upna
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

TEMA 4: EVALUACIÓN DE MODELOS, BIAS/VARIANZA

Mikel Galar Idoate
mikel.galar@unavarra.es

Ciencia de datos con técnicas inteligentes
Experto Universitario en Ciencia de Datos y Big Data

Índice

1. Análisis y evaluación de algoritmos de aprendizaje
   - Conjuntos train / test
2. Selección de modelos
   - Conjuntos de train / val / test
3. Bias / varianza
4. Análisis del error

Índice

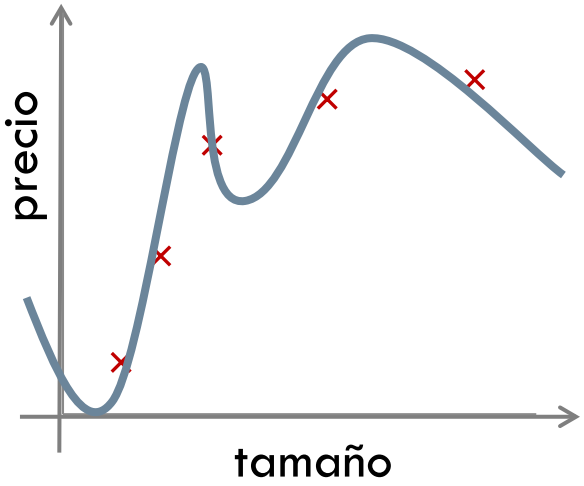
Analizando un algoritmo de aprendizaje

- Supongamos que tenemos un modelo de regresión lineal con regularización para la **valoración de bienes inmuebles**

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

- Sin embargo, al aplicarlo sobre nuevas casas produce **errores inaceptables**, ¿qué podemos hacer?
  - Añadir más ejemplos de entrenamiento
  - Utilizar menos características
  - Añadir nuevas características
  - Añadir características polinomiales (x_1^2, x_2^2, x_1x_2, etc.)
  - Aumentar λ
  - Decrecer λ

Evaluando un algoritmo de aprendizaje



$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

- Nuestra hipótesis **no generaliza bien** ejemplos que no están en el conjunto de entrenamiento
  - Con una variable podemos verlo gráficamente
    - ¿Pero con 100?
  - Los resultados en train son **optimistas**

Evaluando un algoritmo de aprendizaje

- Necesitamos conocer cómo de bien es capaz de clasificar nuevos ejemplos nuestro algoritmo
  - **Conjunto de test** (≠ train)

Tamaño	Precio	
2104	400	70%
1600	330	
2400	369	
1416	232	
3000	540	
1985	300	
1534	315	
1427	199	30%
1380	212	
1494	243	

Conjunto de train →

$$\left(x^{(1)}, y^{(1)}\right)$$
$$\left(x^{(2)}, y^{(2)}\right)$$
$$\vdots$$
$$\vdots$$
$$\left(x^{(m)}, y^{(m)}\right)$$

Conjunto de test →

$$\left(x_{test}^{(1)}, y_{test}^{(1)}\right)$$
$$\left(x_{test}^{(2)}, y_{test}^{(2)}\right)$$
$$\vdots$$
$$\left(x_{test}^{(m_{test})}, y_{test}^{(m_{test})}\right)$$

m_{test} Número de ejemplos en el conjunto de test

Evaluando un algoritmo de aprendizaje

- Proceso de aprendizaje y evaluación:
  1. Aprender los parámetros θ usando los ejemplos en el conjunto de train (minimizando el error en entrenamiento $J(\theta)$)
  2. Calcular el error en el conjunto de test

$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \left(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)} \right)^2$$

Evaluando un algoritmo de aprendizaje

- En clasificación el proceso es equivalente
  1. Aprender los parámetros θcon el conjunto de train
  2. Calcular el error en test
     - Regresión logística:

$$J(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log\left(h_\theta\left(x_{test}^{(i)}\right)\right) + \left(1 - y_{test}^{(i)}\right) \log\left(1 - h_\theta\left(x_{test}^{(i)}\right)\right)$$

     - Más general:
       - Error 0/1 = porcentaje de fallos en test

Índice

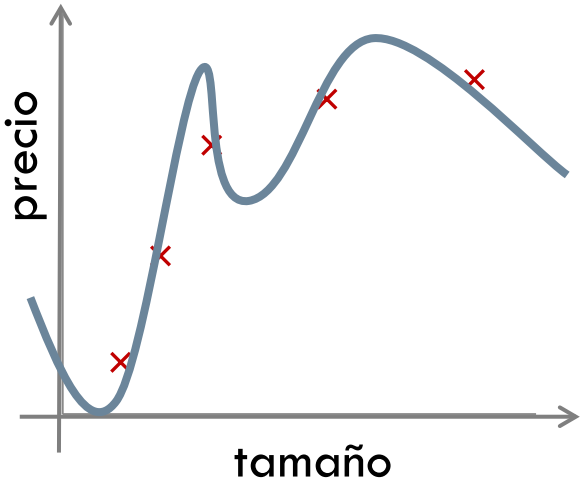
Selección de modelos

- ¿Qué grado de polinomio debemos utilizar?
- ¿Qué valor del parámetro de regularización debemos utilizar?

  - **Selección de modelos**

Selección de modelos

- Problema de sobre-aprendizaje



$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

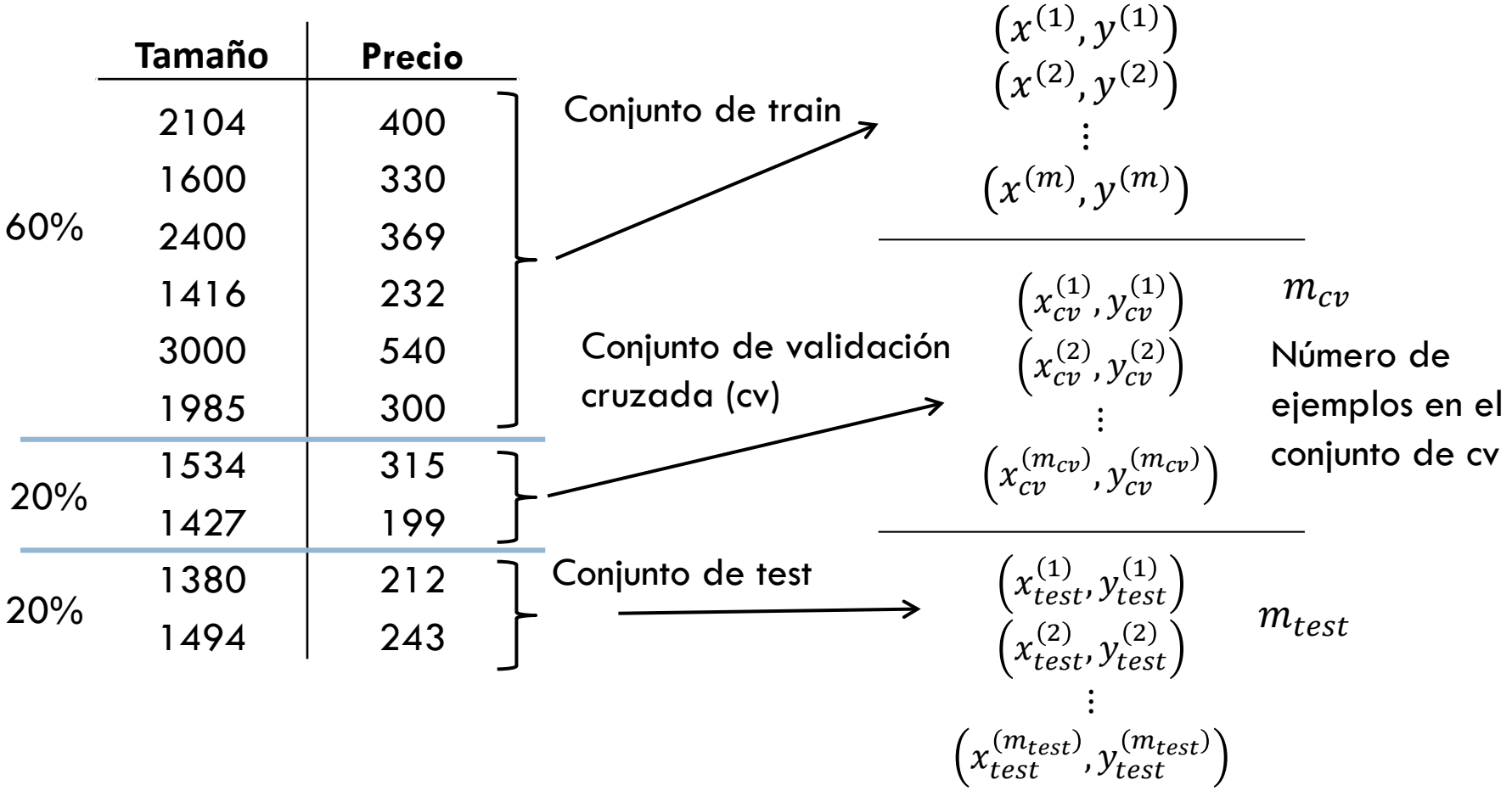
- Parámetros ajustados al conjunto de train
  - $J(\theta) < J_{test}(\theta)$
  - $J(\theta)$ resulta una estimación **OPTIMISTA**
    - Ocurre lo mismo con cualquier conjunto de datos utilizado en el ajuste de parámetros

Selección de modelos

- Selección del grado del polinomio
  - d = 1 $h_\theta(x) = \theta_0 + \theta_1 x$ Calcular $J_{test}(\theta)$
  - d = 2 $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ Calcular $J_{test}(\theta)$
  - d = 3 $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$ Calcular $J_{test}(\theta)$
  - $\vdots$
  - d = 10 $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$ Calcular $J_{test}(\theta)$
- Seleccionar el grado que obtiene el menor $J_{test}(\theta)$
- ¿Cómo generaliza este modelo?
  - Reportamos $J_{test}(\theta)$
  - **¡NO SIRVE!** Es un estimador optimista del error de generalización
    - Ha sido utilizado para ajustar el parámetro d a los datos de test

Selección de modelos

- Particiones de train / cv / test



Selección de modelos

- Error en train / validación / test
  - Error en train

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 = J(\theta)$$

  - Error en validación

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)}\right)^2$$

  - Error en test

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}\left(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)}\right)^2$$

Selección de modelos

- Selección del grado del polinomio
  - d = 1 $h_\theta(x) = \theta_0 + \theta_1 x$ Calcular $J_{cv}(\theta)$
  - d = 2 $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ Calcular $J_{cv}(\theta)$
  - d = 3 $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$ Calcular $J_{cv}(\theta)$
  - $\vdots$
  - d = 10 $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$ Calcular $J_{cv}(\theta)$
- Seleccionar el grado que obtiene el menor $J_{cv}(\theta)$
- ¿Cómo generaliza este modelo?
  - Reportamos $J_{test}(\theta)$
  - Podemos utilizarlo como estimador porque no hemos ajustado los parámetros con el conjunto de test

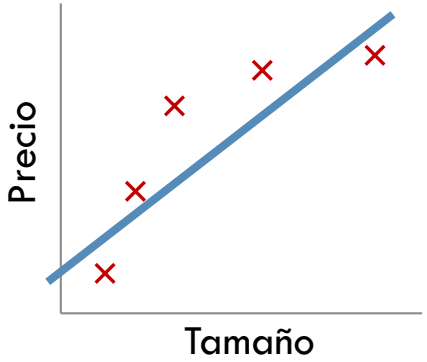
Selección de modelos

- El mismo proceso es válido para seleccionar λ
  1. Probar diferentes valores de λ
  2. Elegir aquel que obtenga el menor $J_{cv}(\theta)$
  3. Reportar $J_{test}(\theta)$ como error de generalización

Índice

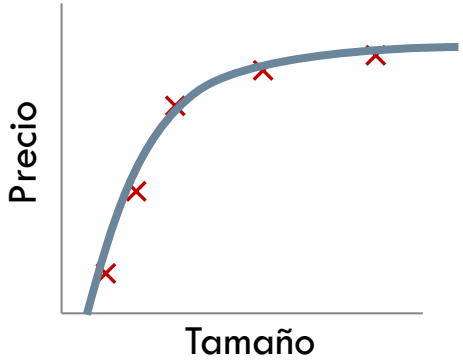
Bias / varianza

- Si un modelo no generaliza bien
  - Puede ser debido a:
    - Bias
    - Varianza



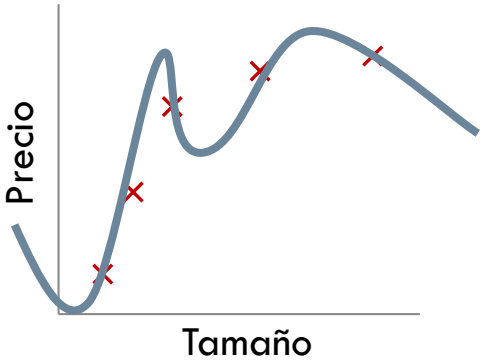
$\theta_0 + \theta_1 x$

Bias alto
(no se ajusta)



$\theta_0 + \theta_1 x + \theta_2 x^2$

Buen ajuste

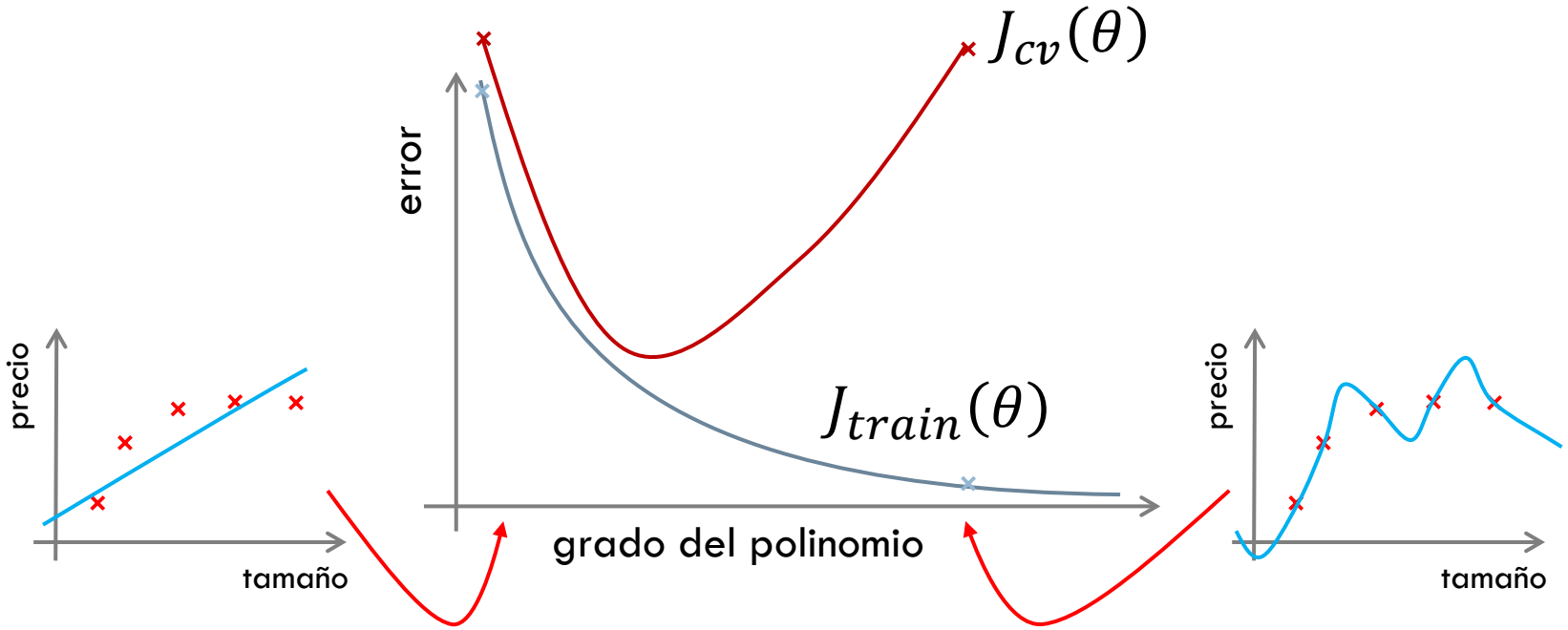


$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Varianza alta
(sobre-aprendizaje)

Bias / varianza

- Error en train: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$
- Error en validación: $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)}\right)^2$

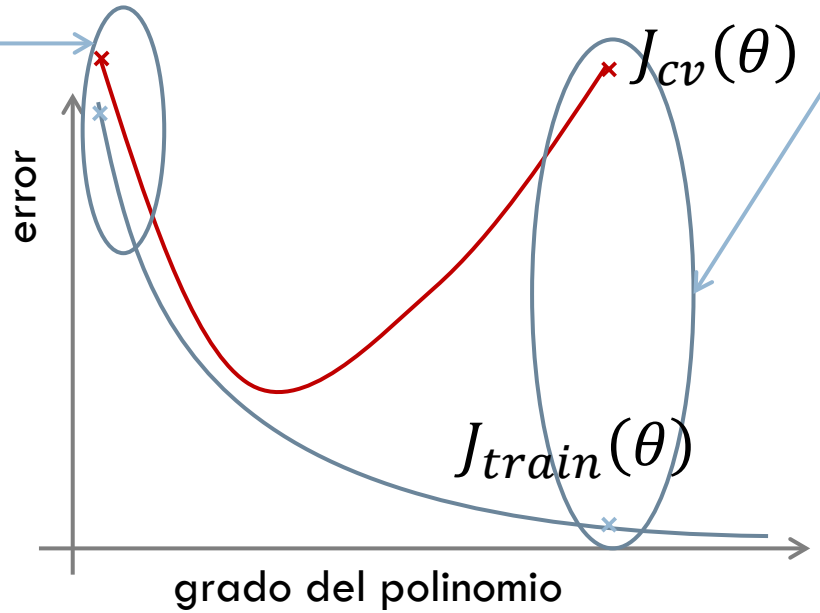


Bias / varianza

- ¿Por qué con valores extremos del grado no generaliza bien?

Bias
(no se ajusta bien a los datos)

$J_{train}(\theta)$ es alto
$J_{cv}(\theta) \approx J_{train}(\theta)$



Varianza
(sobre-aprendizaje)

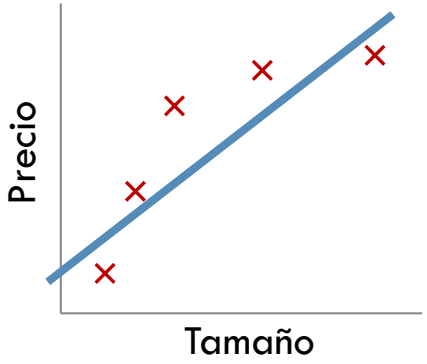
$J_{train}(\theta)$ es bajo
$J_{cv}(\theta) \gg J_{train}(\theta)$

Bias / varianza

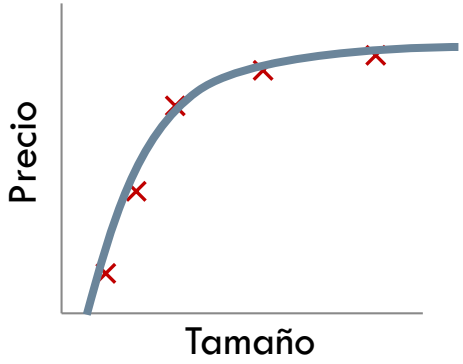
- ¿Qué ocurre si jugamos con λ?

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

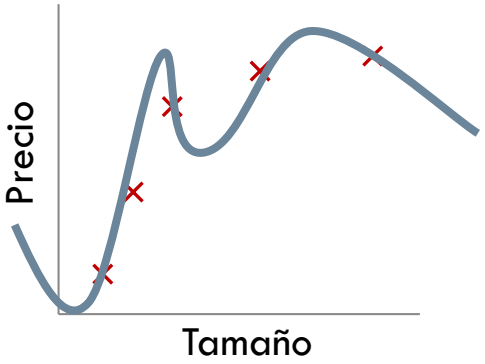
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$



λ alto
Bias alto
(no se ajusta)



λ intermedio
Buen ajuste



λ pequeño
Varianza alta
(sobre-aprendizaje)

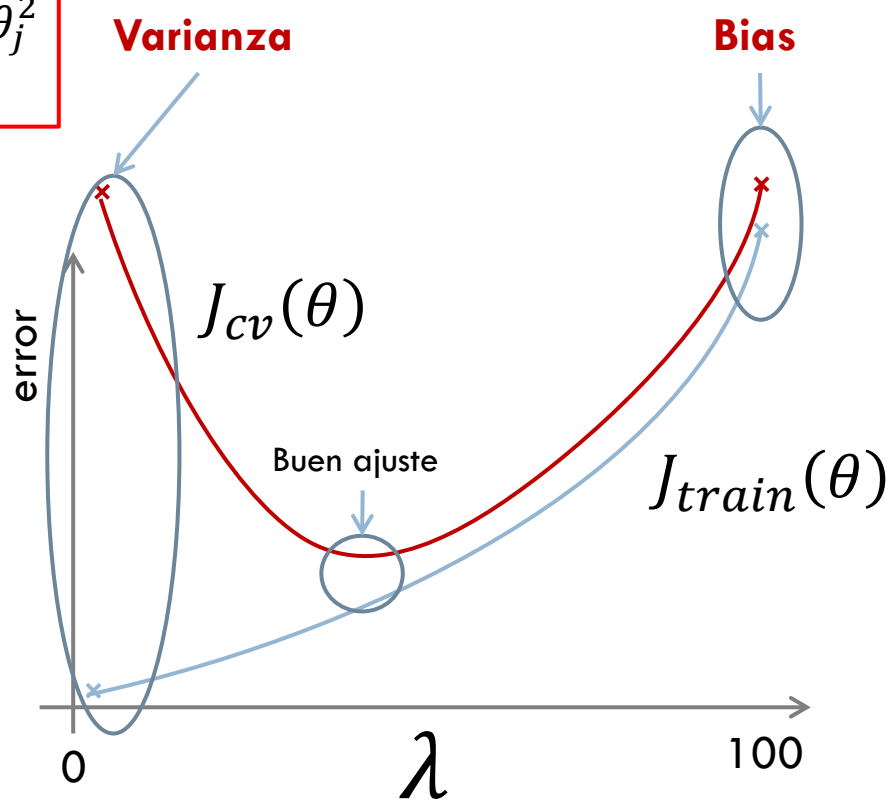
Bias / varianza

- Bias / varianza en función de λ

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \boxed{\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2}$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)}\right)^2$$

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}\left(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)}\right)^2$$

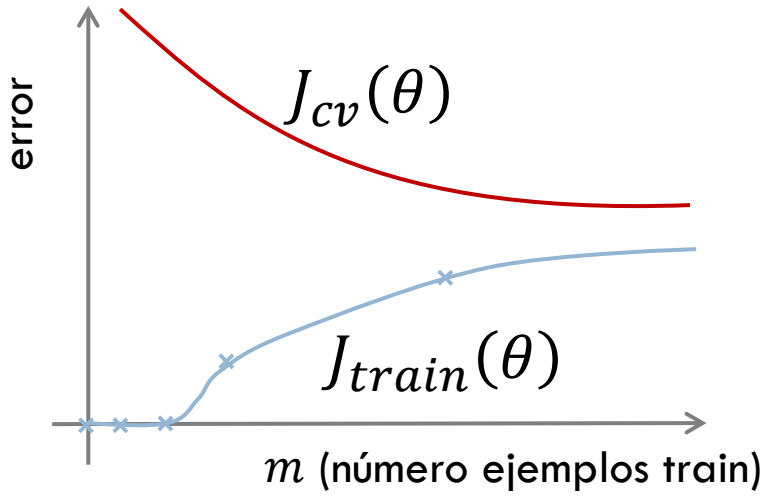


Bias / varianza

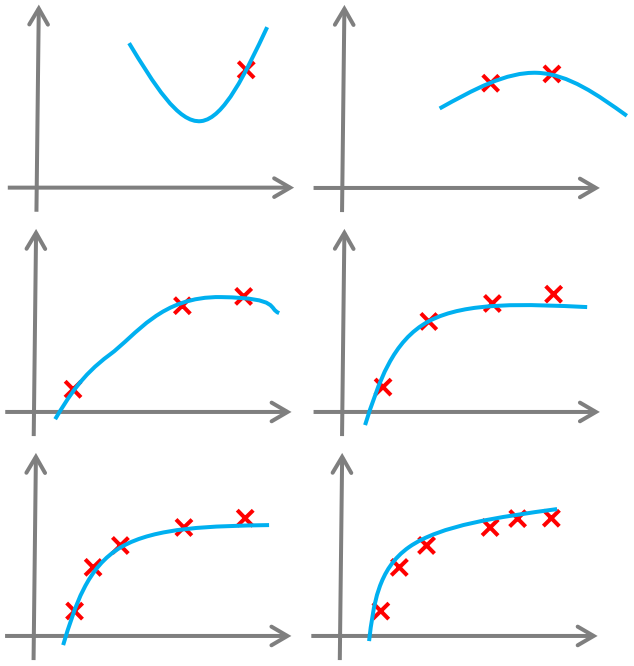
- Curvas de aprendizaje:
  - Error vs. Número ejemplos (p.e.)

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)}\right)^2$$

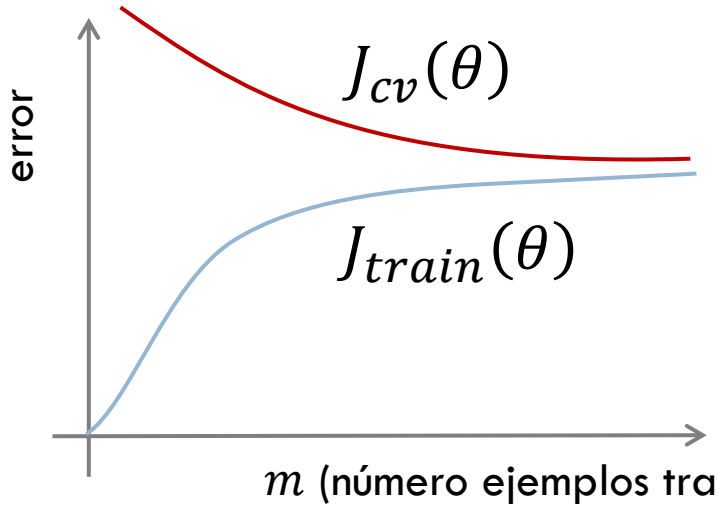


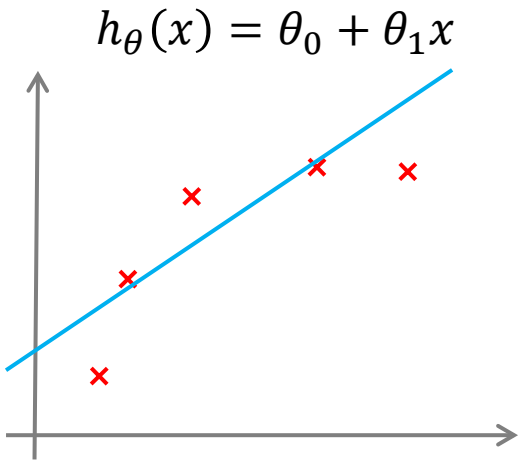
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

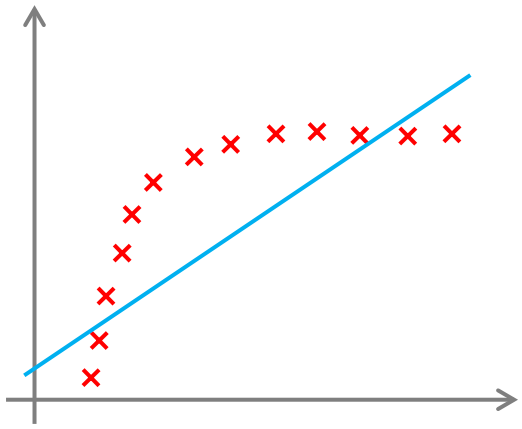


Bias / varianza

- Bias alto



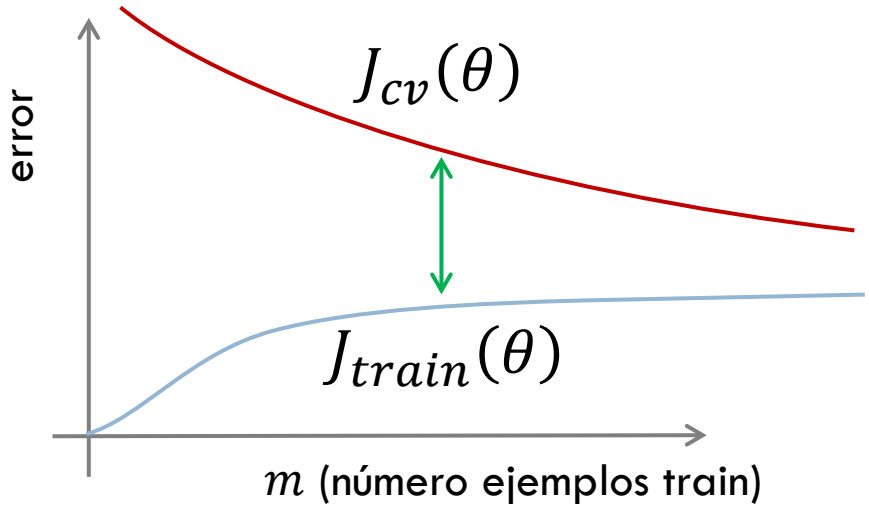




- Más ejemplos no mejoran el ajuste
  - Por sí mismos

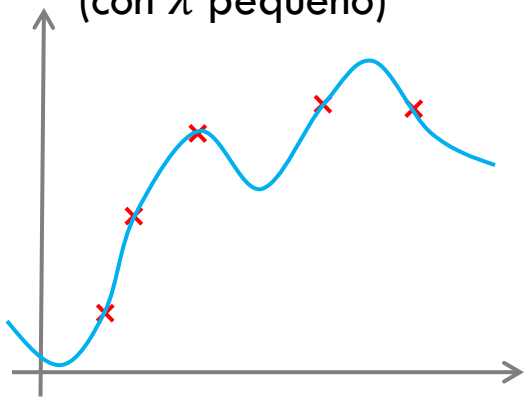
Bias / varianza

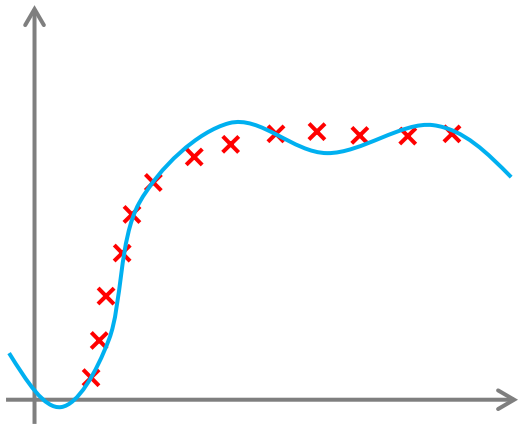
- Varianza alta



$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{100}$

(con λ pequeño)





- Más ejemplos
  - Probablemente ayudarán

Analizando un algoritmo de aprendizaje

- Supongamos que tenemos un modelo de regresión lineal con regularización para la **valoración de bienes inmuebles**

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

- Sin embargo, al aplicarlo sobre nuevas casas produce **errores inaceptables**, ¿qué podemos hacer?
  - Añadir más ejemplos de entrenamiento → mejora varianza alta
  - Utilizar menos características → mejora varianza alta
  - Añadir nuevas características → mejora bias alto
  - Añadir características polinomiales (x_1^2, x_2^2, x_1x_2, etc.) → mejora bias alto
  - Aumentar λ → mejora varianza alta
  - Decrecer λ → mejora bias alto

Índice

1. Análisis y evaluación de algoritmos de aprendizaje
   - Conjuntos train / test
2. Selección de modelos
   - Conjuntos de train / val / test
3. Bias / varianza
4. Análisis del error

Análisis del error

- Supongamos que disponemos de un detector de spam
  - Pero necesitamos reducir su error
  - ¿Cómo lo hacemos?
    - Recolectar más datos de entrenamiento
    - Desarrollar nuevas características
      - Routing
      - Cuerpo del mensaje
      - Usos de puntuación
      - Detección de fallos de escritura
    - …

Análisis del error

- **Estudiar los datos**
  - Crear un modelo rápido y simple
    - Testearlo en los datos de validación
  - Pintar las gráficas de aprendizaje
    - ¿Algo que pueda ayudar?
  - **Análisis del error**
    - Estudiar manualmente los ejemplos fallados
      - ¿Falla sistemáticamente algún tipo de ejemplos?
      - ¿Mejorará tratando esos ejemplos?
    - Inviable para muchos ejemplos/posibilidades
    - Hay que probar la solución y **evaluarla** → **MEDIDAS**

Análisis del error

- **Ratio de clasificación**
  - Porcentaje de **aciertos** (fallos)
  - Útil en muchos problemas
  - Permite establecer si ha habido mejoras al añadir una nueva características o nuevos ejemplos
  - Pero…
    - **No tiene en cuenta el número de ejemplos de cada clase**

Análisis del error

- **Clasificación de cáncer**
  - **1% de error** en test (**99% de aciertos**)
    - ¡Solo el 0.5% de los pacientes tienen cáncer!
    - Prediciendo siempre "no cáncer"
      - **99.5% de acierto**
  - En este marco
    - **El ratio de clasificación no refleja la calidad del clasificador**
    - Se hacen necesarias otras medidas de evaluación

Análisis del error

Precisión / Recall

- Clase **positiva** o **minoritaria** $y = 1$
- Clase **negativa** o **mayoritaria** $y = 0$
- Matriz de confusión

		Predicción	
		Predicción positiva	**Predicción negativa**
Clase real	**Clase positiva**	True Positive (TP)	False Negative (FN) → **Recall**
	Clase negativa	False Positive (FP)	True Negative (TN)

- **Precisión** (de todos los predichos como $y = 1$, ¿qué porcentaje tienen cáncer?)

Precisión

$$\frac{\text{True Positives}}{\text{N° Predichos positivos}} = \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

- **Recall** (de todos los que tienen cáncer, ¿qué porcentaje se ha detectado correctamente?)

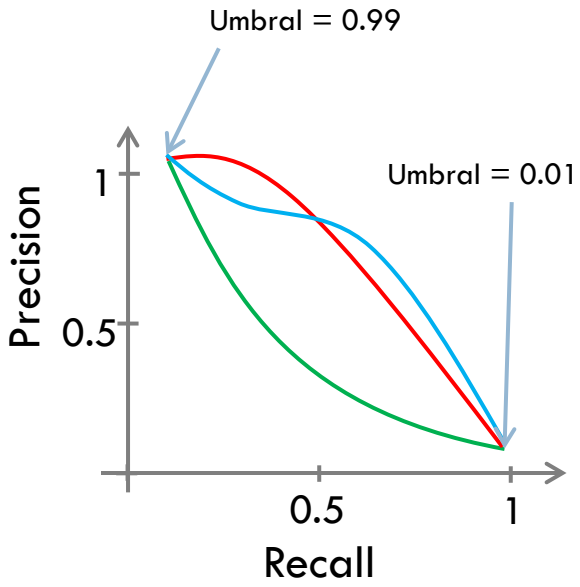
$$\frac{\text{True Positives}}{\text{N° Positivos reales}} = \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

Análisis del error

- **Balance entre precisión y recall**
- Regresión logística $0 \leq h_\theta(x) \leq 1$
  - Predecir 1 si $h_\theta(x) \geq 0.5$
  - Predecir 0 si $h_\theta(x) < 0.5$
- Cambiar 0.5 por otro umbral, predecir 1 si $h_\theta(x) \geq umbral$

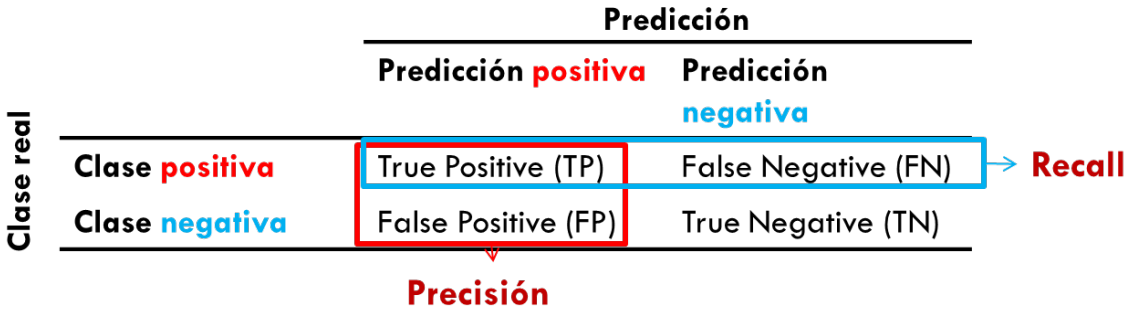
$$\text{precision} = \frac{\text{True Positives}}{\text{N° Predichos positivos}}$$

$$\text{recall} = \frac{\text{True Positives}}{\text{N° Positivos reales}}$$



Análisis del error

- **Precisión / Recall**



$$Precision = \frac{TP}{TP + FP} = \frac{True\ positives}{Nº\ predichos\ positivos}$$

$$Precision = \frac{TP}{TP + FP} = \frac{True\ positives}{Nº\ predichos\ positivos}$$

Análisis del error

- **Precisión / Recall**
  - Casos extremos

Predecimos todo como de la clase positiva (umbral muy bajo)

		Predicción	
		+ (y = 1)	- (y = 0)
Clase real	+ (y = 1)	3	0
	- (y = 0)	3	0

$$Precision = \frac{3}{3+3} = 0.5$$

$$Recall = \frac{3}{3+0} = 1$$

En este caso concreto y al ser un problema balanceado la precisión se mantiene alta.
Veamos qué ocurre en un caso con muchos más ejemplos de la clase negativa…

Análisis del error

- **Precisión / Recall**
  - Casos extremos

Predecimos todo como de la clase positiva (umbral muy bajo)

		Predicción	
		+ (y = 1)	**- (y = 0)**
Clase real	**+ (y = 1)**	10	0
	- (y = 0)	500	0

$$Precision = \frac{10}{10 + 500} = 0.0196$$

$$Recall = \frac{10}{10 + 0} = 1$$

La precisión se ve reducida a casi 0

Análisis del error

- **Precisión / Recall**
  - Casos extremos

Predecimos todo como de la clase negativa (umbral muy alto)

		Predicción	
		+ (y = 1)	- (y = 0)
Clase real	+ (y = 1)	0	3
	- (y = 0)	0	3

$$Precision = \frac{0}{0+0} = Indefinido$$

$$Recall = \frac{0}{0+3} =$$

La precisión no se puede calcular porque todo se predice como negativo
Veamos qué pasa con un caso un poco menos extremo…

Análisis del error

- **Precisión / Recall**
  - Casos extremos

Predecimos todo como de la clase negativa (umbral muy alto)

		Predicción	
		+ (y = 1)	- (y = 0)
Clase real	+ (y = 1)	1	2
	- (y = 0)	0	3

$$Precision = \frac{1}{1+0} = 1$$

$$Recall = \frac{1}{1+2} = 0.333$$

La predicción de los ejemplos positivos se obtiene con mucha fiabilidad y aunque pocos sean predichos como positivos, realmente lo son

Análisis del error

- **Precisión / Recall**
  - Casos extremos

Predecimos todo como de la clase negativa (umbral muy alto)

		Predicción	
		+ (y = 1)	- (y = 0)
Clase real	+ (y = 1)	1	9
	- (y = 0)	0	500

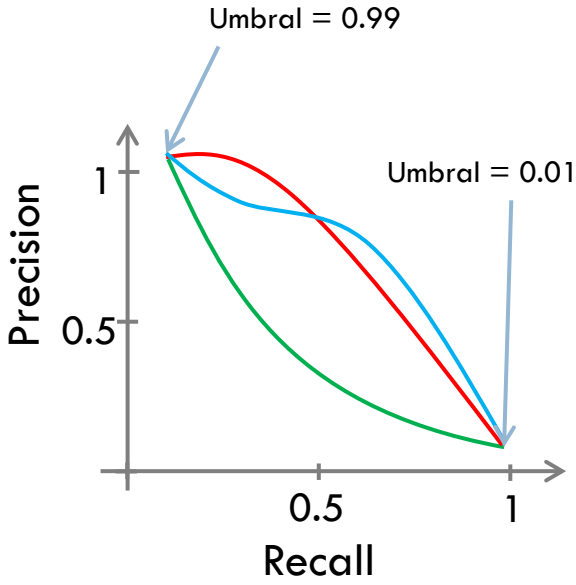
$$Precision = \frac{1}{1+0} = 1$$

$$Recall = \frac{1}{1+9} = 0.1$$

Con más ejemplos podemos observar mejor el balance entre precisión y recall (comparando con la otra tabla con el mismo caso)

Análisis del error

- **Balance entre precisión y recall**
  - Queremos predecir $y = 1$ (cáncer) con **mucha fiabilidad**
    - Queremos estar seguros de que los que predecimos como cáncer realmente lo tienen
  - **Umbral alto → ↑ precisión ↓recall**
    - Al aumentar el umbral, solo predecimos como positivos unos pocos casos → probablemente cáncer
      - Buena precisión (no tenemos falsos positivos)
      - A cambio perdemos en recall
        - Más falsos negativos
          - Tienen cáncer y decimos que no



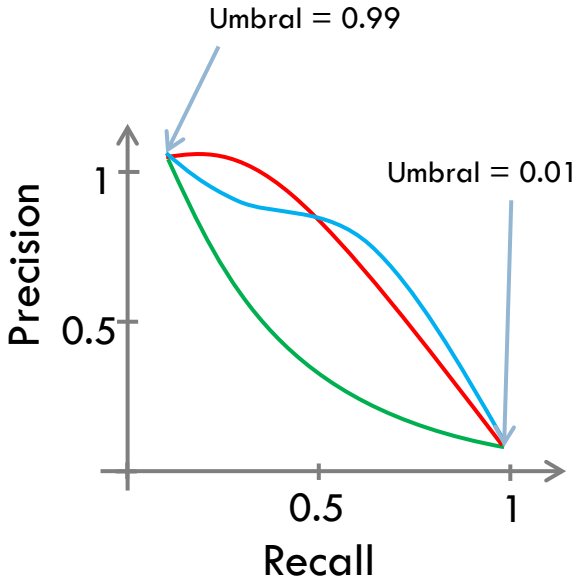
		Predicción	
		+ (y = 1)	- (y = 0)
Clase real	+ (y = 1)	1	9
	- (y = 0)	0	500

$$Precision = \frac{1}{1+0} = 1$$

$$Recall = \frac{1}{1+9} = 0.1$$

Análisis del error

- **Balance entre precisión y recall**
  - Queremos **evitar pasar por alto casos de cáncer**
    - Queremos predecir todos los que tienen cáncer aunque haya algunos que no lo tienen y les digamos que lo tienen
  - **Umbral bajo → ↑ recall ↓ precisión**
    - Conseguiremos predecir todos los positivos correctamente (recall)
      - A cambio nos llevaremos unos cuantos falsos positivos
        - Pérdida en precisión



		Predicción	
		+ (y = 1)	- (y = 0)
Clase real	+ (y = 1)	10	0
	- (y = 0)	500	0

$$Precision = \frac{10}{10 + 500} = 0.0196$$

$$Recall = \frac{10}{10 + 0} = 1$$

Análisis del error

- ¿Cómo comparamos la precisión (P) / recall (R) de dos algoritmos?
  - **F_1Score** $\mathbf{F_{score}} = 2\frac{P \cdot R}{P+R}$
  - Media **armónica** en vez aritmética
  - Mejor balance entre precisión y recall
    - Si P = 0 o R = 0 → F_1Score = 0
    - Si P = 1 y R = 1 → F_1Score = 1

	Precisin(P)	Recall (R)	Media	F_1 Score
Algoritmo 1	0.5	0.4	0.45	0.444
Algoritmo 2	0.7	0.1	0.4	0.175
Algoritmo 3	0.02	1.0	0.51	0.0392

¡¡¡¡Siempre predice $y = 1$!!!!