
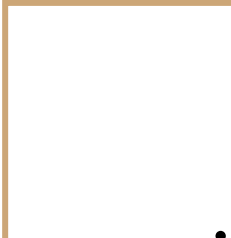


Evaluando Algoritmos de Aprendizaje




Outline

- En ML tenemos 3 cosas
 - Un problema
 - Herramientas
 - Herramientas de desempeño
 - Métricas y técnicas



¿Qué haces
después?



Problema

- Implementas un modelo para saber si es spam o no es spam
- El error en tu testing set es muy alto
- ¿Qué haces?

Problema

- Implementas un modelo para saber si es spam o no es spam
- El error en tu testing set es muy alto
- ¿Qué haces?
 - Conseguir más data (training samples)

Problema

- Implementas un modelo para saber si es spam o no es spam
- El error en tu testing set es muy alto
- ¿Qué haces?
 - Conseguir más data (training samples)
 - Probar menos features o más features

Problema

- Implementas un modelo para saber si es spam o no es spam
- El error en tu testing set es muy alto
- ¿Qué haces?
 - Conseguir más data (training samples)
 - Probar menos features o más features
 - Agregar features polinomiales

Problema

- Implementas un modelo para saber si es spam o no es spam
- El error en tu testing set es muy alto
- ¿Qué haces?
 - Conseguir más data (training samples)
 - Probar menos features o más features
 - Agregar features polinomiales
 - Cambiar lambda (regularización)

Diagnóstico de ML

- Prueba para tener intuición de qué funciona y qué no funciona.
- Guía para saber cómo mejorar desempeño
- **Toma tiempo**



¿Cómo evaluar
hipótesis?



¿Cómo evaluamos una hipótesis?

¿Cómo evaluamos una hipótesis?



- Hacer una gráfica
 - Problema: No se puede hacer cuando hay muchos features

Supervised Learning: Classification

training set

Observation #	Input image (X)	Label (Y)
1		"dog"
2		"cat"
3		"dog"
...
N		"dog"

test set

1		???
2		???

Evaluando Hipótesis

- Aprendemos los parámetros (θ) de la información de entrenamiento.
 - El objetivo aquí es minimizar el error de entrenamiento (J)
- Luego, calculamos el error del modelo ya entrenado con el test set J_{test}
 - El error puede ser el mismo (como MSE).
 - Para clasificación, podemos utilizar el error de misclasificación (0 o 1).

Selección de modelo

- ¿De qué grado elegimos el polinomio? (d)

$$d = 1 \quad h_{\theta} = \theta_0 + \theta_1 x \quad \rightarrow \theta^{(1)} \quad J_{\text{test}}(\theta^{(1)})$$

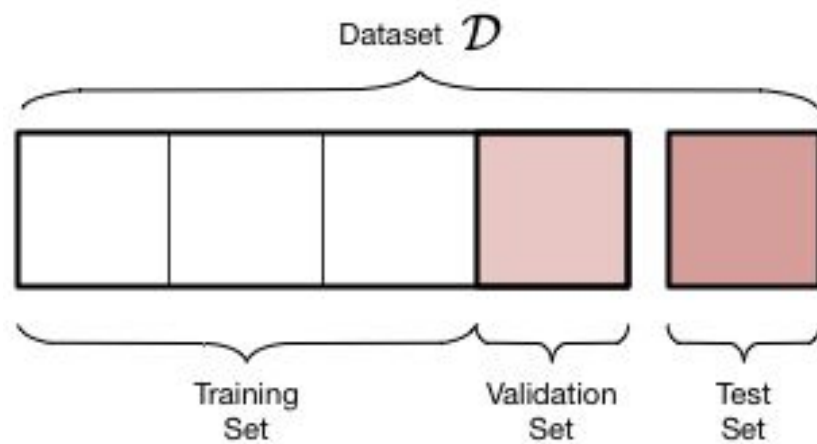
$$d = 2 \quad h_{\theta} = \theta_0 + \theta_1 x + \theta_2 x^2 \quad \rightarrow \theta^{(2)} \quad J_{\text{test}}(\theta^{(2)})$$

...

$$d = 10 \quad h_{\theta} = \theta_0 + \dots + \theta_{10} x^{10} \quad \rightarrow \theta^{(10)} \quad J_{\text{test}}(\theta^{(10)})$$

Selección de modelo

- Asumamos que elegimos el modelo con $d=5$
- $J_{\text{test}}(\theta^{(5)})$ es un estimado optimista para generalizar error
 - Elegimos d basándonos en el error de testing, por lo que no podemos generalizar para nueva información
 - θ no es predictivo para nuevos ejemplos
 - Fue un fit, al final, del test set
- **Nunca uses el test set para tomar decisiones**



Selección de modelo

- ¿De qué grado elegimos el polinomio? (d)

$$d = 1 \quad h_{\theta} = \theta_0 + \theta_1 x \quad \rightarrow \theta^{(1)} \quad J_{cv}(\theta^{(1)})$$

$$d = 2 \quad h_{\theta} = \theta_0 + \theta_1 x + \theta_2 x^2 \quad \rightarrow \theta^{(2)} \quad J_{cv}(\theta^{(2)})$$

...

$$d = 10 \quad h_{\theta} = \theta_0 + \dots + \theta_{10} x^{10} \quad \rightarrow \theta^{(10)} \quad J_{cv}(\theta^{(10)})$$

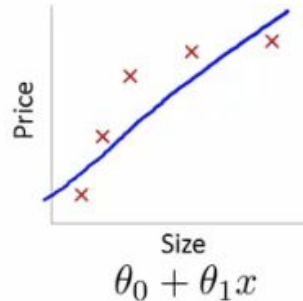


Bias y Variance

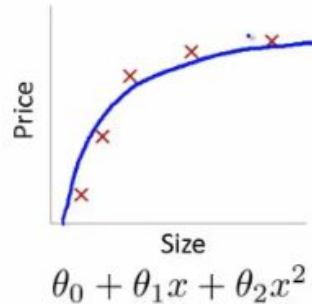


Overfitting y Underfitting

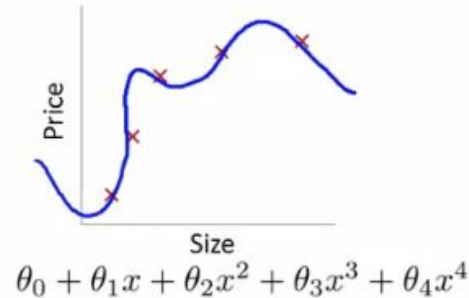
- Bias: Error introducido por aproximar un fenómeno real con un modelo simple.
- Variance: Qué tanto cambia el error en el testing set cuando cambiamos la información de entrenamiento



High bias
(underfit)



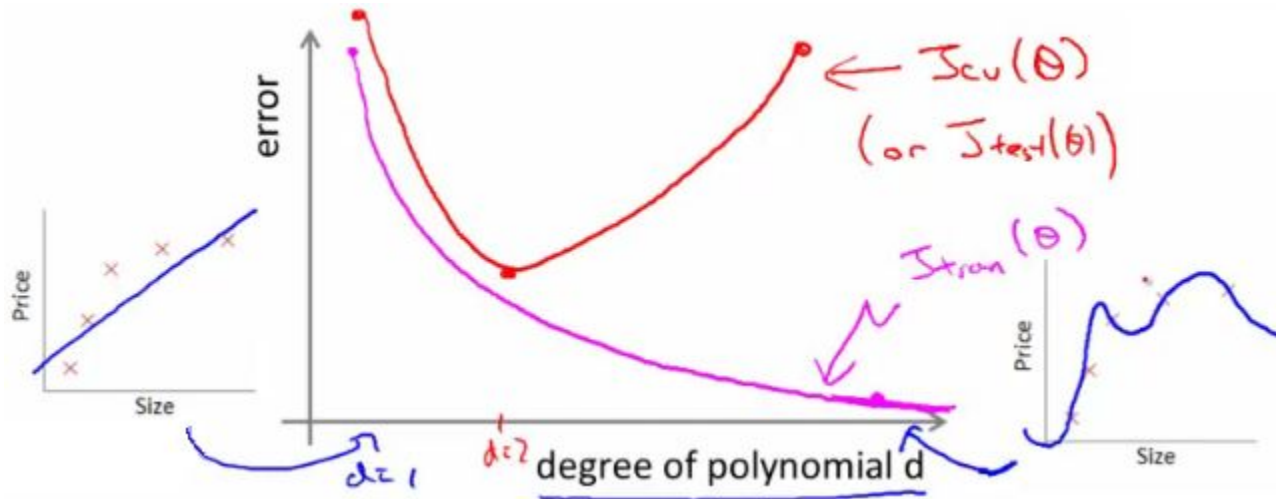
"Just right"



High variance
(overfit)

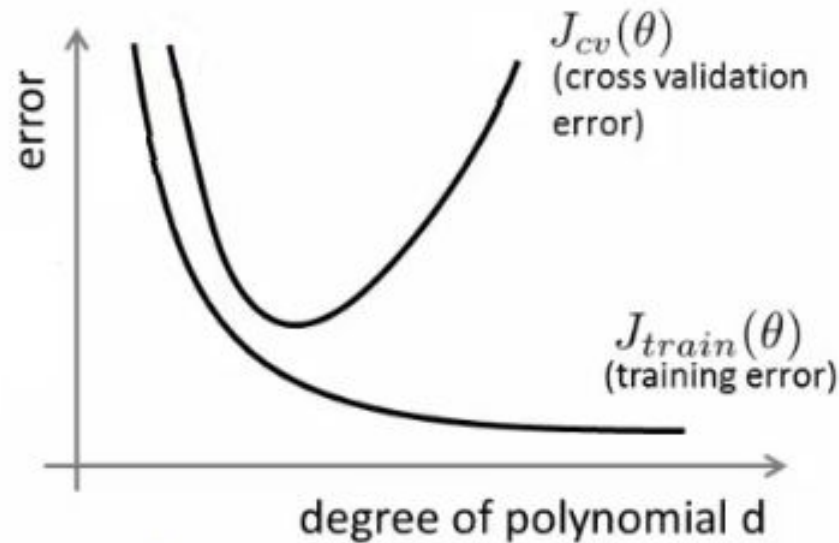
Bias y Variance para grado del polinomio

- Modelos complejos son propensos a hacer overfitting
- Error en testing y en CV son muy similares
- ¿Cuál d elegirían a partir de la gráfica?



Bias y Variance para grado del polinomio

Si J_{cv} es alto significa o que estamos en la región de bias o en la de variance



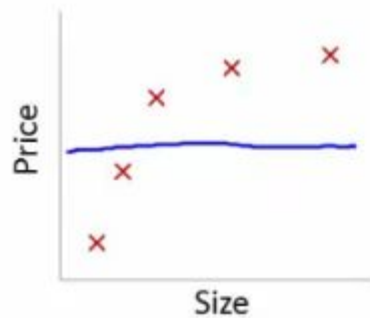
Bias y Variance para Regularización

Linear regression with regularization

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

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

- ¿Qué pasa si el factor de regularización es muy grande?
- ¿Qué pasa si el factor de regularización es muy pequeño?



Large λ

High bias (underfit)

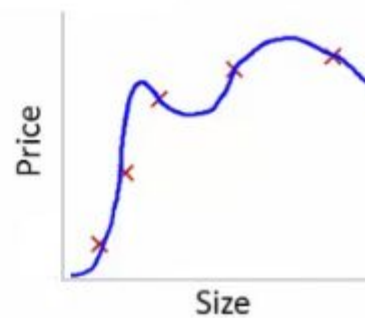
$\lambda = 10000$. $\theta_1 \approx 0, \theta_2 \approx 0, \dots$

$h_{\theta}(x) \approx \theta_0$



Intermediate λ

"Just right"



Small λ

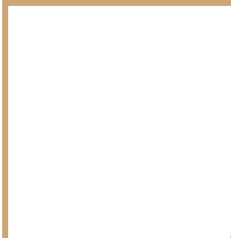
High variance (overfit)

$\lambda \approx 0$


Eligiendo Lambda

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

- Ten un conjunto de valores (normalmente se van duplicando)
 - 0, 0.01, 0.02, 0.04, 0.08, ..., 10.24
- Itera sobre los modelos y calcula el J_{cv} sin regularización.
- Elige el modelo que minimice ese error.
- Aplica J_{test} para medir desempeño
- ¿Cómo se grafica?

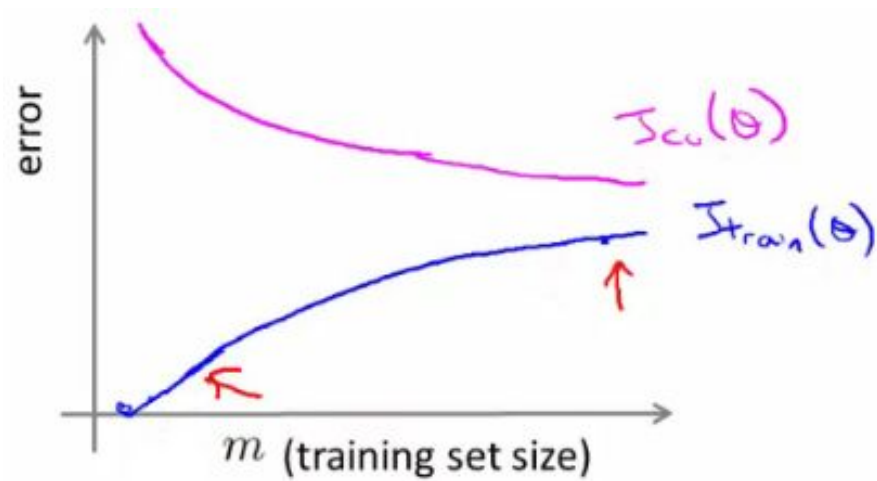


Curvas de Aprendizaje



Curvas de Aprendizaje

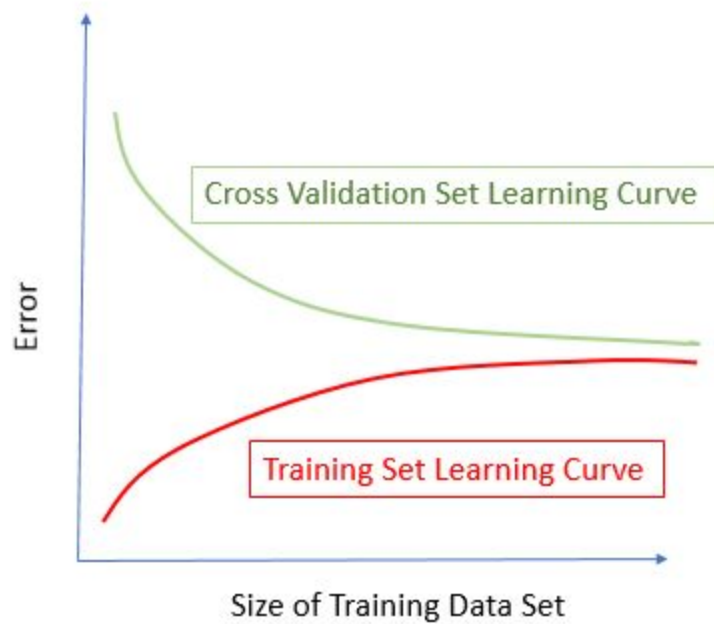
- Es una gráfica que nos permite entender qué está pasando y mejorar el desempeño.
- Se grafica el error en entrenamiento y en testing vs el número de ejemplos de entrenamiento (m).
- Aquí utilizamos m como una constante.
 - Iniciamos con m baja (10).
 - Entrenamos con 10 y calculamos el error.
 - Incrementamos m



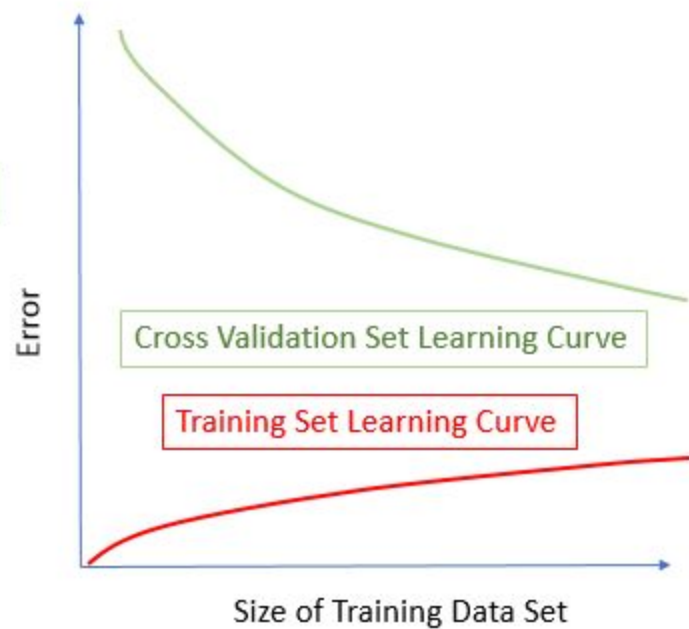
Curvas de Aprendizaje

- Es una gráfica que nos permite entender qué está pasando y mejorar el desempeño.
- Se grafica el error en entrenamiento y en testing vs el número de ejemplos de entrenamiento (m).
- Aquí utilizamos m como una constante.
 - Iniciamos con m baja (10).
 - Entrenamos con 10 y calculamos el error.
 - Incrementamos m
- ¿Cómo se ve nuestro modelo si tiene bias o variance?

High Bias



High Variance



Conclusiones

- Para el bias
 - El error tanto en entrenamiento como en validation es alto
 - Agregar más datos no va a funcionar porque el modelo es malo
 - Si pasa esto, significa que necesitas cambiar tu modelo
- Para el variance
 - Hay gran diferencia en el error de training y validation.

¿Qué hacer después?

- Conseguir más ejemplos -> arregla si hay variance
- Menos features -> arregla si hay variance
- Más features -> arregla si hay bias
 - (hace hipótesis más específica)
- Agrega features polinomiales -> arregla si hay bias
- Reducir regularización -> arregla bias
- Incrementar regularización -> arregla variance



Construyendo un Clasificador de Spam



Construyendo un Clasificador de Spam

- Objetivo: Crear un clasificador de spam utilizando aprendizaje supervisado
 - x = features
 - y = spam (1) o no spam (0)
- Features x
 - Elige 100 palabras que sean indicativas de si un correo es o no es spam.
 - Normalmente, elegiríamos las palabras más frecuentes (10k a 50k)

$$x = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \\ \vdots \end{bmatrix} \begin{matrix} \text{andrew} \\ \text{buy} \\ \text{deal} \\ \text{discount} \\ \vdots \\ \text{now} \\ \vdots \end{matrix}$$

¿Cómo dedicarías tu tiempo?

- ¿Qué prioridades tendrías?

¿Cómo dedicarías tu tiempo?

- Colecciona muchos datos
 - Proyecto honeypot para juntar correos falsos
 - Ejemplo: Campos invisibles
 - Ya vimos que no siempre funciona
- Desarrollar features sofisticados
 - Basados en información de enrutamiento de correo
 - Rutas extrañas del correo
- Desarrollar features que analicen el texto
 - ¿Discount es igual discounts?
 - ¿Nos importa la puntuación?
- Desarrollar algoritmos para detectar errores ortográficos

Sugerencia

- Muchas veces nos enfocamos en sólo una de esas opciones.
 - Planea bien el camino
- Comienza con un simple algoritmo que puedas programar rápido
 - Un día como mucho de crear tu primera opción
 - Haz el testing
 - Grafica curvas de aprendizaje
 - A partir de esto, determina qué necesitas
 - Más datos no sirven siempre
 - Previene optimizar prematuramente
- Usa evidencia para seguir el camino

Análisis de error

- Tienes 500 correos
- Algoritmo clasifica mal 100 de los correos
- Examina manualmente los errores y categorízalos
 - ¿Qué tipo de correos fallaron? (relacionados con drogas, contraseñas, ...)
 - ¿Qué features pudieron haber ayudado?
 - Mala puntuación
- Busca patrones sistemáticos. Esto permite elegir nuevos features.
- **Ten una manera de medir numéricamente**
 - Utiliza **un** solo valor
 - Puede ser complicado, pero permite entender el impacto de cambios en tu algoritmo.



Skewed Data Datos Sesgados



Ejemplo

- Tenemos un clasificador de cáncer con 1% de error.

Ejemplo

- Tenemos un clasificador de cáncer con 1% de error.
 - Sólo 0.5% de los pacientes tienen cáncer. (dato sesgado)
 - Modelo que siempre predice 0 va a tener 0.5% de error.
- ¿Proponer un modelo que vaya de 99.2% a 99.5 una mejora?

Precision & Recall

- **Precisión:** De los pacientes que fueron predecidos con cáncer, ¿cuántos tenían en verdad?

$$TP/(TP + FP)$$

- **Recall:** De los pacientes con cáncer, cuántos fueron clasificados correctamente

$$TP/(TP + FN)$$

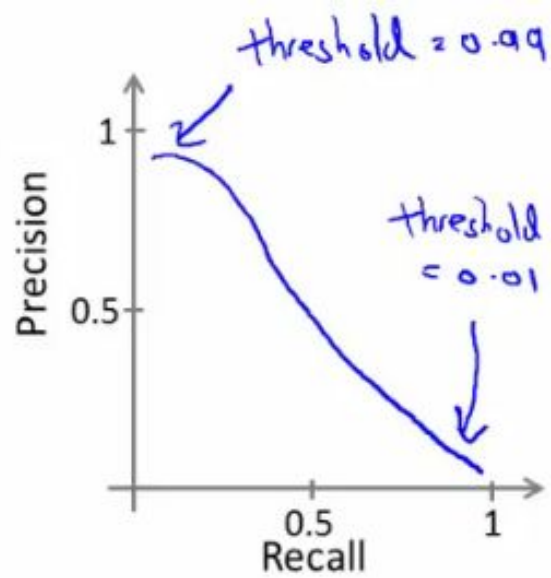
predicted

real

	1	0
1	TP	FP
0	FN	TN

Trade Off entre Precision y Recall

- Regresión logística $0 < h_{\theta}(x) < 1$
 - Predecimos 1 si $h_{\theta}(x) > 0.5$
 - Predecimos 0 si $h_{\theta}(x) < 0.5$
- Si cambiamos 0.5 a 0.7, aumentamos la confianza en el modelo
 - Significa que no vamos a diagnosticar cáncer apenas que estemos muy seguros que el paciente lo tenga.
 - Aumenta la precision
- Si cambiamos 0.5 a 0.3, aumentamos el recall
 - Nos dice que hay más gente que tiene cáncer.



FN o FP

- ¿Qué es peor para medicina?
- ¿Qué es peor para spam?

F_1 score

- Combina precision y recall
- ¿Cómo?

F₁ score

- Combina precision y recall
- ¿Cómo?
 - Usa promedio armónico.
 - Sigue siendo un promedio, pero da más importancia a los valores pequeños.

$$(10.1) \text{ Accuracy} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

$$(10.2) \text{ Precision} = \frac{T_p}{T_p + F_p}$$

$$(10.3) \text{ Recall} = \frac{T_p}{T_p + T_n}$$

$$(10.4) F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

F beta score

- Si es grande, prefiere recall
- Si es pequeño, prefiere precisión

F-Beta Score

$$F_{\beta} = (1 + \beta^2) \frac{\text{Precision} * \text{Recall}}{(\beta^2 * \text{Precision}) + \text{Recall}}$$

$$F_1 = 2 \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Choice of β depends on the specific problem.

Métricas

- Hay muchas
- Para regresión
 - Mean Absolute Error
 - Mean squared Error
- R2 Score
- ROC - Receiver Operating Characteristic

¿Cómo manejamos muchos datos?

- Asumimos que x tiene suficiente información para predecir.
- Si damos el input a un experto humano, ¿este podrá predecir y ?
- Como son muchos datos, estamos seguros de overfitting

Grid Search

- Ahora que empezemos a programar, hay parámetros que tienen muchos valores
- Por ejemplo, para Support Vector Machine
 - C
 - kernel
 - degree
 - Gamma
 - ...
- A veces configurar todos es muy complicado
- Podemos hacer una tabla con todos esos parámetros