

# Regresión

**Matias Vera - Juan Zuloaga**

Centro de Simulación Computacional para Aplicaciones Tecnológicas

# Agenda

- 1 Introducción al problema de regresión
- 2 Regresión Lineal
- 3 Gradiente Descendente

# Teoría de Regresión

## Bases

**Objetivo:** Predecir el valor de  $Y$  a partir del valor de  $X \rightarrow \hat{Y} = \varphi(X)$

**Función costo:** Error cuadrático  $\rightarrow \ell(x, y) = (y - \varphi(x))^2$

**Riesgo Esperado:** MSE  $\rightarrow \mathbb{E}[\ell(X, Y)] = \mathbb{E}[(Y - \varphi(X))^2]$

# Teoría de Regresión

## Bases

**Objetivo:** Predecir el valor de  $Y$  a partir del valor de  $X \rightarrow \hat{Y} = \varphi(X)$

**Función costo:** Error cuadrático  $\rightarrow \ell(x, y) = (y - \varphi(x))^2$

**Riesgo Esperado:** MSE  $\rightarrow \mathbb{E}[\ell(X, Y)] = \mathbb{E}[(Y - \varphi(X))^2]$

## Optimalidad

$$\mathbb{E}[(Y - \varphi(X))^2] \geq \mathbb{E}[\text{var}(Y|X)]$$

con igualdad si y solo si  $\varphi(x) = \mathbb{E}[Y|X = x]$ .

**Regresor óptimo:**  $\varphi(x) = \mathbb{E}[Y|X = x]$

**Error Bayesiano:**  $\mathbb{E}[\text{var}(Y|X)]$

# Reconocimiento de patrones

## Objetivo

Quiero buscar  $\varphi(\cdot)$  que minimice  $\mathbb{E}[\ell(X, Y)]$ . Es decir aprender la “esperanza condicional”.

# Reconocimiento de patrones

## Objetivo

Quiero buscar  $\varphi(\cdot)$  que minimice  $\mathbb{E}[\ell(X, Y)]$ . Es decir aprender la “esperanza condicional”.

## Empirical Risk Minimization (ERM)

Propongo buscar  $\varphi(\cdot)$  que minimice el riesgo empírico:  $\frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i)$

# Reconocimiento de patrones

## Objetivo

Quiero buscar  $\varphi(\cdot)$  que minimice  $\mathbb{E}[\ell(X, Y)]$ . Es decir aprender la “esperanza condicional”.

## Empirical Risk Minimization (ERM)

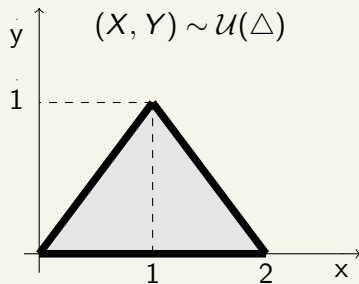
Propongo buscar  $\varphi(\cdot)$  que minimice el riesgo empírico:  $\frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i)$

## Tradeoff: Sesgo/Varianza

$$\underbrace{\mathbb{E}[\ell(X, Y)]}_{\text{Riesgo esperado}} = \underbrace{\frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i)}_{\text{Riesgo empírico}} + \underbrace{\left( \mathbb{E}[\ell(X, Y)] - \frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i) \right)}_{\text{Gap de generalización}}$$

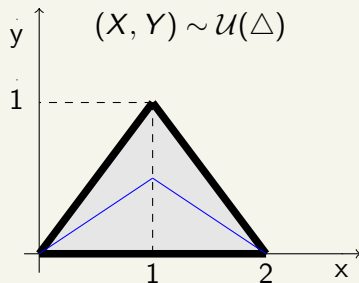
**Nota:** El riesgo empírico se considera grande o pequeño comparándolo con el error bayesiano.

# Overfitting y Underfitting





# Overfitting y Underfitting

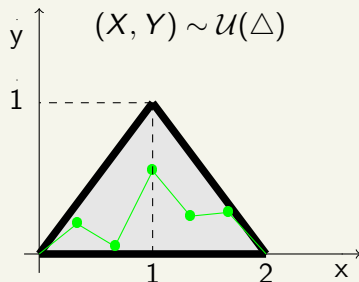


## Solución Óptima

- El regresor elegido es efectivamente la esperanza condicional.
- El riesgo esperado alcanza el límite bayesiano

$$\mathbb{E}[\text{var}(Y|X)] = \frac{1}{24}$$

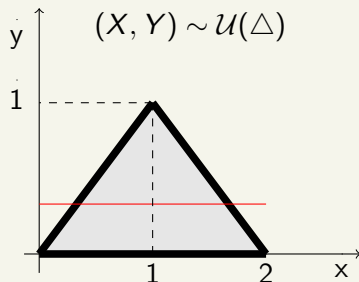
# Overfitting y Underfitting



## Problema de overfitting

- Riesgo empírico muy bajo (puede ser menor incluso que el bayesiano)
- Se detecta por el alto gap de generalización.
- Exceso de complejidad en el modelado.
- Se dice que el algoritmo tiene un problema de varianza.

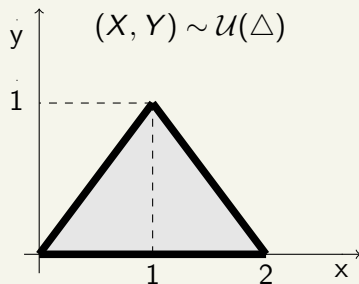
# Overfitting y Underfitting



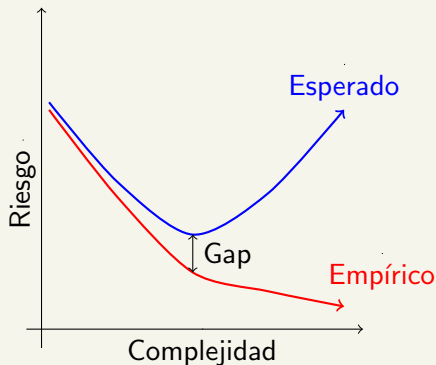
## Problema de underfitting

- Suele tener bajo gap de generalización.
- Riesgo empírico muy superior al error bayesiano.
- Escasez de complejidad en el modelado.
- Se dice que el algoritmo tiene un problema de sesgo.

# Overfitting y Underfitting



## Teoría Clásica de Generalización



# Tarea

Para

$$p_{XY}(x, y) = \frac{3}{4} \mathbb{1} \{0 < y < 1 + x^2, 0 < x < 1\}$$

- 1 Calcular y graficar en una misma figura el soporte, la esperanza condicional  $\mathbb{E}[Y|X = x]$  y la recta de regresión.
- 2 Calcular el error bayesiano.

# Conjuntos de datos

- Conjunto de entrenamiento (*train set*): Datos utilizados para minimizar el costo. Sobre estos se produce el “aprendizaje”. Las variables definidas a partir de este conjunto se llaman parámetros.
- Conjunto de validación (*validation or development set*): Datos utilizados para comparar modelos. Las variables definidas a partir de este conjunto se llaman hiperparámetros.
- Conjunto de testeo (*test set*): Datos utilizados para evaluar la performance final del algoritmo. Su única función es presentar estimadores insesgados de las métricas de error y no es imprescindible.

**Si la base de datos esta dividida, respetar la división!**

# Conjuntos de datos

- Conjunto de entrenamiento (*train set*): Datos utilizados para minimizar el costo. Sobre estos se produce el “aprendizaje”. Las variables definidas a partir de este conjunto se llaman parámetros.
- Conjunto de validación (*validation or development set*): Datos utilizados para comparar modelos. Las variables definidas a partir de este conjunto se llaman hiperparámetros.
- Conjunto de testeo (*test set*): Datos utilizados para evaluar la performance final del algoritmo. Su única función es presentar estimadores insesgados de las métricas de error y no es imprescindible.

**Si la base de datos esta dividida, respetar la división!**

**Enfoque clásico:** 60%/20%/20% - Típico para 100, 1K, 10K muestras.

# Conjuntos de datos

- Conjunto de entrenamiento (*train set*): Datos utilizados para minimizar el costo. Sobre estos se produce el “aprendizaje”. Las variables definidas a partir de este conjunto se llaman parámetros.
- Conjunto de validación (*validation or development set*): Datos utilizados para comparar modelos. Las variables definidas a partir de este conjunto se llaman hiperparámetros.
- Conjunto de testeo (*test set*): Datos utilizados para evaluar la performance final del algoritmo. Su única función es presentar estimadores insesgados de las métricas de error y no es imprescindible.

**Si la base de datos esta dividida, respetar la división!**

**Enfoque clásico:** 60%/20%/20% - Típico para 100, 1K, 10K muestras.

**Big Data:** Para 1M muestras, quizás alcanza con 98%/1%/1% (por eso cross-validation o k-flods no es tan frecuente acá).



# Regresión Lineal: $\hat{Y} = w^T \cdot X + b$

## Idea

Me aseguro mantener acotado el problema de overfitting proponiendo una solución de extremadamente baja complejidad. Si se alcanza bajo error empírico, entonces tengo ciertas garantías de que el algoritmo alcanza un buen desempeño.

# Regresión Lineal: $\hat{Y} = w^T \cdot X + b$

## Idea

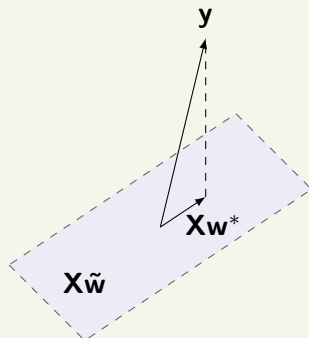
Me aseguro mantener acotado el problema de overfitting proponiendo una solución de extremadamente baja complejidad. Si se alcanza bajo error empírico, entonces tengo ciertas garantías de que el algoritmo alcanza un buen desempeño.

## Empirical Risk Minimization

$$\begin{aligned}(w, b) &\in \arg \min_{(w, b)} \sum_{i=1}^n (w^T \cdot X_i + b - Y_i)^2 \\ &= \arg \min_{\mathbf{w}} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2,\end{aligned}$$

$$\mathbf{X} = \begin{pmatrix} 1 & X_1^T \\ 1 & X_2^T \\ \vdots & \vdots \\ 1 & X_n^T \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} b \\ w \end{pmatrix}$$

# Regresión Lineal

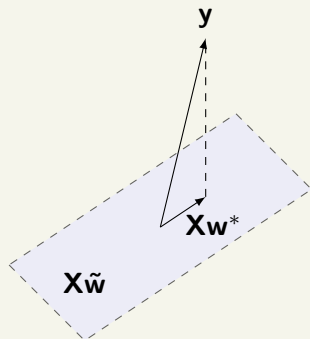


$$(\mathbf{X}\tilde{\mathbf{w}})^T (\mathbf{y} - \mathbf{X}\mathbf{w}^*) = 0 \quad \forall \tilde{\mathbf{w}}$$



$$\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

# Regresión Lineal



$$(X\tilde{w})^T (y - Xw^*) = 0 \quad \forall \tilde{w}$$



$$w^* = (X^T X)^{-1} X^T y$$

Solución matricial óptima: Recta de Regresión

$$w = \underbrace{(X^T X)^{-1}}_{\text{varianza}} \underbrace{X^T y}_{\text{covarianza}}$$

# Tarea

## Derivar respecto a una matriz

Utilizar propiedades sobre las derivadas matriciales para demostrar que  $\nabla_{\mathbf{w}} (\|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2) = \mathbf{0}$  tiene como única solución la pseudo inversa  $\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ .

## Propiedades

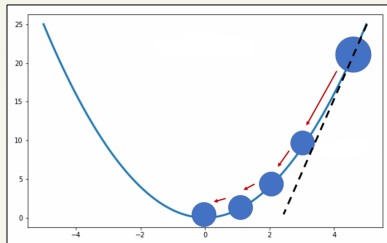
$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{a}) = \mathbf{a}, \quad \frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{B} \mathbf{x}) = (\mathbf{B} + \mathbf{B}^T) \mathbf{x}$$

# Gradiente Descendente

Problema a resolver:  $\min_{\theta \in \Theta} J(\theta)$ .

Solución:

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t)$$



---

*Cauchy 1847*: “Méthode générale pour la résolution de systèmes d’équations simultanées”.

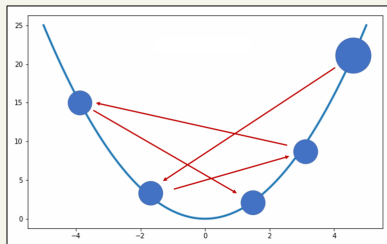
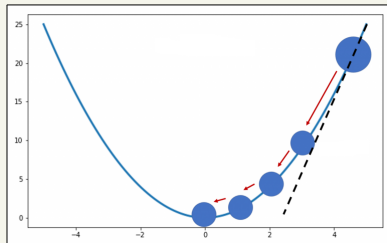
# Gradiente Descendente

Problema a resolver:  $\min_{\theta \in \Theta} J(\theta)$ .

Solución:

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t)$$

- Si  $\alpha$  es chico la convergencia es lenta.
- Si  $\alpha$  es grande puede no converger.



---

*Cauchy 1847*: “Méthode générale pour la résolution de systèmes d’équations simultanées”.

# Convergencia del modelo lineal

## Modelo

- $J(\mathbf{w}) = \frac{1}{n} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2.$
- $\nabla J(\mathbf{w}) = \frac{2}{n} \mathbf{X}^T \mathbf{X} \mathbf{w} - \frac{2}{n} \mathbf{X}^T \mathbf{y}.$
- $\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$
- $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla J(\mathbf{w}_t).$



# Convergencia del modelo lineal

## Modelo

- $J(\mathbf{w}) = \frac{1}{n} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2.$
- $\nabla J(\mathbf{w}) = \frac{2}{n} \mathbf{X}^T \mathbf{X} \mathbf{w} - \frac{2}{n} \mathbf{X}^T \mathbf{y}.$
- $\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$
- $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla J(\mathbf{w}_t).$

$$\begin{aligned} \mathbf{w}_{t+1} - \mathbf{w}^* &= \mathbf{w}_t - \mathbf{w}^* - \frac{2\alpha}{n} (\mathbf{X}^T \mathbf{X} \mathbf{w}_t - \mathbf{X}^T \mathbf{y}) \\ &= \left( I - \frac{2\alpha}{n} \mathbf{X}^T \mathbf{X} \right) (\mathbf{w}_t - \mathbf{w}^*) \end{aligned}$$

# Convergencia del modelo lineal

## Modelo

- $J(\mathbf{w}) = \frac{1}{n} \|\mathbf{X} \cdot \mathbf{w} - \mathbf{y}\|^2.$
- $\nabla J(\mathbf{w}) = \frac{2}{n} \mathbf{X}^T \mathbf{X} \mathbf{w} - \frac{2}{n} \mathbf{X}^T \mathbf{y}.$
- $\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$
- $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla J(\mathbf{w}_t).$

$$\begin{aligned} \mathbf{w}_{t+1} - \mathbf{w}^* &= \mathbf{w}_t - \mathbf{w}^* - \frac{2\alpha}{n} (\mathbf{X}^T \mathbf{X} \mathbf{w}_t - \mathbf{X}^T \mathbf{y}) \\ &= \left( I - \frac{2\alpha}{n} \mathbf{X}^T \mathbf{X} \right) (\mathbf{w}_t - \mathbf{w}^*) \end{aligned}$$

## Diagonalización ortogonal

Toda matriz real, cuadrada y simétrica puede escribirse como  $A = Q^T \Lambda Q$  con  $\Lambda$  diagonal y  $Q^T Q = Q Q^T = I$ .

## Convergencia del modelo lineal

Sean  $Q$  y  $\Lambda$  las matrices correspondientes a la diagonalización ortogonal de  $\mathbf{X}^T \mathbf{X}$ :

$$\begin{aligned}\mathbf{w}_{t+1} - \mathbf{w}^* &= \left( I - \frac{2\alpha}{n} Q^T \Lambda Q \right) (\mathbf{w}_t - \mathbf{w}^*) \\ &= Q^T \left( I - \frac{2\alpha}{n} \Lambda \right) Q (\mathbf{w}_t - \mathbf{w}^*)\end{aligned}$$

## Convergencia del modelo lineal

Sean  $Q$  y  $\Lambda$  las matrices correspondientes a la diagonalización ortogonal de  $\mathbf{X}^T \mathbf{X}$ :

$$\begin{aligned}\mathbf{w}_{t+1} - \mathbf{w}^* &= \left( I - \frac{2\alpha}{n} Q^T \Lambda Q \right) (\mathbf{w}_t - \mathbf{w}^*) \\ &= Q^T \left( I - \frac{2\alpha}{n} \Lambda \right) Q (\mathbf{w}_t - \mathbf{w}^*)\end{aligned}$$

Defino  $v_t = Q (\mathbf{w}_t - \mathbf{w}^*)$ , luego:

$$v_{t+1} = \left( I - \frac{2\alpha}{n} \Lambda \right) v_t,$$

## Convergencia del modelo lineal

Sean  $Q$  y  $\Lambda$  las matrices correspondientes a la diagonalización ortogonal de  $\mathbf{X}^T \mathbf{X}$ :

$$\begin{aligned}\mathbf{w}_{t+1} - \mathbf{w}^* &= \left( I - \frac{2\alpha}{n} Q^T \Lambda Q \right) (\mathbf{w}_t - \mathbf{w}^*) \\ &= Q^T \left( I - \frac{2\alpha}{n} \Lambda \right) Q (\mathbf{w}_t - \mathbf{w}^*)\end{aligned}$$

Defino  $v_t = Q (\mathbf{w}_t - \mathbf{w}^*)$ , luego:

$$v_{t+1} = \left( I - \frac{2\alpha}{n} \Lambda \right) v_t, \quad v_t = \left( I - \frac{2\alpha}{n} \Lambda \right)^t v_0$$

## Convergencia del modelo lineal

Sean  $Q$  y  $\Lambda$  las matrices correspondientes a la diagonalización ortogonal de  $\mathbf{X}^T \mathbf{X}$ :

$$\begin{aligned}\mathbf{w}_{t+1} - \mathbf{w}^* &= \left( I - \frac{2\alpha}{n} Q^T \Lambda Q \right) (\mathbf{w}_t - \mathbf{w}^*) \\ &= Q^T \left( I - \frac{2\alpha}{n} \Lambda \right) Q (\mathbf{w}_t - \mathbf{w}^*)\end{aligned}$$

Defino  $\mathbf{v}_t = Q (\mathbf{w}_t - \mathbf{w}^*)$ , luego:

$$\mathbf{v}_{t+1} = \left( I - \frac{2\alpha}{n} \Lambda \right) \mathbf{v}_t, \quad \mathbf{v}_t = \left( I - \frac{2\alpha}{n} \Lambda \right)^t \mathbf{v}_0$$

### Condición y velocidad de convergencia

El GD convergerá si  $|1 - \frac{2\alpha}{n} \lambda_j| < 1$  para todo  $j = \{1, \dots, d_x + 1\}$  y el learning rate óptimo estará asociado al criterio de peor caso:

$$\min_{\alpha} \max_j \left| 1 - \frac{2\alpha}{n} \lambda_j \right| \quad \text{s.t.} \quad \left| 1 - \frac{2\alpha}{n} \lambda_j \right| < 1 \quad \forall j$$

# Convergencia del modelo lineal

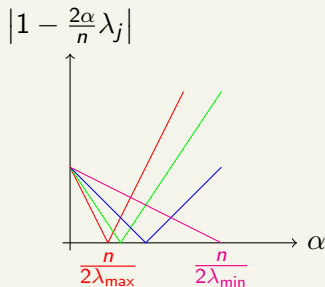
## Condición de convergencia

$|1 - \frac{2\alpha}{n} \lambda_j| < 1$  para todo  $j = \{1, \dots, d_x + 1\}$  equivale a pedir  $\alpha < \frac{n}{\lambda_{\max}}$ .

# Convergencia del modelo lineal

## Condición de convergencia

$|1 - \frac{2\alpha}{n} \lambda_j| < 1$  para todo  $j = \{1, \dots, d_x + 1\}$  equivale a pedir  $\alpha < \frac{n}{\lambda_{\max}}$ .



## Velocidad de convergencia

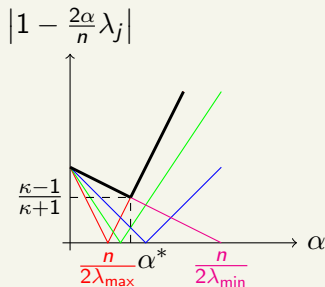
El óptimo learning rate en este caso es  $\alpha^* = \frac{n}{\lambda_{\max} + \lambda_{\min}}$  y su velocidad asociada  $\left(\frac{\kappa-1}{\kappa+1}\right)^t$  depende del número de condición  $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ .



# Convergencia del modelo lineal

## Condición de convergencia

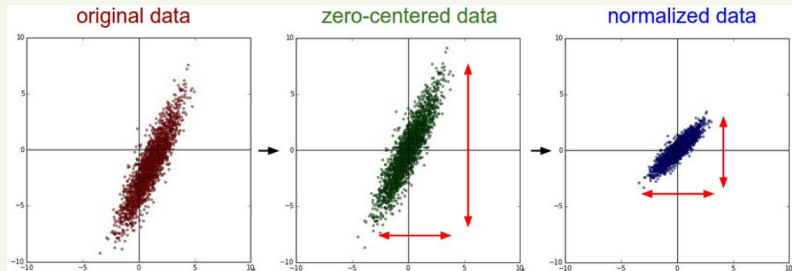
$|1 - \frac{2\alpha}{n} \lambda_j| < 1$  para todo  $j = \{1, \dots, d_x + 1\}$  equivale a pedir  $\alpha < \frac{n}{\lambda_{\max}}$ .



## Velocidad de convergencia

El óptimo learning rate en este caso es  $\alpha^* = \frac{n}{\lambda_{\max} + \lambda_{\min}}$  y su velocidad asociada  $\left(\frac{\kappa-1}{\kappa+1}\right)^t$  depende del número de condición  $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ .

# Normalización de la entrada



Normalizar *cada componente* de la entrada tiene sus beneficios:

$$(\mathbf{x})_k \leftarrow \frac{(\mathbf{x})_k - \mu_k}{\sigma_k}$$

donde las  $\mu_k$  y  $\sigma_k$  son calculadas previo al entrenamiento (para validación y testeo se usan las mismas) como:

$$\mu_k = \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} (\mathbf{x}_i)_k, \quad \sigma_k = \sqrt{\frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} [(\mathbf{x}_i)_k - \mu_k]^2}$$

# Normalización de la entrada

Me permite usar learning rates más grandes!

