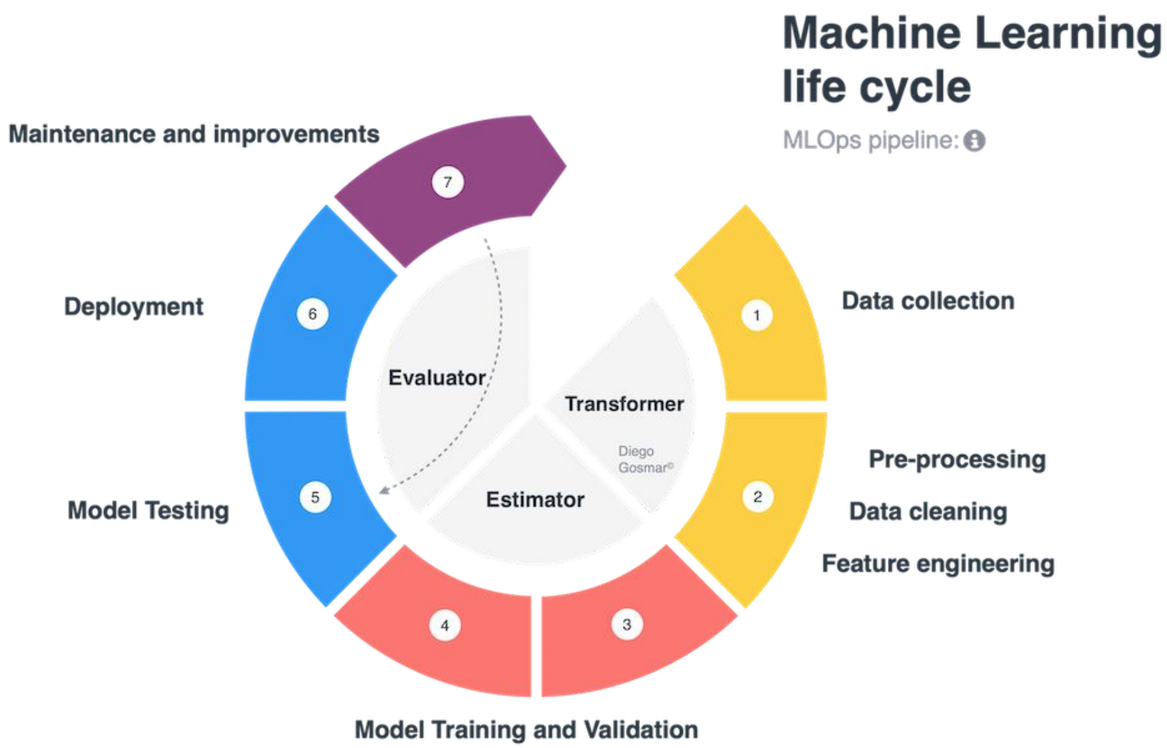
Leave P Out: selecciona una cantidad P determinada (ej: 50). Se separarán de a 50 muestras contra las cuales validar y se iterará de igual manera que las otras variantes. Si el valor P es pequeño, esto resultará en muchísimas iteraciones de entrenamiento con un alto coste computacional (y seguramente en tiempo). Si el valor P es muy grande, podría contener más muestras que las usadas para entrenamiento, lo cual sería absurdo.

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:

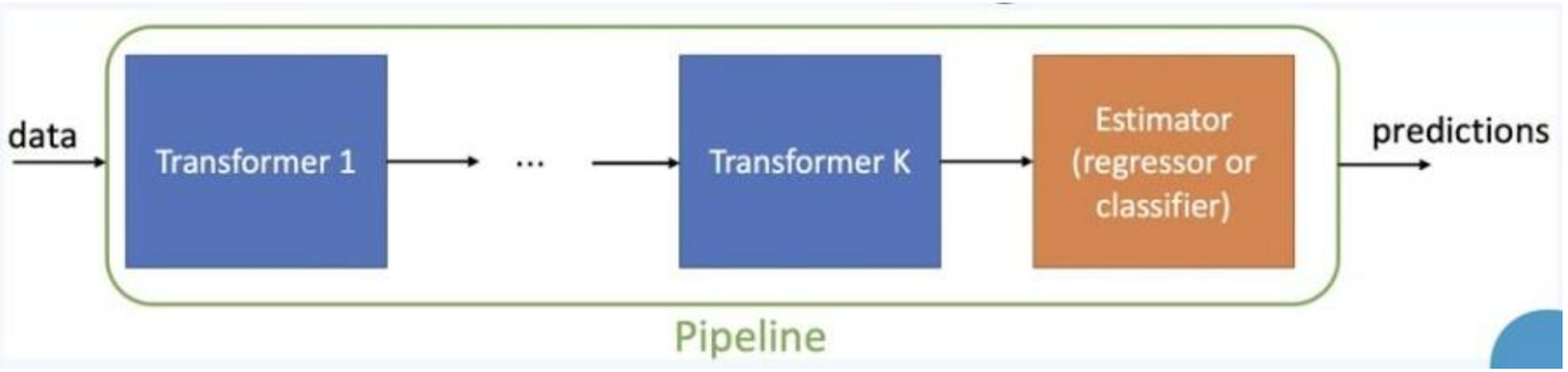
- Procesar datos nulos / erroneos
- One hot encoding
- Discretización
- Escalar los datos
- Entrenar modelos
- Etc



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

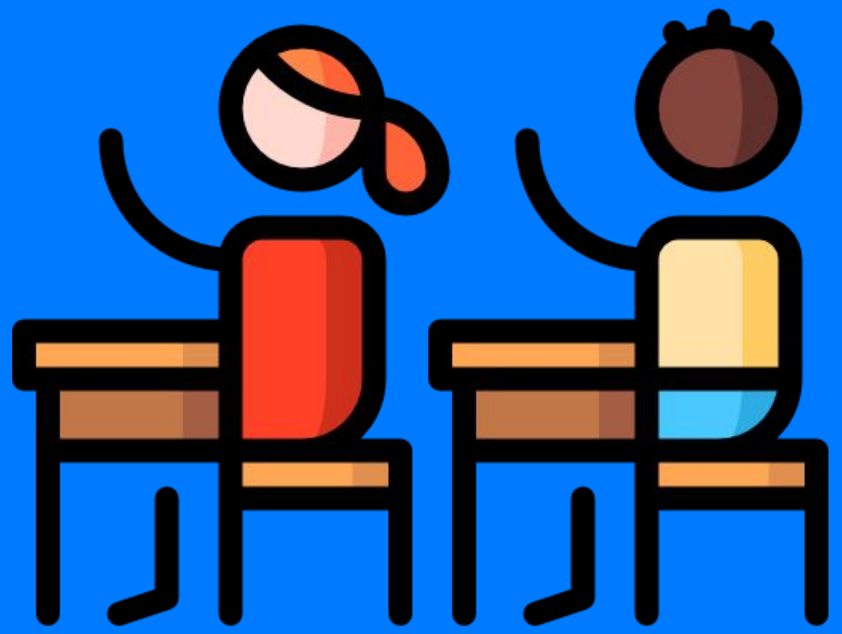


Esto nos va a permitir tener un único objeto al cual hacerle "fit" y "transform" o "predict"

Pipelines

Finalmente el resultado es un objeto que recibió todas las instrucciones de lo que tiene que hacer, y solo le pasaremos los datos de entrada utilizando el .fit con los dato de test.

Mismo sci-kit learn nos provee de diferentes métodos en caso de que querramos saber que pasos se realizaron, que score se obtuvo e incluso realizar las predicciones a partir de este pipeline



Práctic
a