Estadística multivariada

Por Tomás Olarte y Santiago Hernández

02 Supervisado básico

Hola!

Soy Tomás Olarte

tolarteh@eafit.edu.co

Ingeniero matemático y estudié inteligencia artificial en mi maestría porque me encanta que las máquinas trabajen por mí en las tareas repetitivas.

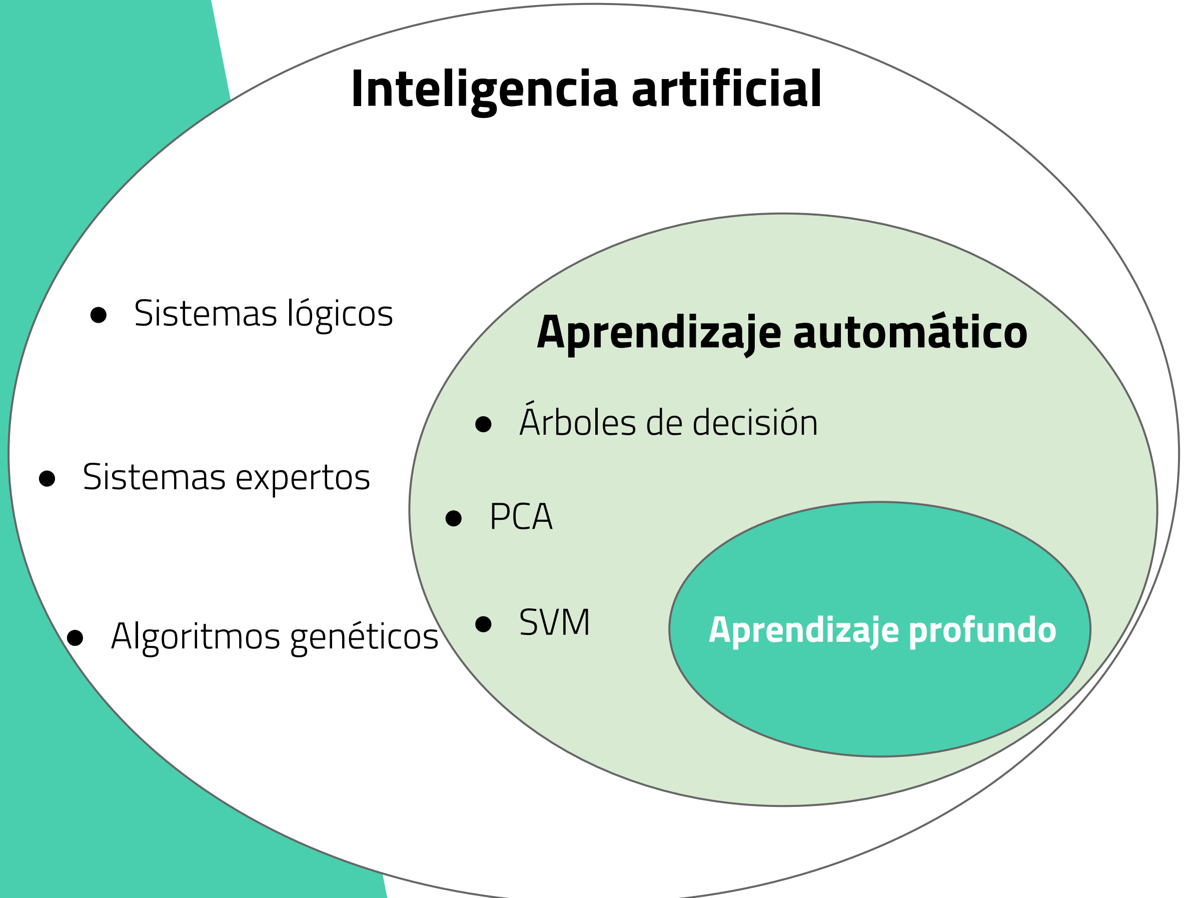
Prof. Diego Kuonen @DiegoKu...
"**AI is not about** replacing the human with a robot. It is about taking the robot out of the human."

Hoy aprenderemos...

Aprendizaje supervisado

- Notación para el aprendizaje supervisado
- Conocer el problema de regresión
- Solucionar la regresión lineal
- Extender a la regresión NO lineal
- [TAREA] Implementar una regresión lineal.

Conceptos básicos



La **inteligencia artificial (AI)** estudia la simulación de comportamientos inteligentes.

El **machine learning (ML)** estudia los algoritmos que mejoran su rendimiento incorporando nuevos datos a un modelo existente.

El **deep learning** usa algoritmos de redes neuronales artificiales para extraer características cada vez más complejas a partir de los datos de entrada, buscando imitar al cerebro.

ML Supervisado

$$D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \subseteq \mathcal{R}^d \times \mathcal{C}$$

D es el conjunto de datos
R^d es el espacio de atributos *d*-dimensional
$\mathbf{X}_i$ es el vector de entrada del ejemplo *i*
y_i es la etiqueta del ejemplo *i*
C es el espacio de etiquetas

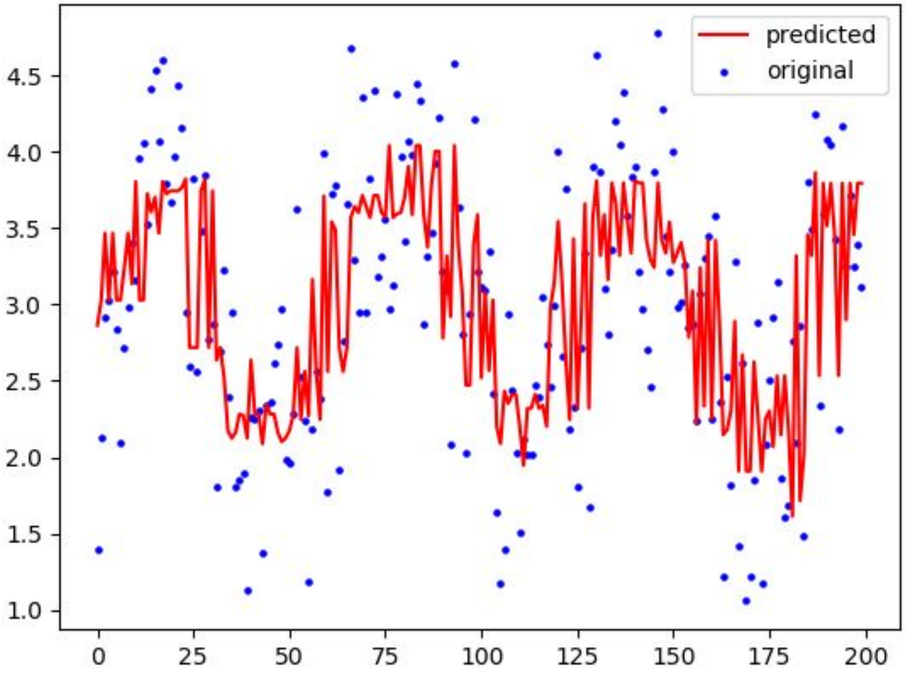
Regresión	$\mathcal{C} = \mathbb{R}$	**Ej.** Predecir la altura futura de una persona
Clasificación binaria	$\mathcal{C} = \{0, 1\}$ $\mathcal{C} = \{-1, +1\}$	**Ej.** Realizar un filtro de correo spam
Clasificación multi-clase	$\mathcal{C} = \{1, 2, \ldots, K\}$	**Ej.** Clasificar la cara de una persona donde se tienen *K* identidades (Obama, Bush, etc.)

Regresión

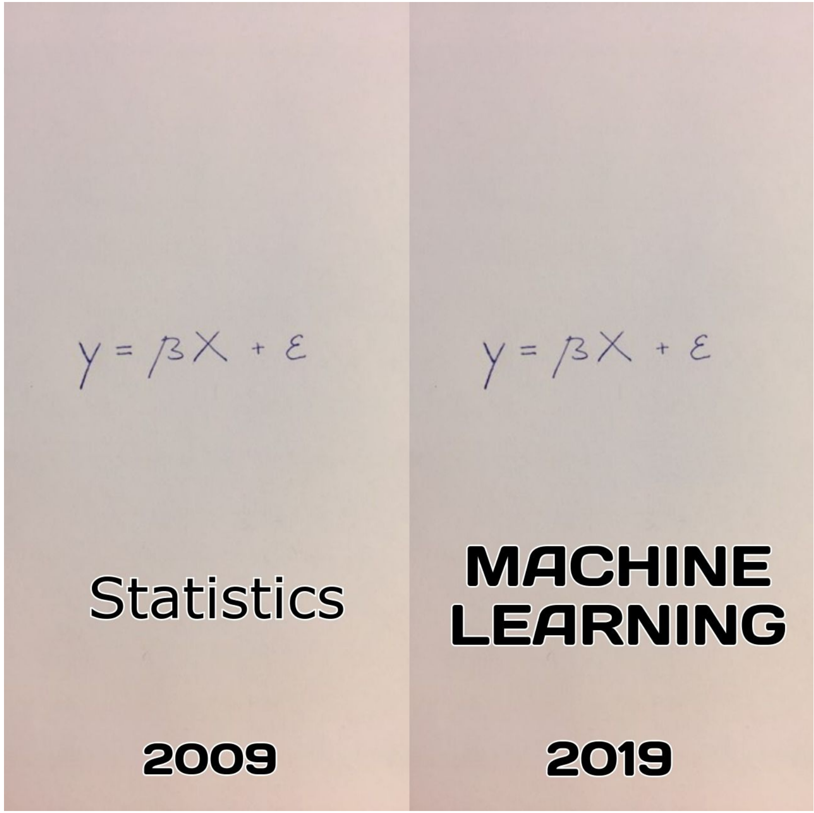
Quiero encontrar una relación

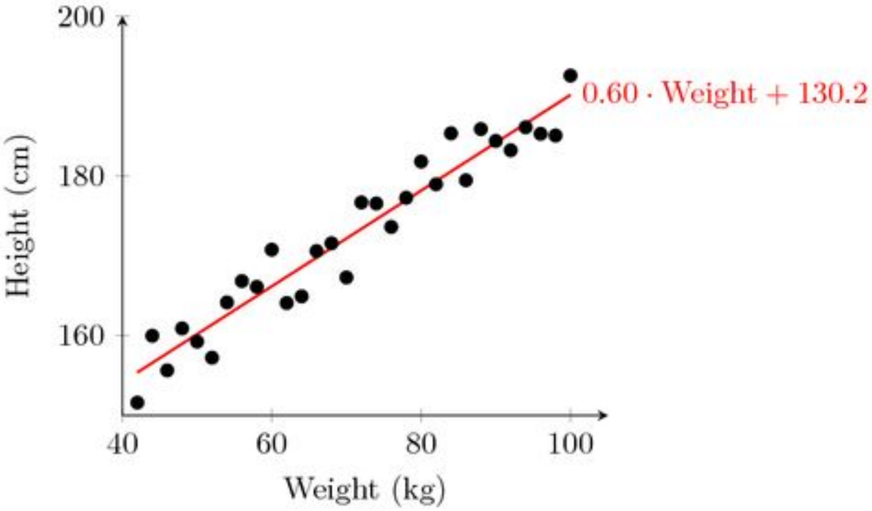
$$y = f(x)$$

donde ***y*** es una variable continua.



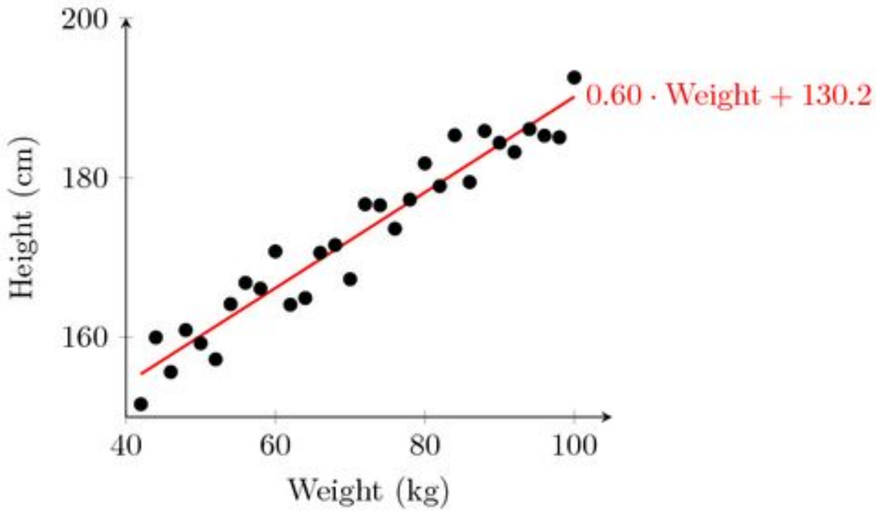
Regresión lineal





Regresión lineal

- Ajustar una **línea** a las observaciones
- Usar esta línea para **predecir** valores no observados
- Usualmente se encuentra con el método de los **mínimos cuadrados**
- Si la relación no es lineal, se puede **procesar las entradas** para reflejar esta relación.



Regresión lineal

Supuesto del modelo: una relación lineal

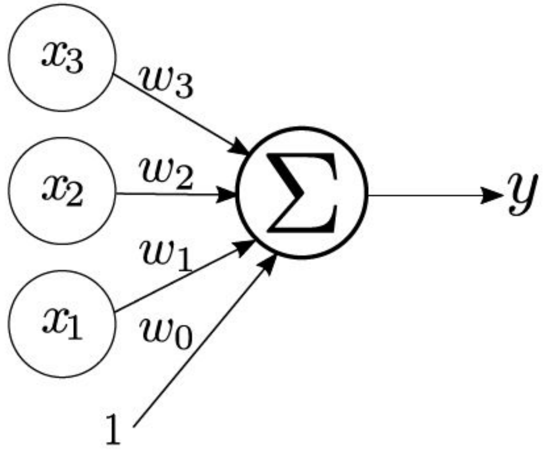
$$y = f(\mathbf{x}) = w_0 + \sum_{j=1}^{d} w_j \times x_j$$

Al predecir:

$$\hat{y} = \hat{f}(\mathbf{x}) = w_0 + \sum_{j=1}^{d} w_j \times x_j$$

$$\hat{y} = \hat{f}(\mathbf{x}) = \mathbf{x}^T \mathbf{w}$$

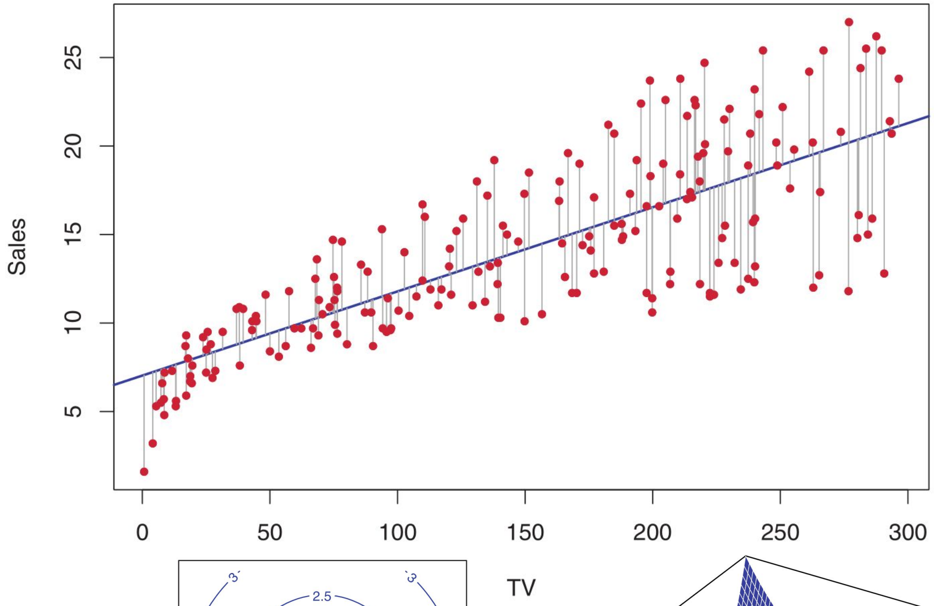
Representación alternativa

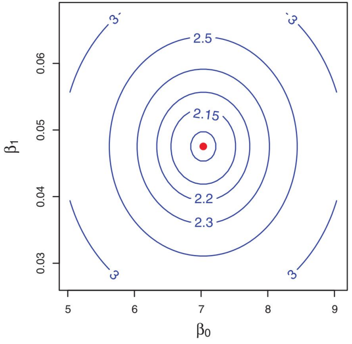


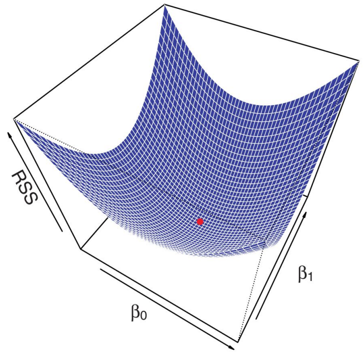
Mínimos cuadrados

Minimiza el error de ajuste de la línea a cada punto (en realidad el error promedio):

- No me interesa si se equivoca por encima o por debajo (**error al cuadrado** o error absoluto)
- **Sumo** los errores entre todos los puntos (suma de errores cuadrados)
- Encuentro el **óptimo** valor de dicha suma, esto lo puedo hacer de manera analítica (*ecuación normal*) o iterativa (*descenso por el gradiente*).







Mínimos cuadrados

$$
\begin{array}{ll}
y_1 & \mathbf{x}_1 \\
y_2 & \mathbf{x}_2 \\
\vdots & \\
y_n & \mathbf{x}_i \\
\vdots & \\
y_i & \mathbf{x}_n
\end{array}
\qquad\qquad
\begin{aligned}
\hat{y}_1 &= \hat{f}(\mathbf{x}_1) = \mathbf{x}_1^T\mathbf{w} \\
\hat{y}_2 &= \hat{f}(\mathbf{x}_2) = \mathbf{x}_2^T\mathbf{w} \\
&\vdots \\
\hat{y}_i &= \hat{f}(\mathbf{x}_i) = \mathbf{x}_i^T\mathbf{w} \\
&\vdots \\
\hat{y}_n &= \hat{f}(\mathbf{x}_n) = \mathbf{x}_n^T\mathbf{w}
\end{aligned}
$$

Debemos encontrar $\mathbf{w}$ que haga mínima la diferencia entre el valor real y la predicción:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T\mathbf{w})^2$$

Mínimos cuadrados

Teniendo esta función objetivo:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \mathbf{w})^2$$

$$= \arg\min_{\mathbf{w}} (Y - \mathbf{X}\mathbf{w})^T (Y - \mathbf{X}\mathbf{w})$$

Podemos encontrar una solución analítica (ecuación normal):

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

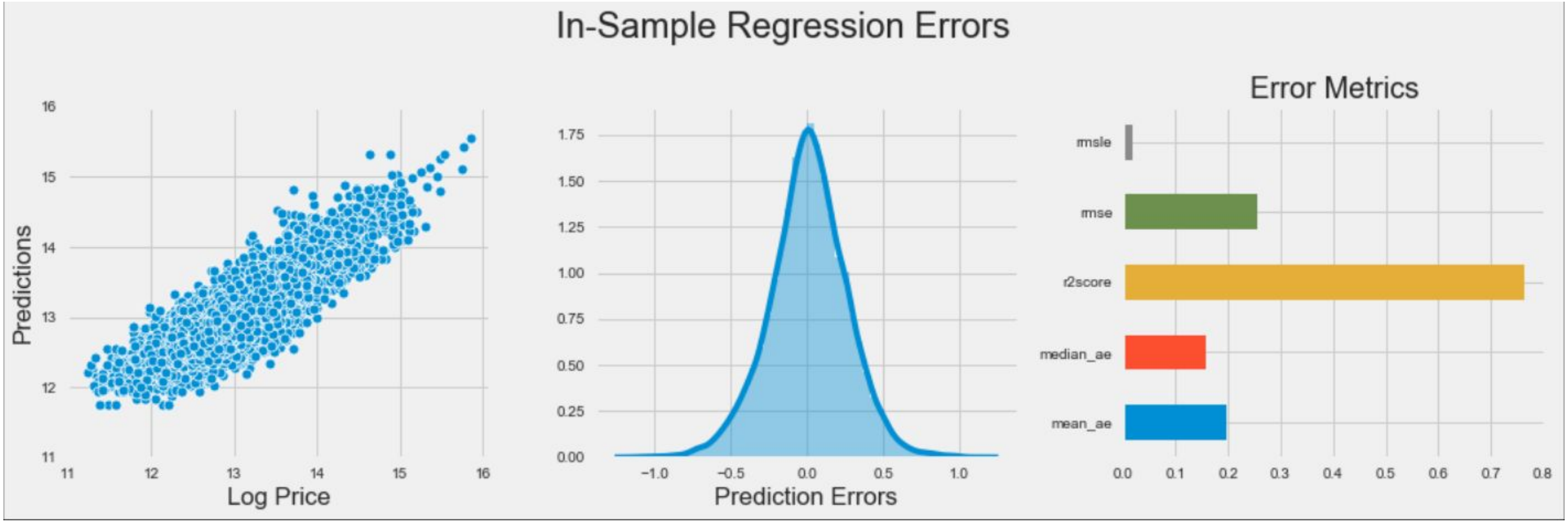
Métricas

Diferentes métricas para los problemas de regresión pueden ser usadas, pero entre las más usadas están:

- MSE
- R2

Name	Formula	sklearn
Mean squared error	$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$	`mean_squared_error`
Mean squared log error	$\frac{1}{N}\sum_{i=1}^{N}(\ln(1+y_i) - \ln(1+\hat{y}_i))^2$	`mean_squared_log_error`
Mean absolute error	$\frac{1}{N}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i \rvert$	`mean_absolute_error`
Median absolute error	$\text{median}\left(\lvert y_1 - \hat{y_1}\rvert, \ldots, \lvert y_n - \hat{y_n}\rvert\right)$	`median_absolute_error`
Explained variance	$1 - \frac{\text{var}(y-\hat{y})}{\text{var}(y)}$	`explained_variance_score`
R2 score	$1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$	`r2_score`

Métricas



Al usar las diferentes métricas, resultados ligeramente diferentes pueden ocurrir, pero **los supuestos** del modelo deben mantenerse para considerar el modelo adecuado.

Regresión no lineal

Una **regresión no lineal** supone que la relación entre la variable dependiente y las independientes es una combinación lineal de transformaciones no lineales de las variables predictores.

$$\hat{y}_i = \hat{f}(\mathbf{x}_i) = \mathbf{x}_i^T \mathbf{w} \longrightarrow \hat{y}_i = \hat{f}(\phi(\mathbf{x}_i)) = \phi(\mathbf{x}_i)^T \mathbf{w}$$

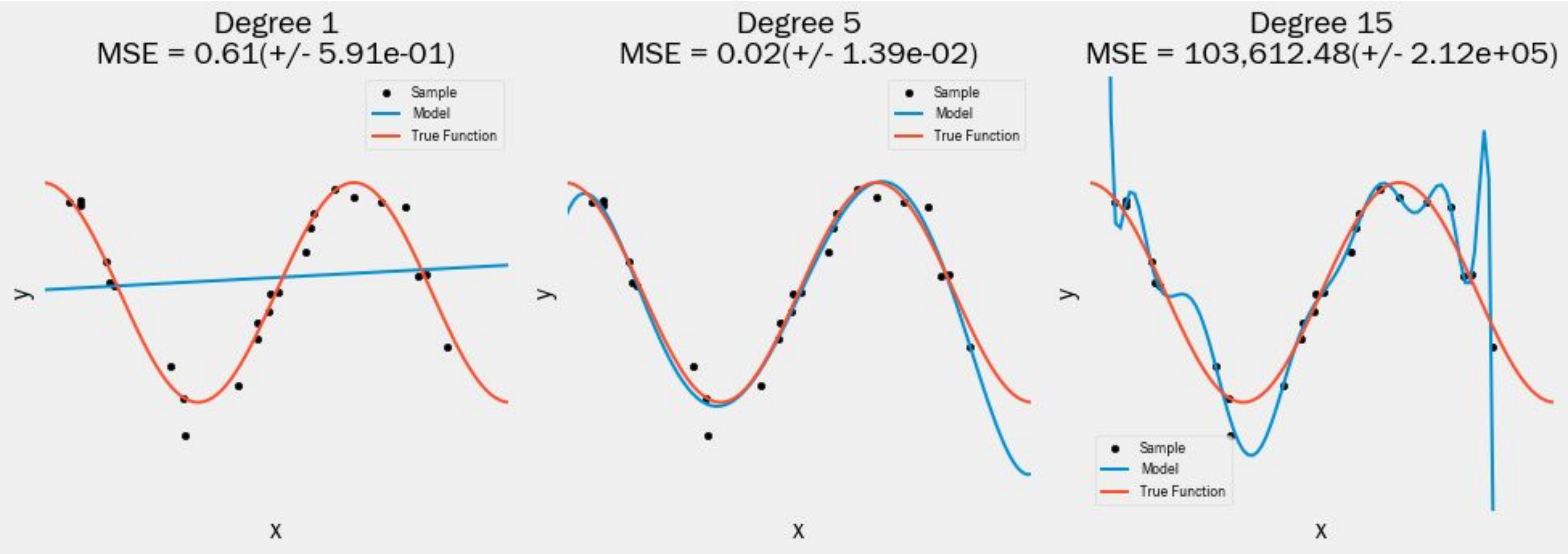
ϕ : Cualquier función que haga una transformación no lineal sobre las variables de entrada al modelo.

Ejemplo:

$$\langle x_1, x_2 \rangle \longrightarrow \langle x_1^2, \sin x_2, x_1 x_2 \rangle$$

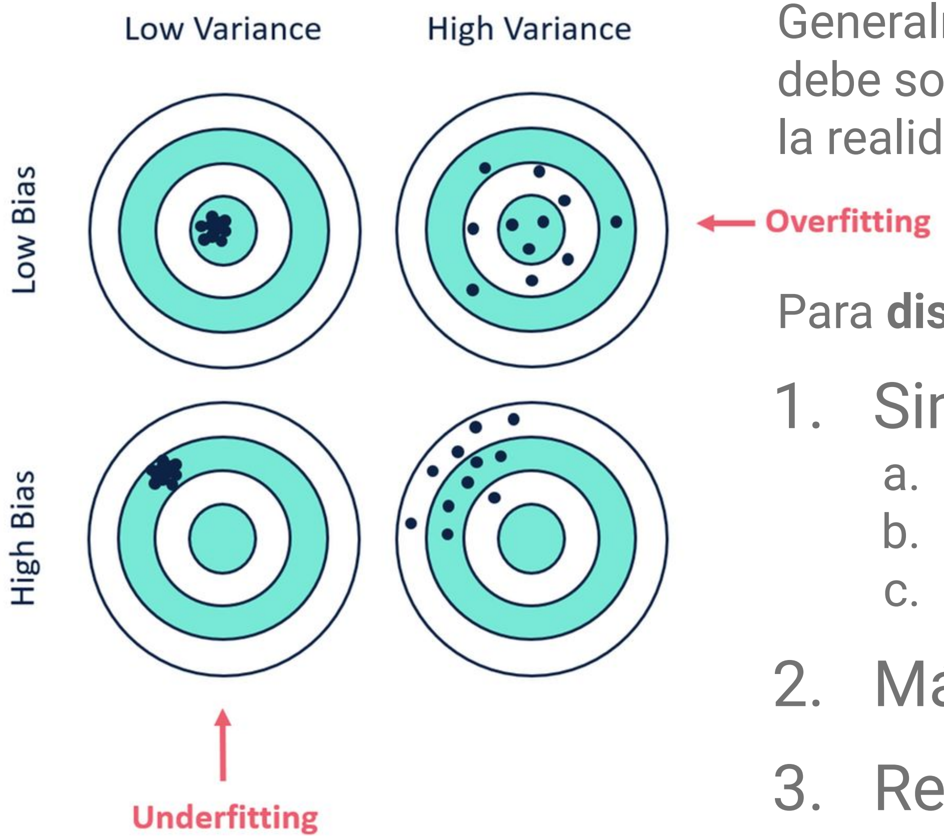
Regresión no lineal

sklearn.preprocessing.PolynomialFeatures — scikit-learn 0.22.1 documentation



A mayor complejidad del modelo, más capacidad de “memorizar” los datos de entrenamiento, llevando al modelo a generalizar en menor grado (pobres resultados en pruebas): **sobreajuste**.

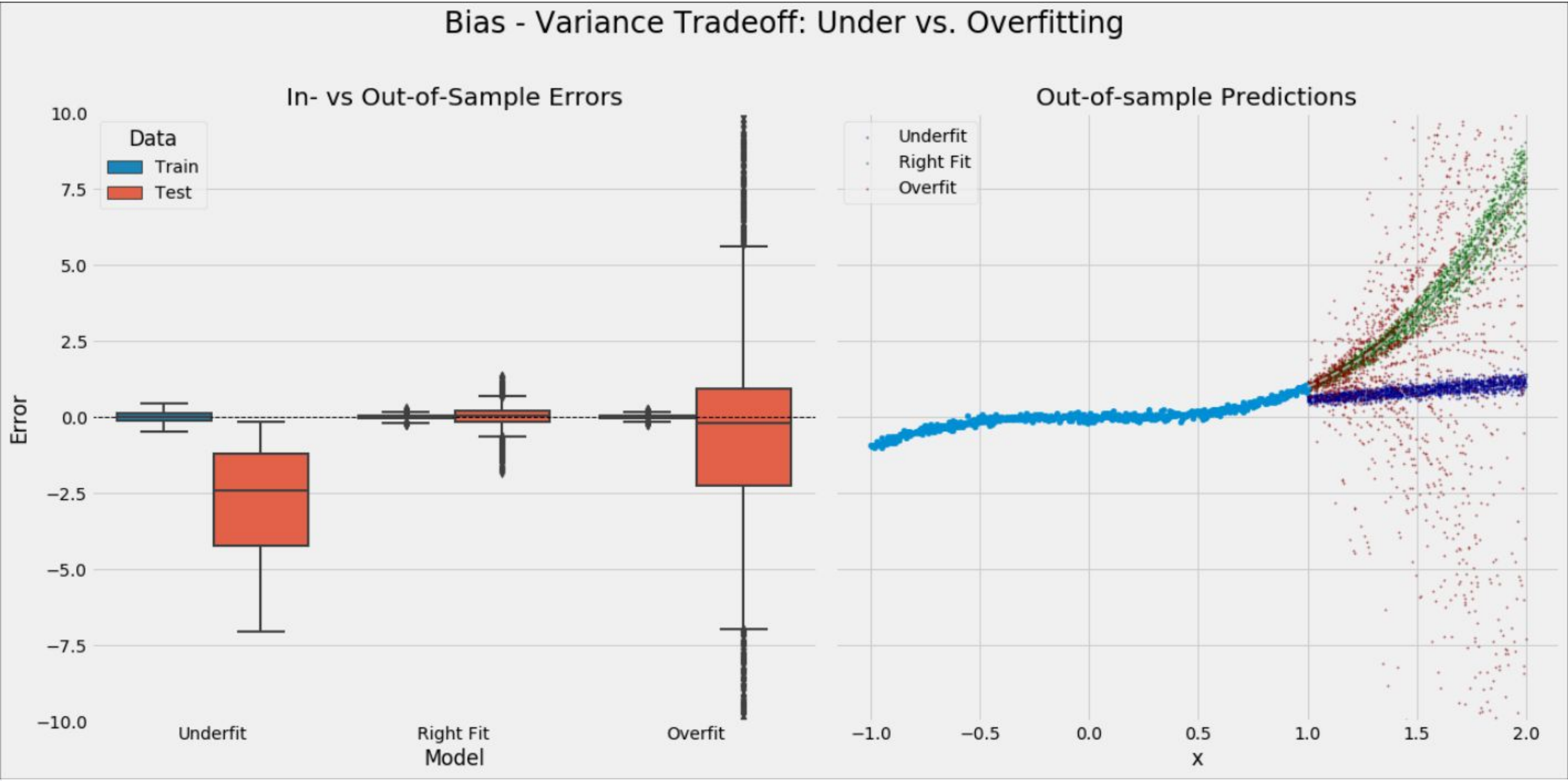
Básico: Sesgo vs. Varianza



Generalmente en el proceso de modelación se debe sopesar el **sesgo** del modelo respecto a la realidad y su **varianza** en nuevas muestras.

Para **disminuir** esta varianza se recomienda:

1. Simplificar el modelo
    a. Quitar variables
    b. Cambiar hiper-parámetros
    c. Cambiar el tipo de modelo
2. Más datos de entrenamiento
3. Regularización en el modelo.


Bias - Variance Tradeoff: Under vs. Overfitting
In- vs Out-of-Sample Errors
Data
Train
Test
Error
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
Underfit
Right Fit
Overfit
Model
Out-of-sample Predictions
Underfit
Right Fit
Overfit
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x

Un poco de práctica

Programar la solución analítica de regresión lineal

http://bit.ly/eafit-reglineal