# 09481: Inteligencia Artificial

Profesor del curso: Breyner Posso, Ing. M.Sc.
e-mail: breyner.posso1@u.icesi.edu.co

Programa de Ingeniería de Sistemas.
Departamento TIC.
Facultad de Ingeniería.
Universidad Icesi.
Cali, Colombia.

# Agenda

- Introducción

- Aprendizaje Supervisado.
  - Técnicas de regularización.
    - ¿Qué son? ¿para qué sirven? ¿cómo usarlas?
    - Regularización L2 (regresión Ridge).
    - Regularización L1 (regresión Lasso).

# Introducción

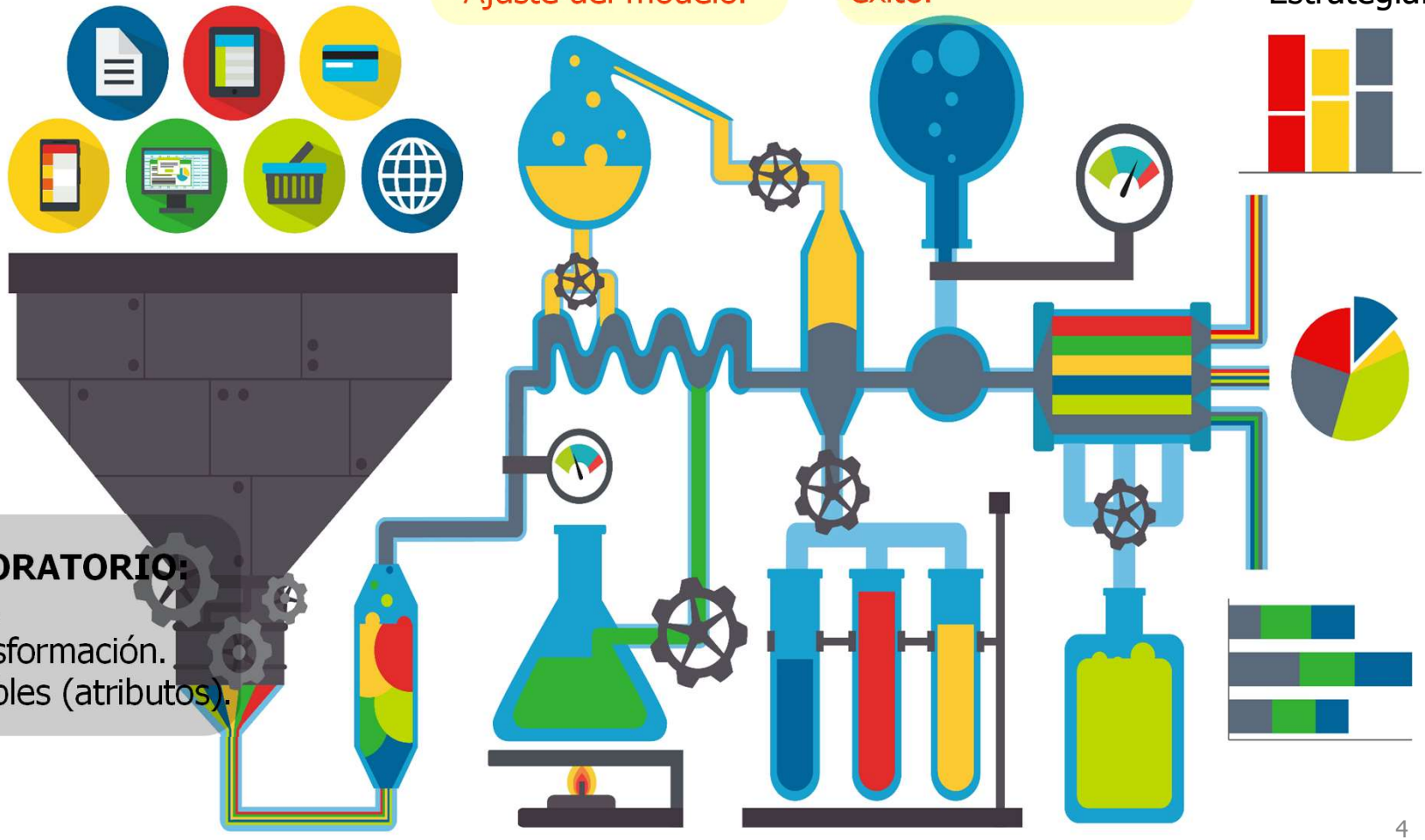
**DATOS:**
Materia prima.

**MODELO:**
Implementación del modelo de analítica.
Ajuste del modelo.

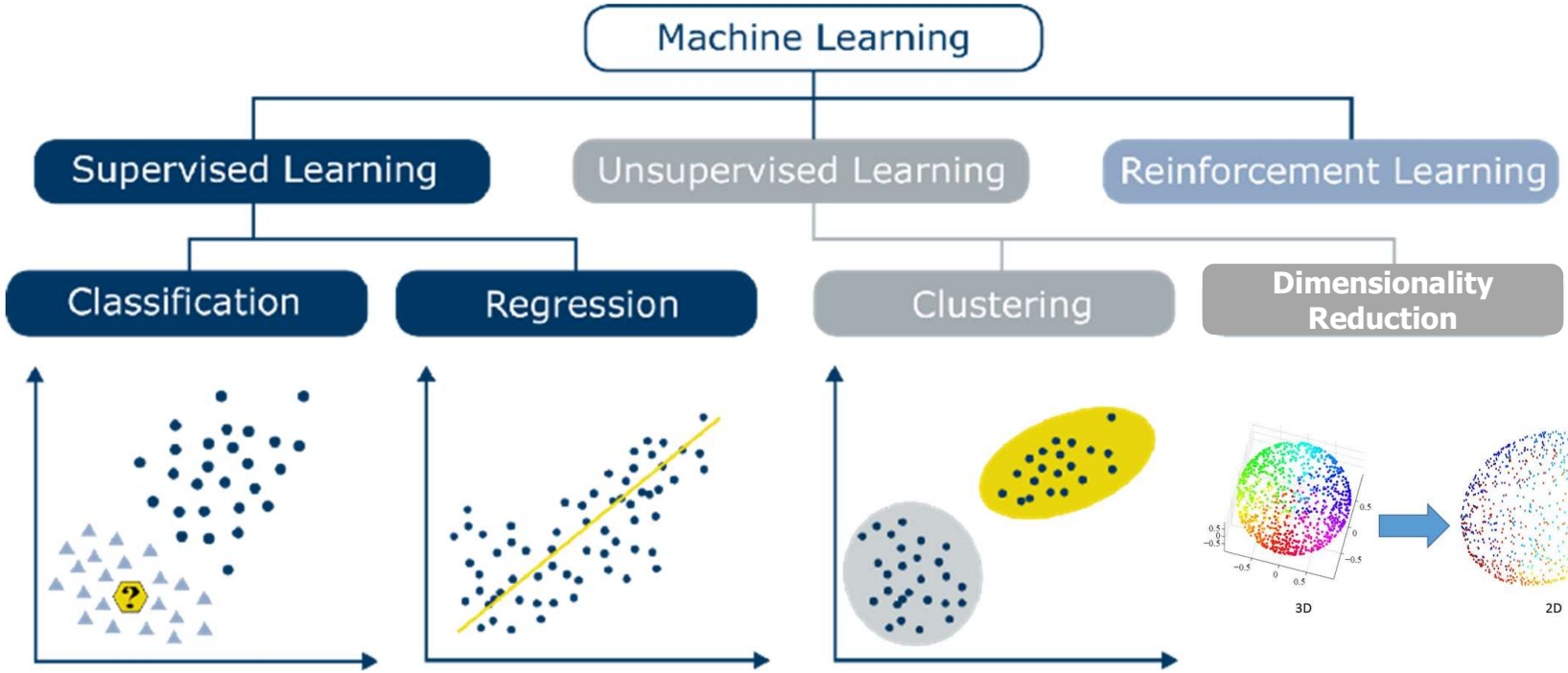
**EVALUACION:**
Viabilidad del negocio.
Validación criterios de éxito.

**DESPLIEGUE:**
Resultados.
Conocimiento.
Estrategia.

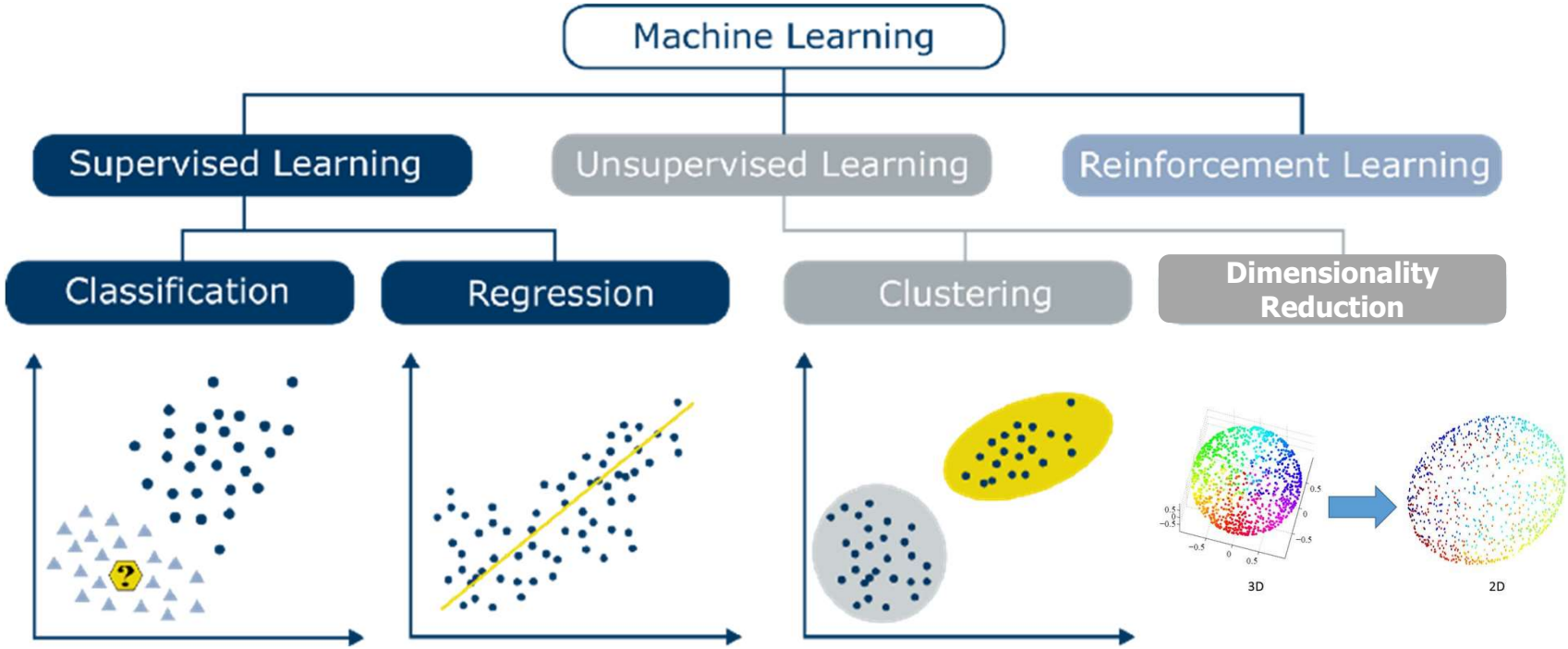
**PREGUNTAS:**
¿Cómo?
¿Cuáles?
¿Cuándo?
¿Por qué? *



**ANALISIS EXPLORATORIO:**
Limpieza de datos.
Preparación / transformación.
Selección de variables (atributos).

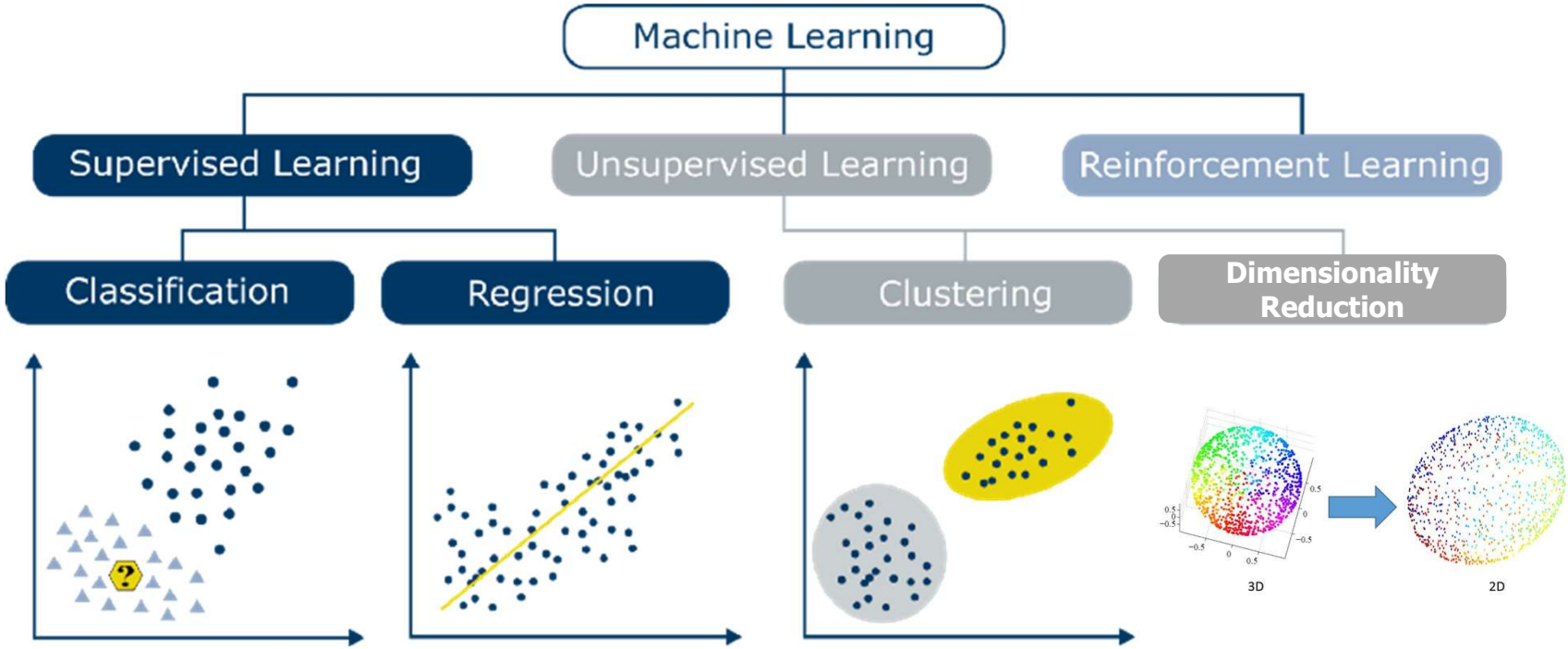



| | | | |
|---|---|---|---|
| **Métodos** | • KNN<br>• Regresión Logística<br>• Bayes Ingenuo | • Regresión Lineal Simple<br>• Regresión Lineal Múltiple<br>• Regresión Polinomial | • K-Means |
| **Evaluación** | Accuracy, Precision, Recall, F1 score, ROC Curve, AUC, etc. | MSE, RMSE, $R^2$, etc. | Método del codo<br>Método de la silueta<br>Calinski-Harabasz |




**Métodos**

- KNN
- Regresión Logística
- Bayes Ingenuo
- Regresión Lineal Simple
- Regresión Lineal Múltiple
- Regresión Polinomial
- K-Means
- Árboles de decisión
- Máquinas de soporte vectorial (SVM)
- Redes neuronales
- …
- Regresión Ridge
- Regresión Lasso




**Métodos**

- KNN
- Regresión Logística
- Bayes Ingenuo

- Regresión Lineal Simple
- Regresión Lineal Múltiple
- Regresión Polinomial

- K-Means

- Árboles de decisión
- Máquinas de soporte vectorial (SVM)
- Redes neuronales
- …

- Regresión Ridge
- Regresión Lasso

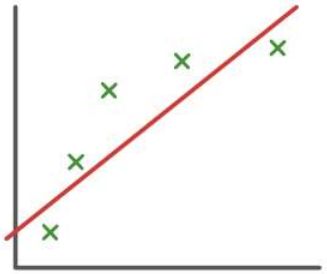
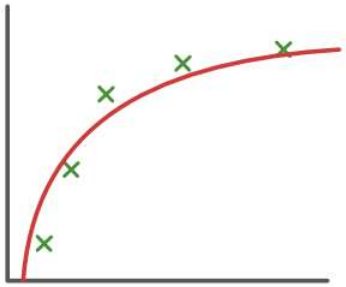
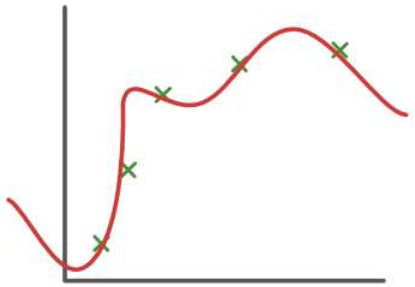
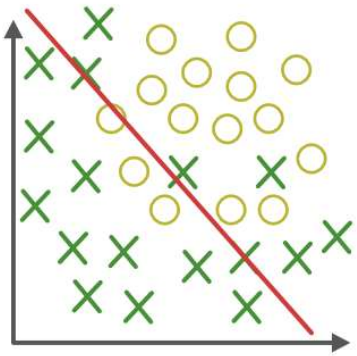
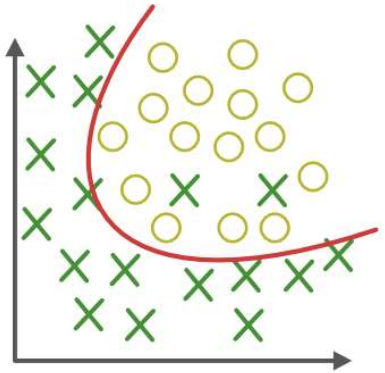
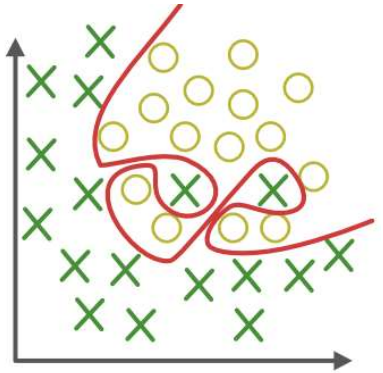
# Técnicas de Regularización

# Errores de sesgo (bias) y varianza (variance)

Al evaluar un modelo de aprendizaje supervisado puede presentarse una de las siguientes situaciones:

- Sub aprendizaje (underfitting)
- Ajuste óptimo.
- Sobre aprendizaje (overfitting)

# Errores de sesgo (bias) y varianza (variance)

| | **Sub aprendizaje (High Bias and Low Variance. (underfitting)** | **Ajuste óptimo** | **Sobre aprendizaje (Low Bias and High Variance. (overfitting)** |
|---|---|---|---|
| **Regresión** |  |  |  |
| **Clasificación** |  |  |  |
| **Evaluación en datos de entrenamiento** | Pobre. | Buena. | Muy buena. |
| **Evaluación en datos de validación y prueba** | Pobre | Buena. | Pobre. |

# Errores de sesgo (bias) y varianza (variance)

Se pueden adoptar varios métodos para combatir el sobre aprendizaje:

- Validación cruzada (cross-validation)
- Reducción del número de atributos.
- Reducción de la complejidad del modelo.
- ***Regularización.***
- Etc.

# Antes de comenzar, ¿qué es la norma *p* de un vector?

Sea **w**=($w_1$,$w_2$,$w_3$, ….,$w_n$) un vector en $\boldsymbol{R}^n$.

Para un número real $p \geq 1$, la norma *p*, o norma $L_p$ del vector ***w*** está definida por:

$$\|\boldsymbol{w}\|_p = \left(|w_1|^p + |w_2|^p + \ldots + |w_n|^p\right)^{1/p}$$

La norma $L_\infty$ (también conocida como norma máxima, o norma uniforme), es el límite de las normas *p* cuando $p \to \infty$. Este límite corresponde con la siguiente definición:

$$\|\boldsymbol{w}\|_\infty = \max\left(|w_1|, |w_2|, \ldots, |w_n|\right)$$

# Regularización en la regresión

- En un modelo de regresión lineal, los parámetros $w_i$ se estiman teniendo como objetivo la minimización de la suma de los residuos al cuadrado. Donde los residuos son las diferencias entre los los valores reales ($y$) y las predicciones ($\hat{y}$):

Predicción en un modelo de regresión lineal múltiple

$$\hat{y}^{(j)} = w_0 + w_1 x_1^{(j)} + w_2 x_2^{(j)} + \ldots + w_i x_i^{(j)} + \ldots + w_n x_n^{(j)}$$

# Regularización en la regresión

- En un modelo de regresión lineal, los parámetros $w_i$ se estiman teniendo como objetivo la minimización de la suma de los residuos al cuadrado. Donde los residuos son las diferencias entre los los valores reales ($y$) y las predicciones ($\hat{y}$):

Predicción en un modelo de regresión lineal múltiple

$$\hat{y}^{(j)} = w_0 + w_1 x_1^{(j)} + w_2 x_2^{(j)} + \ldots + w_i\, x_i^{(j)} + \ldots + w_n\, x_n^{(j)}$$

***Función de costo J* sin regularización**

$$J(\omega) = \sum_{j=1}^{m} [y^{(j)} - \hat{y}^{(j)}]^{\,2} = \sum_{j=1}^{m} [y^{(j)} - (w_0 + w_1 x_1^{(j)} + w_2 x_2^{(j)} + \ldots + w_i\, x_i^{(j)} + \ldots + w_n\, x_n^{(j)})]^{\,2}$$

# Regularización en la regresión

- En un modelo de regresión lineal, los parámetros $w_i$ se estiman teniendo como objetivo la minimización de la suma de los residuos al cuadrado. Donde los residuos son las diferencias entre los los valores reales ($y$) y las predicciones ($\hat{y}$):

Predicción en un modelo de regresión lineal múltiple

$$\hat{y}^{(j)} = \boldsymbol{w_0 + w_1 x_1^{(j)} + w_2 x_2^{(j)} + \ldots + w_i\, x_i^{(j)} + \ldots + w_n\, x_n^{(j)}}$$

***Función de costo J* sin regularización**

$$\boldsymbol{J(\omega) = \sum_{j=1}^{m} [y^{(j)} - \hat{y}^{(j)}]^{\,2} = \sum_{j=1}^{m} [y^{(j)} - (w_0 + w_1 x_1^{(j)} + w_2 x_2^{(j)} + \ldots + w_i\, x_i^{(j)} + \ldots + w_n\, x_n^{(j)})]^{\,2}}$$

$$\textbf{arg min} \sum_{j=1}^{m} [y^{(j)} - (w_0 + w_1 x_1^{(j)} + w_2 x_2^{(j)} + w_i x_i^{(j)} \ldots + w_n x_n^{(j)})]^{\,2}$$

***R**esidual **S**um of **S**quares (**RSS**)*

donde $x_i^{(j)}$ se interpreta como el valor de la *i*-ésima variable independiente de la la *j*-ésima observación, ejemplo, o instancia. Se asume que el conjunto de datos de entrenamiento tiene *m* observaciones.

# Regularización en la regresión

- **Las técnicas de regularización** modifican la función de costo de forma tal que además de penalizar los errores de predicción, se penalicen **magnitudes grandes** de los **coeficientes** $w_j$ (para $j \neq 0$) que caracterizan al modelo.
- Algunos tipos de regularización como la $L_1$ (que da origen a la regresión Lasso: *least absolute shrinkage and selection operator*), permiten reducir el número de variables, y con ello, los requerimientos computacionales para entrenar los modelos y generar nuevas predicciones.

# Regularización en la regresión

- **Las técnicas de regularización** modifican la función de costo de forma tal que además de penalizar los errores de predicción, se penalicen **magnitudes grandes** de los **coeficientes** $w_j$ (para $j \neq 0$) que caracterizan al modelo.
- Algunos tipos de regularización como la $L_1$ (que da origen a la regresión Lasso: *least absolute shrinkage and selection operator*), permiten reducir el número de variables, y con ello, los requerimientos computacionales para entrenar los modelos y generar nuevas predicciones.

***J* sin regularización**
**(se usa en la regresión lineal múltiple):**

$$J(\boldsymbol{w}) = \left[\sum_{i=1}^{m}\left(y^{(i)} - \tilde{y}^{(i)}\right)^2\right] = \left[\sum_{i=1}^{m}\left(y^{(i)} - w_0 - w_1 x_1^{(i)} - .... - w_n x_n^{(i)}\right)^2\right]$$

***J* con regularización $L_2$**
**(se usa en la regresión Ridge):**

$$J(\boldsymbol{w}) = \left[\sum_{i=1}^{m}\left(y^{(i)} - \tilde{y}^{(i)}\right)^2\right] + \frac{\lambda}{2}\sum_{j=1}^{n} w_j^{\,2}$$

**No** se penaliza $w_0$

***J* con regularización $L_1$**
**(se usa en la regresión Lasso):**

$$J(\boldsymbol{w}) = \left[\sum_{i=1}^{m}\left(y^{(i)} - \tilde{y}^{(i)}\right)^2\right] + \frac{\lambda}{2}\sum_{j=1}^{n} |w_j|$$

*n*: número de variables independientes.
*m*: número de observaciones.
λ: factor de regularización.

# Regularización en la regresión

- El parámetro de regularización $\lambda$ sirve para controlar el impacto relativo de los dos términos en la función de costo (el que minimiza el error de predicción, y el que limita la complejidad del modelo para evitar el sobre aprendizaje).
  - Cuando $\lambda = 0$, la penalidad no tiene efecto.
  - Entre más grande $\lambda$, los coeficientes estimados van a ser más pequeños.
  - Seleccionar un valor adecuado de $\lambda$ es crítico. Se puede usar validación cruzada para optimizar este ***hiperparámetro***.
  - Al introducir el término de regularización en la función de costo, se cambia la expresión para el gradiente, y por ende se debe actualizar el código que encuentra los coeficientes usando el gradiente descendente.
  - Obtener el nuevo gradiente es fácil para el caso de regresión Ridge, no tanto en el caso de regresión Lasso.
- La red elástica (*ElasticNet*) es un modelo de regresión que combina las regularizaciones $L_1$ y $L_2$. En este caso se requieren dos parámetros de regularización.

# Regularización en la regresión: consideraciones [1]

- Regresión Ridge obtiene coeficientes pequeños.
  - ✓ Todos los atributos predictores están presentes en el modelo.
  - ✓ El modelo resultante es difícil de interpretar si hay muchas variables predictivas.

- Regresión Lasso hace desaparecer los coeficientes menos importantes.
  - ✓ Sirve como método de selección de atributos, forzando los coeficientes de los atributos eliminados a 0, si $\lambda$ es suficientemente grande.

# Regularización en la regresión: consideraciones [2]

- Regresión Ridge:
  - ✓ En caso de correlación de las variables independientes: funciona bien, pues aunque incluya todas las variables sus coeficientes van a distribuirse considerando sus correlaciones.
  - ✓ Se usa para prevenir el sobre entrenamiento (*overfitting*).
- Regresión Lasso:
  - ✓ En caso de correlación de las variables independientes:
    - Escoge arbitrariamente cualquiera de las variables altamente correlacionadas, poniendo en 0 los coeficientes de las demás.
    - Puede haber problemas en caso de términos polinomiales que puedan desaparecer al estar correlacionados entre ellos.
  - ✓ Produce modelos más simples, robustos con respecto al sobre entrenamiento, y fáciles de interpretar.
  - ✓ Se usa como método de selección de atributos cuando hay muchas variables independientes. Produce modelos "poco densos" (*sparse*).

# ¿Cómo usar la regresión Ridge, Lasso, o la red elástica desde Scikit-learn en Python?

*from sklearn import linear_model*

*linear_model.Ridge*

*linear_model.Lasso*

*linear_model.ElasticNet*

- Para mayor información, consulte:
  - ✓ https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html#sklearn.linear_model.Ridge
  - ✓ https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html
  - ✓ https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html#sklearn.linear_model.ElasticNet

# Veamos un ejemplo para entender la importancia de la regularización

# Ejemplo: regresión polinomial de una variable

$x$ : Variable de entrada real.

$t$ : Variable de salida real.

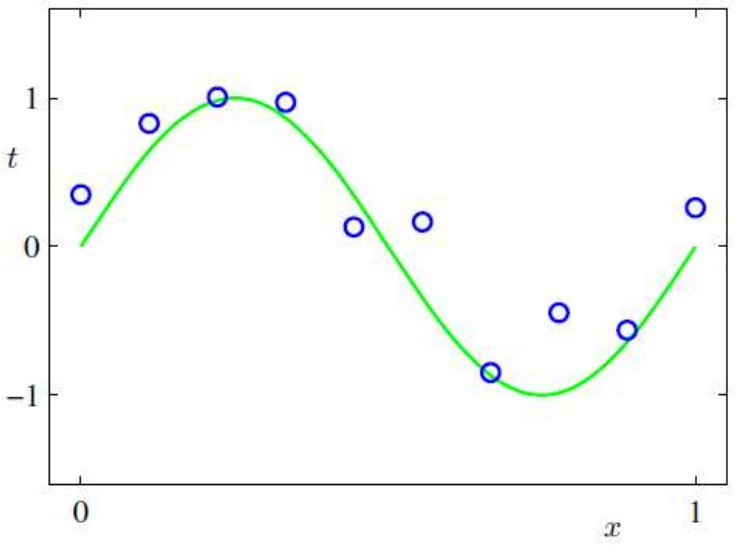
$t = \sin(2\pi x) + \mathcal{N}(\mu, \sigma^2)$ : Modelo que se usó para generar los datos artificiales.

$N$ : Número de observaciones de $x$ y $t$.

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^T$$

$$\boldsymbol{t} = [t_1, t_2, \ldots, t_N]^T$$

**Figure 1.2** Plot of a training data set of $N = 10$ points, shown as blue circles, each comprising an observation of the input variable $x$ along with the corresponding target variable $t$. The green curve shows the function $\sin(2\pi x)$ used to generate the data. Our goal is to predict the value of $t$ for some new value of $x$, without knowledge of the green curve.



# Regresión

- Regresión polinomial en una variable (predicción):

$$y(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

- Donde:

$M$ : orden del polinomio.

$w_0, w_1, w_2, w_3, \ldots, w_M$ : coeficientes del polinomio.

$\boldsymbol{w}$ : vector de coeficientes.

- $y(x, \mathbf{w})$ es una función no lineal de $x$, pero es una función lineal de los coeficientes $\mathbf{w}$.

- Cuando las funciones son lineales en los coeficientes o parámetros desconocidos, estas se conocen como modelos lineales.

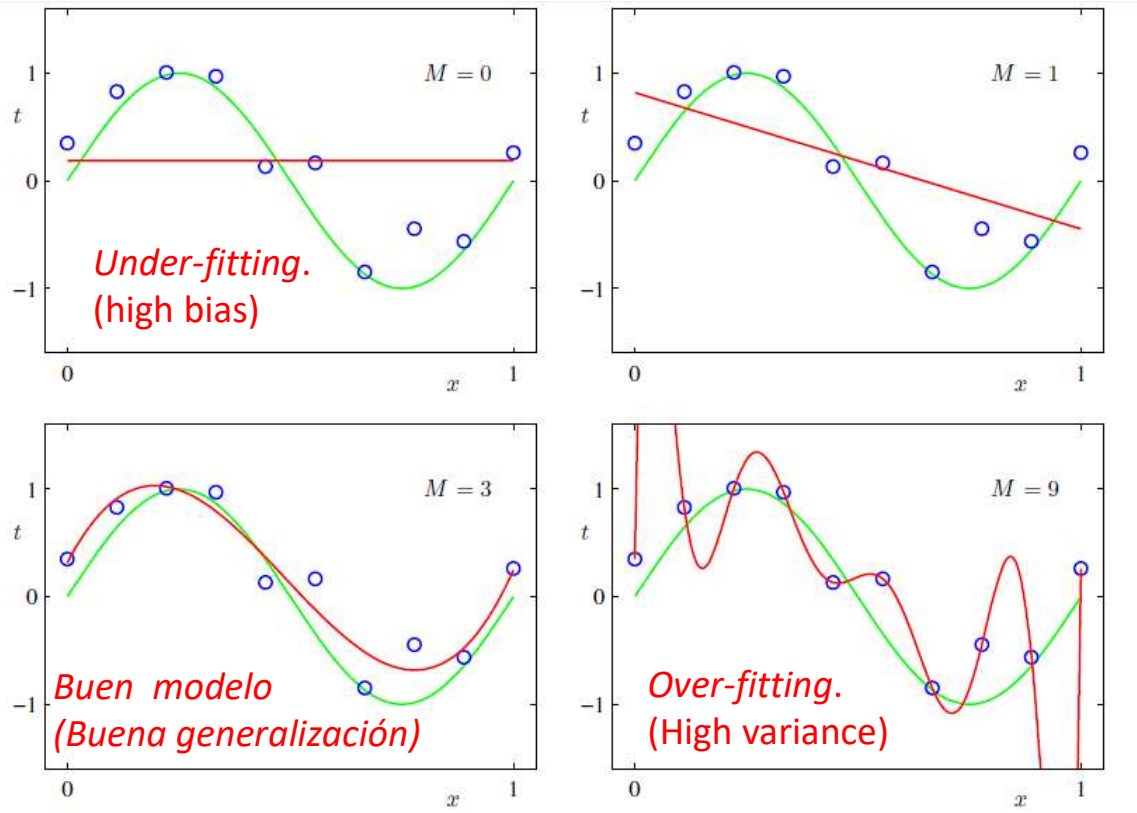



**Figure 1.4** Plots of polynomials having various orders $M$, shown as red curves, fitted to the data set shown in Figure 1.2.

# Coeficientes $\boldsymbol{w}$ del polinomio para diferentes valores de $M$

**Table 1.1** Table of the coefficients $\mathrm{w}^\star$ for polynomials of various order. Observe how the typical magnitude of the coefficients increases dramatically as the order of the polynomial increases.

| | $M=0$ | $M=1$ | $M=3$ | $M=9$ |
|---|---|---|---|---|
| $w_0^\star$ | 0.19 | 0.82 | 0.31 | 0.35 |
| $w_1^\star$ | | -1.27 | 7.99 | 232.37 |
| $w_2^\star$ | | | -25.43 | -5321.83 |
| $w_3^\star$ | | | 17.37 | 48568.31 |
| $w_4^\star$ | | | | -231639.30 |
| $w_5^\star$ | | | | 640042.26 |
| $w_6^\star$ | | | | -1061800.52 |
| $w_7^\star$ | | | | 1042400.18 |
| $w_8^\star$ | | | | -557682.99 |
| $w_9^\star$ | | | | 125201.43 |

# Efecto de usar más datos

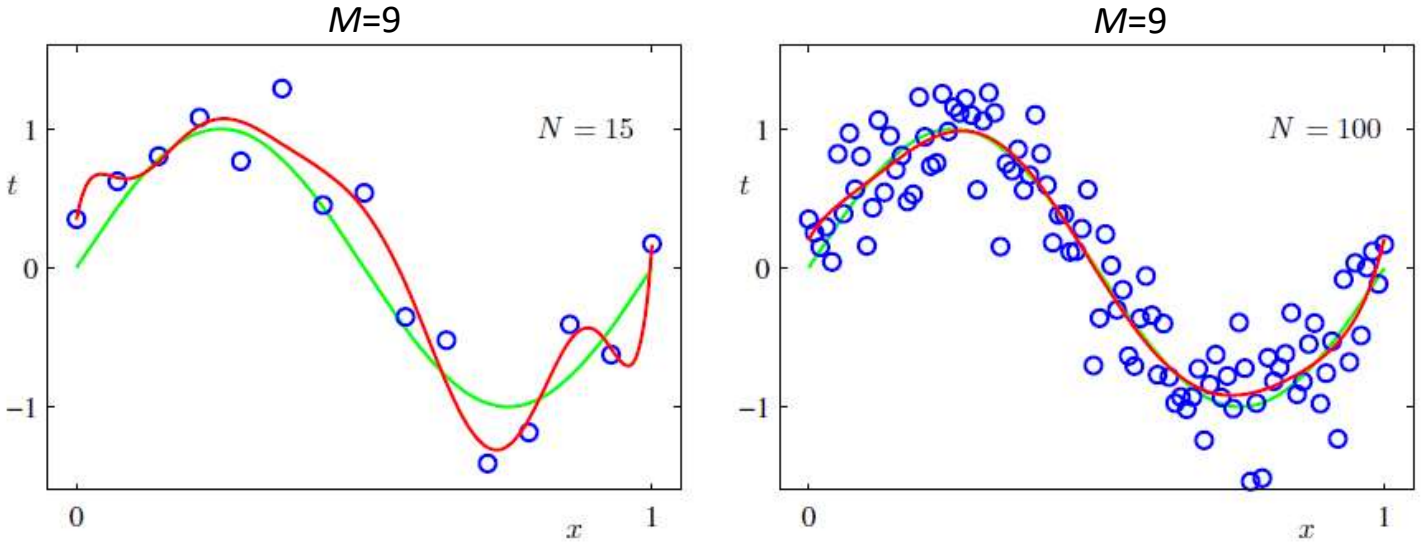



**Figure 1.6** Plots of the solutions obtained by minimizing the sum-of-squares error function using the $M = 9$ polynomial for $N = 15$ data points (left plot) and $N = 100$ data points (right plot). We see that increasing the size of the data set reduces the over-fitting problem.

# Efecto de usar regularización $L_2$

- En las siguientes gráficas se usaron 10 datos de entrenamiento, y M=9.

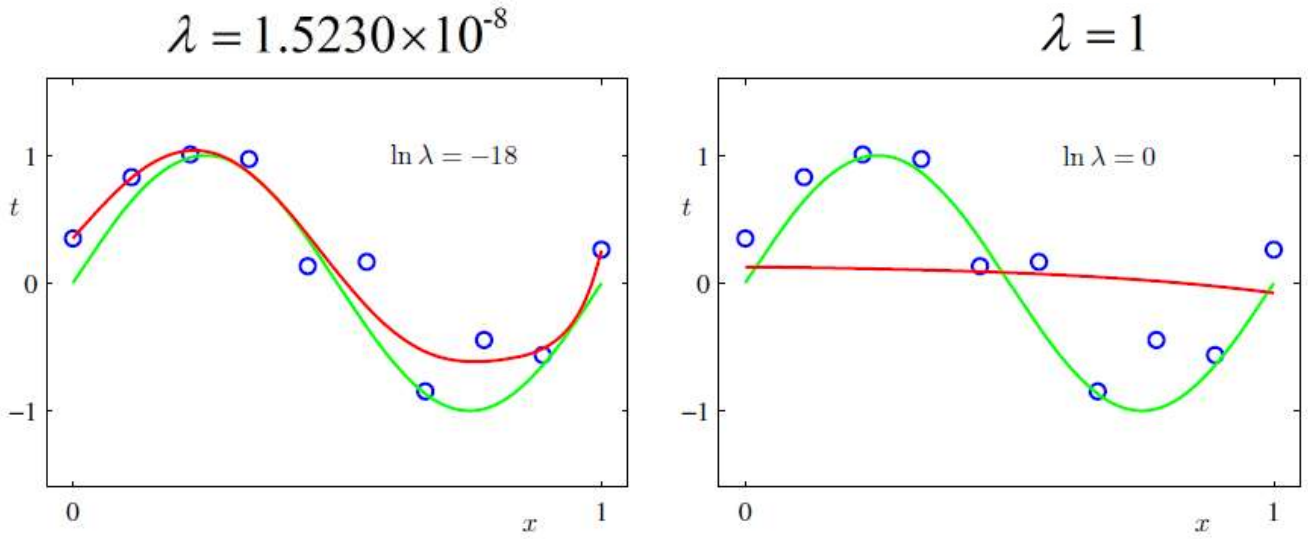



**Figure 1.7** Plots of $M = 9$ polynomials fitted to the data set shown in Figure 1.2 using the regularized error function (1.4) for two values of the regularization parameter $\lambda$ corresponding to $\ln \lambda = -18$ and $\ln \lambda = 0$. The case of no regularizer, i.e., $\lambda = 0$, corresponding to $\ln \lambda = -\infty$, is shown at the bottom right of Figure 1.4.

# Coeficientes ***w*** del polinomio para diferentes valores de λ, con *M=9*

$\lambda = 0$ $\lambda = 1.5230\times10^{-8}$ $\lambda = 1$

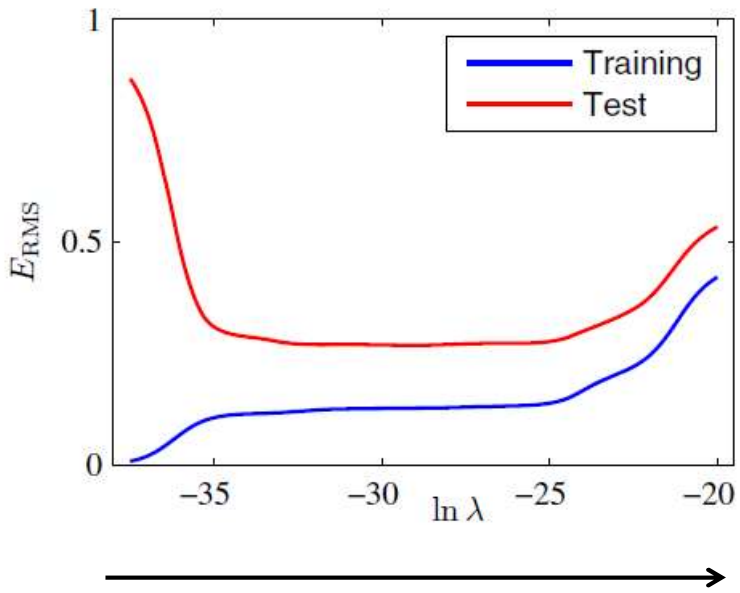
**Table 1.2** Table of the coefficients $\mathrm{w}^\star$ for $M = 9$ polynomials with various values for the regularization parameter $\lambda$. Note that $\ln\lambda = -\infty$ corresponds to a model with no regularization, i.e., to the graph at the bottom right in Figure 1.4. We see that, as the value of $\lambda$ increases, the typical magnitude of the coefficients gets smaller.

| | $\ln\lambda = -\infty$ | $\ln\lambda = -18$ | $\ln\lambda = 0$ |
|---|---|---|---|
| $w_0^\star$ | 0.35 | 0.35 | 0.13 |
| $w_1^\star$ | 232.37 | 4.74 | -0.05 |
| $w_2^\star$ | -5321.83 | -0.77 | -0.06 |
| $w_3^\star$ | 48568.31 | -31.97 | -0.05 |
| $w_4^\star$ | -231639.30 | -3.89 | -0.03 |
| $w_5^\star$ | 640042.26 | 55.28 | -0.02 |
| $w_6^\star$ | -1061800.52 | 41.32 | -0.01 |
| $w_7^\star$ | 1042400.18 | -45.95 | -0.00 |
| $w_8^\star$ | -557682.99 | -91.53 | 0.00 |
| $w_9^\star$ | 125201.43 | 72.68 | 0.01 |

- A medida que aumenta λ, disminuye la magnitud de los coeficientes.

# Efecto del parámetro de regularización en el aprendizaje

**Figure 1.8** Graph of the root-mean-square error (1.3) versus $\ln \lambda$ for the $M = 9$ polynomial.




λ aumenta en esta dirección