2.4 El paradigma del aprendizaje supervisado

Entrenamiento y optimización de modelos.

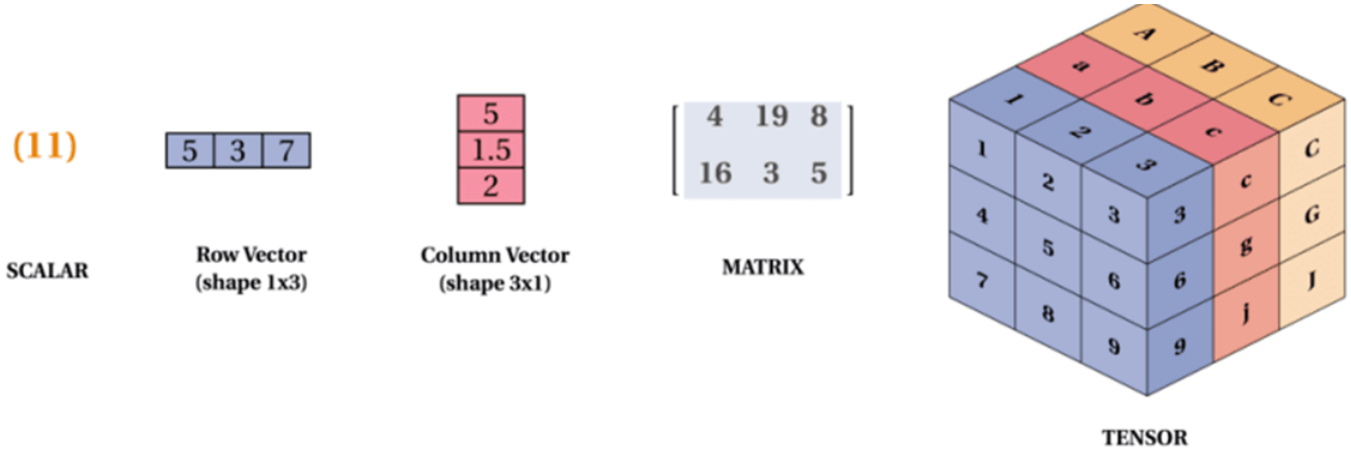
Marc Palomer Cadenas

Indice

- Tipos de datos
- Dimension de los datos
- Aprendizaje supervisado y no supervisado
- Aprendizaje supervisado: clasificación vs. regression
- División de los datos: Train/Test. k-fold Cross-validation
- Metricas de evaluación: regresión y clasificación
- Overfitting y underfitting
- Hyperparameter tunning

Tipos de datos

- Dimension matemática de los datos:
  - **Numero**: 0 dimensiones (E.g. valor aislado de presión arterial)
  - **Vector**: 1 dimension. Ees una columna de nuestros datos. E.g. Valores de presión arterial medidos cada minuto (serie temporal) o conjunto de presiones arteriales de los pacientes de mi URPA.
  - **Matriz**: 2 dimensiones. Columnas puestas una al lado de otra. E.g. Un caso tipico es una table o excel file con las variables como columnas y los pacientes como filas. También és el caso de una images, que tiene un valor de intensidad para cada uno de sus pixels.
  - **Tensor**: n dimensiones



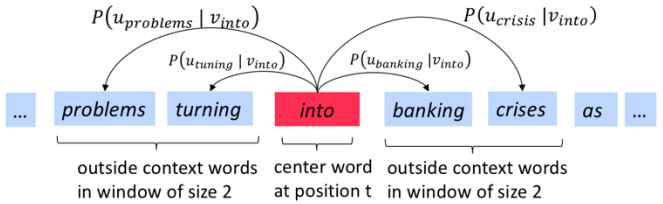
Dimensión de los datos

- **Tabulares**: Son la gran mayoría de datos con los que nos encontraremo en la práctica clínica. Variables como columnas y pacientes (o observaciones) como filas.
- **Temporales**: Datos que tienen una secuencia temporal. E.g. Una derivación de ECG es el valor de potencial que se mide cada ~4 milisegundos al paciente. Portanto es la serie temporal del potencial de su corazón a lo largo del tiempo
- **Imagen**: Una radiografia és una matriz donde cada posición nos indica la intensidad con que el cuerpo ha absorbido o no los Rayos X administrados al paciente.
- **Video**: Un video es una imagen que cambia en el tiempo.
- **Texto**: Secuencias de texto natural. E.g. Una respuesta de ChatGPT, lectura de las enfermedades presents dentro de una EHR…

Variables/predictors

target

	A	B	C	D	E
1	sepal-l	sepal-w	petal-l	petal-w	species
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa
11	4.9	3.1	1.5	0.1	setosa
12	5.4	3.7	1.5	0.2	setosa
13	4.8	3.4	1.6	0.2	setosa
14	4.8	3	1.4	0.1	setosa
15	4.3	3	1.1	0.1	setosa
16	5.8	4	1.2	0.2	setosa
17	5.7	4.4	1.5	0.4	setosa
18	5.4	3.9	1.3	0.4	setosa

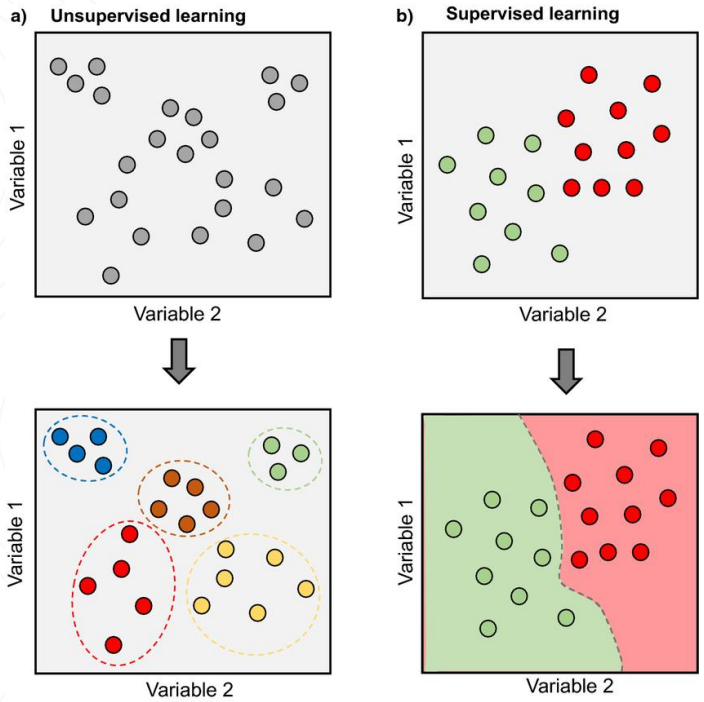


CEMP CENTRO EUROPEO DE MÁSTERES Y POSGRADOS

Aprendizaje supervisado y aprendizaje no supervisado

- Aprendizaje supervisado: En los datos existe una columna target (o objetivo), la cual mostramos a el modelo durante la fase de entrenamiento, para que aprenda la relación entre nuestras variables y esa columna. E.g. Todos los algoritmos de regresión o clasificación intentan aprender como predecir la variable target, que puede ser continua en regresión o una categoría en clasificación

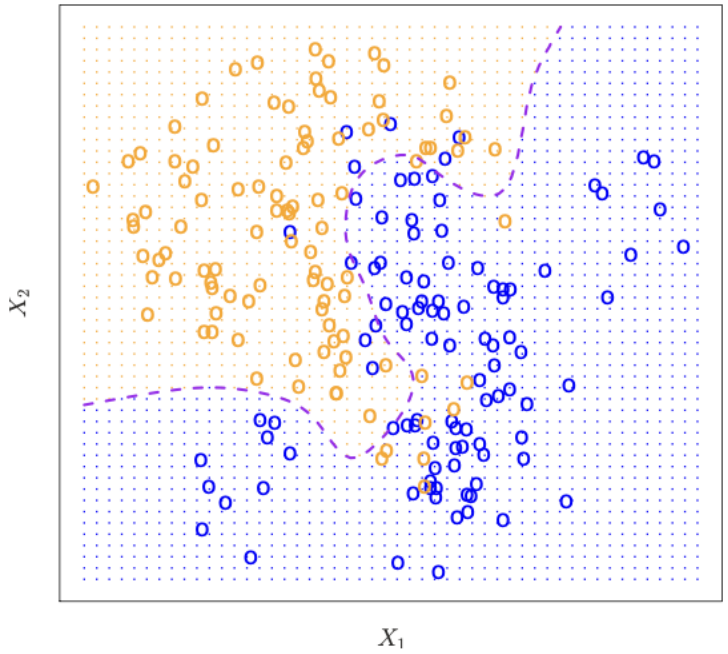
- Aprendizaje no supervisado: No existe la columna target. El modelo usa técnicas para agrupar los datos de manera natural. E.g. PCA, K-means etc.

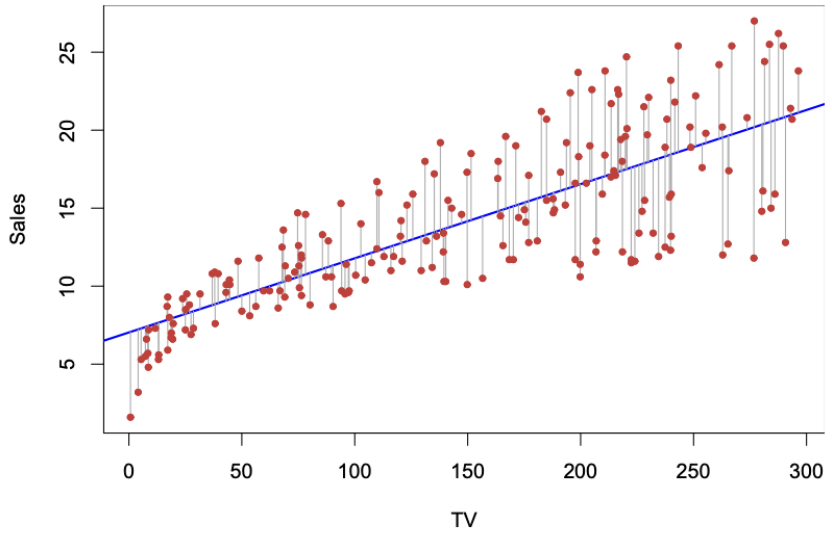


Ejemplo clasificación: En la izquierda el modelo se "inventa" que la major manera de agrupar los datos o observaciones es a partir de esos cinco grupos. En la derecha el algoritmo aprende de los datos la major manera de separarlos en dos grupos, en base a saber de antemano que paciente pertenecia a qué grupo

Aprendizaje supervisado: Clasificación vs. Regresión

- La diferencia entre clasificación y regresión está en **que es lo que queremos predecir/obtener**
- En el caso de la clasificación, queremos saber donde caería (**en que clase**) un paciente nuevo… Si en la clase cancer (azul) o no cancer (amarillo).
- En el caso de la regresión queremos saber para un paciente nuevo **que valor** tendrá para una variable desconocida (E.g. ¿Cuantas horas se estará el paciente en urgencias?)

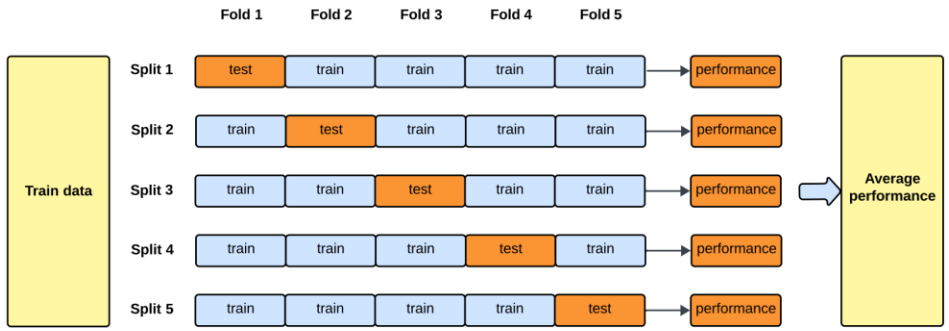




CEMP CENTRO EUROPEO DE MÁSTERES Y POSGRADOS

División de mis datos: Train/Val/Test/CV

- **Train/Test Split:** Pongamos que tenemos 1000 pacientes para un estudio diagnostico de cancer de mama a través de orina. Puedo usar 700 (70%) para **entrenar** un modelo que diferencie dos clases: cancer de mama y No cancer de mama. Luego usar los últimos 300 pacientes (30%) para **testear** como de bien ha aprendido el modelo. Reservar pacientes para testear nos deja con menos pacientes para entrenar, pero nos da un gran beneficio que es intentar averiguar como lo hará el modelo cuando lo estemos usando en el hospital. Por ejemplo, digamos que el modelo separa bien cancer de no cancer el 80% de los pacientes del grupo de test. Diremos que tiene una accuracy del 80%, y esperamos que ese 80% sea parecido a la accuracy que tendría en el hospital.
- **Cross-validation**: Como hemos dicho en el Train/test Split, la finalidad del test es probar el modelo y ver como de bien lo haría en la realidad… La validación cruzada pretende hacer lo mismo pero haciendo agrupaciones repetidas de los datos, para en vez de tener una sola estimación de la accuracy tener muchas, y así ser capaces de estimar mejor como lo hará el modelo en el hospital. En el ejemplo hemos hecho 5 grupos, por tanto tendremos 5 valores de accuracy. Esta sería un 5 fold cross-validation



CEMP CENTRO EUROPEO DE MÁSTERES Y POSGRADOS

▲ Métricas de evaluación: Clasificación

		True class		
		− or Null	+ or Non-null	Total
Predicted class	− or Null	True Neg. (TN)	False Neg. (FN)	N^*
	+ or Non-null	False Pos. (FP)	True Pos. (TP)	P^*
	Total	N	P	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P^*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N^*	

TABLE 4.7. *Important measures for classification and diagnostic testing, derived from quantities in Table 4.6.*

Desbalanceo de Clases

- **¿Por qué es problemático?**
  - Enfermedades raras: 1 caso positivo por cada 1000 negativos
  - El modelo puede alcanzar 99.9% accuracy simplemente prediciendo siempre "no enfermo"
  - Las métricas tradicionales engañan
- **Consecuencias:**
  - Modelo no aprende la clase minoritaria
  - Falsos negativos peligrosos en medicina
  - Métricas infladas pero inútiles
- **Soluciones:**
- **Balanceo de datos**: SMOTE, undersampling, oversampling
- **Pesos de clase**: Penalizar más errores en clase minoritaria
- **Métricas apropiadas**: Precision, Recall, F1, AUC-ROC
- **Umbrales ajustados**: Modificar punto de corte de decisión

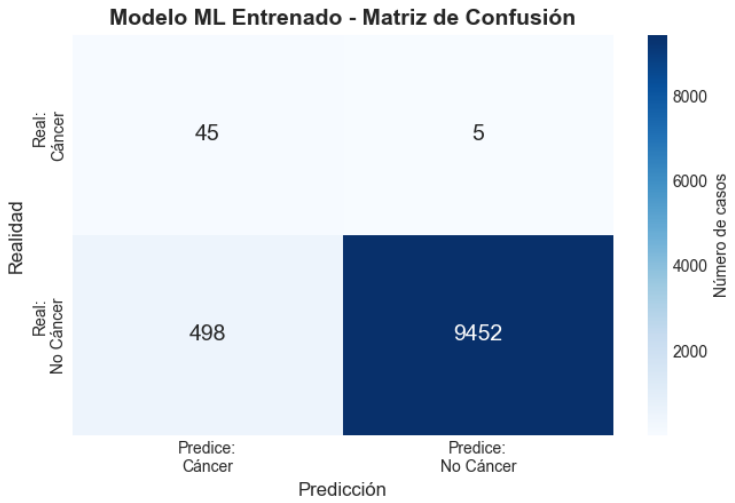
Ejemplo: Screening de Cáncer de Páncreas

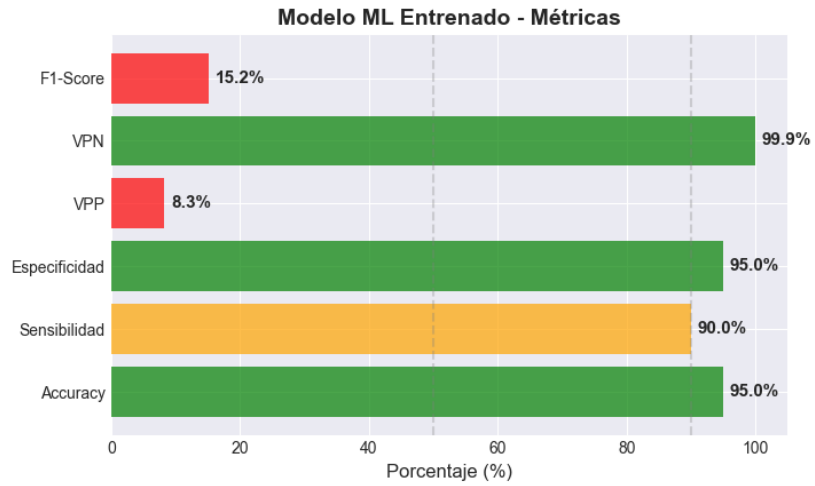
- **Configuración del experimento**
  - **Población**: 10,000 pacientes asintomáticos
  - **Prevalencia real**: 0.5% (50 casos con cáncer)
  - **Test diagnóstico**: Sensibilidad 90%, Especificidad 95

- **Métricas obtenidas:**
  - **Accuracy: 95.0%** → ¡Parece excelente!
  - **Sensibilidad: 90.0%** → Detecta 45/50 cánceres
  - **Especificidad: 95.0%** → Identifica bien los sanos
  - **VPP: 8.3%** → Solo 45/543 positivos son reales
  - **VPN: 99.95%** → Negativo = casi seguro sano

- **Impacto, por cada cáncer detectado:**
  - Se generan **11 falsas alarmas**
  - **498 pacientes** con ansiedad innecesaria
  - **498 biopsias** potencialmente innecesarias
  - **5 cánceres no detectados** (potencialmente fatales)

¿Por qué Accuracy no es suficiente?
Cáncer con prevalencia 0.5%



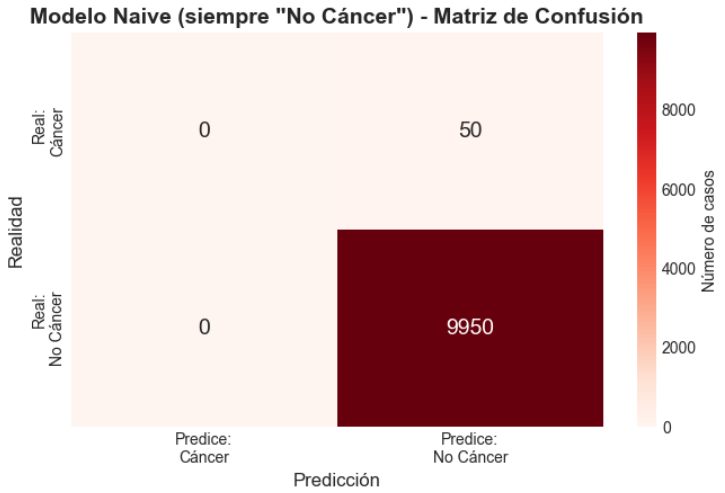


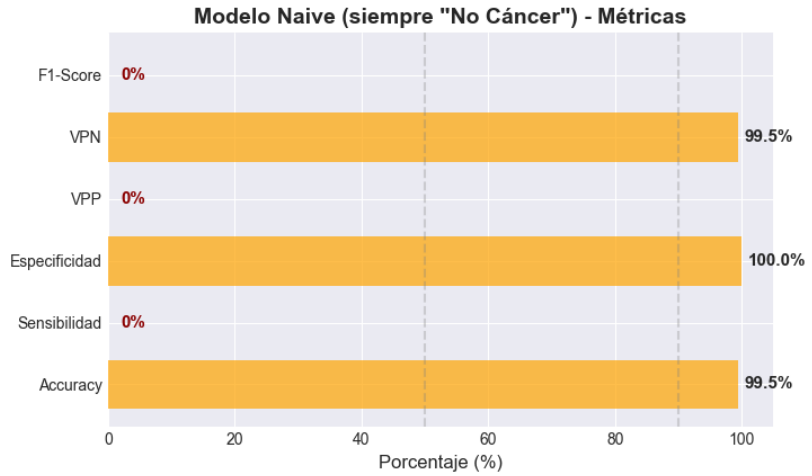
CEMP CENTRO EUROPEO DE MÁSTERES Y POSGRADOS

Ejemplo: Screening de Cáncer de Páncreas

- **La paradoja del modelo "tonto" Si siempre predecimos "No Cáncer":**
  - Accuracy: **99.5%** (¡más alta que nuestro modelo!)
  - Sensibilidad: **0%** (no detecta ningún cáncer)
  - Muertes evitables: **50** pacientes
  - **Conclusión:** Un modelo con 99.5% de accuracy pero completamente inútil

- **Mensajes clave**
  - **En enfermedades raras, accuracy engaña. Hay que mirar más metricas**
  - **Evalúa siempre**: Sensibilidad, Especificidad, VPP
  - Balance → F1-score, AUC-ROC





CEMP CENTRO EUROPEO DE MÁSTERES Y POSGRADOS

Métricas de evaluación: Regresión

- MAE - Mean Absolute Error
  - Average of the absolute difference between y and ŷ
  - 0 value is best. Unit dependant. Robust to outliers

$$MAE = \frac{1}{n}\Sigma\left|y - \hat{y}\right|$$

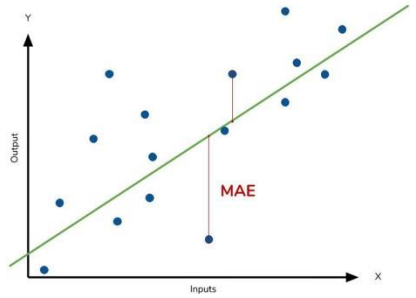
Divide by the total number of data points

Predicted output value

Actual output value

Sum of

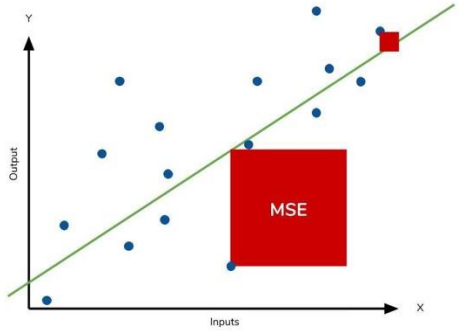
The absolute value of the residual



- RMSE - Root Mean Square Error
  - Sample Standard Deviation of the differences between y and ŷ.
  - 0 is value is best. Unit dependant. Sensitive to outliers
  - **Most used**: is computationally simple, easily differentiable and present as default metric for most of the models

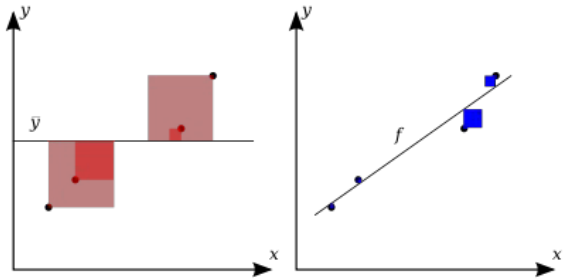
$$MSE = \frac{1}{n}\Sigma\left(y - \hat{y}\right)^2$$

The square of the difference between actual and predicted



- R^2 (adj-R^2)
  - The numerator is MSE (average of the squares of the residuals) and the denominator is the variance in Y values. Higher the MSE, smaller the R^2 and poorer is the model (1 best)

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$



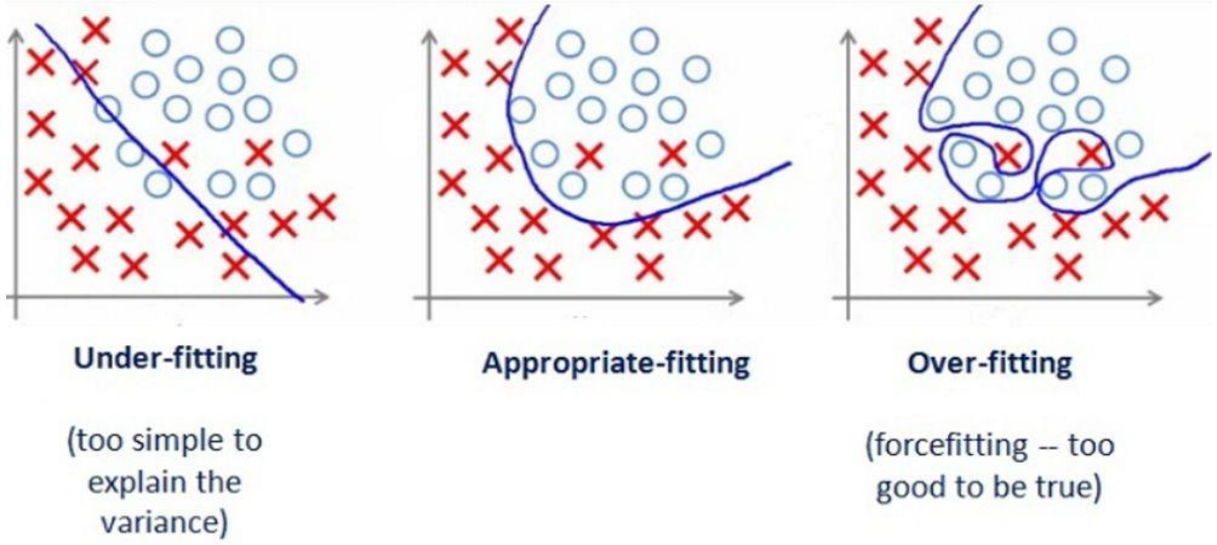
Overfitting y underfitting

- Overfitting is produced when you **fit too closely to a particular set of data and** may therefore fail to fit additional data or predict future observations reliably.

  An overfitted model is a statistical model that contains more parameters than can be justified by the data. The essence of overfitting is to have unknowingly extracted some of the residual variation (i.e. the noise) as if that variation represented underlying model structure.

- Underfitting occurs when a statistical model **cannot adequately capture the underlying structure of the data**.

  An underfitted model is a model where some parameters or terms that would appear in a correctly specified model are missing. Underfitting would occur, for example, when fitting a linear model to non-linear data. Such a model will tend to have poor predictive performance.



Bias /Variance tradeoff is a central problem in supervised learning. Ideally, one wants to choose a model that both accurately captures the regularities in its training data but also generalizes well to unseen data. Unfortunately, it is typically impossible to do both simultaneously. High-variance learning methods may be able to represent their training set well but are at risk of overfitting to noisy or unrepresentative training data. In contrast, algorithms with low variance typically produce simpler models that don't tend to overfit but may *underfit* their training data, failing to capture important regularities.

Valores Faltantes (NaN)

¿Qué es?

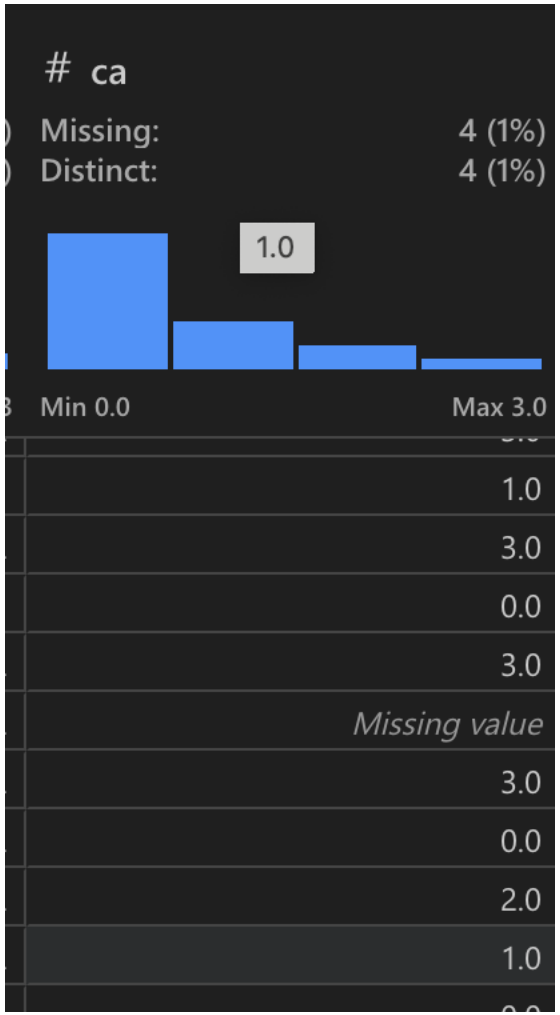
- Missing data: Valores ausentes en nuestro dataset
- Pueden ser MCAR (Missing Completely At Random), MAR (Missing At Random) o MNAR (Missing Not At Random)

¿Por qué es importante?

- En la práctica clínica, es común tener datos incompletos (pacientes que no se hacen todas las pruebas, valores fuera de rango, errores de registro, pacientes que se salen de un estudio…)
- Muchos modelos ML no pueden procesar valores faltantes directamente
- Una mala gestión puede sesgar completamente los resultados

¿Cómo se maneja?

- **Eliminación**: Quitar filas/columnas con muchos faltantes (>50%)
- **Imputación simple**: Media, mediana, moda, forward-fill
- **Imputación avanzada**: KNN imputer, MICE, modelos predictivos
- **Indicador de ausencia**: Crear variable binaria que indique si faltaba



CEMP CENTRO EUROPEO DE MÁSTERES Y POSGRADOS

Errores Comunes en ML Médico

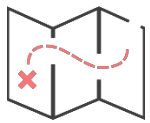
- **Data Leakage:**
  - Incluir información del futuro
  - Variables que contienen el outcome
  - Ejemplo: Incluir "medicamento X prescrito" para predecir "enfermedad Y" cuando X solo se prescribe para Y
- **Selection Bias:**
  - Entrenar solo con pacientes graves (los que llegaron a UCI)
  - Validar en población diferente
  - Missing not at random
- **Métricas inadecuadas:**
  - Usar accuracy en datos desbalanceados
  - No considerar costos asimétricos (falso negativo vs falso positivo)
- **Validación insuficiente:**
  - No validar en otros hospitales
  - Overfitting al test set por múltiples iteraciones

Optimización de parámetros: Hyperparameter tunning

- La mayoría de los modelos tienen parámetros “arbitratios” que tenemos que definir. Por ejemplo la cantidad de vecinos a tener en cuenta en un KNN, o la profundidad de los arboles de un RF.

- Estos parámetros se estiman o bien a partir de la **experiencia**, o probando varios valores hasta encontrar los que nos dan mejor metrica de performance. Existen métodos para probar muchos valores de forma optimizada como la optimización bayesiana o por gradient descent.






METODOLOGÍA DE PROYECTO ML: Pasos genéricos a seguir

FASE 1: DEFINICIÓN DEL PROBLEMA

- Identificación de la variable objetivo (target)
- Clasificación del problema: Clasificación vs Regresión

FASE 2: RECOLECCIÓN DE DATOS

- Carga de datos desde fuentes disponibles
- Inspección preliminar de estructura y dimensiones

FASE 3: ANÁLISIS EXPLORATORIO DE DATOS (EDA)

3.1 Análisis Univariado

- Variables numéricas: Estadísticas descriptivas, distribuciones, detección de outliers
- Variables categóricas: Frecuencias y distribución de categorías

3.2 Análisis Bivariado

- Correlaciones entre variables numéricas
- Relaciones entre variables categóricas y numéricas
- Tablas cruzadas para variables categóricas

3.3 Análisis de la Variable Target

- Distribución de la variable objetivo
- Evaluación de desbalanceo de clases (clasificación)
- Identificación de valores atípicos (regresión)

3.4 Relación Predictores-Target

- Análisis de la relación entre variables predictoras y objetivo
- Identificación de variables relevantes para la predicción

METODOLOGÍA DE PROYECTO ML: Pasos genéricos a seguir

FASE 1: DEFINICIÓN DEL PROBLEMA

- Identificación de la variable objetivo (target)
- Clasificación del problema: Clasificación vs Regresión

FASE 2: RECOLECCIÓN DE DATOS

- Carga de datos desde fuentes disponibles
- Inspección preliminar de estructura y dimensiones

FASE 3: ANÁLISIS EXPLORATORIO DE DATOS (EDA)

3.1 Análisis Univariado

- Variables numéricas: Estadísticas descriptivas, distribuciones, detección de outliers
- Variables categóricas: Frecuencias y distribución de categorías

3.2 Análisis Bivariado

- Correlaciones entre variables numéricas
- Relaciones entre variables categóricas y numéricas
- Tablas cruzadas para variables categóricas

3.3 Análisis de la Variable Target

- Distribución de la variable objetivo
- Evaluación de desbalanceo de clases (clasificación)
- Identificación de valores atípicos (regresión)

3.4 Relación Predictores-Target

- Análisis de la relación entre variables predictoras y objetivo
- Identificación de variables relevantes para la predicción

METODOLOGÍA DE PROYECTO ML: Pasos genéricos a seguir (II)

- **FASE 4: PREPROCESAMIENTO DE DATOS**
- Tratamiento de valores nulos mediante imputación o eliminación
- Feature engineering: Creación de variables derivadas
- Codificación de variables categóricas
- Normalización o estandarización de variables numéricas
- **FASE 5: DIVISIÓN DE DATOS**
- Train/Test Split: Separación en conjuntos de entrenamiento y prueba
- K-Fold Cross-Validation: Validación cruzada para evaluación robusta
- **FASE 6: SELECCIÓN Y ENTRENAMIENTO DEL MODELO**
- Selección de algoritmos según naturaleza del problema
- Entrenamiento del modelo con datos de entrenamiento
- Optimización de hiperparámetros (Hyperparameter Tuning)
- **FASE 7: EVALUACIÓN DEL MODELO**
- Clasificación: Accuracy, Precision, Recall, F1-Score, ROC-AUC
- Regresión: MAE, RMSE, R^2

FASE 8: VALIDACIÓN Y DIAGNÓSTICO
- Evaluación con datos de test no vistos
- Diagnóstico de Overfitting y Underfitting
- Comparación de métricas entre conjuntos de entrenamiento y prueba

FASE 9: INTERPRETACIÓN Y MEJORA ITERATIVA
- Análisis de importancia de variables
- Refinamiento del modelo mediante ajustes y nuevas iteraciones

FASE 10: DOCUMENTACIÓN Y REPORTE
- Generación de reportes con resultados y visualizaciones
- Documentación de metodología, decisiones y conclusiones