Machine Learning
Redes Neuronales

Inmaculada Gutiérrez García-Pardo

Comparación de modelos

En esta sección se presentan los distintos elementos que intervienen en la comparación de modelos: estructuras de validación, medidas de bondad de ajuste, problemas que pueden surgir y formas de evitarlos

Medidas básicas de comparación de modelos

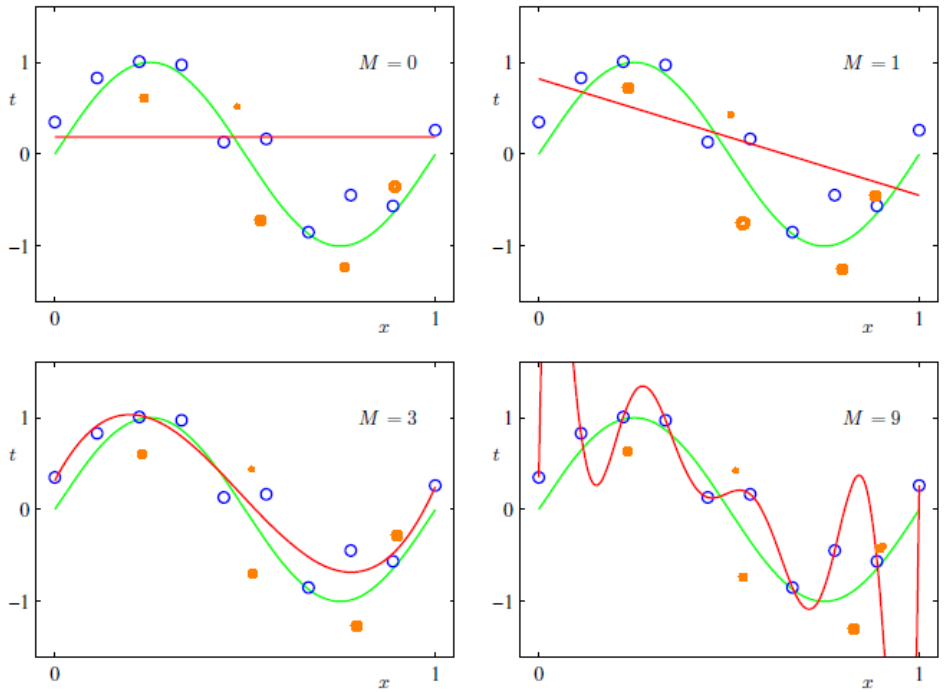
- En general, cuando se quiere evaluar o comparar un **MISMO MODELO** aplicado con **DISTINTOS DATOS**, mejor **AIC, BIC, AICC**

- Cuando se quieren comparar **DISTINTOS MODELOS** para **MISMOS DATOS**, mejor **validación cruzada, validación cruzada repetida, train/test**.

¿Son suficientes las medidas de penalización (AIC, AICC, BIC, Cp, etc. ?

En general, estas medidas no funcionan mal para **selección de variables dentro de un mismo tipo de modelo**, pero hay opciones más apropiadas para determinar qué modelo de entre varios modelos de diferente tipo (redes, regresión, árboles, etc.) funcionará mejor para **datos nuevos.**

Definición 1: Sobreajuste (Overfitting): cuando un modelo está demasiado ajustado a los datos utilizados para su construcción, y funciona relativamente mal para **nuevos datos**. Se da este fenómeno cuando el modelo es **excesivamente complejo** o con un **número de parámetros excesivo**.



Datos entrenamiento azul

Nuevos datos en naranja

El modelo verde es bueno

- M=0, M=1: el modelo **rojo** es insuficientemente complejo**. La predicción es muy pobre para los datos utilizados y para nuevos datos.**
- M=3: el modelo **rojo** es adecuado (recordar la multiplicidad de modelos: no existe un único modelo correcto)**. La predicción para nuevos datos es correcta.**
- M=9: El modelo **rojo** es excesivamente complejo: **está sobreajustado. La predicción para nuevos datos será muy pobre.**

Definición 2: Capacidad de generalización, eficacia predictiva: capacidad de un modelo de predecir bien nuevos datos tomados de la misma población que los datos training. Estos conceptos deben tenerse en cuenta paralelamente o por encima, de los clásicos conceptos de ajuste.

Métodos más habituales para medir o comparar la capacidad predictiva de uno o varios modelos DISTINTOS

TRAINING, VALIDATION, TEST

Se dividen aleatoriamente (o por procedimientos de muestreo) los datos en tres grupos. Se necesitan muchas observaciones. Dependiendo de la cantidad de observaciones, algunas particiones típicas son (70, 20, 10), (60, 20,20), (80, 10, 10), donde cada porcentaje se reserva a los datos:

- **Datos Training**: se utilizan para estimar los posibles modelos
- **Datos Validation**: se aplican los modelos sobre esos datos y se observa su performance. En algunos métodos como redes neuronales o gradient boosting, intervienen de alguna forma en el proceso de construcción de modelos.
- **Datos Test**: se utilizan para calibrar el funcionamiento predictivo real de los modelos. Sirven para evaluar el error de predicción del modelo "óptimo".

El error de predicción puede estar subestimado (en realidad puede ser mayor, pues se ha elegido el modelo que mejor funciona en los datos test).

Métodos más habituales para medir o comparar la capacidad predictiva de uno o varios modelos DISTINTOS

TRAINING, TEST

Se dividen aleatoriamente los datos en dos grupos. Dependiendo de la cantidad de observaciones, algunas particiones típicas son (80, 20); (67, 33); (50, 50), donde cada porcentaje se reserva a los datos

- **Datos Training:** se utilizan para estimar los posibles modelos
- **Datos Test:** se aplican los modelos sobre esos datos y se observa su performance.

El error de predicción puede estar subestimado (en realidad puede ser mayor, pues se ha elegido el modelo que mejor funciona en los datos test).

VALIDACIÓN CRUZADA (k grupos)

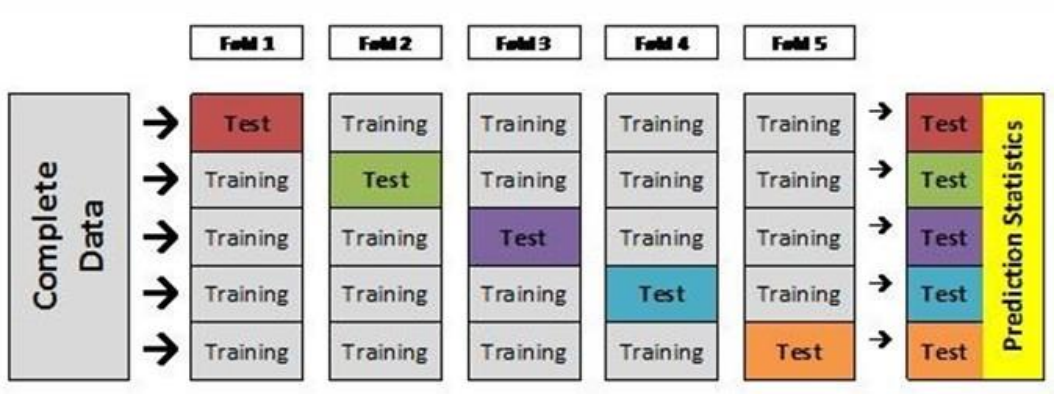
Si no se tienen muchas observaciones, la división en Training-Validación-Test arrojará resultados muy variables según la selección aleatoria de esos grupos.

Algoritmo de validación cruzada

1) Se dividen los datos aleatoriamente en *k* grupos. La idea es que **todos los datos pasen** por alguna fase de **train** y por alguna de **test**.
2) Se realiza iterativamente:

Desde i=1 hasta k
Dejar aparte el grupo i;
Construir el modelo con los grupos restantes;
Estimar el error (por ejemplo ASE) al predecir el grupo i: $Error_i$
Fin

3) Una medida de error de predicción del modelo será la suma o media de los errores, $Error_i$



Con distintos conjuntos de train/test, usamos la validación cruzada para evaluar cómo se comporta el modelo

Ventajas de la validación cruzada:

- Es el mejor esquema de validación, pues utiliza todos los datos
- La validación cruzada repetida permite evaluar la variabilidad del modelo

Desventajas de la validación cruzada:

- A veces el resultado (qué modelo funciona mejor) depende del número de grupos *k*
- El método también depende de la asignación aleatoria a los grupos (habría que probar con varias semillas)
- El error de predicción también suele estar subestimado
- Algunos investigadores proponen incluir la selección de variables para cada iteración

Selección de variables en cada paso de la validación cruzada

- Seguramente mejora el modelo, pero puede tener mucho coste computacional

En problemas de regresión cuantitativa: se suele usar ASE-MSE (error cuadrático medio) como medida básica del **error de predicción**

En problemas de clasificación: se suele usar la tasa de fallos como **error de predicción,** el accuracy o la tasa de acierdos. No obstante, lo ideal es fijarse en la **matriz de confusión** y establecer la medida de ajuste mejor para el problema abordado…*¿es mejor que me equivoque en falsos positivos que en falsos negativos?*

Para evaluar la capacidad predictiva del modelo, **de peor a mejor metodología**:

a) Training-test una sola vez

Recomendado solo para datos muy grandes, y siempre dependiendo de la complejidad del modelo (número de variables, número de parámetros)>>10.000 observaciones.

b) Training-test repetido, variando la semilla de selección

Mejor que el anterior esquema, pero solo para datos muy grandes, y siempre dependiendo de la complejidad del modelo (número de variables, número de parámetros)>>5.000 observaciones.

c) Validación cruzada una sola vez

Mejor que el anterior esquema, para datos moderados o grandes, según la potencia del ordenador, y siempre dependiendo de la complejidad del modelo (número de variables, número de parámetros)>>5.000 observaciones.

d) Validación cruzada repetida, variando la semilla de ordenación

El mejor esquema, para datos pequeños, moderados o grandes, según la potencia del ordenador, y siempre dependiendo de la complejidad del modelo (número de variables y parámetros)>>500 observaciones. Se repite el esquema de la validación cruzada tantas veces como se indique.

El intercambio Sesgo-Varianza

(The Bias-Variance Trade-off)

- El **sesgo** representa el **error** cometido al ajustar el modelo durante el **entrenamiento**
- La **varianza** representa el **error** cometido cuando se aplica el error obtenido a **datos nuevos**
- Lo ideal sería que SESGO y VARIANZA fueran lo MÁS PEQUEÑOS POSIBLE (buscar un **equilibrio**)

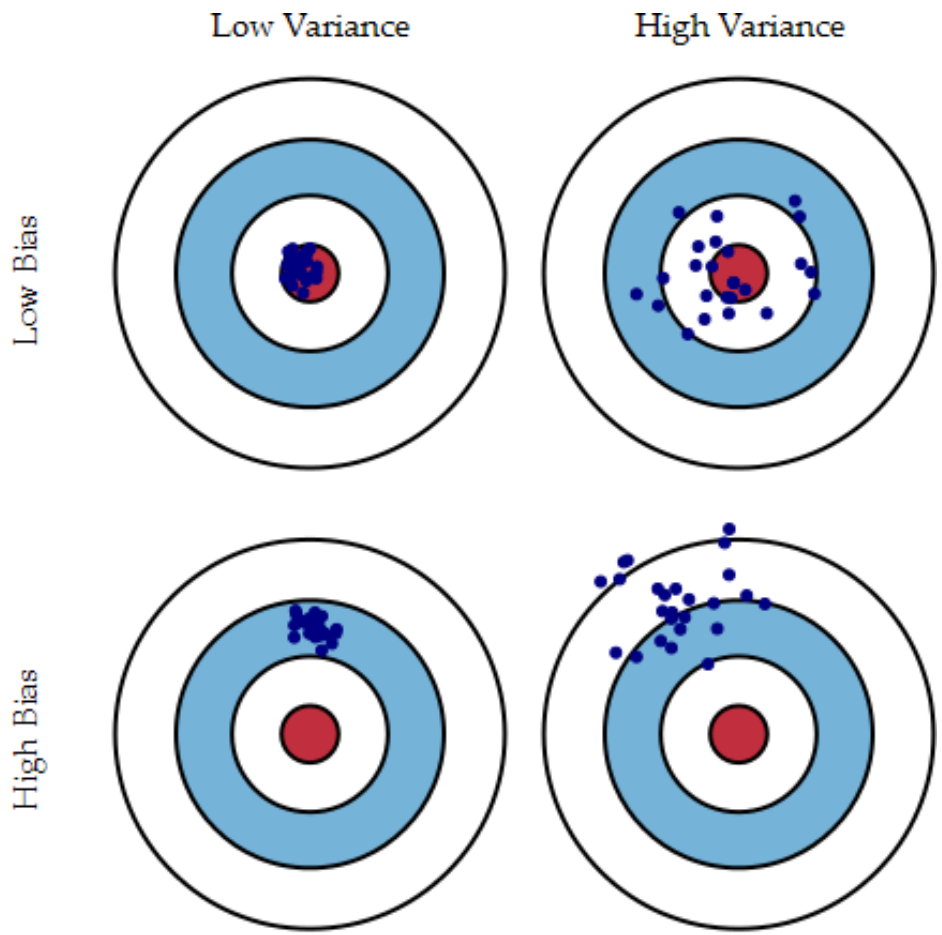


Fig. 1 Graphical illustration of bias and variance.

Al utilizar un modelo predictivo, el **error global** cometido por éste se debe a tres factores que se suman:

- El error "irreducible" o no explicable
- El sesgo[2], que es el error promedio cometido por desfase de los verdaderos valores.
- La varianza, que es la variabilidad de las predicciones del modelo al variar los datos utilizados para construirlo.

We may estimate a model $\hat{f}(X)$ of $f(X)$ using linear regressions or another modeling technique. In this case, the expected squared prediction error at a point x is:

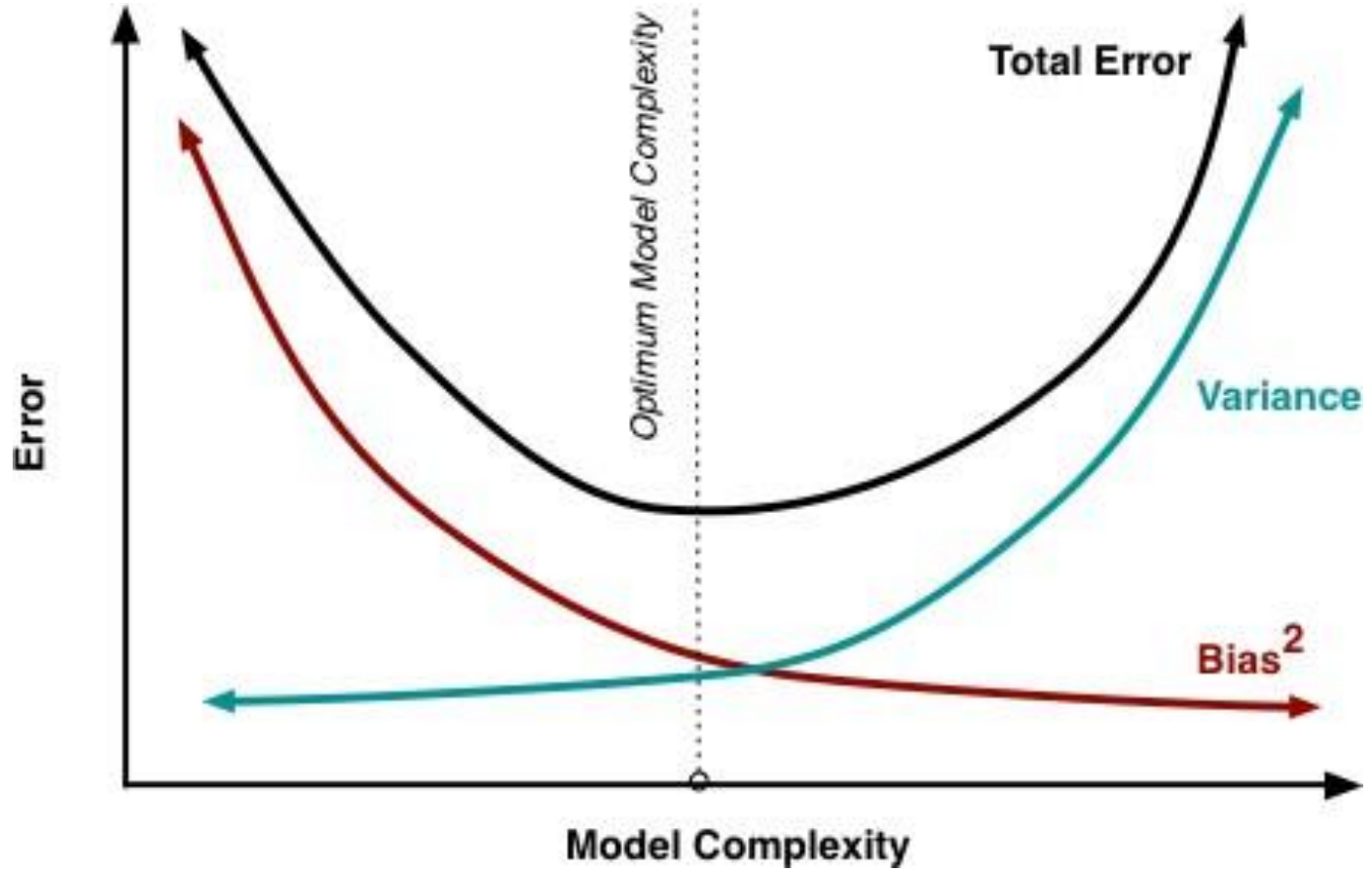
$$Err(x) = E\left[(Y - \hat{f}(x))^2\right]$$

This error may then be decomposed into bias and variance components:

$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\hat{f}(x) - E[\hat{f}(x)]\right]^2 + \sigma_e^2$$

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

¡¡¡EQUILIBRIO ENTRE SESGO Y VARIANZA!!!



- Cuanto más complejo es el modelo, más se ajusta a los datos usados para su construcción, por lo que mayor es la varianza al usar datos nuevos.
- Cuanto más simple es el modelo, más general es para datos nuevos, por lo que inferior es la varianza, pero como no se ajusta perfectamente a los datos usados para su construcción, aumenta el sesgo.

INCISO: interpretación del gráfico de cajas y bigotes

Los gráficos de cajas y bigotes son métodos gráficos para representar una serie de datos numéricos a partir de sus cuartiles: de un vistazo se puede ver la **mediana**, **cuartiles**, y **valores atípicos**.

Los **bigotes** son las líneas que se extienden fuera de la caja. Indican la variabilidad fuera de los cuartiles superior e inferior: cualquier punto fuera de esas líneas o bigotes se considera un valor atípico.

En nuestro caso, en el eje X representamos cada modelo. En el eje Y representamos la media del error, con lo que lo utilizaremos para evaluar el valor del sesgo. Cuanto más "abajo" en el eje Y aparezca una caja, menor será su sesgo. Cuanto más "corta" sea la caja, menor será su variabilidad.

En esta tabla se presentan varios estudios, en los que se han utilizado distintas medidas de error, tamaños de datos y métodos de validación, así como la comparativa entre métodos clásicos y métodos de machine learning.

Table 1
Applications in accounting and finance

Reference	Statistical model	No. of variables	Sample size	Validation Method	Error measure	Finding
Odom and Sharda (1990)	DA	5	129	Tr-Ts /R-3 times	Confusion matrix	[A]
Duliba (1991)	Reg	5-10	600	Tr-Ts	R^2 Value	[A]-Random effect [C]-Fixed effect
Salchenberger et al. (1992)	Logit	29	3479	Tr-Va-Ts	Confusion matrix	[A]*
Tam and Kiang (1992)	k-NN, DA, ID3	19	118	Jackknifing	Confusion matrix	[A]
Fletcher and Goss (1993)	LR	3	36	18-fold CV	Confusion matrix, MSE	[A]
Yoon et al. (1993)	DA	4	151	Tr-Ts (50–50)	Confusion matrix	[A]
Altman et al. (1994)	DA	10- DA 15-NN	1108	Tr-Ts (70–30)	Confusion matrix	[C]
Dutta et al. (1994)	Reg, LR	6, 10	47	Tr-Ts (70–30)	Confusion matrix	[A]
Wilson and Sharda (1994)	DA	5	129	Tr-Ts/R-3 times	Confusion matrix	[A]*
Boritz and Kennedy (1995)	Logit, Probit, DA	5, 9	342	Tr-Ts (70–30) / R-5 times	Confusion matrix	[B]
Lenard et al. (1995)	LR	4 & 8	80	Tr-Ts (50–50)	Confusion matrix	[A]*
Desai et al. (1996)	LR, DA	18	2733	Tr-Ts (70–30) / R-10 times	Confusion matrix	[B]*
Leshno and Spector (1996)	DA	41	88	Tr-Ts	Confusion matrix	[A]*
Jo et al. (1997)	DA, CBR	20	564	Tr-Ts	Confusion matrix	[A]*
Spear and Leis (1997)	DA, LR, Reg	4	328	Tr-Va-Ts (76-12-12)	Confusion matrix	[B]
Zhang et al. (1999)	LR	6	220	5-fold CV	Confusion matrix	[A]*
Lee and Jung (2000)	LR	11	21678	Tr-Ts	C-index, Some measure for degree of separation	[A]-Rural customer [C]-Urban customer
Limsombunchai et al. (2005)	LR	11	16560	Tr-Ts	Confusion matrix	[B]
Lee et al. (2005)	DA, LR	5	168	4-fold CV	Confusion matrix	[A]*
Pendharkar (2005)	C4.5, DA	3	100- sim 200-real	Bootstrapping	Confusion matrix	[A]*
Landajo et al. (2007)	Robust reg, Loglinear reg	9	Multiple models	Tr-Ts	MAE	[C]*

Conceptos básicos sobre optimización

Objetivo del algoritmo de optimización: minimizar la función de error o función objetivo (en este caso, diferencia entre valor real y valor estimado):

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

La predicción $\hat{y}_i$ depende de la forma funcional de la red. En el primer ejemplo

$$\begin{aligned}\hat{y}_i =& \tanh(w_{1,out}(\tanh(w_{11}X1+w_{21}X2+w_{31}X3+w_{41}X4+b_1))\\ &+w_{2,out}(\tanh(w_{12}X1+w_{22}X2+w_{32}X3+w_{42}X4+b_2))+\\ &+w_{3,out}(\tanh(w_{13}X1+w_{23}X2+w_{33}X3+w_{43}X4+b_3))+\\ &+w_{4,out}(\tanh(w_{14}X1+w_{24}X2+w_{34}X3+w_{44}X4+b_4))+\\ &+b_{out})\end{aligned}$$

Y habrá que hallar los **parámetros** (que llamaremos pesos o weights) $w_{1,out}$, w_{11}, w_{21}, …, etc. que **minimicen el MSE**. Es decir, se trata de minimizar MSE como función de los parámetros $w_{1,out}$, w_{11}, w_{21}, …, etc.

Es decir

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

$$min_{w_{ij}} \sum_{i=1}^{n} \frac{1}{n} (y_i - \tanh(w_{1,out}(\tanh(w_{11}x_{1i} + \cdots) + w_{2,out}(\tanh(w_{21}x_{1i} + \cdots) + \cdots))))^2$$

- Se trata de una función de los parámetros wij.
- Hay que hallar los valores de los wij que hagan mínima esa función.
- Todos los valores x_{1i}, x_{2i}, etc. son valores reales de nuestros datos: x_{1i} valor en la observación *i* de la variable x_1; x_{2i} valor en la observación *i* de la variable x_2, etc. El sumatorio tiene tantos términos como observaciones (por ejemplo, 5000 términos).
- La complejidad y dificultad del proceso de optimización viene dada por el número de parámetros a estimar.
- OJO: para encontrar los valores w_{ij} que minimizan el MSE no hace falta tener en cuenta $\frac{1}{n}$

Como se trata de una función de los parámetros **no lineal** y **compleja**, es necesario utilizar **algoritmos numéricos de aproximación**.

En general, el funcionamiento es similar en todos ellos:

1) Tomar unos **valores iniciales** para los pesos (se suelen aleatorizar)
2) Calcular la **función de error** con esos valores
3) Calcular una **dirección de decrecimiento** de la función de error
4) **Retocar** los valores de los pesos en esa dirección de decrecimiento
5) Volver al paso 2)

El algoritmo se detiene al llegar a cualquier **criterio de parada** preestablecido.

En general los algoritmos que utilizaremos se basarán en el
Algoritmo básico Gradient Descent

El objetivo es hallar los pesos w_{ij} que minimicen el error (MSE)
Proceso básico:
1) Se inicializan los pesos wij aleatoriamente

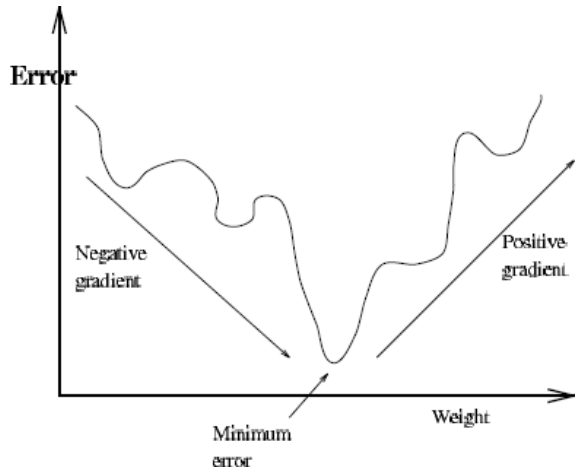
2) Se calcula la derivada de la función del error en ese punto (esos pesos concretos w_{ij}): $\frac{\partial E(n)}{\partial \omega_{ij}}$

3) Se hacen variar los pesos en la dirección de descenso del error (recordemos que wij es un vector) (los pesos varían según un learning rate ε):

$$\Delta\omega_{ij}(n) = -\epsilon\frac{\partial E(n)}{\partial \omega_{ij}}$$

$$\omega_{ij}(n+1) = \omega_{ij}(n) + \Delta\omega_{ij}(n)$$

4) Se repiten los pasos 2 y 3 hasta llegar a un criterio de parada.



En ciertas versiones se introduce una constante α llamada weight decay o momentum:

$$\Delta\omega_{ij}(n) = -\epsilon\frac{\partial E(n)}{\partial\omega_{ij}} + \alpha\Delta\omega_{ij}(n-1)$$

OJO:

1) **El learning rate ε refleja en qué medida vamos a cambiar los pesos w_{ij} en cada iteración** (interesa que sea pequeño):

a) Learning rate muy bajo (p.ej. 0.001) hace que el proceso de optimización sea lento, y con el peligro de quedarse enganchado en óptimos locales (el entrenamiento preliminar reduce este riesgo)

b) Learning rate muy alto (p. ej.0.5) hace diverger demasiado los valores provocando inconstancia y mala optimización

2) El **weight decay** o momentum (<1 siempre) hace que en cada iteración del algoritmo los pesos se vayan cambiando algo menos, acercándose "con cuidado" al mínimo, más despacio cada vez. **Medida para limitar el efecto de los pesos altos**.

Nota 1

La **inicialización aleatoria de los pesos** puede tener gran influencia en los resultados, cambiando mucho los parámetros finales de la red, si ésta no está bien calibrada o el programa de optimización no es demasiado fino. (**probar a variar la semilla de la red en el primer ejemplo cstrengh –age, water y observar el MSE sobre datos test**).

Una manera de corregir esto es construir la red de manera repetida con diferentes semillas y promediar la predicción.

Nota 2

Aumentar mucho el número de iteraciones en algunos casos puede afectar negativamente a la red, ajustando "demasiado bien" los pesos y provocando mala generalización (la red funciona mal para datos test a partir de un número de iteraciones). Reducir el número de iteraciones para evitar esto es una técnica que se denomina "**early stopping**". No siempre es necesario, depende de los casos.

Ejemplo de los valores calculados para los parámetros de una red

```
a 2-3-1 network with 13 weights
options were - linear output units
 b->h1 i1->h1 i2->h1
  3.05   3.31   0.13
 b->h2 i1->h2 i2->h2
 -0.96   0.13  -1.89
 b->h3 i1->h3 i2->h3
-73.01   0.02  36.16
  b->o  h1->o  h2->o  h3->o
-51.42  88.52  28.69   8.51
```

Todo lo que va al nodo 1:

```
 b->h1 i1->h1 i2->h1
  3.05   3.31   0.13
```

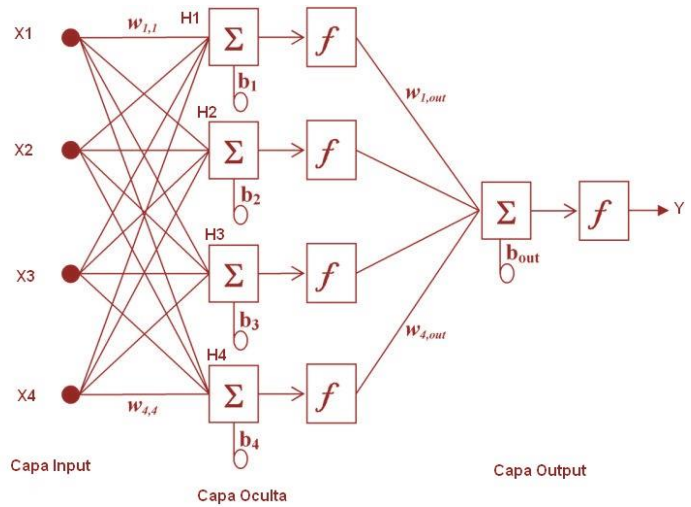
Esto significa:

h1=w11*age+w21*water+b1
h1=3.31*age+0.13*water+3.05

luego activación:

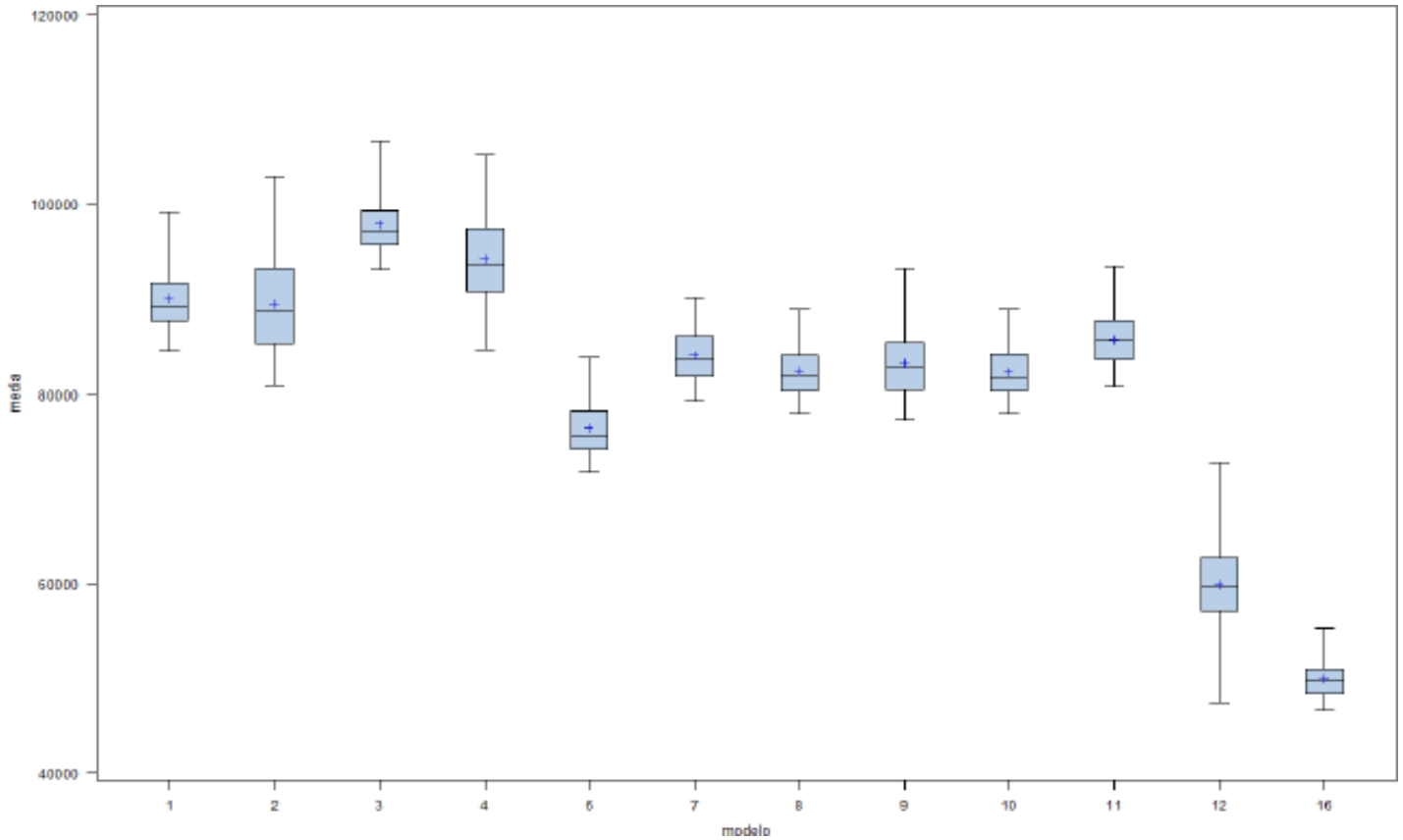
h1=tanh(h1)

```
# weights:  13

initial  value 1601216.121533
iter     value 166917.729534
       1
0
iter     value 147054.970173
       2
0
iter     value 144565.081887
       3
0
iter     value 143638.206741
       4
0
iter     value 143070.915174
       5
0
iter     value 141797.145971
       6
0
iter     value 141156.059581
       7
0
iter     value 140535.605141
       8
0
iter     value 140427.683543
       9
0
iter 100 value 140017.488970

final  value 140017.488970
stopped after 100 iterations
```



Una manera de explorar sesgo y varianza es haciendo **validación cruzada repetida (k-fold repeated crossvalidation)** y observando el error promedio cometido (sesgo) y su variabilidad (altura de la caja). Se puede realizar a través de boxplot .

El modelo 12 tiene sesgo bajo pero alta varianza; si es un modelo complicado puede ser peligroso/inestable a pesar de su bajo sesgo. El modelo 16 tiene bajo sesgo y baja varianza: **parece óptimo**.



		Quiero bajar el Sesgo	**Quiero bajar la Varianza**
General (1)	número de variables	+	-
	introducir interacciones entre variables, creación de dummies	+	-
	Complejidad del modelo	+	-
	tamaño muestral	+	+
Logística , regresión			
	p-valor (stepwise, backward, forward) (2)	+	-
	categorías poco representadas en variables input	+=	-
Redes			
	número de nodos	+	-
	linealidad (3)	-	-
	número de iteraciones en algoritmo de optimización (usar early stopping puede reducir la varianza)	+	-
	en algoritmo de optimización: learn rate,momentum (en general si subimos momentum bajamos learn rate). learn rate bajos requieren nº de iteraciones más altas	-	+

1) En general, hacer **el modelo más complejo suele reducir el sesgo y aumentar la varianza**. Esto incluye aumentar el número de variables, utilizar variables categóricas con muchas categorías, incluir interacciones, utilizar modelos no lineales (por ejemplo regresión polinómica), etc.
Aumentar el tamaño muestral tiene incidencia positiva sobre el sesgo y la varianza: ambas cantidades disminuyen aumentando n.

2) El **p-valor de corte** en procesos de selección de variables mide la **exigencia** en la inclusión de variables o efectos en el modelo. A menor p-valor, más exigente es la inclusión de variables; reducir el p-valor implica entonces menos variables y por lo tanto más sesgo pero menos varianza. Aumentar el p-valor permite la inclusión de más variables y por lo tanto modelos más complejos que pueden tener menos sesgo pero más varianza.

3) La **linealidad** en modelos de regresión o separabilidad lineal en modelos de clasificación mejora en general ambos sesgo y varianza. Si se puede conseguir esta linealidad con transformaciones sencillas y utilizar regresión o regresión logística (en clasificación) puede ser mejor que probar algoritmos más complicados como redes.

4) Monitorizar modelos con **excesivo número de aspectos** a probar (activación, algoritmo, etc.) y centrándonos en el resultado que se tiene en datos test puede llevar a un sobreajuste implícito. Esto se suele denominar **Sesgo de Selección de Modelos**.

Medidas de bondad de ajuste: Clasificación

Indican cómo de bien el modelo predice las **clases** en comparación con los **datos reales**. Se mide usando métricas específicas, dependiendo de lo que quieras evaluar

- **MATRIZ DE CONFUSIÓN:** resumen de las predicciones correctas e incorrectas de cada clase

	PREDICCIÓN (Clase 0)	PREDICCIÓN (Clase 1)
REAL (Clase 0)	Verdadero Negativo (VN/TN)	Falso Positivo (FP/FP)
REAL (Clase 1)	Falso Negativo (FN/FN))	Verdadero Positivo (VP/TP)

- **ACCURAY (Precisión Global):** porcentaje de predicciones correctas sobre el total

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

- **PRECISIÓN (Precisión por Clase):** ¿cuántas predicciones han sido correctas? Especialmente importante cuando los datos son desbalanceados.

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

- **RECALL/SENSIBILIDAD:** qué porcentaje de los casos positivos reales son correctamente clasificados

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

- **F1-SCORE:** promedio armónico entre precisión y sensibilidad. Especialmente importante cuando los datos son desbalanceados.

$$F1 = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}$$

Medidas de bondad de ajuste: Clasificación

- **ESPECIFIDAD (Tasa Verdaderos Negativos):** mide qué proporción de los negativos reales son correctamente identificados como negativos. Se enfoca en los negativos reales

$$Especifidad = \frac{VN}{VN + FP}$$

- **La Curva ROC:** representa el porcentaje de verdaderos positivos (**Eje Y: True Positive Rate, *TPR***), también conocido como **Recall**, contra el ratio de falsos positivos (**Eje X: False Positive Rate, *FPR=1-Especifidad***). muestra gráficamente cómo de bien un modelo de clasificación puede separar dos clases al cambiar el umbral de decisión

$$Recall = TPR = \frac{VP}{VP + FN}; \qquad 1 - Especifidad = FRP = \frac{FP}{VN + FP}$$

- **AUC** (*Área Bajo la Curva/Area Under Curve*): métrica sólida y muy útil para problemas de clasificación binaria. Toma valores entre 0 y 1, donde 0.5 indica el resultado de una clasificación aleatoria, 0 una clasificación aún peor que el azar, y 1 indica el resultado de la clasificación perfecta.

Medidas de bondad de ajuste: Clasificación

- **Datos:** *Real={Spam, Spam, Spam, No Spam, Spam, No Spam}; Predicción={Spam, Spam, No Spam, Spam, No Spam, No Spam}; Spam: 1, No Spam: 0*

- **MATRIZ DE CONFUSIÓN:**

	PREDICCIÓN (Clase 0)	PREDICCIÓN (Clase 1)
REAL (Clase 0)	1 (VN/TN)	1 (FP/FP)
REAL (Clase 1)	2 (FN/FN)	2 (VP/TP)

- **ACCURAY (Precisión Global):** porcentaje de predicciones correctas sobre el total

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{VP + VN}}{\boldsymbol{Total\ predicciones = VP + VN + FP + FN}} = \frac{\mathbf{2+1}}{\mathbf{2+1+1+2}} = \frac{\mathbf{3}}{\mathbf{6}} = \mathbf{50\%}$$

- **PRECISIÓN (Precisión por Clase):** ¿cuántas predicciones han sido correctas? Especialmente importante cuando los datos son desbalanceados.

$$\boldsymbol{Precision} = \frac{\boldsymbol{VP}}{\boldsymbol{VP + FP}} = \frac{\mathbf{2}}{\mathbf{2+1}} = \mathbf{66.6\%}$$

- **RECALL/SENSIBILIDAD:** qué porcentaje de los casos positivos reales son correctamente clasificados

$$\boldsymbol{Recall} = \frac{\boldsymbol{VP}}{\boldsymbol{VP + FN}} = \frac{\mathbf{2}}{\mathbf{2+2}} = \mathbf{50\%}$$

- **F1-SCORE:** promedio armónico entre precisión y sensibilidad. Especialmente importante cuando los datos son desbalanceados.

$$\boldsymbol{F1} = \mathbf{2}\ \boldsymbol{x}\ \frac{\boldsymbol{Precision\ x\ Recall}}{\boldsymbol{Precision +\ Recall}} = \mathbf{2}\boldsymbol{x}\frac{\mathbf{0.66\ x\ 0.5}}{\mathbf{0.66 + 0.5}} = \mathbf{57.1\%}$$

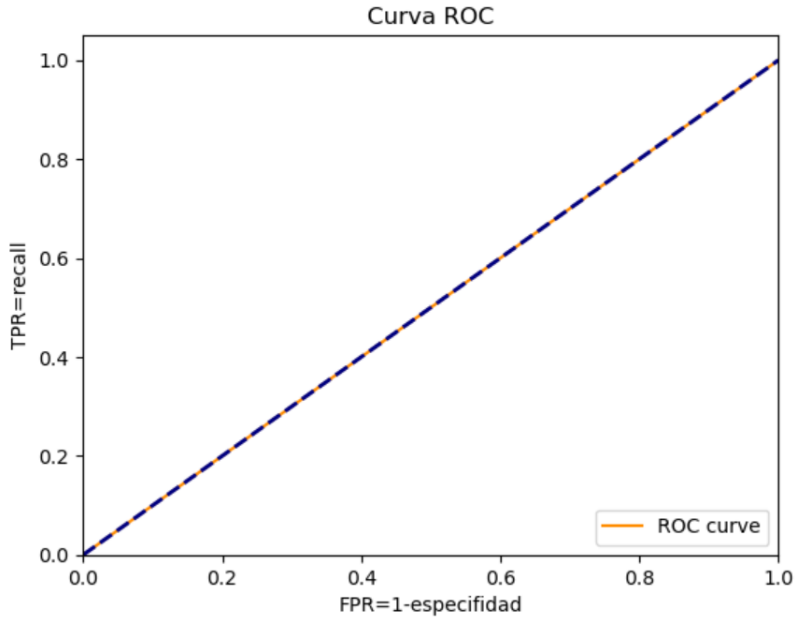
Medidas de bondad de ajuste: Clasificación

- **ESPECIFIDAD (Tasa Verdaderos Negativos):**

$$Especifidad = \frac{VN}{VN + FP} = \frac{1}{1+1} = 50\%$$

- **La Curva ROC:**

$$Recall = TPR = \frac{VP}{VP + FN} = \frac{2}{2+2} = 50\%; \qquad 1 - Especifidad = FRP = \frac{FP}{VN + FP} = \frac{1}{1+1} = 50\%$$



- **AUC** (*Área Bajo la Curva/Area Under Curve*): 0.5

Medidas de bondad de ajuste. ¿Cómo afectan los ajustes?

Ajuste de parámetros: afecta a la capacidad general del modelo para adaptarse a los datos.

- **Sensibilidad (TPR o Recall):** si el modelo es demasiado restrictivo o simple (bajo ajuste), podría no detectar correctamente los positivos reales, disminuyendo la sensibilidad. *Ejemplo*: En k-NN, un valor muy grande de k podría suavizar demasiado las predicciones, afectando la detección de positivos.

- **Especificidad:** un modelo muy complejo (sobreajustado) podría clasificar erróneamente más negativos reales, disminuyendo la especificidad. *Ejemplo*: Un árbol de decisión muy profundo podría clasificar demasiados ejemplos negativos como positivos.

- **Precisión:** los parámetros que reducen los falsos positivos aumentan la precisión, ya que esta mide qué porcentaje de las predicciones positivas son correctas. *Ejemplo*: En SVM, aumentar el valor de C puede reducir los falsos positivos, aumentando la precisión.

- **F1-Score:** como combina precisión y sensibilidad, un cambio que mejore una métrica pero afecte negativamente la otra podría reducir el F1-Score.

Medidas de bondad de ajuste. ¿Cómo afectan los ajustes?

Threshold o punto de corte

El *threshold* define a partir de qué probabilidad un ejemplo se clasifica como positivo. Cambiar este umbral tiene los siguientes efectos:

- **Sensibilidad (TPR)**
  - **Umbral bajo:** Detecta más positivos, aumentando la sensibilidad, pero podría incrementar los falsos positivos.
  - **Umbral alto:** Detecta menos positivos, reduciendo la sensibilidad, pero mejora la especificidad.
- **Especificidad**
  - **Umbral bajo:** Clasifica menos negativos correctamente, reduciendo la especificidad.
  - **Umbral alto:** Clasifica más negativos correctamente, aumentando la especificidad.
- **Precisión**
  - **Umbral bajo:** Aumenta los falsos positivos, reduciendo la precisión.
  - **Umbral alto:** Reduce los falsos positivos, aumentando la precisión.
- **F1-Score:** cualquier cambio extremo puede desbalancear sensibilidad y precisión, reduciendo el F1-Score. Se maximiza cuando el balance entre ambas métricas es óptimo.

Medidas de bondad de ajuste. ¿Cómo afectan los ajustes?

Supongamos que un correo se clasifica como *Spam* con una probabilidad de ≥0.5

Reducir el umbral (por ejemplo, 0.30)**:**

Detectará más correos spam (alta sensibilidad).

Aumentará los falsos positivos (baja precisión y especificidad).

Aumentar el umbral (por ejemplo, 0.70):

Detectará menos correos spam (baja sensibilidad).

Mejorará la precisión y la especificidad, ya que reducirá los falsos positivos.

Medidas de bondad de ajuste. ¿Cuál escoger?

- **Prioridad: minimizar FALSOS NEGATIVOS (ALTA SENSIBILIDAD):** no identificar un positivo es más costoso/grave que clasificar incorrectamente un negativo. *Ejemplos*:
  - Diagnóstico médico: es más grave no identificar un paciente con cáncer (falso negativo) que generar una alerta innecesaria (falso positivo).
  - Seguridad: identificar correos con malware. No marcar un malware como peligroso puede hundir un sistema.
  - **Métrica recomendada: recall/sensibilidad**, si fuera necesario equilibrar, **F1**.
- **Prioridad: minimizar FALSOS POSITIVOS (ALTA PRECISIÓN):** clasificar un negativo como positivo tiene un alto coste/gravedad *Ejemplos*:
  - Detección de fraude: clasificar una transacción legítima como fraudulenta puede provocar la huida de clientes.
  - Clasificación de spam: identificar un correo importante como spam (falso positivo), puede provocar que el usuario pierda información importante.
  - **Métrica recomendada: precisión**, si fuera necesario equilibrar, **F1**.
- **Prioridad: BALANCEAR FALSOS POSITIVOS Y FALSOS NEGATIVOS:** es importante evitar tanto los falsos positivos como los falsos negativos. *Ejemplos*:
  - Reconocimiento facial: en accesos restringidos, un error podría negar la entrada a un usuario legítimo (falso negativo) o permitir el acceso a un intruso (falso positivo)
  - Sistemas de recomendación: clasificar incorrectamente un producto como relevante o no relevante afecta la experiencia del usuario.
  - **Métrica recomendada: F1**, para considerar el equilibrio entre precisión y sensibilidad, **Accuracy** (si las clases están más o menos balanceadas, **ROC-AUC**.
- **Problemas con clases DESBALANCEADAS: cuando** una clase es mucho más frecuente que la otra, las métricas estándar como la exactitud (accuracy) pueden ser engañosas. *Ejemplos*:
  - Detección de fallos en maquinaria: si los fallos son eventos raros (clase minoritaria). Un modelo que prediga siempre "no falla" podría tener alta exactitud pero sería inútil.
  - Sistemas de recomendación: clasificar incorrectamente un producto como relevante o no relevante afecta la experiencia del usuario.
  - **Métrica recomendada: ROC-AUC** para medir la capacidad del modelo de diferenciar entre las clases. También se puede usar **Precision-Recall AUC** si la clase positiva es extremadamente minoritaria.
- **Problema de clasificación MULTI-CLASE:** hay más de una clase y todas tienen una importancia relativa. *Ejemplos*:
  - **Clasificación de noticias:** Etiquetar artículos en categorías como deportes, política, tecnología.
  - **Reconocimiento de imágenes:** Identificar objetos en imágenes (perro, gato, coche, etc.).
  - **Métrica recomendada: Acuracy**, si las clases están balanceadas; **Macro F1-Score si**i las clases están desbalanceadas y se quiere evaluar el desempeño promedio en todas las clases.

Medidas de bondad de ajuste: Regresión

1. **Error Cuadrático Medio (MSE)**:

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

donde: - y_i son los valores reales. - $\hat{y}_i$ son los valores predichos. - n es el número total de observaciones.

2. **Raíz del Error Cuadrático Medio (RMSE)**:

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

La diferencia principal con el MSE es que el RMSE toma la raíz cuadrada del MSE, lo que lo hace más interpretable al estar en las mismas unidades que la variable objetivo.

3. **Error Absoluto Medio (MAE)**:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Aquí, la diferencia es simplemente la media de las diferencias absolutas entre los valores reales y los predichos.

4. **R^2 (Coeficiente de Determinación)**:

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

donde: - $\bar{y}$ es la media de los valores reales. - El numerador es la suma de los cuadrados de los errores (SSE), y el denominador es la suma de los cuadrados totales (SST).

Medidas de bondad de ajuste: Regresión

Otras opciones son:

•**Error Relativo Absoluto Medio (MAPE)**: promedio de los errores absolutos expresados como porcentaje de los valores reales.

•**Error Absoluto Mediano (MedAE)**: mediana de los errores absolutos. Menos sensible a los valores atípicos que el MAE.

•**Error Cuadrático Medio Logarítmico (MSLE)**: MSE calculado sobre los logaritmos de los valores predichos y reales. Es útil cuando los datos tienen un rango amplio.

•**Coeficiente de Correlación de Pearson (r)**: Mide la fuerza y la dirección de la relación lineal entre los valores reales y predichos.

•**Prueba F de ANOVA**: Utilizada para comparar los modelos y ver si hay diferencias significativas entre los resultados predichos y los reales.

Medidas de bondad de ajuste: Regresión

Supongamos que tenemos los siguientes valores reales y y predichos $\hat{y}$:

$$y = [3, -0.5, 2, 7]; \quad \hat{y} = [2.5, 0.0, 2, 8]$$

1. Error Cuadrático Medio (MSE)

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

Calculamos las diferencias cuadradas:

$$(3 - 2.5)^2 = 0.25$$
$$(-0.5 - 0.0)^2 = 0.25$$
$$(2 - 2)^2 = 0$$
$$(7 - 8)^2 = 1$$

Sumamos estos valores y dividimos entre el número total de observaciones ($n = 4$):

$$\text{MSE} = \frac{1}{4}(0.25 + 0.25 + 0 + 1) = \frac{1.5}{4} = 0.375$$

2. Raíz del Error Cuadrático Medio (RMSE)

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

La suma de los errores cuadráticos es 1.5. Ahora aplicamos raíz cuadrada:

$$\text{RMSE} = \sqrt{\frac{1.5}{4}} = \sqrt{0.375} \approx 0.612$$

Medidas de bondad de ajuste: Regresión

3. Error Absoluto Medio (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Calculamos las diferencias absolutas:

$$|3 - 2.5| = 0.5$$
$$|-0.5 - 0.0| = 0.5$$
$$|2 - 2| = 0$$
$$|7 - 8| = 1$$

Sumamos estos valores y dividimos entre $n = 4$:

$$\text{MAE} = \frac{1}{4}(0.5 + 0.5 + 0 + 1) = \frac{2}{4} = 0.5$$

4. R^2 (Coeficiente de Determinación)

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

Primero calculamos la media de y:

$$\bar{y} = \frac{3 + (-0.5) + 2 + 7}{4} = \frac{11.5}{4} = 2.875$$

Luego, calculamos la suma de los cuadrados totales (SST):

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = (3 - 2.875)^2 + (-0.5 - 2.875)^2 + (2 - 2.875)^2 + (7 - 2.875)^2$$

$$= (0.125)^2 + (-3.375)^2 + (-0.875)^2 + (4.125)^2$$
$$= 0.015625 + 11.390625 + 0.765625 + 17.003125 = 29.175$$

Ya tenemos la suma de los cuadrados de los errores (SSE), que calculamos anteriormente como 1.5. Ahora podemos calcular R^2:

$$R^2 = 1 - \frac{1.5}{29.175} \approx 1 - 0.0514 = 0.9486$$

Medidas de bondad de ajuste. ¿Cuál escoger?

- **Predicción de valores continuos con distribuciones similares y datos 'limpios':** no identificar un positivo es más costoso/grave que clasificar incorrectamente un negativo. *Ejemplos*:
  - Predicción de precios de viviendas: estimar el precio de una vivienda en base a sus características.
  - Pronóstico de demanda: Predecir la cantidad de productos a vender en el próximo mes.
  - **Métrica recomendada:**
    - **Mean Squared Error (MSE):** penaliza más los errores grandes; útil cuando los errores pequeños son menos relevantes.
    - **Mean Absolute Error (MAE):** penaliza linealmente los errores; útil cuando los errores grandes y pequeños tienen el mismo impacto

- **Predicción de valores continuos con distribuciones desiguales o datos ruidosos** *Ejemplos*:
  - Análisis financiero: predicción de precios de acciones, donde los outliers pueden ser frecuentes.
  - Meteorología: predicción de temperaturas en regiones con valores extremos.
  - Sector energético: predicción del consumo eléctrico en regiones con patrones irregulares o eventos atípicos.
  - **Métrica recomendada:**
    - **Mediana del Error Absoluto:** Más robusta frente a outliers.
    - **R^2** (Coeficiente de determinación): Mide qué tan bien el modelo explica la variabilidad de los datos.
    - **Root Mean Squared Error (RMSE):** Si el objetivo es penalizar más los errores grandes, aunque no es tan robusta frente a outliers