

M2.2.2 Modelos Supervisados y No Supervisados

Programa Big Data y Business Intelligence

Enrique Onieva

enrique.onieva@deusto.es

<https://twitter.com/EnriqueOnieva>

<https://www.linkedin.com/in/enriqueonieva/>

Modelos Supervisados **(Predictivos)**

- Problemas de clasificación vs. regresión
- Curva de Aprendizaje
- Overfitting
- Validación de modelos
- Evaluación de modelos
 - Clasificación
 - Regresión

Aprendizaje Supervisado

● Primer Paso a tomar

- Identificar la pregunta que queremos responder
- Identificar qué es lo que queremos predecir
- Dependiendo del tipo de respuesta, podremos aplicar unas técnicas u otras
 - Si la pregunta se responde con SI/NO o la pregunta admite sólo un conjunto de respuestas discreto: problema de clasificación
 - Si la pregunta es sobre la predicción de una cantidad, generalmente real, estamos en un problema de regresión
- Los datos deberán prepararse acorde a ella
 - Cada fila deberá contener atributos relevantes de la instancia sobre la que vamos a hacer predicciones
 - Si vamos a predecir fuga de clientes → fila=cliente
 - Si vamos a predecir precios de venta de productos → fila=producto
 - Si vamos a predecir ventas en una fecha → fila=fecha

Aprendizaje Supervisado

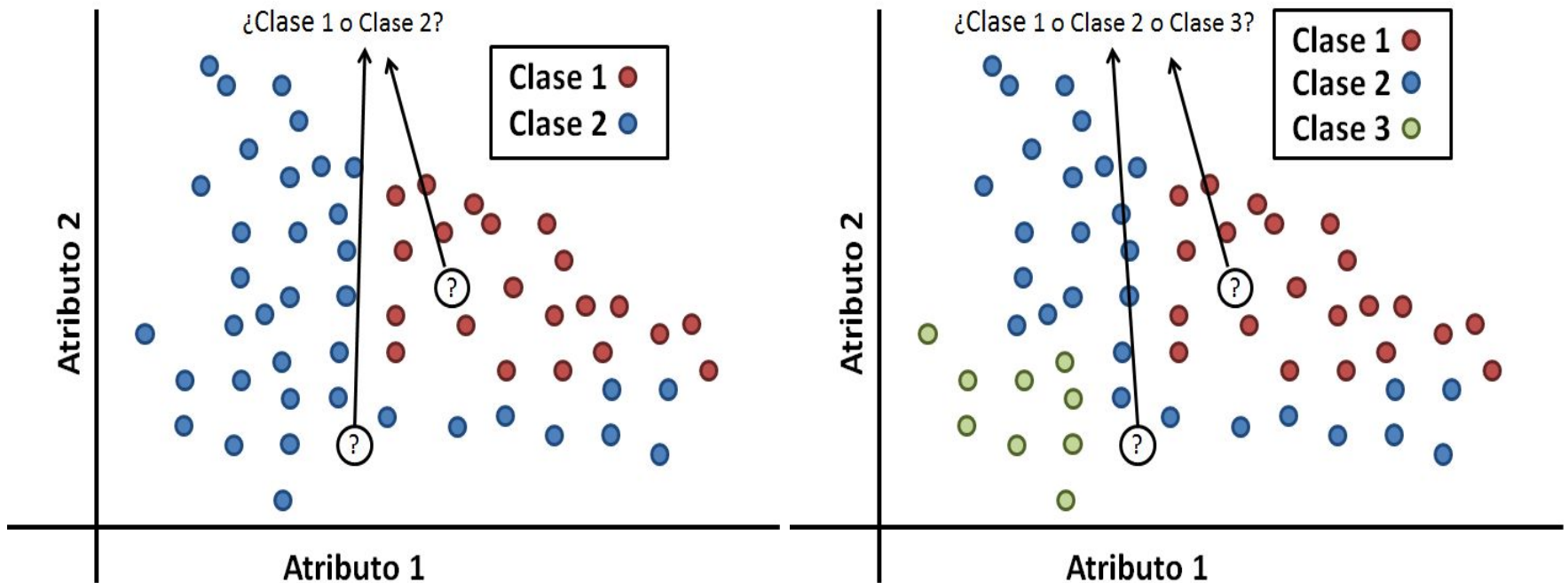
- Algunos ejemplos:

- Dado el perfil de un cliente y su actividad pasada, ¿en qué productos estaría más interesado?
- Dados los resultados del test, ¿sufre de <<enfermedad>>?
- Dada una resonancia magnética, ¿hay un tumor?
- Con la actividad de una tarjeta, ¿es la operación fraudulenta?
- Dada la descripción de un piso, ¿cuál es el valor de un piso?
- Dado el historial de transacciones, ¿cuáles serán las ventas el próximo año?

Problemas de Clasificación

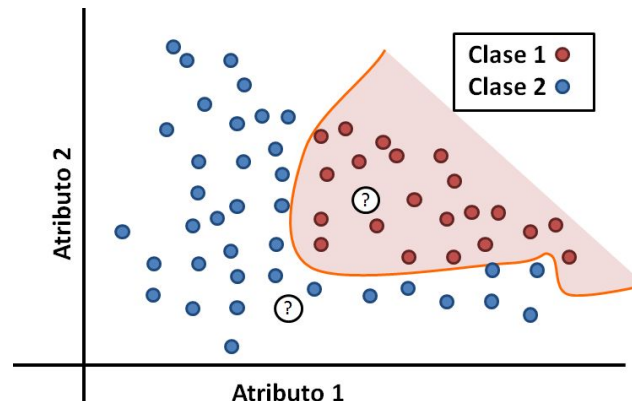
- En un problema de clasificación, dado un conjunto de ejemplos con su correspondiente etiqueta
 - El objetivo es predecir la pertenencia de una observación a un conjunto de clases predefinidas
- Formalmente se puede describir como:
 - Conjunto de datos de entrenamiento de tamaño N con d características de cada observación, con su correspondiente **etiqueta** y .
- En el caso más sencillo de dos clases $y=\{0,1\}$

Problemas de Clasificación



Problemas de Clasificación

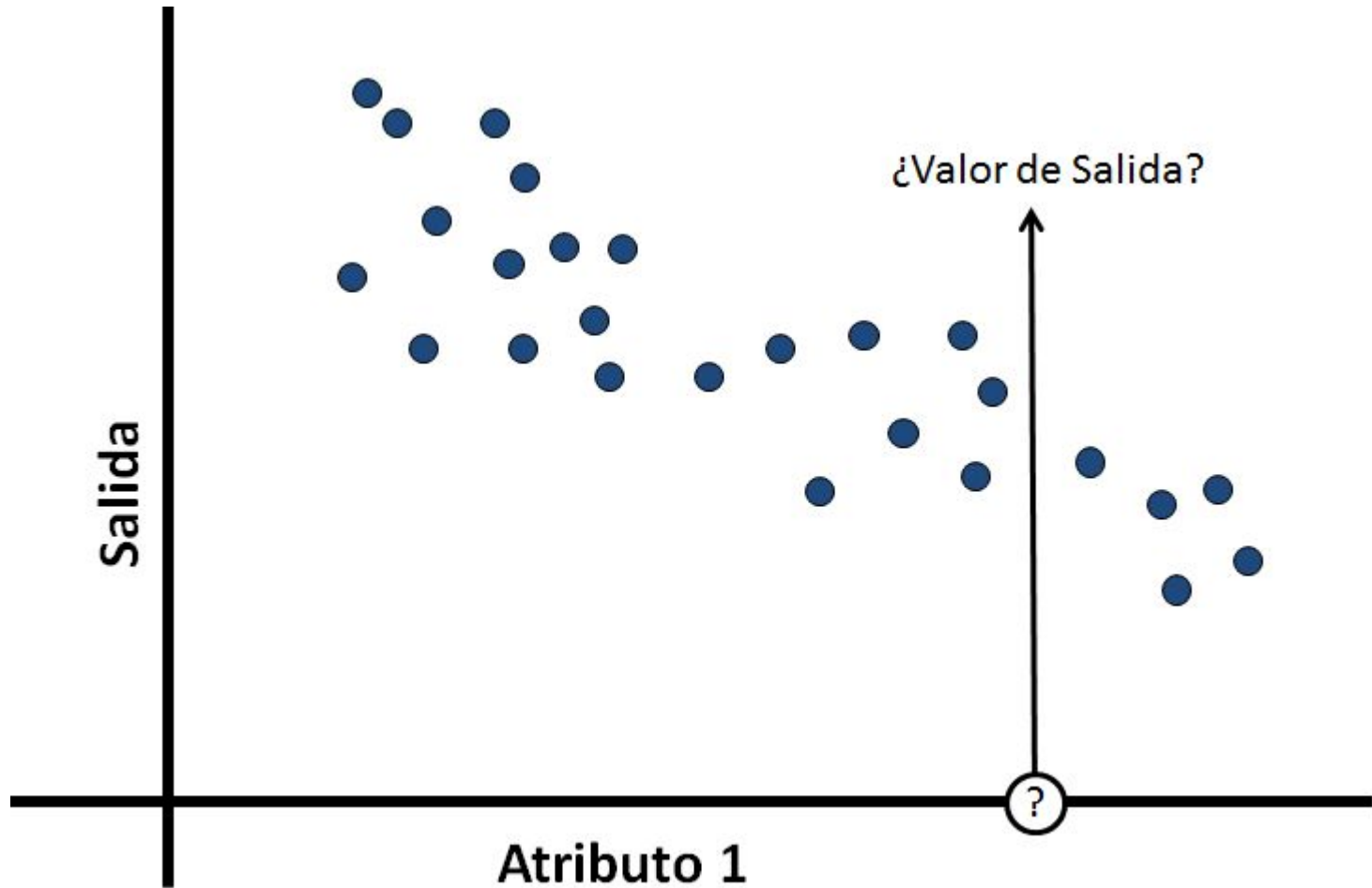
- El objetivo es encontrar un modelo que sea capaz de, ante valores de entrada, decir si dicho elemento debería pertenecer a la clase 1, 2, 3, ...
 - Ese modelo puede ser cualquier...
 - Función matemática, Serie de reglas, Salida de red neuronal,...
 - Se construirá en base a los datos disponibles



Problemas de Regresión

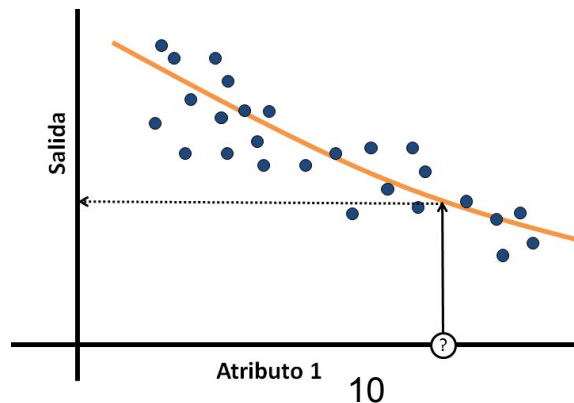
- En un problema de regresión, dado un conjunto de ejemplos con su correspondiente valor de salida.
 - El objetivo es predecir la salida para observación determinada
- Formalmente se puede describir como:
 - Conjunto de datos de entrenamiento de tamaño N con d características de cada observación, con su correspondiente **valor** y .

Problemas de Regresión



Problemas de Regresión

- El objetivo es encontrar un modelo que sea capaz de, ante unos valores de entrada, predecir el valor que debería de tomar la salida...
 - Ese modelo puede ser cualquier...
 - Función matemática, Serie de reglas, Salida de red neuronal, ...
 - Se construirá en base a los Datos disponibles



Modelos Supervisados

- Algunos problemas se pueden adaptar
- Para ser resueltos tanto mediante técnicas de clasificación como de regresión
 - Problemas de clasificación Binaria
 - Clasificación {"NO", "SI"} →
 - Regresión (con redondeo de salida) en $[0, 1]$
 - Problemas de "Clasificación Ordenada" / "Regresión Truncada"
 - Regresión para la edad →
 - Clasificación si edad = {"<20", "20-29", "30-39"...}

Modelos Supervisados

- En ambos casos el objetivo es similar
 - Construir un modelo utilizando los datos disponibles
 - Utilizar el modelo para predecir la salida con nuevos datos
- En Machine Learning, estas dos fases se llaman:
 - Entrenamiento (training): dados unos datos, y su etiqueta/valor, se construye (entrena) un modelo
 - Prueba (test): se utiliza el modelo entrenado para hacer predicciones sobre datos nuevos

Modelos Supervisados

- La fase de test se utiliza para probar lo bueno que es el modelo ante datos nuevos
 - Pero, por lo general, para datos nuevos, no tenemos el valor real de esa salida...
 - No podemos saber si las predicciones serán buenas o malas
- Utilizamos una porción de los datos para entrenar nuestro modelo
- Utilizamos la otra porción para probar cómo de bien predice el modelo

Modelos Supervisados

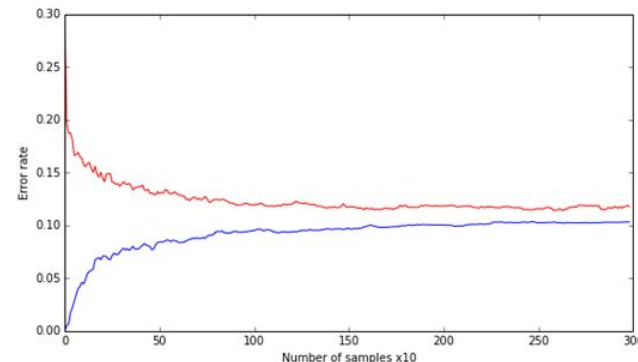
- Train-error: Error cometido sobre la muestra de entrenamiento
 - Estimador optimista de la tasa de error. Overfitting
- Test-error: Error cometido sobre datos “nuevos”
- Por lo general:
 - $\text{Train-error} < \text{Test-error}$

Curva de Aprendizaje

- ¿Cómo varían las tasas de error de entrenamiento y validación para una complejidad dada, conforme más datos tenemos?
 - Supongamos que generamos unos datos aleatorios con muestras aleatorias y entrenamos para cada muestra un árbol de clasificación de profundidad máxima 5
 - La profundidad del árbol es la medida de la complejidad del modelo

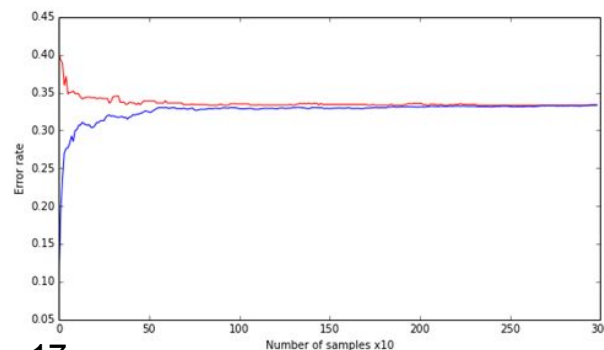
Curva de Aprendizaje

- Tasas de error en muestras de entrenamiento y validación, árbol de profundidad 5
 - A medida que aumenta el número de muestras de entrenamiento, los errores tienden al SESGO
 - Cuando los datos de entrenamiento son pocos, el error de entrenamiento es pequeño, y el de validación grande
- Más Datos no tiene por qué implicar mejores modelos



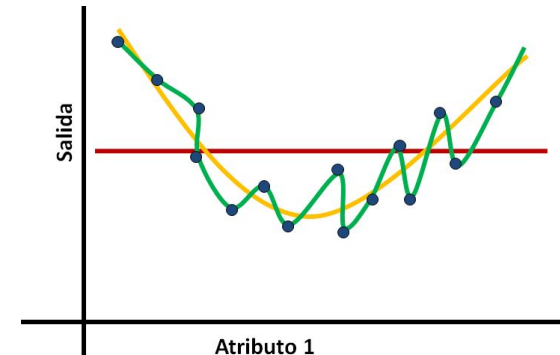
Curva de Aprendizaje

- Mismo ejercicio con un árbol de profundidad 1 (menor complejidad)
 - Con modelos más sencillos, la tasa error converge antes
 - Sin embargo el error es mayor
- Sesgo y Varianza
 - Sesgo: valor hacia el que convergen los dos errores
 - Varianza: diferencia entre ese valor y el error en la muestra de validación



Curva de Aprendizaje

- ¿Cómo varían las tasas de error de entrenamiento y validación para una complejidad dada?
 - Supongamos que generamos polinomios de grado creciente
 - $Salida = a$
 - $Salida = a \cdot Entrada + b$
 - $Salida = a \cdot Entrada^2 + b \cdot Entrada + c$
 - (Ese grado nos medirá la complejidad del modelo)
 - ¿Cuál es mejor?

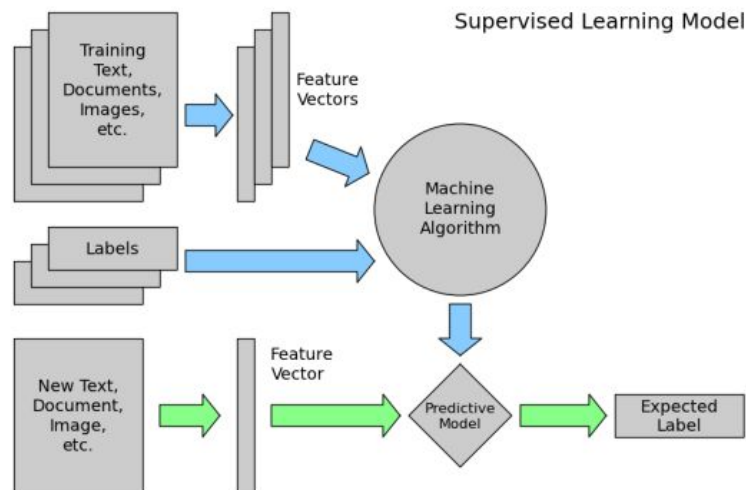


Overfitting

- El modelo rojo:
 - No se ajusta a los datos de entrada, y posiblemente no dará buenos resultados ante valores nuevos
 - Se dice que sufre de “underfitting” (no tiene suficiente complejidad para representar a los datos)
- El modelo amarillo
 - Se ajusta (moderadamente) a los datos de entrenamiento
 - Funcionará bien ante datos nuevos
- El modelo verde
 - Se ajusta perfectamente a los datos de entrenamiento, no lo hará ante datos nuevos
 - Se dice que sufre de “overfitting”

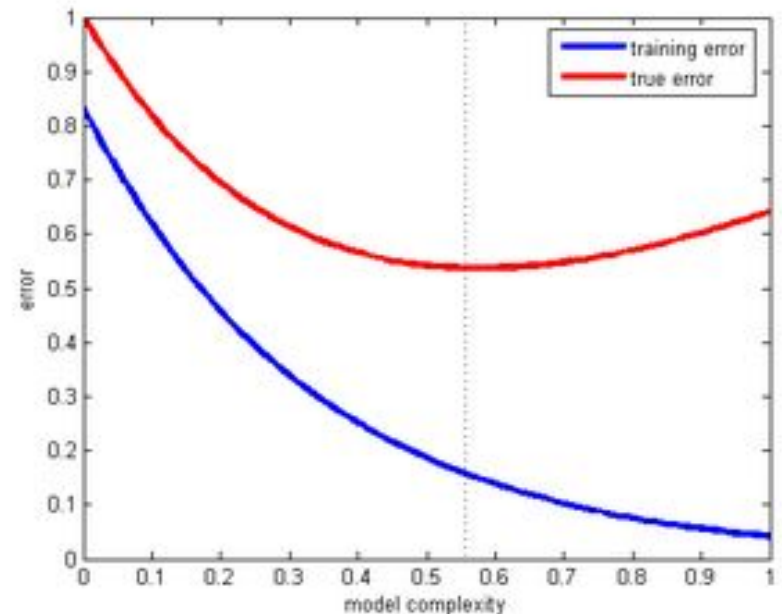
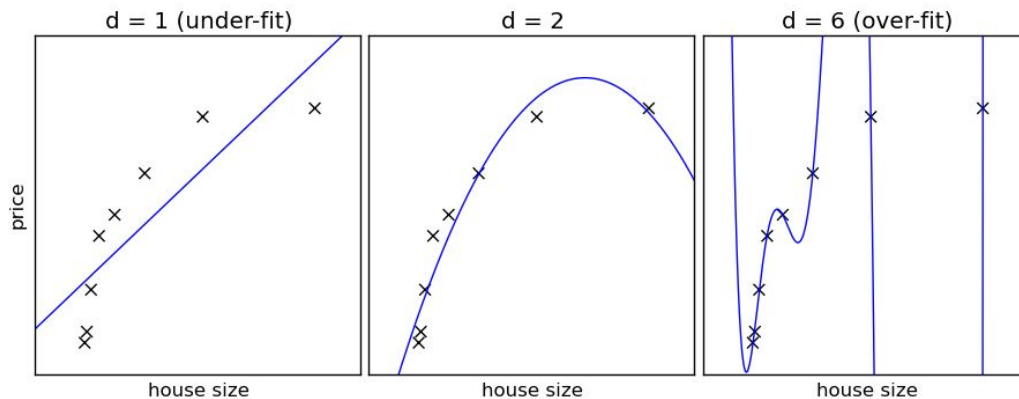
Overfitting

- Es por esto (entre otras cosas), que para medir la calidad de un modelo, se utilizan parte de los datos para entrenar y parte para validar (testear)
 - Válido para problemas de clasificación como de regresión
 - Normalmente (80-20%) ó (90-10%)



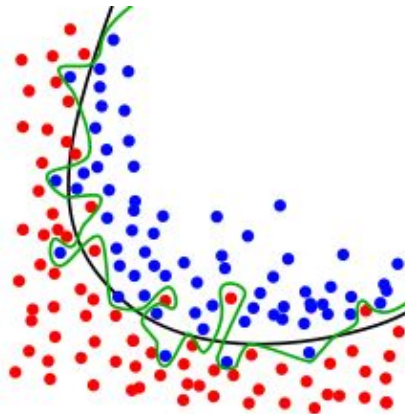
Overfitting

- Cuanto más complejo sea el modelo resultante:
 - Mejor se ajustará a los datos de entrenamiento
 - Mejor se ajustará a los datos de test (hasta cierto punto)
 - Ese punto es la complejidad óptima del modelo



Overfitting

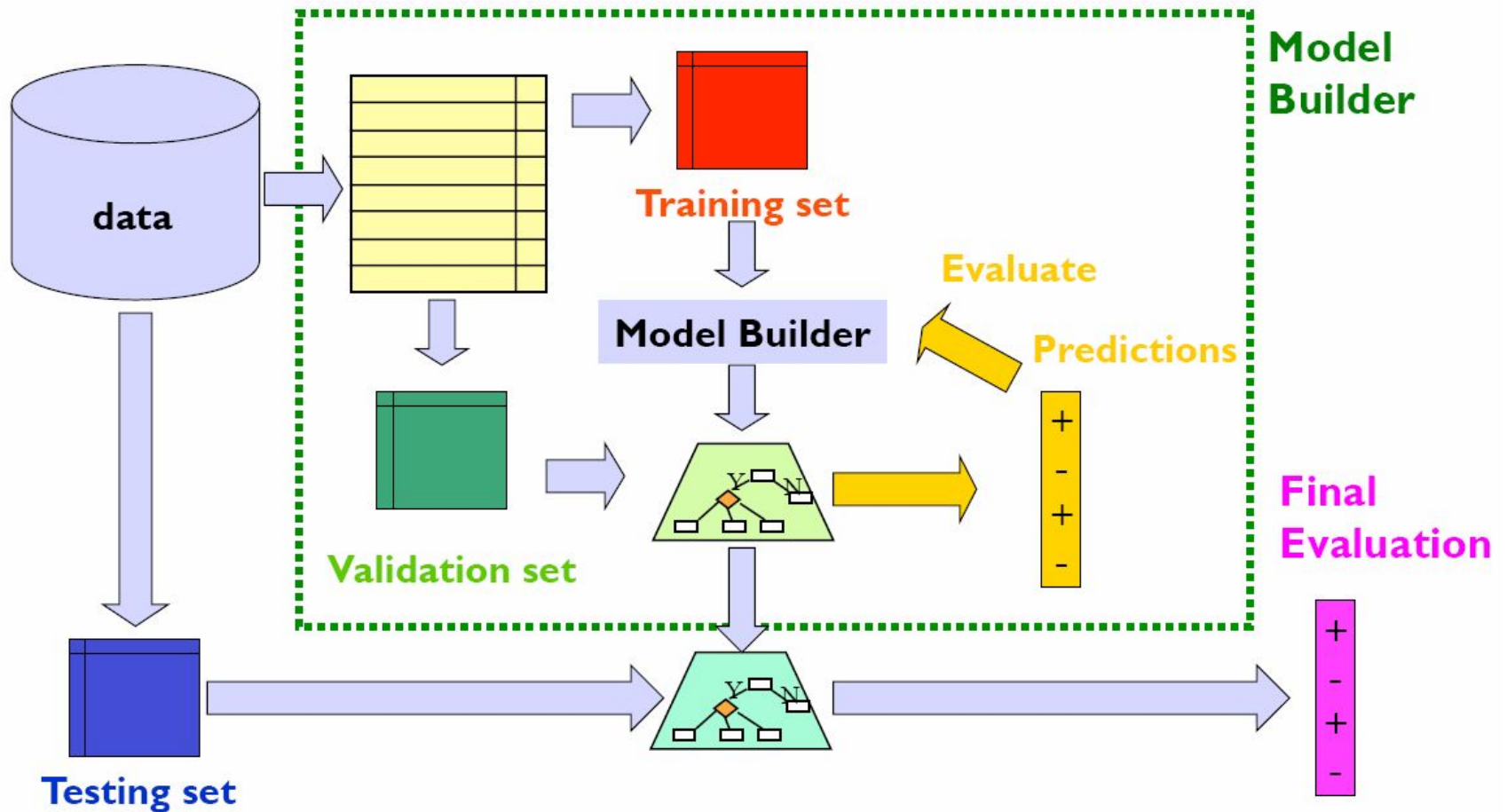
- La línea verde se ajusta mejor a los datos con los que hemos entrenado
- Pero está demasiado ajustada a ellos
- Ante nuevos datos probablemente dará más errores que la clasificación usando la línea negra.



Validación de Modelos

- Cuando tenemos suficientes datos los podemos particionar en 3 pedazos aleatorios
 - Training: Datos con los que se entrenan los modelos
 - Validation: Datos con los que probamos los modelos y elegimos el que tenga menor error
 - Test: Si queremos hacer una estimación de cómo generaliza nuestro modelo
- Por lo general, se habla únicamente de entrenamiento y test

Validación de Modelos



Validación de Modelos

- Una alternativa es dividir los datos disponibles en entrenamiento/train ($2/3$) y test ($1/3$)
 - Test es independiente de train y representativo puesto que train y test vienen de la misma distribución subyacente
- Condiciones que debe cumplir el conjunto de evaluación (test):
 - Independiente del conjunto usado para construir el modelo
 - Pero representativo del conjunto de entrenamiento
 - Lo mas grande que podamos para que sea preciso

Validación de Modelos

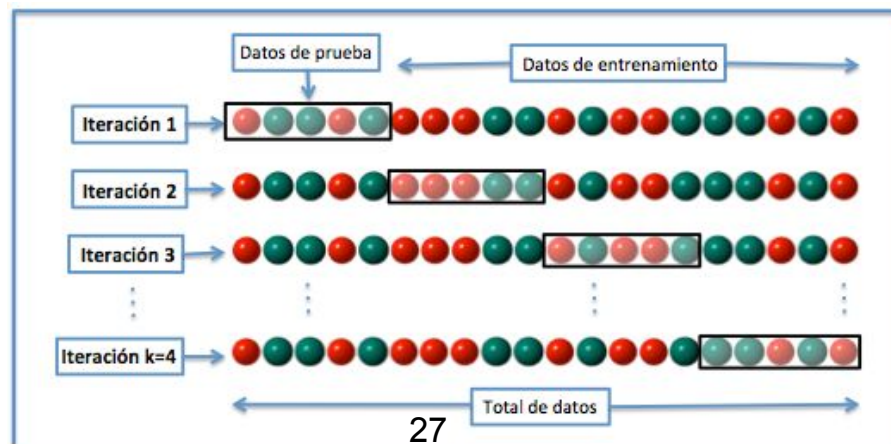
- La división train 2/3 test 1/3 es algo arbitraria, pero común. Tenemos un dilema:
 - Cuanto más grande sea el conjunto de test, más preciso será el cómputo del error de test, pero tendremos menos datos en train para construir el modelo
 - Construir el modelo con muchos datos (train) pero tener poca seguridad sobre si el modelo es bueno o malo
 - Construir el modelo con pocos datos (será malo), pero tendremos gran seguridad sobre que el modelo es, efectivamente, malo

Validación de Modelos

- Validación cruzada

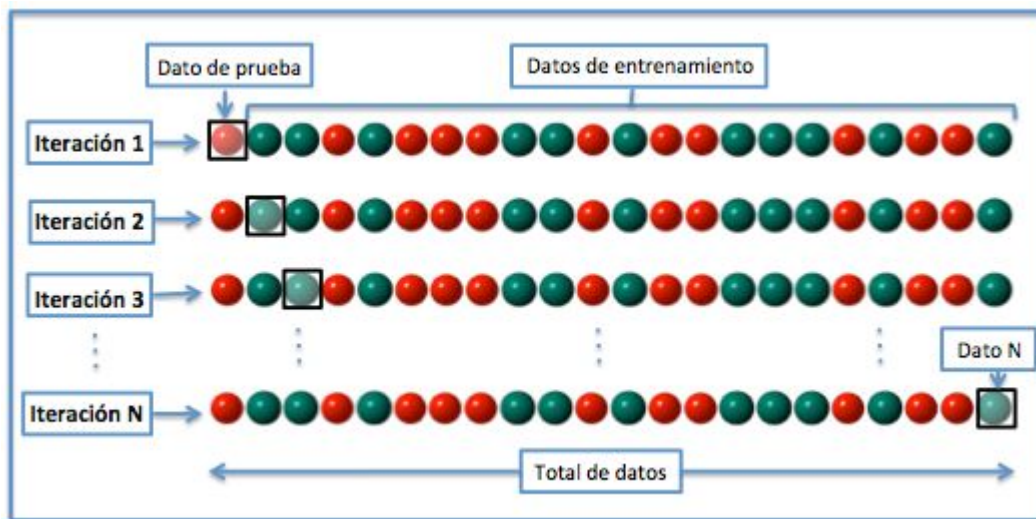
- Dividir el conjunto de datos de entrenamiento en k divisiones independientes.
 - $k-1$ divisiones se usan para entrenar y la restante para evaluar.
 - El proceso se repite K veces dejando cada vez una división fuera.
 - Se promedian los resultados.

- Se suele tomar el valor de $k=5$ ó 10



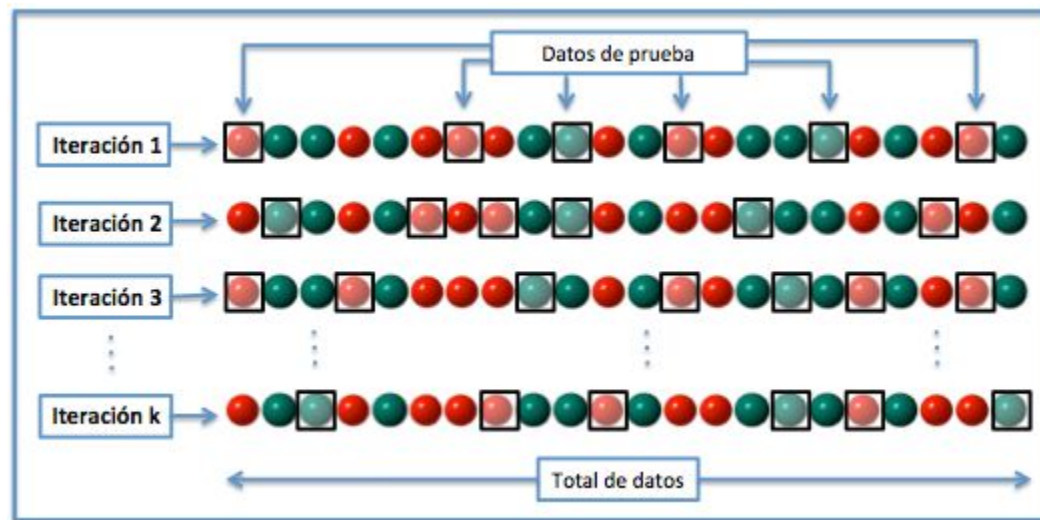
Validación de Modelos

- Validación leave-one-out
 - Es un caso extremo de validación cruzada donde k es igual al número de datos
 - Computacionalmente costoso (o imposible) el entrenar tantos modelos como datos tenemos en nuestro conjunto



Validación de Modelos

- Validación random k-fold
 - Similar a la validación k-fold, pero los datos se parten aleatoriamente



Validación de Modelos

- Validación stratified k-fold cross
 - Variante de k-fold que se asegura de que la muestra de test sea representativa con respecto al conjunto global
 - Para ello, realiza la selección de datos de manera que la distribución de los datos se mantenga en los conjuntos de training y test
 - Si tenemos un dataset con 10 de una clase y 90 de otra
 - Se asegura de que, tanto en el conjunto de entrenamiento y test, la proporción 10-90 se mantenga.

Evaluando Clasificadores

- La manera más inmediata de medir cómo de bien funciona un clasificador puede ser hacer un porcentaje del número de predicciones correctas
- Y quedarnos con el modelo con mayor número
 - Es lo que se conoce como la Precisión
 - También podemos medir el porcentaje de error (Buscando el valor menor)

Evaluando Clasificadores

- Para clasificación (binaria), una vez hecho el modelo, podemos encontrarnos ante 4 casos
 - El sistema predice “0” y la salida correcta es “0”
 - El sistema predice “1” y la salida correcta es “0”
 - El sistema predice “0” y la salida correcta es “1”
 - El sistema predice “1” y la salida correcta es “1”
- Lo ideal es que los casos de fallo no ocurran nunca
 - ¿0 no es tan “ideal”?

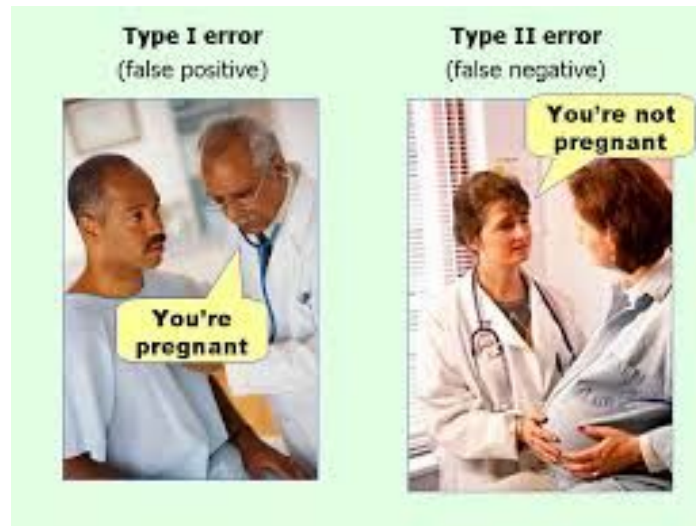
Evaluando Clasificadores

- Estas 4 situaciones pueden ponerse en una matriz, con el fin de analizar el resultado de un modelo
 - Matriz de confusión
 - TP: True positives
 - FN: False negatives
 - FP: False positives
 - TN: True negatives
 - La medida más inmediata para medir la bondad de un clasificador es su “porcentaje de aciertos”
 - $\text{Accuracy} = (TP+TN)/(TP+FN+FP+TN)$
 - Para más de 2 clases, añadir más filas/columnas

	Predicted 1	Predicted 0
True 1	TP	FN
True 0	FP	TN

Evaluando Clasificadores

- Pero, ¿valen lo mismo TP y TN? ¿Valen lo mismo FP y FN?
 - La respuesta depende de la aplicación a realizar



Evaluando Clasificadores

- ¿Cuál es mejor clasificador?
 - Ambos aciertan en el 75% de los casos
 - Podríamos establecer una matriz de costes sobre cada uno de los posibles errores
 - ¿Qué aspecto tendría?

Enfermedad	Predicho SI	Predicho NO
Real SI	90	40
Real NO	10	60

Enfermedad	Predicho SI	Predicho NO
Real SI	60	10
Real NO	40	90

Evaluando Clasificadores

- Evaluación sensible al coste
 - En muchos casos reales, necesitaremos una “medida” de la calidad del modelo especial
 - Matriz de coste

Cost Matrix	PREDICTED CLASS		
	$C(i j)$	+	-
	ACTUAL CLASS		
	+	-1	100
	-	1	0

Model M_1	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	150	40
	-	60	250

Accuracy = 80%
Cost = 3910

Model M_2	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	250	45
	-	5	200

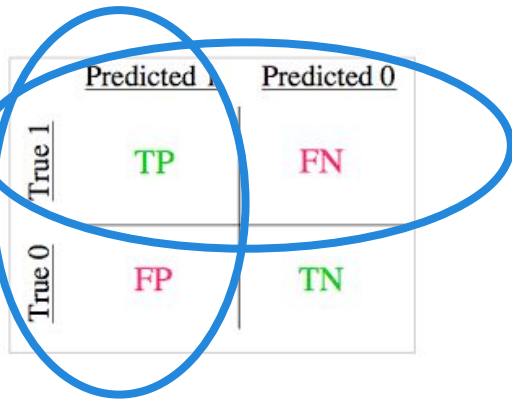
Accuracy = 90%
Cost = 4255

Evaluando Clasificadores

- Evaluación sensible a la distribución
 - Algunas veces, las medidas tienen sus inconvenientes en casos reales
 - Datos imbalanceados
 - Si hago un sistema que predice a quién le va a tocar la lotería, puedo acertar en el 99.99% de los casos
 - (Pista: Mi modelo siempre respondería <<NO>>)

Evaluando Clasificadores

- Dada la matriz de confusión,
 - Hay diferentes medidas que podemos obtener para evaluar la bondad de los modelos
 - Accuracy: porcentaje de aciertos, la más inmediata
 - Precision: expresa la probabilidad de que, cuando el clasificador diga “clase 1”, el caso realmente pertenezca a dicha clase
 - Recall: expresa la probabilidad de que, para un ejemplo de la “clase 1”, el clasificador lo identifique como tal



	Predicted 1	Predicted 0
True 1	TP	FN
True 0	FP	TN

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Evaluando Clasificadores

- F-score:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- Combina precision y recall

- (Cada una de ellas sólo usa 2 celdas de la matriz de confusión)

- G-media

$$G = \sqrt{\text{precision} \cdot \text{recall}}$$

- Media geométrica entre ambas medidas

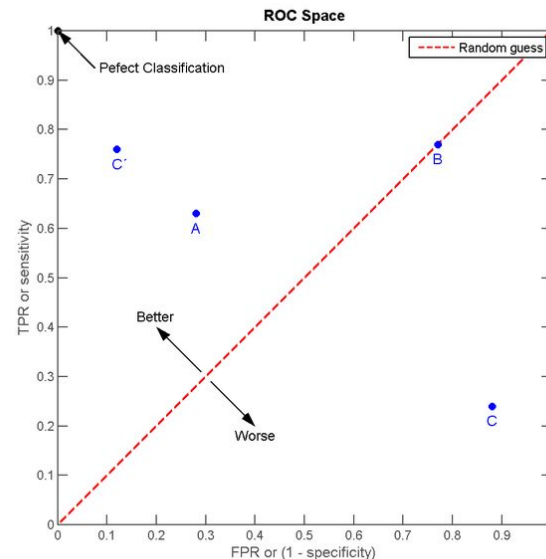
Evaluando Clasificadores

- Cuando tenemos más de dos clases
 - Hablamos de Precision y Recall, para cada una de las clases
 - $\text{Precision}(C_i) = n_{ii}/P_i$
 - $\text{Recall}(C_i) = n_{ii}/R_i$

		PREDICTED CLASS					
		C1	C2	Cn	Sum	Recall
ACTUAL CLASS	C1	n_{11}	n_{12}	n_{1n}	R_1	n_{11}/R_1
	C2	n_{21}	n_{22}	n_{2n}	R_2	n_{22}/R_2
	Cn	n_{n1}	n_{n2}	n_{nn}	R_n	n_{nn}/R_n
	Sum	P_1	P_2	P_n	T	\bar{R}
Precision		$\frac{n_{11}}{P_1}$	$\frac{n_{22}}{P_2}$	$\frac{n_{nn}}{P_n}$	\bar{P}	$\frac{n_{11} + n_{22} + \dots + n_{nn}}{T}$

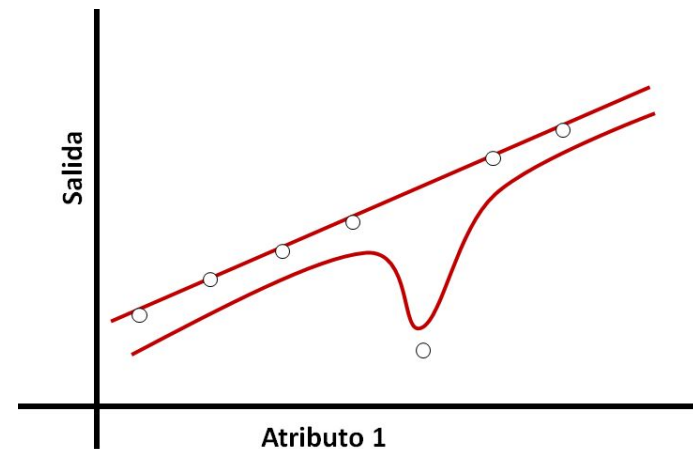
Evaluando Clasificadores

- Cuando tenemos más de dos clases
 - Podemos calcular promedios de la precisión o recall
 - O promedios ponderados en función del número de ejemplos de cada clase
 - O la raíz cuadrada del producto de los aciertos en cada clase



Evaluando Regresores

- La manera más inmediata sería medir el error (absoluto) promedio, con todos los datos
 - Mean Absolute Error
 - Pero, ¿qué es mejor?:
 - Un sistema siempre perfecto que se equivoca mucho una vez?
 - Un sistema que siempre se equivoca un poco?
 - ¿Es igual equivocarse en 500€?
 - Prediciendo el salario de una persona
 - Prediciendo el salario de Cristiano Ronaldo



Evaluando Regresores

- **MSE: Error cuadrático medio**
 - (Los errores mayores penalizan más)
- **MAPE: Error medio absoluto porcentual**
 - (Los errores sobre cantidades mayores penalizan menos)
- **R²: Coeficiente de Determinación**
 - % de la varianza explicada por la regresión - cómo de bien serán predichos futuros ejemplos
 - Cómo de bueno es con respecto a un promedio simple.

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

Copyright (c) University of Deusto

This work (but the quoted images, whose rights are reserved to their owners*) is licensed under the Creative Commons "Attribution-ShareAlike" License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>



Enrique Onieva

enrique.onieva@deusto.es

<https://twitter.com/EnriqueOnieva>

<https://www.linkedin.com/in/enriqueonieva/>