Aprendizaje automático: Introducción

Marta Caro Martínez

Adaptado de Enrique Martín y Javier Arroyo

Datos en nuestro conjunto de datos

Sexo	Altura	Peso	Exp.	Act.
H	1,55	45	A+	Fútbol
H	1,67	58	C	Fútbol
M	1,45	45	B	Fútbol
M	1,58	50	A+	Pádel
H	1,20	40	B	Fútbol
M	1,80	60	A	Pádel
M	1,35	39	B	Fútbol
...	...	...	...	...

- Tenemos un conjunto de datos sobre alumnos de un colegio (sexo, altura, peso y nota media del expediente), así como qué actividad extraescolar realizan.

Datos en nuestro conjunto de datos

Sexo	Altura	Peso	Exp.	Act.
H	1,55	45	A+	Fútbol
H	1,67	58	C	Fútbol
M	1,45	45	B	Fútbol
M	1,58	50	A+	Pádel
H	1,20	40	B	Fútbol
M	1,80	60	A	Pádel
M	1,35	39	B	Fútbol
...	...	...	...	...

- Los conjuntos de datos están compuestos por ejemplos, llamados **instancias**.
- Cada fila de la tabla será una instancia.

Datos en nuestro conjunto de datos

Sexo	Altura	Peso	Exp.	Act.
H	1,55	45	A+	Fútbol
H	1,67	58	C	Fútbol
M	1,45	45	B	Fútbol
M	1,58	50	A+	Pádel
H	1,20	40	B	Fútbol
M	1,80	60	A	Pádel
M	1,35	39	B	Fútbol
...	...	...	...	...

- Cada instancia tiene una serie de valores para unos **atributos**, que describen la instancia.
- Cada columna en un conjunto de datos corresponde con un atributo.

Datos en nuestro conjunto de datos

Sexo	Altura	Peso	Exp.	Act.
H	1,55	45	A+	Fútbol
H	1,67	58	C	Fútbol
M	1,45	45	B	Fútbol
M	1,58	50	A+	Pádel
H	1,20	40	B	Fútbol
M	1,80	60	A	Pádel
M	1,35	39	B	Fútbol
...	...	...	...	...
M	**1,42**	**40**	**A**	**¿?**

- En algunos casos, hay un atributo especial, llamado **clase**.
- El objetivo del aprendizaje automático en este caso es predecir el valor de la clase para instancias nuevas.
- En este ejemplo, la clase es la actividad.

Datos en nuestro conjunto de datos

- Los atributos pertenecen a dos tipos principales: **categóricos y continuos.**
- Los atributos **categóricos** toman valores de un conjunto finito de valores posibles. Ejemplos:

  (Nominal) Tiempo: soleado, nublado, lluvioso

  (Ordinal) Valoración: malo < regular < bueno
- Los atributos **continuos** toman valores enteros o reales. Ejemplos: número de hijos (0,1,2…), temperatura (0°C, 35°F)

Aprendizaje supervisado

- Los conjuntos de datos que tienen clase se les llama **datos etiquetados**.
- El aprendizaje automático que usa datos etiquetados es el **aprendizaje automático supervisado**.
  - Si la clase es de tipo categórico: la tarea se conoce como **clasificación**
  - Si la clase es de tipo continuo, la tarea se conoce como **regresión**.

Tipos de clasificación

- **Single class**:
  - Clase que nos interesa
  - Otras
- **Binary class**: categorías excluyentes
  - Positiva y negativa, spam y no spam, aprobado o rechazado
- **Multiclass**:
  - Clase 1
  - Clase 2
  - Clase 3
  - …

Aprendizaje no supervisado

- Los conjuntos de datos que NO tienen clase se les llama **datos no etiquetados**.
- El aprendizaje automático que usa datos no etiquetados es el **aprendizaje automático no supervisado**.
  - Destacan la generación de reglas de asociación, que vinculan los valores de unos atributos con otros.
  - La obtención de grupos de instancias comunes (**clustering**)

Entender los datos

- Antes de aplicar técnicas de aprendizaje automático: **entendimiento de los datos**.
  - **Visualizar los datos**: determinar qué variables están relacionadas, o como se separan las clases
  - **Calcular estadísticos descriptivos**: media, mediana, moda, desviación típica, valores mínimos y máximos…
  - **Representar la distribución de las variables**: observar valores frecuentes y outliers

Análisis de los datos para entenderlos

- **Determinar el problema a resolver**:
  - Clasificación (binaria, muticlass…)
  - Regresión
  - …
- **Analizar el dataset**:
  - Describir los atributos del dataset → ¿cuántos hay? ¿de qué tipo?
  - Describir relación entre los atributos
  - Funciones útiles en python: describe, corr, dibujar histogramas de frecuencias de las variables (hist de matplotlib), pairplot de seaborn

Preparación de los datos

- Después de entender los datos y antes de aplicar técnicas de aprendizaje automático: **preparación de los datos**.
- Esta tarea trata de eliminar anomalías en las instancias del conjunto de datos. Entre otras:
  - El valor de un atributo categórico está mal escrito
  - Falta el valor de un atributo
  - Un valor continuo toma únicamente 5 valores diferenciados → convertir en atributo categórico
  - Un atributo toma siempre el mismo valor.
  - Hay valores extremos para un atributo.

Preparación de los datos

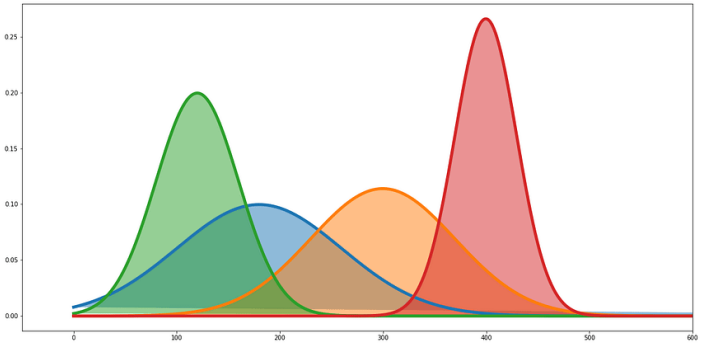
- Borrar filas con valores nulos
- Determinar si se puede eliminar algún atributo (aquellos que tengan correlación muy fuerte con otros)
- Convertir atributos categóricos a numéricos (**excepto para árboles de decisión y random forest**):
  - OneHotEncoder o getdummies: una columna por cada valor de atributo, valores de 0 y 1 si ese valor de atributo aparece en la instancia
    - Se usa cuando tenemos pocos valores de atributo
  - LabelEncoder: una única columna, valor numérico por cada valor categórico (1, 2, 3…)
    - Modelos de IA podrían asumir que hay una relación ordinal
    - Usar cuando hay muchos valores de atributo

Preparación de los datos

- Realizar normalización o estandarización si hace falta (puede ser distinta para cada modelo de IA)
  - **Objetivo: evitar sesgo** que se produce cuando los algoritmos de IA tienen en cuenta las variables con rangos más grandes
  - Normalización: MinMaxScaler [0,1]
  - Estandarización: StandarScaler [-1,1]

Normalización VS Estandarización

- **Normalización**: conocemos el valor mínimo y máximo que necesitamos que tengan los datos. Normalmente:
  - Modelos de IA basados en distancias (kNN, clustering, CBR…)
  - Redes neuronales
  - Datasets de imágenes o time series
- **Estandarización**: cuando los datos van a tener una distribución gaussiana (media cercana a 0, desviación estándar cercana a 1):
  - Variables con diferentes unidades de medida
  - Muchas variables distintas
  - Modelos de IA sensibles a la escala (clustering, SVM, regresión logística…)



Conjuntos de datos

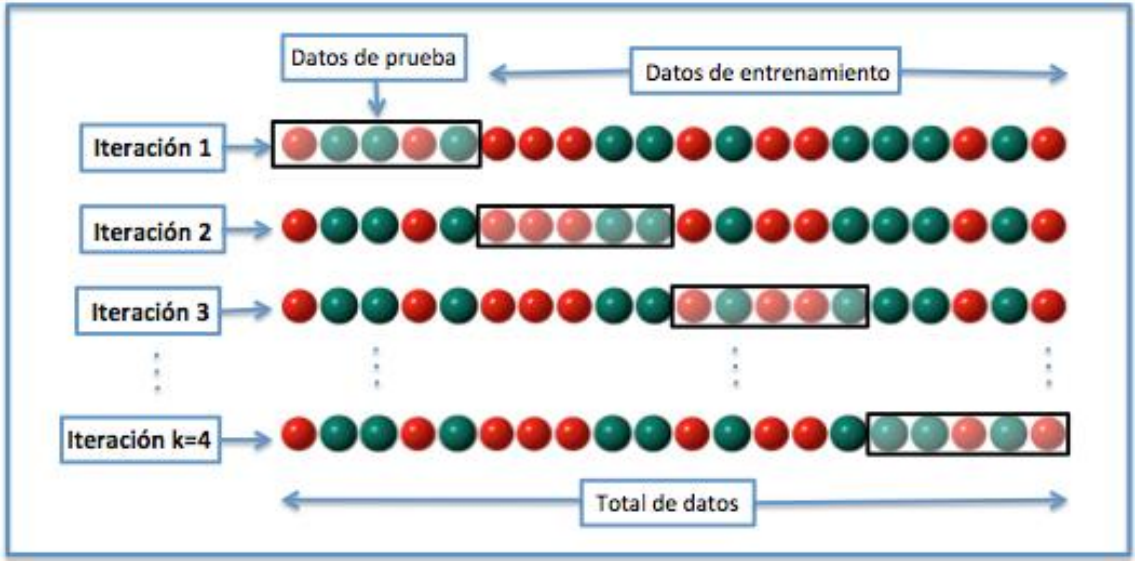
- Para entrenar un modelo de IA y evaluar nuestras técnicas de aprendizaje automático, se divide el conjunto de datos en dos conjuntos de datos:
  - El **conjunto de entrenamiento (training set)**. Sirve para alimentar al algoritmo de aprendizaje máquina.
  - El **conjunto de test (test set)**. Son instancias diferentes al conjunto de entrenamiento. Sirve para medir la calidad de los resultados obtenidos.

Conjuntos de datos

- Se puede hacer una partición simple (aleatoria o no). Por ejemplo:
  - 80% de las instancias → entrenamiento
  - 20% de las instancias → evaluación
- Se puede usar otras técnicas:
  - K-fold-cross-validation
  - Random cross validation
  - Leave-one-out

K-fold-cross-validation

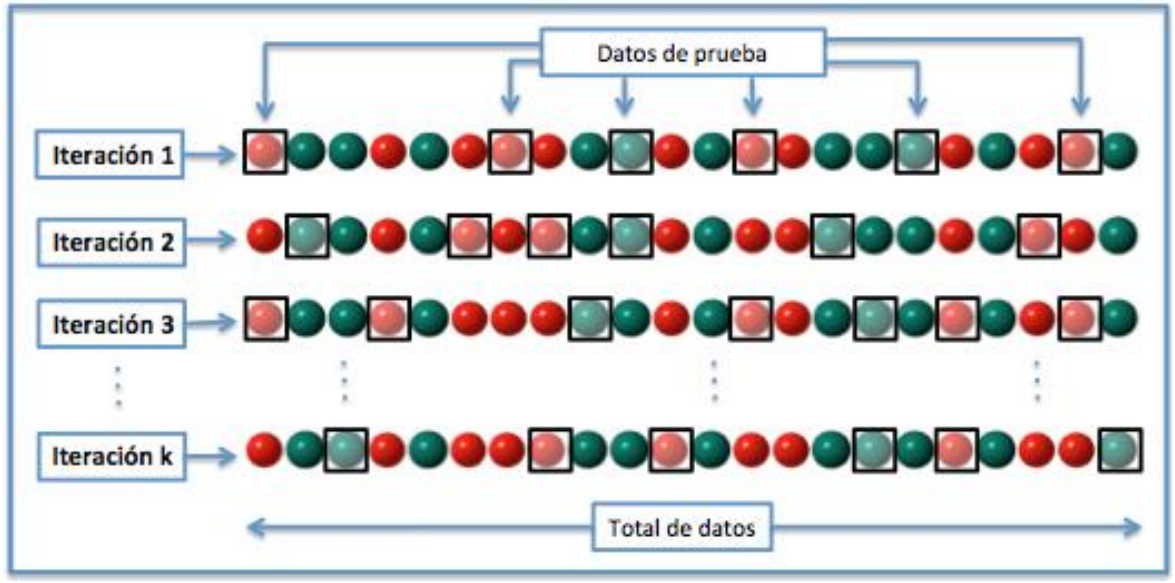
Ejemplo de 4 iteraciones



Wikipedia: adaptado de Salman, M. S., Dey, P. K., Das, P., & Shuvro, R. A. (2016). Breast Cancer Detection and Classification Using Mammogram Images. Technical Report, 1–8.

Random-cross-validation

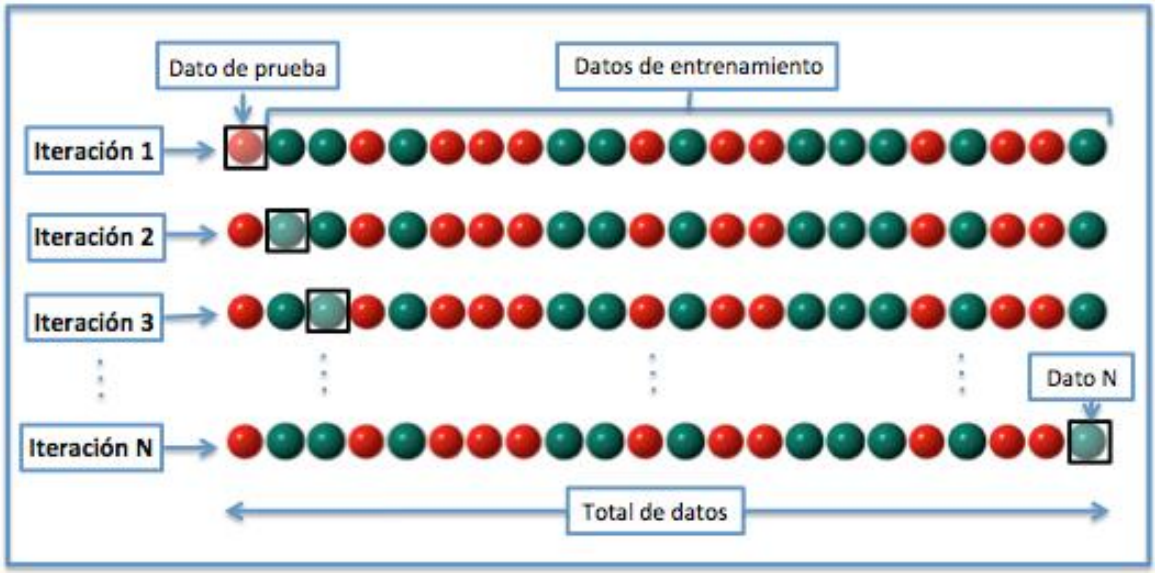
Ejemplo de 4 iteraciones



Wikipedia: adaptado de Salman, M. S., Dey, P. K., Das, P., & Shuvro, R. A. (2016). Breast Cancer Detection and Classification Using Mammogram Images. Technical Report, 1–8.

Leave-one-out

Ejemplo de 4 iteraciones



Wikipedia: adaptado de Salman, M. S., Dey, P. K., Das, P., & Shuvro, R. A. (2016). Breast Cancer Detection and Classification Using Mammogram Images. Technical Report, 1–8.

Métricas de evaluación en clasificación

- **Objetivo**: comprobar que las clases predichas para las instancias en el conjunto de evaluación son las mismas que las reales
- **Matriz de confusión**. Ejemplo para clase binaria:

		Clase observada	
		1	0
Clase pronosticada	1	*Verdaderos Positivos*	*Falsos Positivos*
	0	*Falsos Negativos*	*Verdaderos Negativos*

Métricas de evaluación en clasificación

- **Accuracy**

(exactitud o tasa de aciertos):

Accuracy = $\frac{VP + VN}{VP + VN + FP + FN}$

- **Precision**

(Valor Predictivo Positivo - VPP)

Precision = $\frac{VP}{VP+FP}$

- **Recall** (exhaustividad)

(Tasa de Verdaderos Positivos - TVP):

Recall = $\frac{VP}{VP+FN} = \frac{VP}{P}$

- **F1 score** = $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Métricas de evaluación en clasificación

- **Accuracy =** $\frac{VP + VN}{VP + VN + FP + FN}$

(exactitud o tasa de aciertos):

Cuándo Usarla:

- Mide la **proporción de predicciones correctas sobre el total de predicciones**.
- Es útil cuando todas **las clases tienen importancia similar** y el **desequilibrio** entre las clases **no** es **significativo**.
- En casos de desequilibrio de clases, la accuracy puede ser engañosa.

Métricas de evaluación en clasificación

- **Precision** (Precisión positiva)

(Valor Predictivo Positivo - VPP)

Precision = $\frac{VP}{VP+FP}$

Cuándo Usarla:

- Precision se enfoca en la proporción de instancias clasificadas como positivas que realmente son positivas.
- Es útil cuando tener falsos positivos en el resultado es grave.
- Por ejemplo, en la detección de spam, es importante minimizar los falsos positivos para no marcar correos legítimos como spam.

Métricas de evaluación en clasificación

- **Recall** (exhaustividad – Tasa de verdaderos positivos)

(Tasa de Verdaderos Positivos - TVP):

Recall = $\frac{VP}{VP+FN} = \frac{VP}{P}$

Cuándo Usarla:

- Recall mide la proporción de instancias positivas que fueron correctamente identificadas.
- Es útil cuando el coste de tener falsos negativos es alto.
- En problemas médicos, por ejemplo, es crucial no perder casos positivos, incluso si hay falsos positivos.

Métricas de evaluación en clasificación

F1 (score): $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Cuándo Usarla:

- Métrica que combina precision y recall en un solo valor.
- Es útil cuando hay un equilibrio entre la importancia de los falsos positivos y los falsos negativos.
- **Si tanto la precision como el recall son importantes**

Métricas de evaluación en clasificación

- **Para evaluar sistemas de recuperación de la información y ranking**

- Mean Reciprocal Rank (MRR)

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\mathrm{rank}_i}$$

rank_i: Posición del primer documento relevante para la consulta i
Q: tamaño de las consultas

- DCG (Discounted Cumulative Gain) y NDCG (Normalized DCG)

$$\mathrm{DCG} = \sum_{i=1}^{n}\frac{\mathrm{rel}_i}{\log_2(i+1)} \qquad \mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}}$$

rel: Relevancia del documento en la posición i.
valor ideal (IDCG), es decir, el mejor ranking posible.

MRR se enfoca en el primer resultado relevante
DCG y NDCG consideran la calidad del ranking completo
DCG penaliza posiciones bajas, NDCG es la versión normalizada (comparar distintos rankings)

Métricas de evaluación en clasificación

- **Para evaluar sistemas de recuperación de la información y ranking**
- Precision@k, recall@k, F1@k: métricas calculadas en los primeros k más relevantes
- **MAP (Mean Average Precision)** $\mathrm{MAP} = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{m} \sum_{k=1}^{m} \mathrm{Precision@k}$

Calcula la media de la precisión en cada posición relevante, y luego hace una media de todas las medias

Métricas de evaluación en regresión

- **Objetivo**: comparar si los resultados predichos son los que tenemos en el conjunto de evaluación

- Mean Square Error

(penalizar errores grandes)

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

- Root Mean Square Error

(misma escala que variables: resultados más interpretables)

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

Scikit-learn

- Principal librería para implementar aprendizaje automático.
- Junto a NumPy, pandas, matplotlib y plotly podéis desarrollar y evaluar técnicas de AA

https://scikit-learn.org/stable/