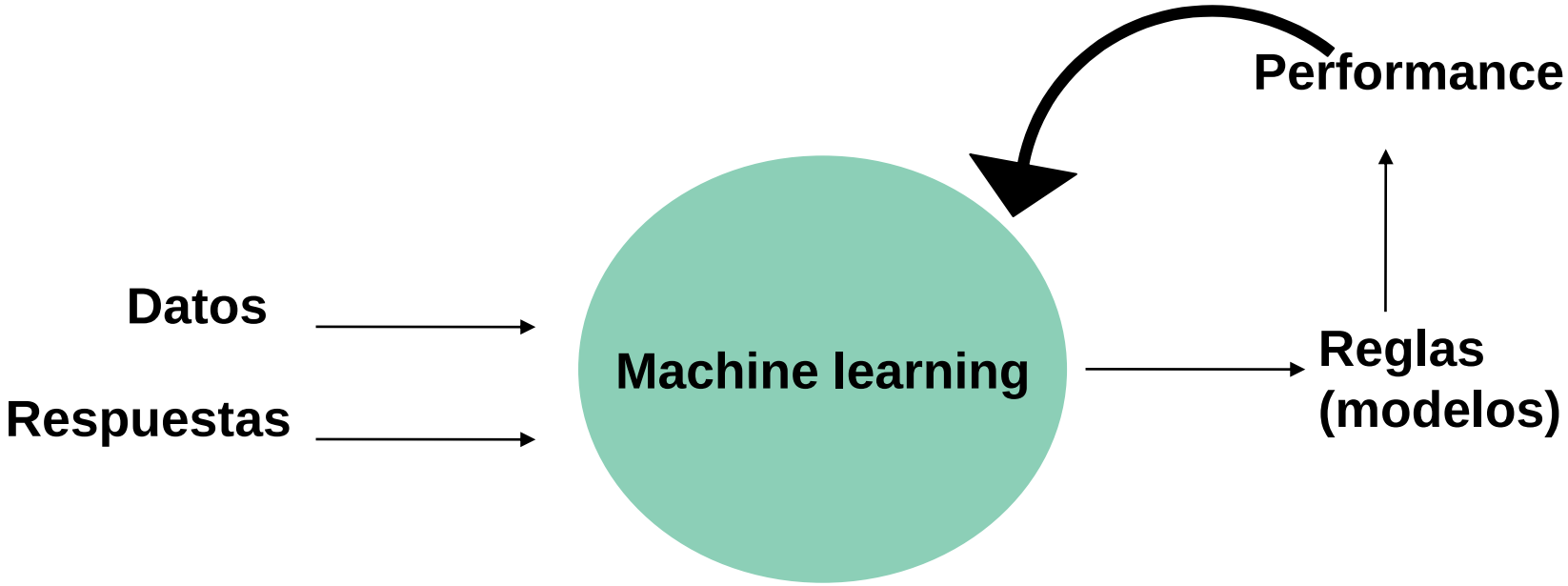
Ciencia y analítica de datos

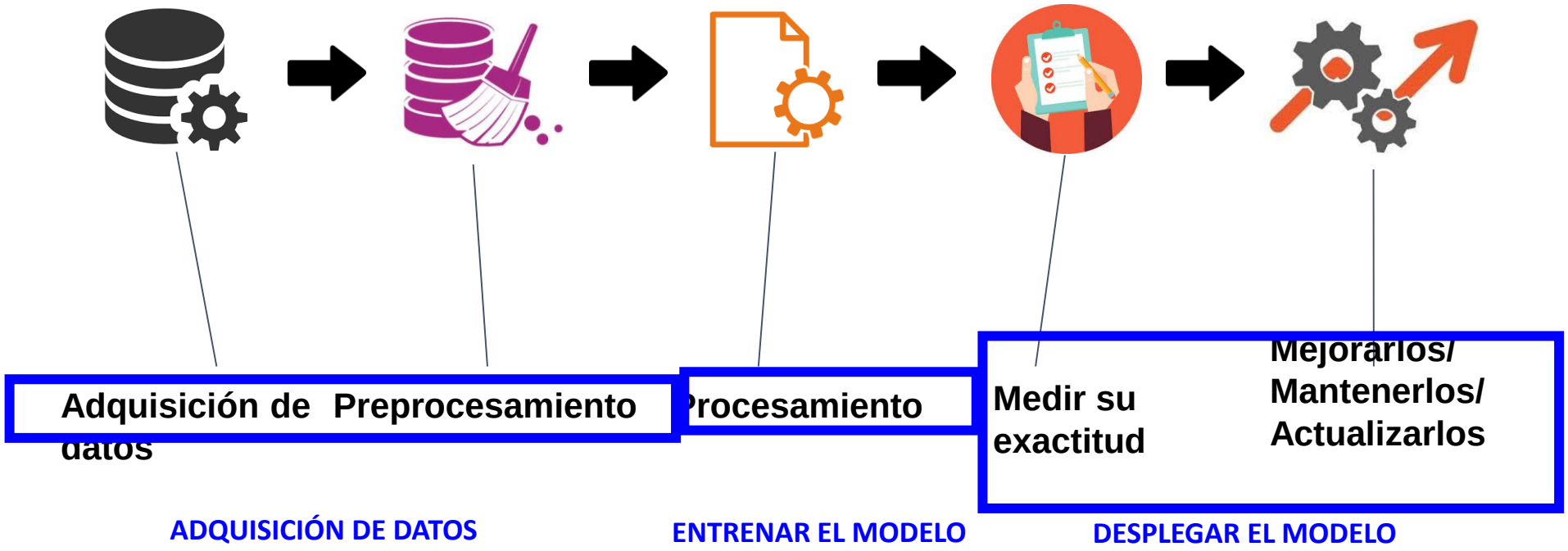
Dra. María de la Paz Rico Fernández.

Tecnológico de Monterrey

- ¿Qué necesitamos para ml?

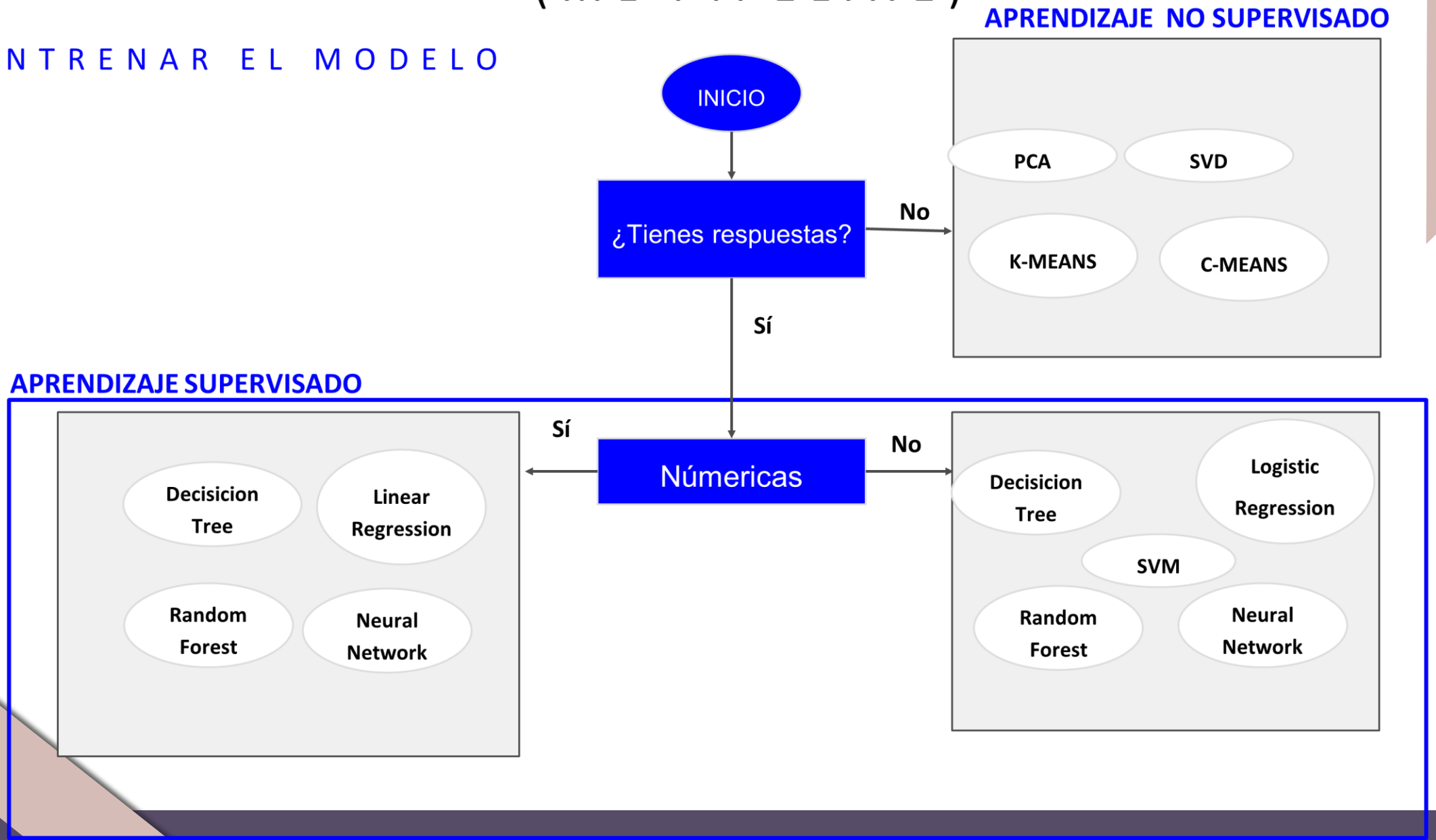


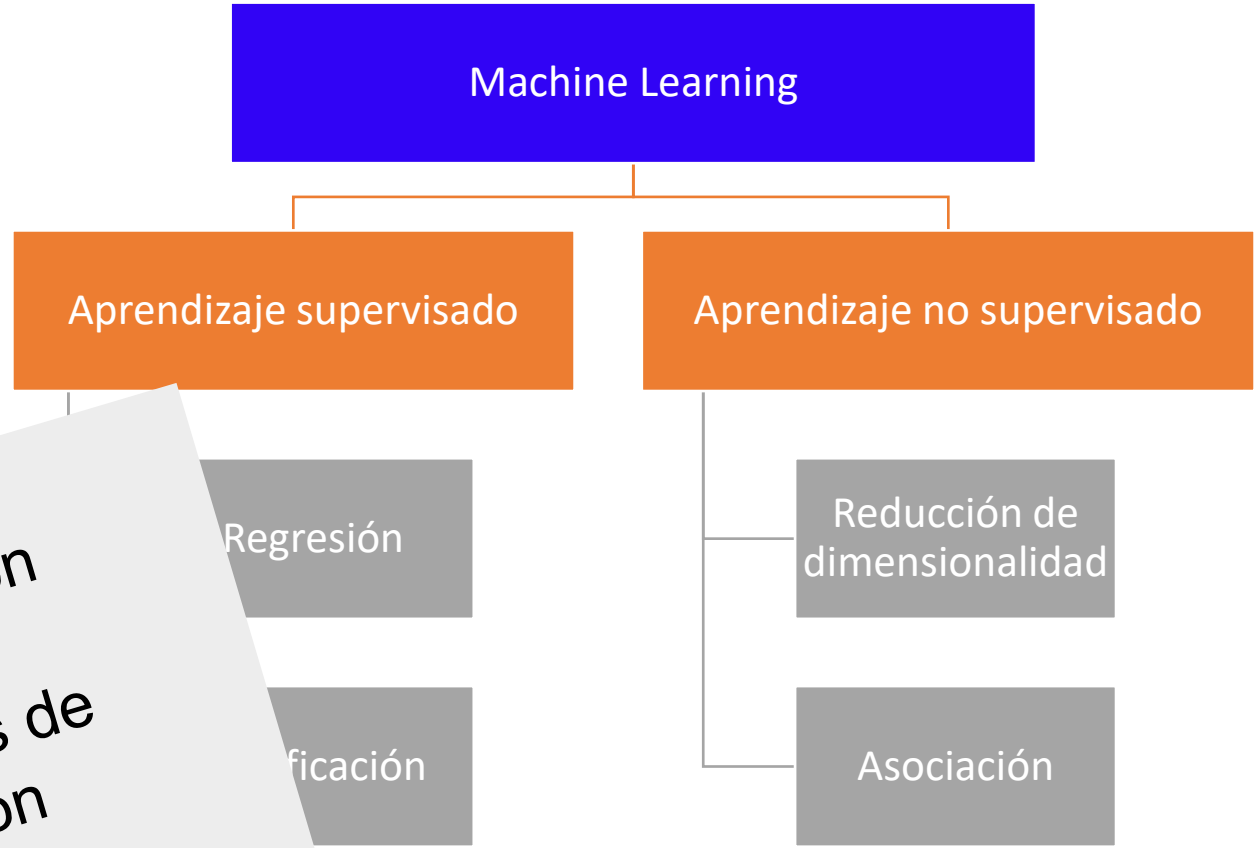
- ml pipeline


Adquisición de datos
Preprocesamiento
Procesamiento
Medir su exactitud
Mejorarlos/ Mantenerlos/ Actualizarlos
ADQUISICIÓN DE DATOS
ENTRENAR EL MODELO
DESPLEGAR EL MODELO

PASOS CLAVE PARA ML PROJECT (ML PIPELINE)

ENTRENAR EL MODELO





Métodos:
- Regresión lineal
- Arboles de decisión
- SVM
- Neural network

Métodos:
- K-means
- PCA

ml supervisado

REGRESIÓN LINEAL

En la regresión, las etiquetas son datos continuos.

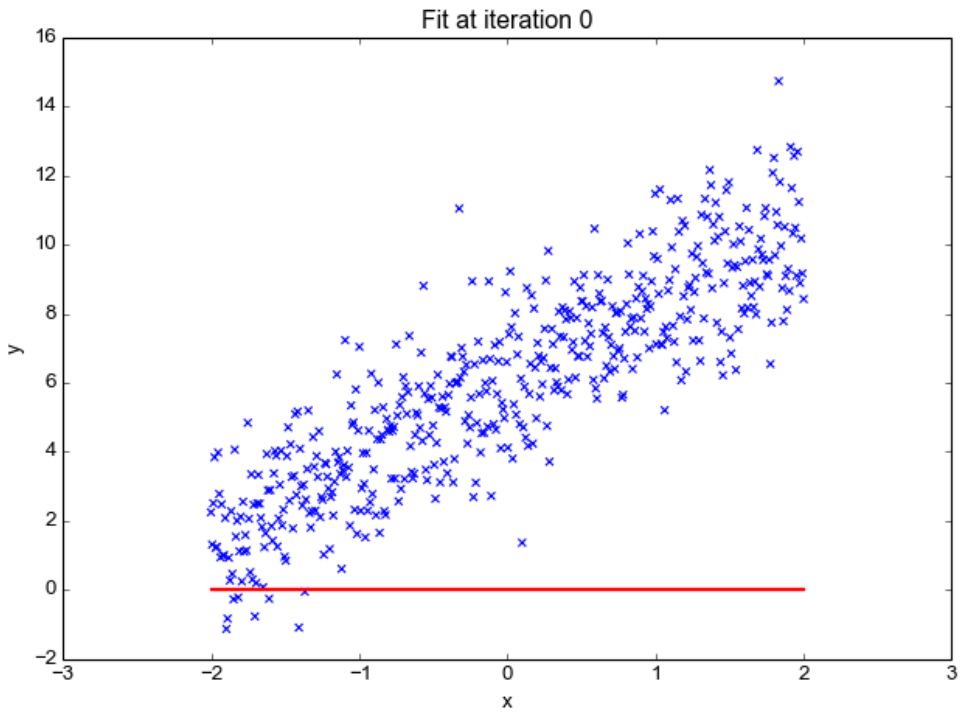
El modelo lineal se forma a través de la suma ponderada de las variables (características), más un sesgo o donde se intercepta.

$$y = ax + b$$

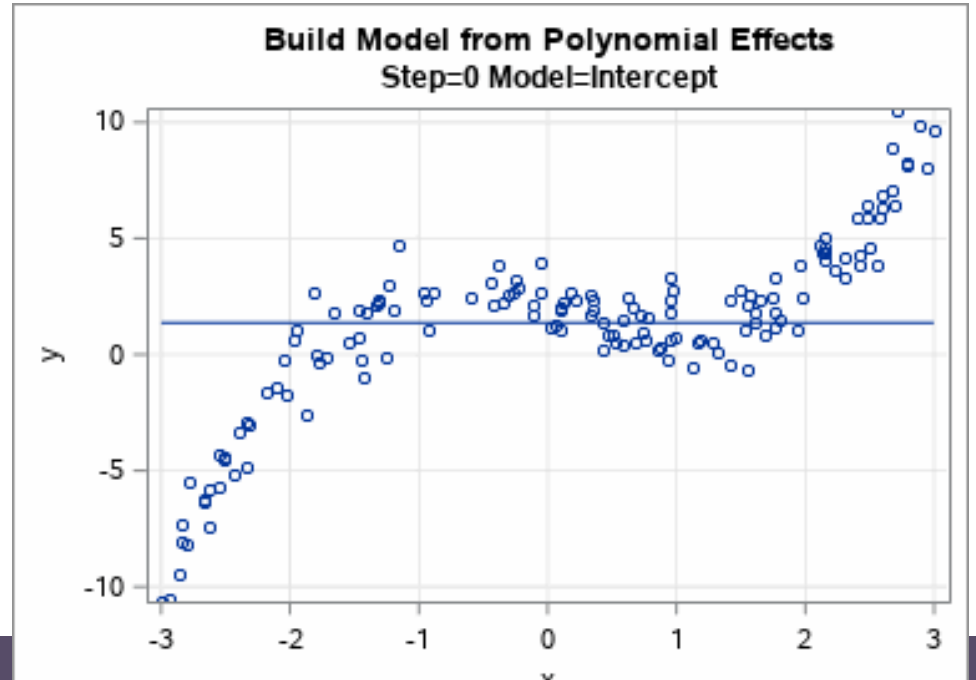
Puede haber regresiones lineales o polinomiales, depende de si los datos son lineales o no respectivamente.

$$y = a_0 + a_1x_1 + a_2x_2 + a_{3a_3x^3}x_3 + \ldots \rightarrow y$$
$$= a_0 + a_1x + a_2x^2 +$$

Lineal



Polinomial



The Normal Equation

To find the value of $\boldsymbol{\theta}$ that minimizes the MSE, there exists a *closed-form solution*—in other words, a mathematical equation that gives the result directly. This is called the *Normal equation* (Equation 4-4).

Equation 4-4. Normal equation

$$\widehat{\boldsymbol{\theta}} = (\mathbf{X}^{\intercal}\mathbf{X})^{-1}\,\mathbf{X}^{\intercal}\,\mathbf{y}$$

In this equation:

- $\widehat{\boldsymbol{\theta}}$ is the value of $\boldsymbol{\theta}$ that minimizes the cost function.
- $\mathbf{y}$ is the vector of target values containing $y^{(1)}$ to $y^{(m)}$.

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*.
" O'Reilly Media, Inc.".

Notice that Scikit-Learn separates the bias term (`intercept_`) from the feature weights (`coef_`). The `LinearRegression` class is based on the `scipy.linalg.lstsq()` function (the name stands for "least squares"), which you could call directly:

```
>>> theta_best_svd, residuals, rank, s = np.linalg.lstsq(X_b, y, rcon
>>> theta_best_svd
array([[4.21509616],
       [2.77011339]])
```

This function computes $\widehat{\boldsymbol{\theta}} = \mathbf{X}^{+}\mathbf{y}$, where $\mathbf{X}^{+}$ is the *pseudoinverse* of $\mathbf{X}$ (specifically, the Moore–Penrose inverse). You can use `np.linalg.pinv()` to compute the pseudoinverse directly:

```
>>> np.linalg.pinv(X_b) @ y
array([[4.21509616],
       [2.77011339]])
```

The pseudoinverse itself is computed using a standard matrix factorization technique called *singular value decomposition* (SVD) that can decompose the training set matrix $\mathbf{X}$ into the matrix multiplication of three matrices $\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{\intercal}$ (see `numpy.linalg.svd()`). The pseudoinverse is computed as $\mathbf{X}^{+} = \mathbf{V}\mathbf{\Sigma}^{+}\mathbf{U}^{\intercal}$. To compute the matrix $\mathbf{\Sigma}^{+}$, the algorithm takes $\mathbf{\Sigma}$ and sets to zero all values smaller than a tiny threshold value, then it replaces all the nonzero values with their inverse, and finally it transposes the resulting matrix. This approach is more efficient than computing the Normal equation, plus it handles edge cases nicely: indeed, the Normal equation may not work if the matrix $\mathbf{X}^{\intercal}\mathbf{X}$ is not invertible (i.e., singular), such as if $m < n$ or if some features are redundant, but the pseudoinverse is always defined.

La clase de Linear Regression que es la que mandamos llamar se basa en los mínimos cuadrados (least squares), donde se calcula la pseudoinversa.

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.".

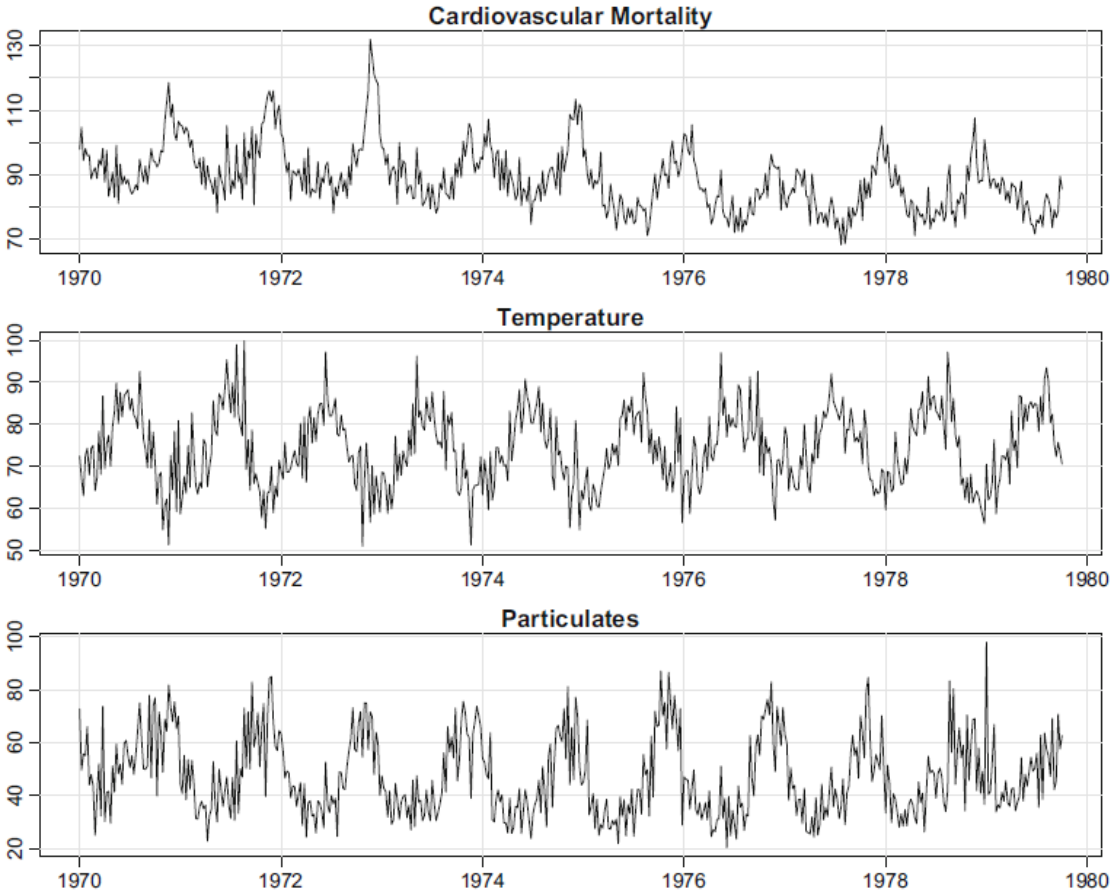
REGRESIÓN MÚLTIPLE

Cuando hay dos o más variables predictoras, el modelo se denomina modelo de regresión múltiple. La forma general de un modelo de regresión múltiple es

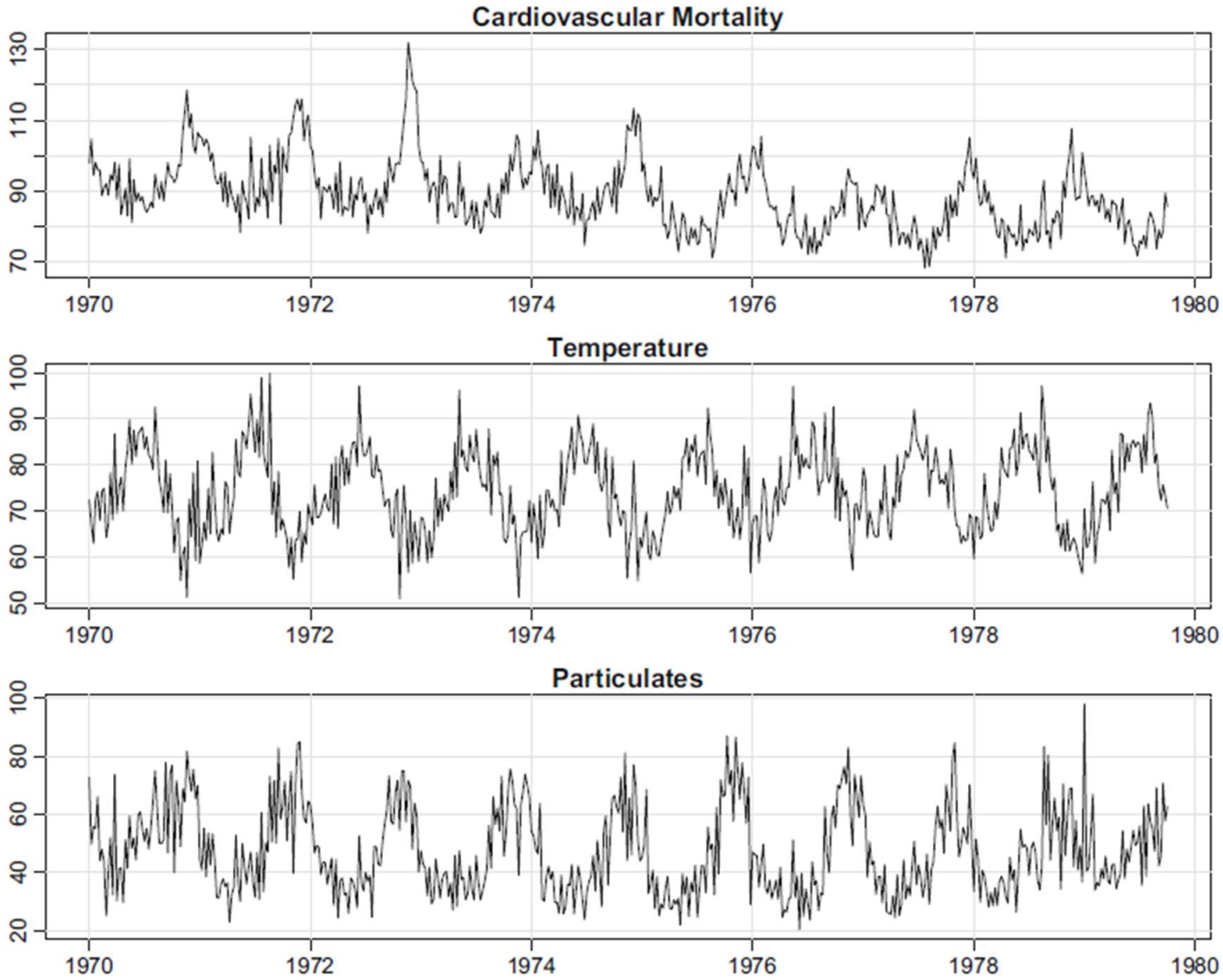
$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t,$$

donde y es la variable a pronosticar y x1,…, x son las k variables predictoras. Cada una de las variables predictoras debe ser numérica.

Los datos que se muestran son series extraídas de un estudio de Shumway de los posibles efectos de la temperatura y la contaminación sobre la mortalidad semanal en Los Ángeles.



$$M_t = \beta_1(T_t - \overline{T}) + \beta_2 P_t + \varepsilon$$


Cardiovascular Mortality
130
110
90
70
1970
1972
1974
1976
1978
1980
Temperature
100
90
80
70
60
50
1970
1972
1974
1976
1978
1980
Particulates
100
80
60
40
20
1970
1972
1974
1976
1978
1980

Notebooks

- 1) Regresión lineal y polinomial
- https://colab.research.google.com/drive/1yMk887XcxVsdgdkl--sUO2y_k2m9cB1s?usp=sharing

- Penalizaciones
- https://drive.google.com/file/d/1lgzTIzDoVBwYXVIzr3wuNNAlbV30Ymwl/view?usp=sharing
- (si aparece solo texto darle en la opción de abrir con google colab)