

# Machine Learning

## Métricas de evaluación

Christian Oliva Moya  
Luis Fernando Lago Fernández

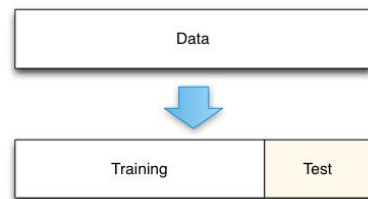
# Evaluación de modelos

---

- Disponemos de un único dataset. El procedimiento natural es separarlo (split) en dos subconjuntos **disjuntos** para generar los datasets de entrenamiento y test



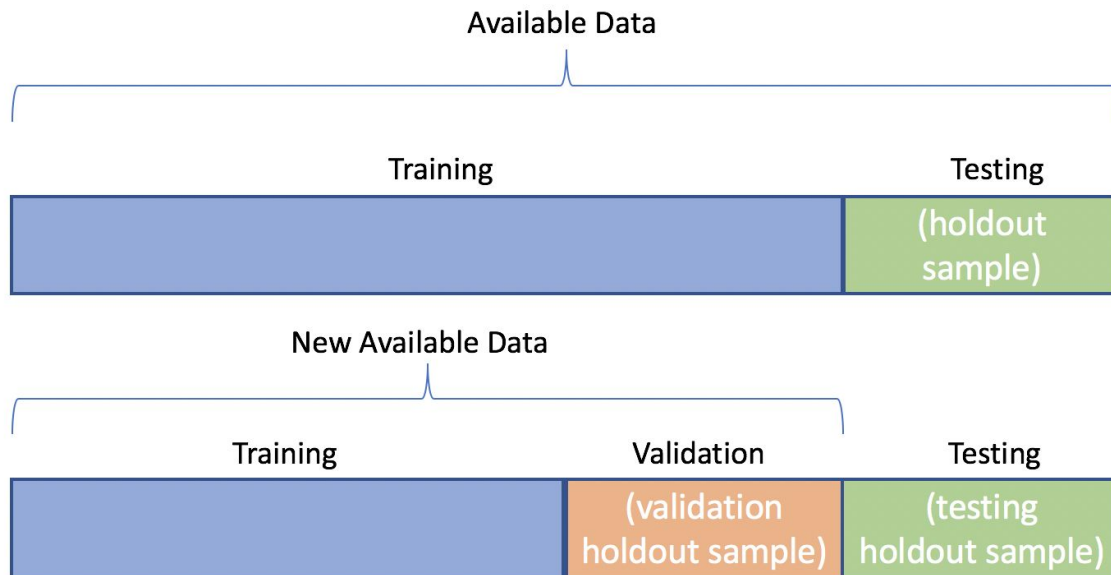
# Evaluación de modelos



- Una vez tenemos el conjunto de entrenamiento y test por separado, solamente podemos trabajar con los datos de training:
  - Si normalizamos, usamos la media y la std de entrenamiento
  - Para seleccionar hiperparámetros, usamos los datos de entrenamiento
  - Para seleccionar el mejor modelo, usamos los datos de entrenamiento
  - **NO podemos decidir qué modelo usamos utilizando test**
- Test se utilizará para simular una prueba en un escenario desconocido (producción)
- ¿Entonces cómo evaluamos los modelos?

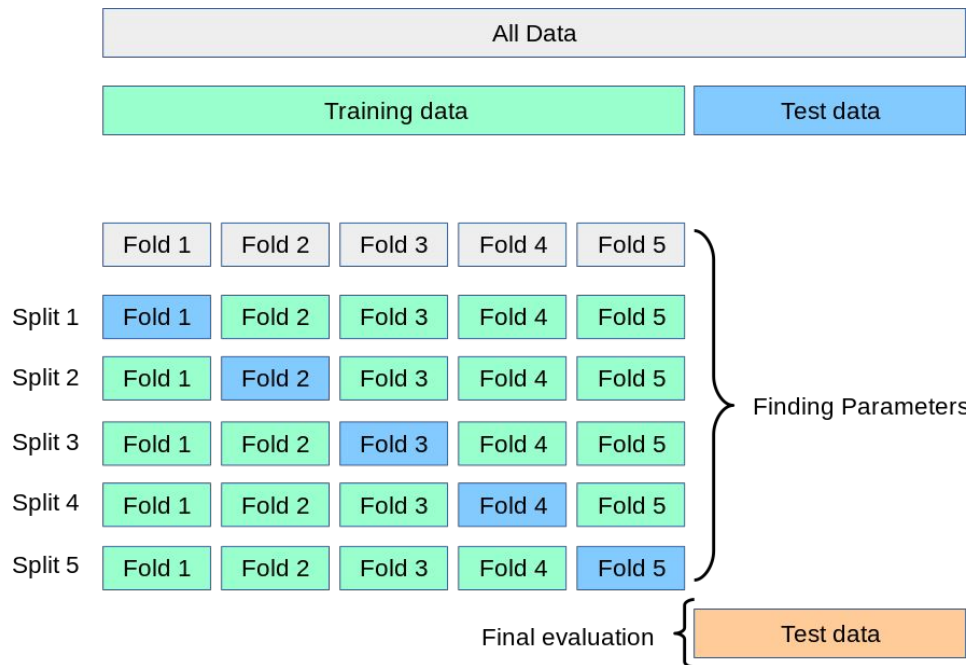
# Evaluación de modelos

- Usamos técnicas de validación. El conjunto de entrenamiento lo volvemos a dividir.
  - **Validación simple:** Un nuevo split del conjunto de entrenamiento.



# Evaluación de modelos

- Usamos técnicas de validación. El conjunto de entrenamiento lo volvemos a dividir.
  - **Validación Cruzada (KFold):** K splits, usamos K-1 para entrenar y 1 para validar



# Evaluación de modelos

---

- ¿Qué métricas utilizamos para validar/testear nuestros modelos?
  - ¿En regresión?
  - ¿En clasificación?

# Evaluación de modelos

- En problemas de **regresión** utilizamos métricas de error:

- **Error Absoluto Medio** (MAE - Mean Absolute Error)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - t_i|$$

- **Error Cuadrático Medio** (MSE - Mean Squared Error)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - t_i)^2$$

- **Coeficiente R2**

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - t_i)^2}{\sum_{i=1}^n (y_i - \mu)^2}$$

# Evaluación de modelos

- En problemas de **clasificación** utilizamos métricas relacionadas con la matriz de confusión:

|                  |          | clase real              |                         |
|------------------|----------|-------------------------|-------------------------|
|                  |          | positiva                | negativa                |
| clase predicción | positiva | verdadero positivo (TP) | falso positivo (FP)     |
|                  | negativa | falso negativo (FN)     | verdadero negativo (TN) |



# Evaluación de modelos

|                  |          | clase real              |                         |
|------------------|----------|-------------------------|-------------------------|
|                  |          | positiva                | negativa                |
| clase predicción | positiva | verdadero positivo (TP) | falso positivo (FP)     |
|                  | negativa | falso negativo (FN)     | verdadero negativo (TN) |

- En problemas de **clasificación** utilizamos métricas relacionadas con la matriz de confusión:

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

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

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

$$F1score = \frac{2 \times precision \times recall}{precision + recall}$$

# Evaluación de modelos

- El accuracy representa el porcentaje de acierto. Cuantos son True frente al total

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

|                  |          | clase real              |                         |
|------------------|----------|-------------------------|-------------------------|
|                  |          | positiva                | negativa                |
| clase predicción | positiva | verdadero positivo (TP) | falso positivo (FP)     |
|                  | negativa | falso negativo (FN)     | verdadero negativo (TN) |

# Evaluación de modelos

- La precisión representa cuántos son realmente True frente a los que digo que son True

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

|                  |          | clase real              |                         |
|------------------|----------|-------------------------|-------------------------|
|                  |          | positiva                | negativa                |
| clase predicción | positiva | verdadero positivo (TP) | falso positivo (FP)     |
|                  | negativa | falso negativo (FN)     | verdadero negativo (TN) |

# Evaluación de modelos

- El recall representa cuántos de los verdaderos True yo he clasificado como True

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

|                  |          | clase real              |                         |
|------------------|----------|-------------------------|-------------------------|
|                  |          | positiva                | negativa                |
| clase predicción | positiva | verdadero positivo (TP) | falso positivo (FP)     |
|                  | negativa | falso negativo (FN)     | verdadero negativo (TN) |

# Evaluación de modelos

- El F1 score representa una media armónica entre precisión y recall

Es alta cuando tanto la precisión como el recall son altos

$$F1score = \frac{2 \times precision \times recall}{precision + recall}$$

|                  |          | clase real              |                         |
|------------------|----------|-------------------------|-------------------------|
|                  |          | positiva                | negativa                |
| clase predicción | positiva | verdadero positivo (TP) | falso positivo (FP)     |
|                  | negativa | falso negativo (FN)     | verdadero negativo (TN) |

# Evaluación de modelos

Clasificamos según un umbral de confianza. Lo normal es 0.5

| Pred | Class | MC |
|------|-------|----|
| 0.95 | 1     | TP |
| 0.85 | 1     | TP |
| 0.75 | 0     | FP |
| 0.65 | 1     | TP |
| 0.55 | 0     | FP |
| 0.45 | 1     | FN |
| 0.35 | 0     | TN |
| 0.25 | 1     | FN |
| 0.15 | 0     | TN |
| 0.05 | 0     | TN |

umbral = 0.5

accuracy = 6 / 10 = 0.6

precision = 3 / 5 = 0.6

recall = 3 / 5 = 0.6

F1 score = 0.6

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

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

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

$$F1score = \frac{2 \times precision \times recall}{precision + recall}$$

# Evaluación de modelos

Si buscamos un clasificador más preciso aumentamos el umbral

| Pred | Class | MC |
|------|-------|----|
| 0.95 | 1     | TP |
| 0.85 | 1     | TP |
| 0.75 | 0     | FP |
| 0.65 | 1     | FN |
| 0.55 | 0     | TN |
| 0.45 | 1     | FN |
| 0.35 | 0     | TN |
| 0.25 | 1     | FN |
| 0.15 | 0     | TN |
| 0.05 | 0     | TN |

umbral = 0.7

accuracy = 6 / 10 = 0.6

precision = 2 / 3 = 0.67

recall = 2 / 5 = 0.4

F1 score = 0.5

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

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

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

$$F1score = \frac{2 \times precision \times recall}{precision + recall}$$

# Evaluación de modelos

Si buscamos un clasificador más preciso aumentamos el umbral

| Pred | Class | MC |
|------|-------|----|
| 0.95 | 1     | TP |
| 0.85 | 1     | FN |
| 0.75 | 0     | TN |
| 0.65 | 1     | FN |
| 0.55 | 0     | TN |
| 0.45 | 1     | FN |
| 0.35 | 0     | TN |
| 0.25 | 1     | FN |
| 0.15 | 0     | TN |
| 0.05 | 0     | TN |

umbral = 0.9

accuracy = 6 / 10 = 0.6

precision = 1 / 1 = 1.0

recall = 1 / 5 = 0.2

F1 score = 0.33

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

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

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

$$F1score = \frac{2 \times precision \times recall}{precision + recall}$$



# Evaluación de modelos

Si buscamos un clasificador más sensible disminuimos el umbral

| Pred | Class | MC |
|------|-------|----|
| 0.95 | 1     | TP |
| 0.85 | 1     | TP |
| 0.75 | 0     | FP |
| 0.65 | 1     | TP |
| 0.55 | 0     | FP |
| 0.45 | 1     | TP |
| 0.35 | 0     | FP |
| 0.25 | 1     | FN |
| 0.15 | 0     | TN |
| 0.05 | 0     | TN |

umbral = 0.3

accuracy = 6 / 10 = 0.6

precision = 4 / 7 = 0.57

recall = 4 / 5 = 0.8

F1 score = 0.66

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

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

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

$$F1score = \frac{2 \times precision \times recall}{precision + recall}$$

# Evaluación de modelos

Si buscamos un clasificador más sensible disminuimos el umbral

| Pred | Class | MC |
|------|-------|----|
| 0.95 | 1     | TP |
| 0.85 | 1     | TP |
| 0.75 | 0     | FP |
| 0.65 | 1     | TP |
| 0.55 | 0     | FP |
| 0.45 | 1     | TP |
| 0.35 | 0     | FP |
| 0.25 | 1     | TP |
| 0.15 | 0     | TN |
| 0.05 | 0     | TN |

umbral = 0.2

accuracy = 7 / 10 = 0.7

precision = 5 / 8 = 0.625

recall = 5 / 5 = 1.0

F1 score = 0.77

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

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

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

$$F1score = \frac{2 \times precision \times recall}{precision + recall}$$

# Clasificación multi-clase

---

- Un problema de ML es multi-clase cuando hay más de 2 posibles etiquetas
  - Problema del Iris: ¿setosa, versicolor o virginica?
  - Trading: ¿compro, vendo o me quedo quieto?
- Algunos clasificadores responden de forma natural al problema multiclase:
  - KNN mira los más cercanos y selecciona la clase más probable
- Otros clasificadores son binarios por naturaleza:
  - SVM
  - Naive Bayes
  - Regresión logística

# Clasificación multi-clase

- Enfoque One-vs-rest:
  - Se entrena un modelo para cada clase
  - Cada modelo predice si pertenece o no a la clase particular
  - Se combinan las salidas para dar la predicción multi-clase

| $x_1 \dots x_n$       | Clase | EsA? | EsB? | EsM? | Predicción |
|-----------------------|-------|------|------|------|------------|
| $v_{11} \dots v_{1m}$ | A     | Si   | No   | No   | A          |
| ...                   | B     | No   | Si   | No   | B          |
| ...                   | M     | Si   | No   | No   | A          |
| $v_{n1} \dots v_{nn}$ | A     | Si   | No   | Si   | ¿?         |

# Clasificación multi-clase

- Enfoque One-vs-rest:

- Podemos predecir la clase de la fila ¿? con el clasificador binario que proporcione la mayor probabilidad
- Para expresar esta probabilidad multi-clase, **normalizamos** las probabilidades de cada modelo individual típicamente usando la función **Softmax**

| $x_1 \dots x_n$       | Clase | EsA? | EsB? | EsM? | Predicción |
|-----------------------|-------|------|------|------|------------|
| $v_{11} \dots v_{1m}$ | A     | Si   | No   | No   | A          |
| ...                   | B     | No   | Si   | No   | B          |
| ...                   | M     | Si   | No   | No   | A          |
| $v_{n1} \dots v_{nm}$ | A     | Si   | No   | Si   | ¿?         |

$$\text{softmax}(c_i|x_i) = \frac{e^{P(c_i|x_i)}}{\sum_{c_j \in C} e^{P(c_j|x_i)}}$$

# Clasificación multi-etiqueta

---

- Un problema de ML es multi-etiqueta cuando puede haber varias clases objetivo a la vez
  - ¿Qué hay en la imagen? Un perro, dos personas y un coche al fondo
  - ¿Temática de un texto? Política, economía y salud mental
  - ¿Categorías de películas? Acción, terror, ciencia ficción
- Algunos clasificadores permiten la estrategia multiclase:
  - Redes neuronales
- Otros no lo permiten:
  - KNN, ¿cómo podría elegir varias clases a la vez si solo mira el más cercano?
  - Los binarios, como SVM o Naive Bayes, ¿cómo eligen varios si solo son 2 posibles?

# Clasificación multi-etiqueta

- Enfoque One-vs-all:
  - Se entrena un clasificador para cada clase
  - Todos los que predigan que pertenece se añade a la respuesta final

| $x_1 \dots x_n$       | <b>Etiquetas</b> | EsA? | EsB? | EsC? | <b>Predicción</b> |
|-----------------------|------------------|------|------|------|-------------------|
| $v_{11} \dots v_{1m}$ | A                | Si   | No   | No   | A                 |
| ...                   | A, B             | Si   | Si   | No   | A,B               |
| ...                   | A, B             | Si   | No   | Si   | A,C               |
| $v_{n1} \dots v_{nn}$ | A, B             | No   | No   | Si   | C                 |

- En esta situación, pueden salir **combinaciones que no existen** o no son válidas
- Además, puede existir **correlación entre clases**

# Clasificación multi-etiqueta

---

- Otras aproximaciones - ranking de etiquetas:
  - Representamos el problema como una tarea de ranking
  - Elegimos las N primeras etiquetas como salida
- Es una estrategia muy utilizada en sistemas de recomendación