Módulo II. Métodos de Machine Learning

Laboratorio Nacional de Informática Avanzada A.C.

CURSO:

Taller de Machine Learning para el análisis y visualización en Power BI

Laboratorio de Inteligencia Artificial

Tema 02 – Aprendizaje Supervisado

Profesor: Saúl Domínguez Isidro, PhD.
Contacto: saul.dominguez@lania.edu.mx

Objetivo

Entender los principales conceptos y algoritmos de aprendizaje supervisado

Contenido

- Introducción
- Técnicas de regresión
- Generalización de modelos
- Técnicas de clasificación

Aprendizaje Automático

- Involucra el aprendizaje de una función a partir de un conjunto de datos
  - La función mapea una variable **x** (la cual puede ser un vector) a una variable **y**
  - El conjunto de entrenamiento (training set) es un conjunto de valores de pares (x, y)

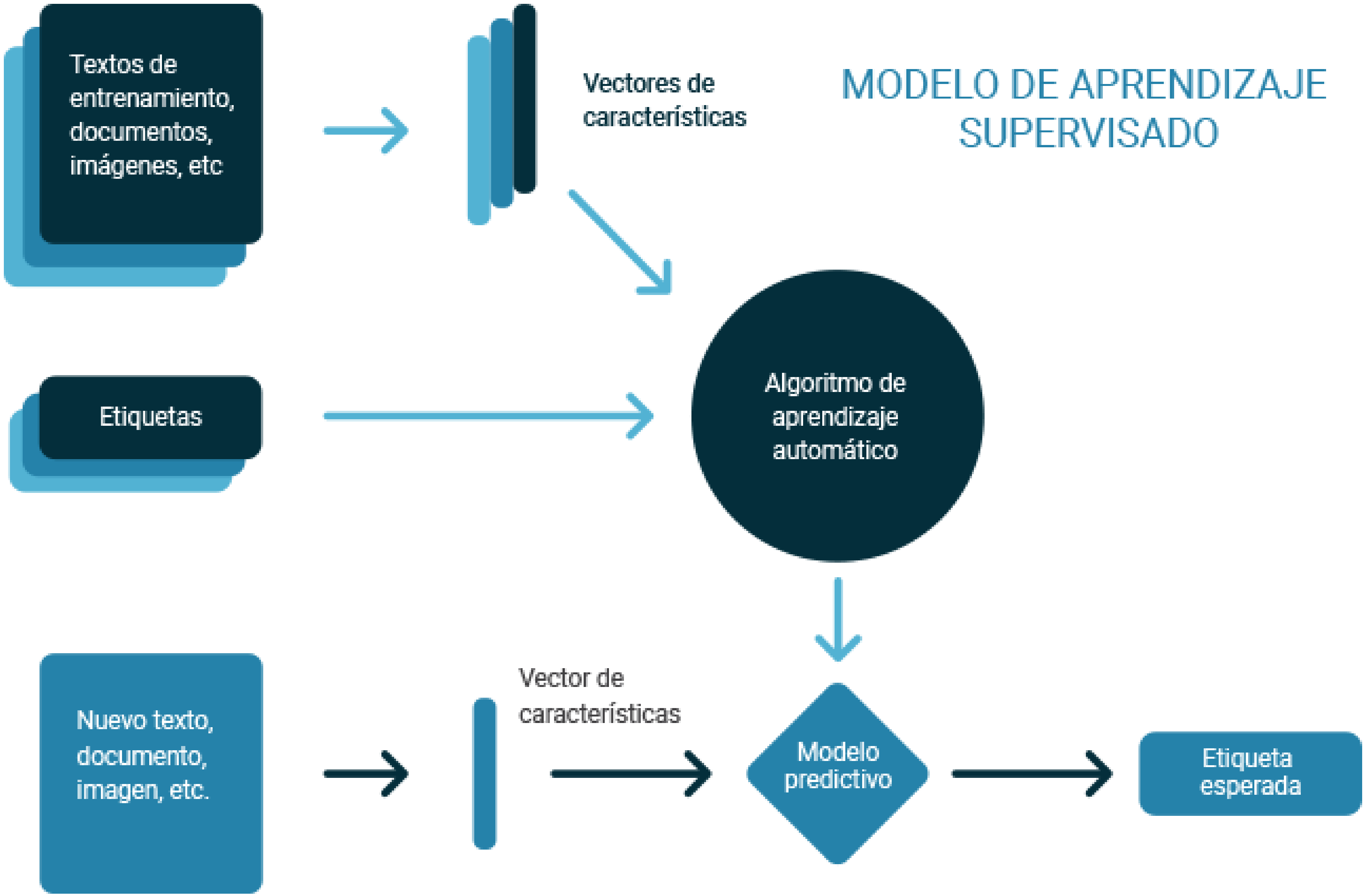
  - Las variables x son llamados predictores, o atributos
  - Las variables y son llamados objetivos (targets), o etiquetas

Aprendizaje Supervisado

Se tienen variables de entrada y salida y se usa un algoritmo para aprender la asignación de la variable de entrada a la variable de salida.

$$Y = f(X)$$

El objetivo es comprender muy bien el mapeo, y eso permite transformar la entrada en la salida. Cuando nuestro programa recibe los datos de entrada, ejecuta la función de mapeo para generar datos de salida.



Tomado de: https://cdn-images-1.medium.com/max/1600/1*0_fuxcGj6FL0Yqsu0rt4gQ.png

Etiquetas

Una etiqueta es el valor que estamos prediciendo.

La etiqueta podría ser el precio futuro del trigo, el tipo de animal que se muestra en una imagen, el significado de un clip de audio o simplemente cualquier cosa.

Atributos

- Un atributo es una variable de entrada

- Un proyecto de aprendizaje automático simple podría usar un solo atributo, mientras que otro más sofisticado podría usar millones de atributos, especificados como:

$$x_1, x_2, \ldots, \mathrm{x}_n$$

Atributos categóricos y continuos

- Aunque la distinción entre diferentes categorías de variables puede ser importante en algunos casos, muchos sistemas prácticos de minería de datos dividen los atributos en solo dos tipos:

- **categóricos**: correspondiente a variables nominales, binarias y ordinales
- **continuos**: correspondiente a las variables enteras, de escala de intervalos y proporcionales.

- Un modelo de **regresión** predice valores continuos. Por ejemplo, los modelos de regresión hacen predicciones que responden a preguntas como las siguientes:
  - ¿Cuál es el valor de una casa en CDMX?
  - ¿Cuál es la probabilidad de que un usuario haga clic en este anuncio?

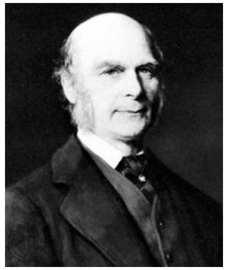
- Un modelo de **clasificación** predice valores categóricos. Por ejemplo, los modelos de clasificación hacen predicciones que responden a preguntas como las siguientes:
  - ¿Un mensaje de correo electrónico determinado es spam o no es spam?
  - ¿Esta imagen es de un perro, un gato o un hámster?

Regresión Lineal

La regresión es una técnica estadística estándar para realizar un aprendizaje supervisado cuando las variables son usualmente, pero no siempre, continuas.

No fue desarrollado por la comunidad de inteligencia artificial, sino que tiene sus raíces en Francis Galton en el siglo XIX.

Se usa para investigar la relación funcional entre dos o más variables, ajustando algún modelo matemático.



Regresión lineal simple

- En la regresión lineal simple, suponemos que tenemos una variable aleatoria **independiente** X y una variable aleatoria **dependiente** Y de tal manera que:

$$y = \mathrm{B}_0 + \mathrm{B}_1 x + \epsilon_x$$

Donde ϵ_x es una variable aleatoria que depende del valor x de X, con las siguientes propiedades:

- Para cada valor x de X, ϵ_x se distribuye normalmente con media 0
- Para cada valor x de X, ϵ_x tiene la misma desviación estándar σ
- Las variables aleatorias ϵ_x para todas las x son mutuamente independientes

- Estos supuestos implican que el valor esperado de Y dado un valor x de X viene dado por:

$$E(Y|X = x) \ = \mathrm{B}_0 + \mathrm{B}_1 x$$

- La idea es que el valor esperado de Y es una función lineal determinista de x.
- Sin embargo, el valor real y de Y no está determinado únicamente por el valor de X debido a un término de error aleatorio ϵ_x.

- Una vez que hacemos estas suposiciones sobre dos variables aleatorias, usamos una regresión lineal simple para tratar de descubrir la relación lineal a partir de una muestra aleatoria de valores de X y Y.

- En el caso de múltiples variables:

$$y = \mathrm{B}_0 + \mathrm{B}_1 x_1 + \mathrm{B}_2 x_2 + \cdots + \mathrm{B}_k x_k + \epsilon_x$$

Ejemplo

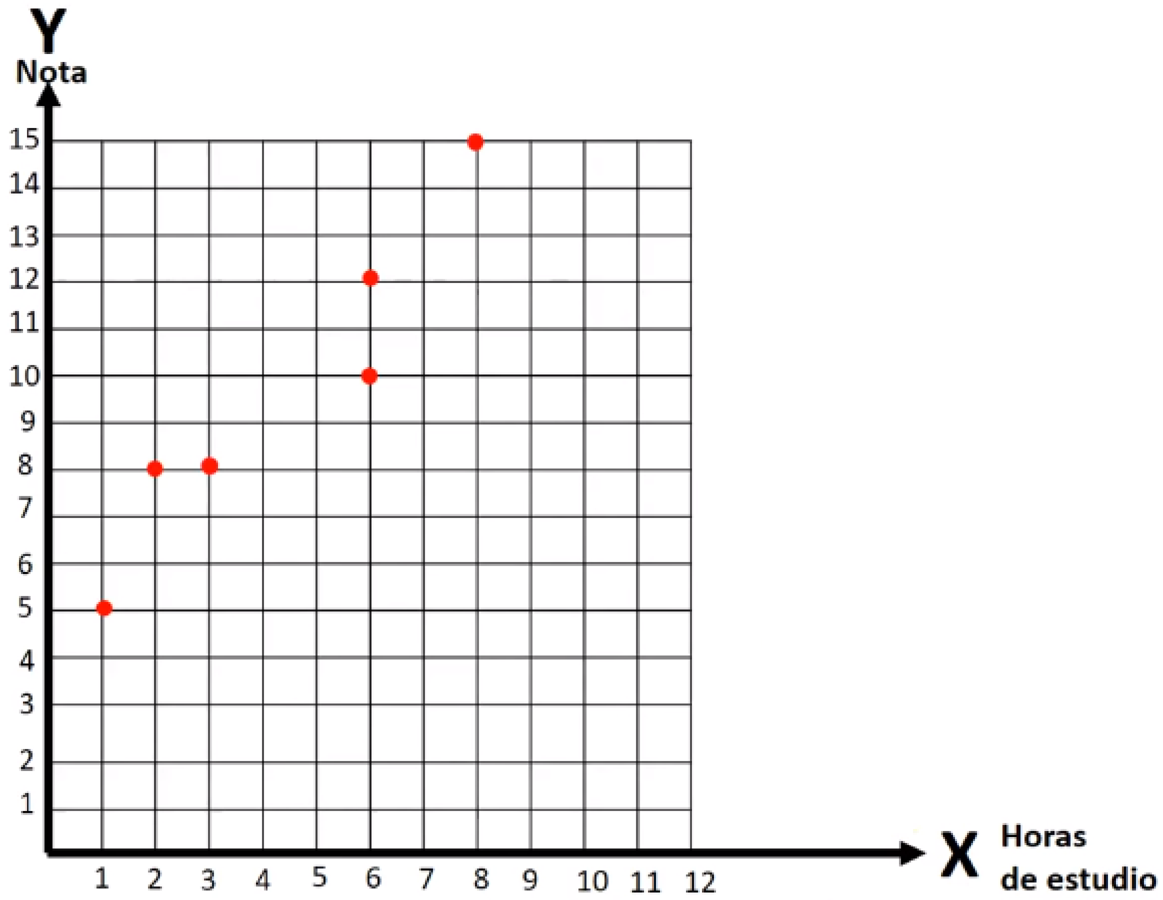
- Encontrar la relación entre las horas de estudio y las notas obtenidas :
  - Training set consiste de pares (x, y), donde:
    - x ← Horas de estudio
    - y ← Nota obtenida

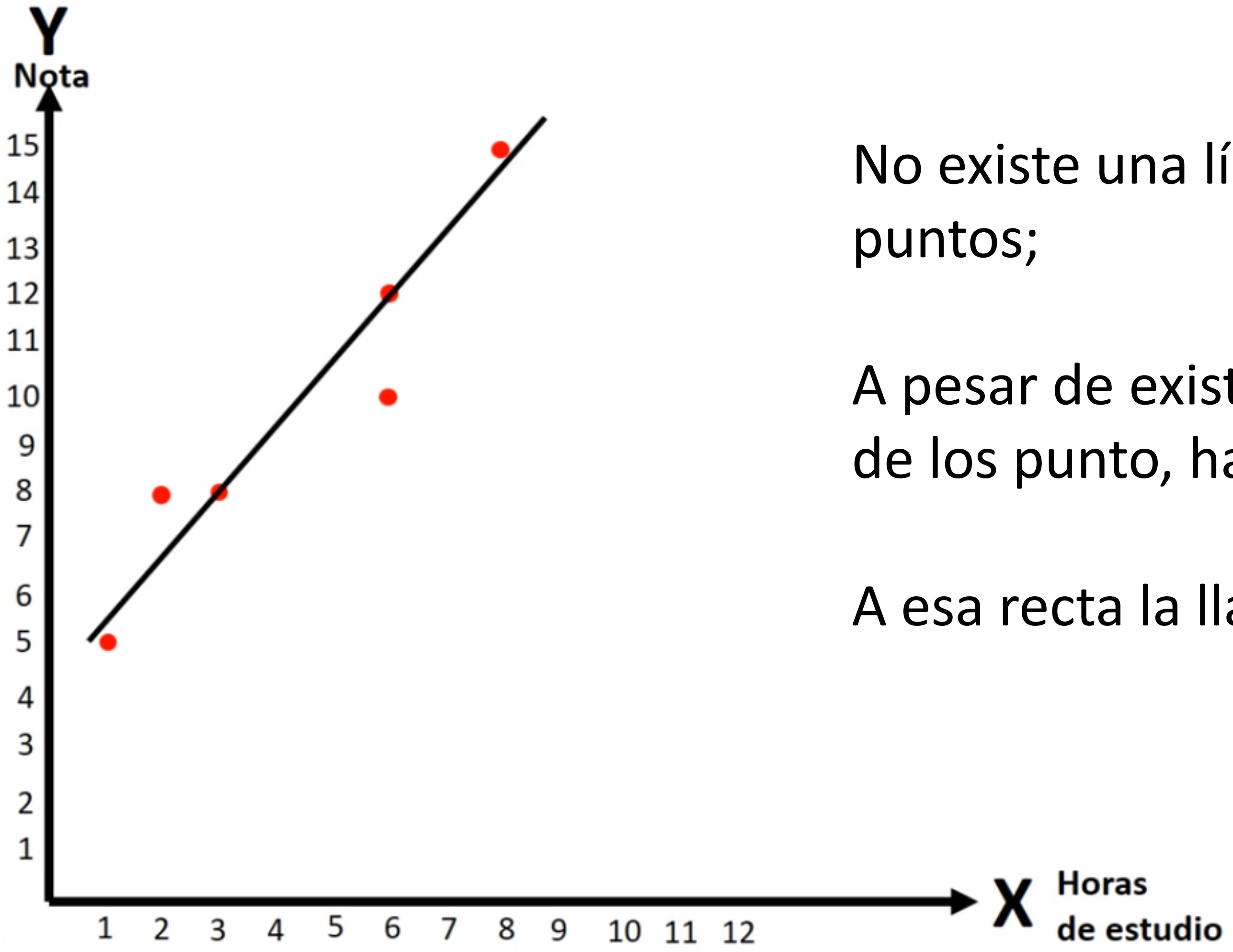
horas-estudio	nota
3	8
6	10
8	15
2	8
1	5
6	12

- Para estimar los valores de B_0 y B_1, encontramos los valores de b_0 y b_1 que minimizan el error cuadrático medio (MSE), que es

$$\frac{\sum_{i=1}^{n}[y_i - \overbrace{(b_0 + b_1 x_i)}^{\text{Predicción}}]^2}{n}$$

horas-estudio	nota
3	8
6	10
8	15
2	8
1	5
6	12



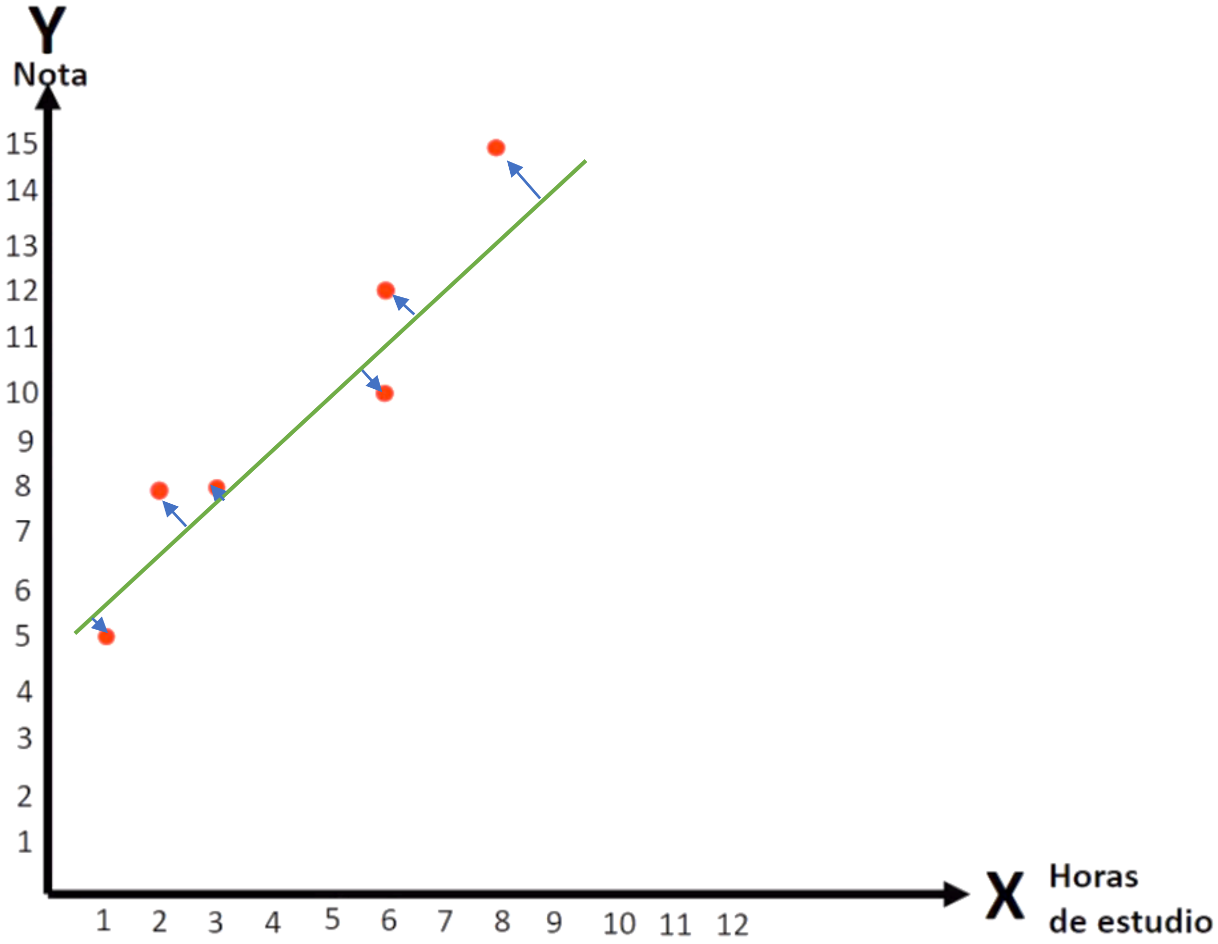


No existe una línea recta que pase por todos los puntos;

A pesar de existir diferentes líneas que pasen cerca de los punto, habrá una que es la mejor de todas.

A esa recta la llamaremos ecuación estimada $\hat{y}$

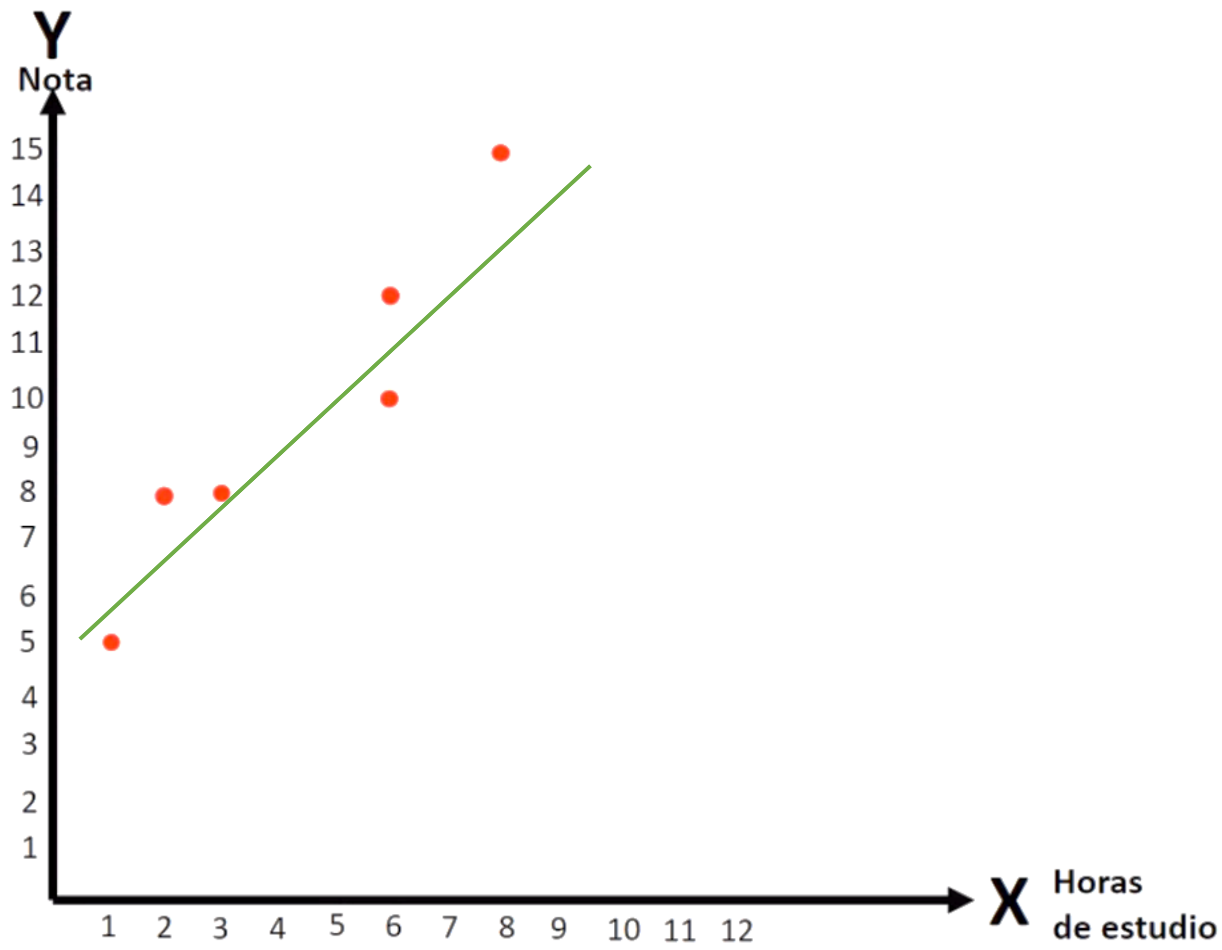
Ecuación estimada



$$\hat{y} = b_0 + b_1 x$$

Es necesario calcular los valore b0 y b1 con el método de mínimos cuadrados

Ecuación estimada

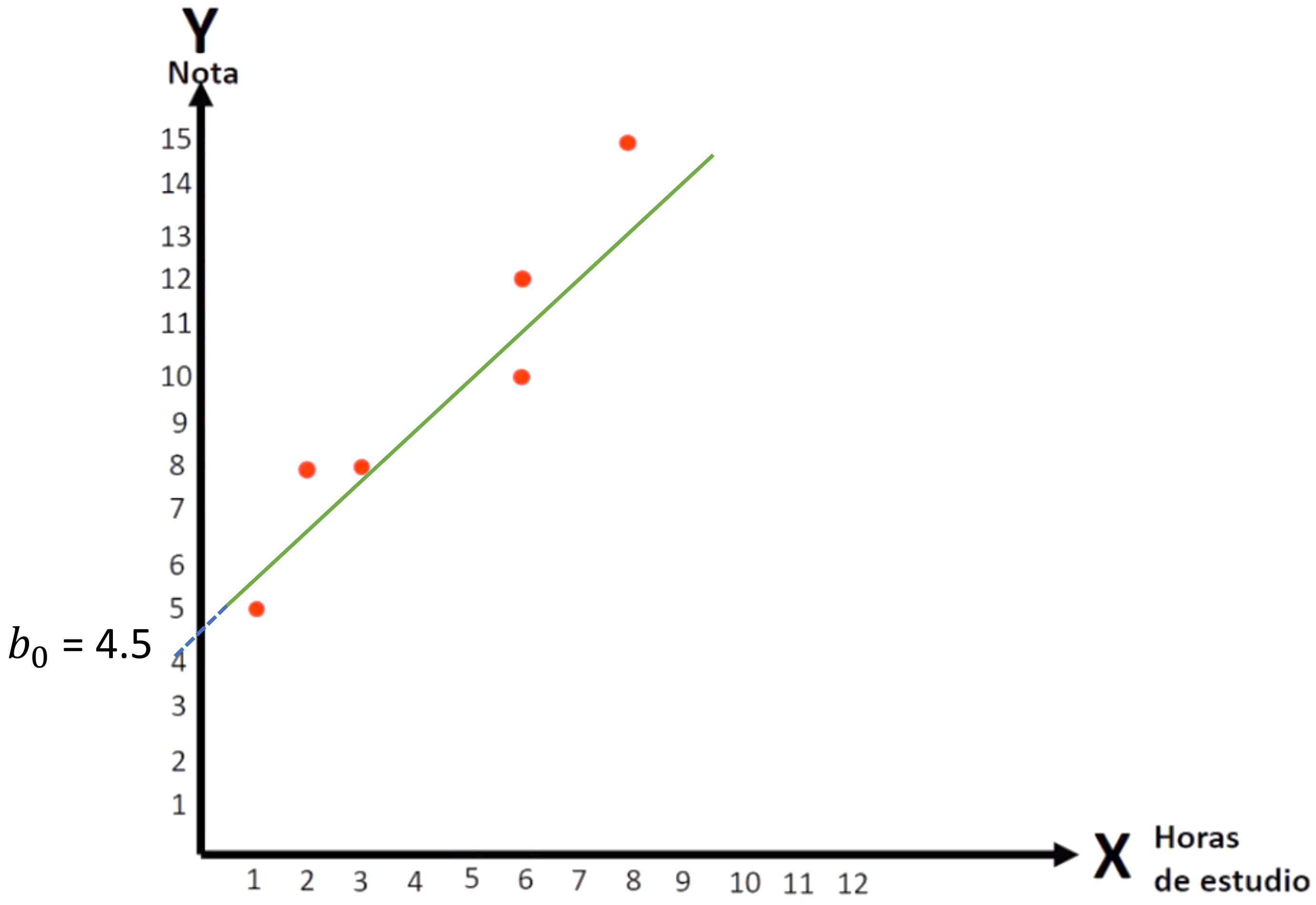


$$\hat{y} = b_0 + b_1 x$$

Intercepto

Pendiente

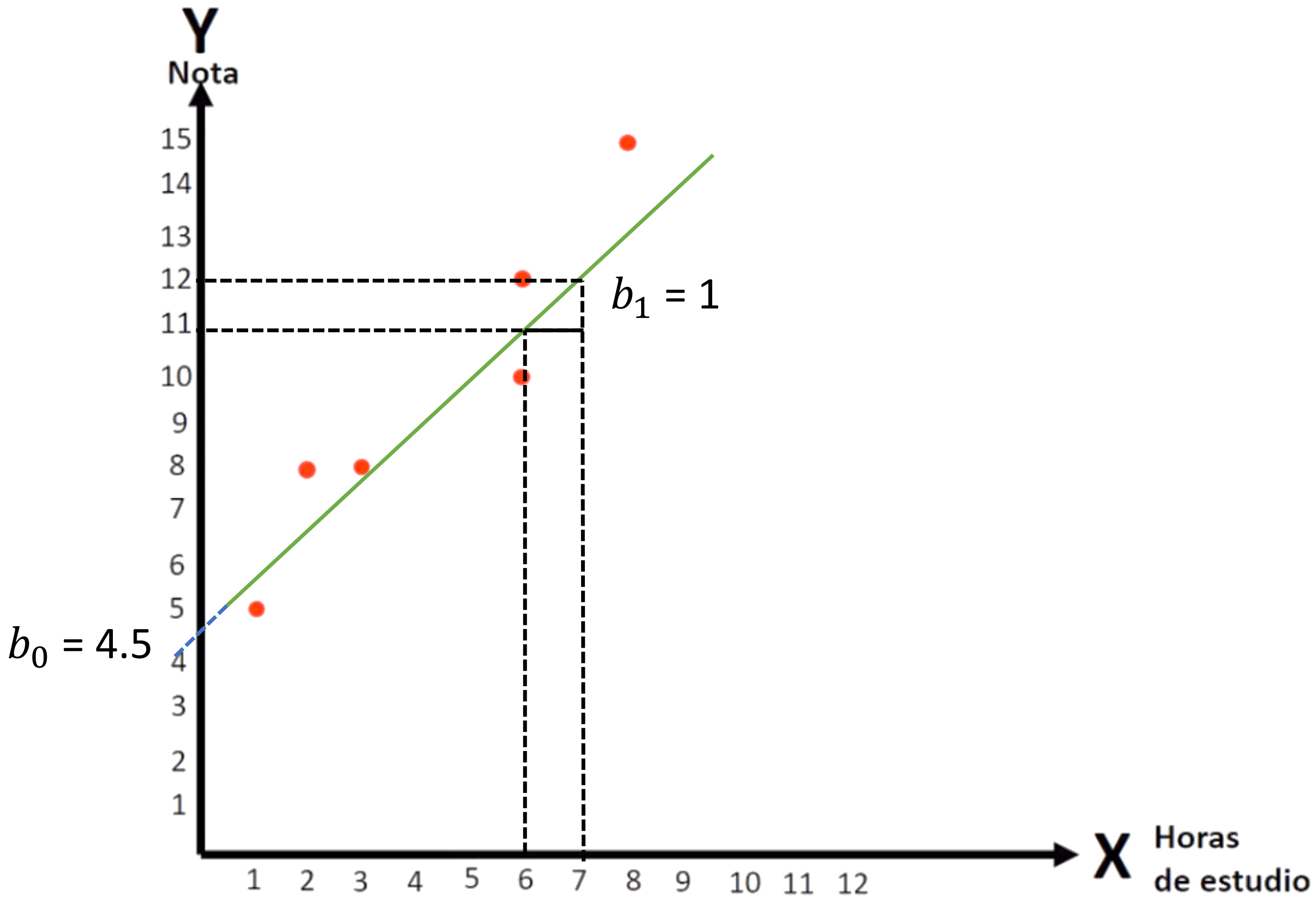
Ecuación estimada



$$\hat{y} = b_0 + b_1 x$$

Intercepto

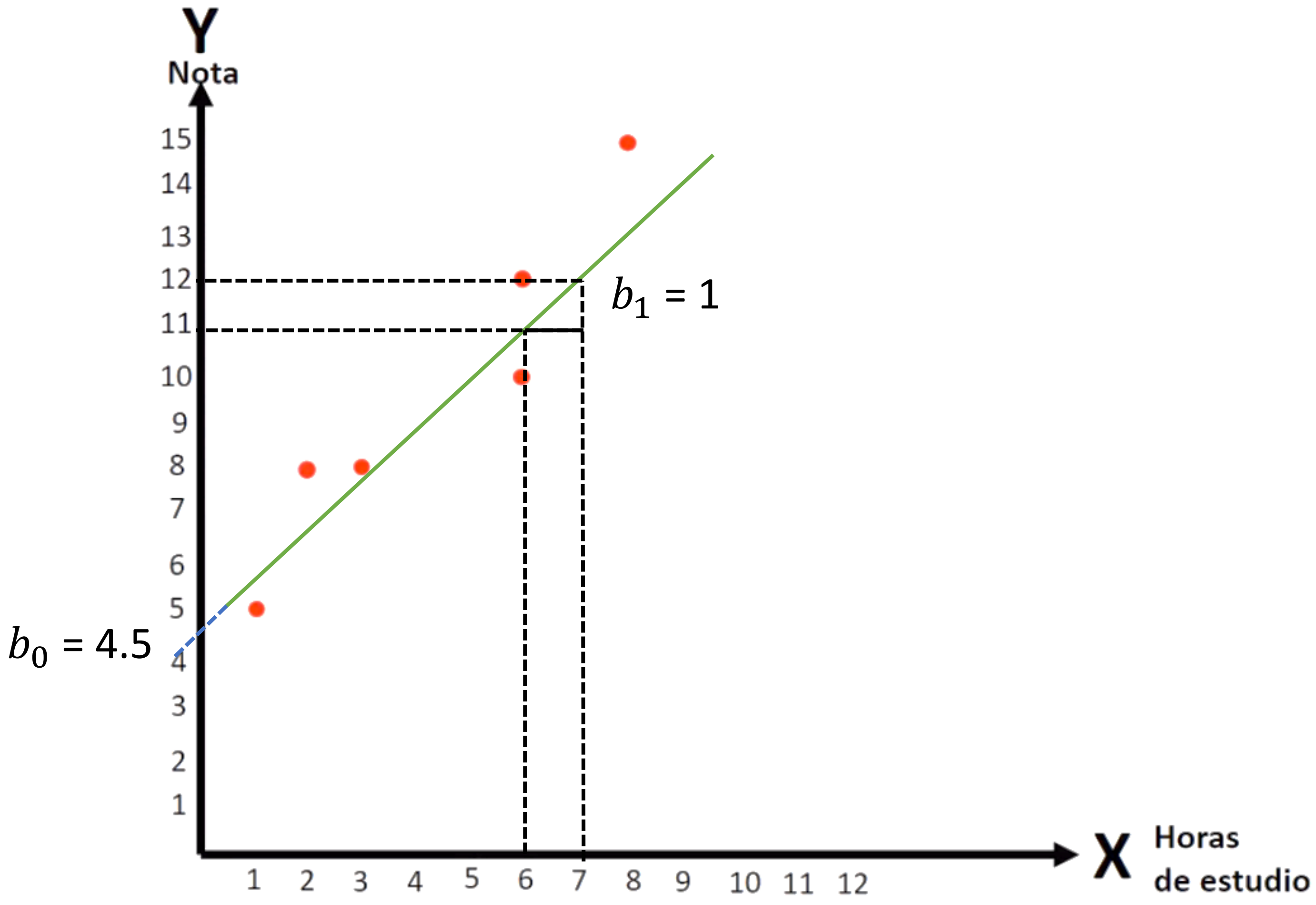
Ecuación estimada



$$\hat{y} = b_0 + b_1 x$$

Pendiente

Ecuación estimada



$$\hat{y} = b_0 + b_1 x$$

$$\hat{y} = 4.5 + 1x$$

Estimación de parámetros

- La **función de pérdida**, en cierto sentido mide la diferencia en un valor observado y del resultado y la estimación $\hat{y}$ obtenida del modelo. En el caso de regresión lineal, establecemos:

$$Loss(y, \hat{y}) = (y - \hat{y})^2 = (y - (b_0 + b_1 x))^2$$

- La suma de la función de pérdida sobre los datos observados se llama función de costo

$$Cost([y_1, \hat{y}_1], [y_2, \hat{y}_2], ..., [y_n, \hat{y}_n]) = \sum_{i=1}^{n} Loss(y_i, \hat{y}_i).$$

- En el caso de regresión lineal simple, tenemos

$$Cost([y_1, \hat{y}_1], [y_2, \hat{y}_2], ..., [y_n, \hat{y}_n]) = \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_i))^2$$

Estimación de los parámetros para la regresión lineal simple

- Los valores de b0 y b1 que minimizan la función de costo en la ecuación anterior, son los siguientes:

$$
\begin{aligned}
b_1 &= \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \\
b_0 &= \bar{y} - b\bar{x}.
\end{aligned}
$$

Laboratorio Nacional de Informática Avanzada A.C.

Descenso del gradiente

Algorithm 5.1 Gradient_Descent_z^2

Output: Value of z that minimizes z^2.

Function *Minimizing_Value*;
$z = arbitrary\_value$;
$\lambda = learning\_rate$;
repeat *number_iterations* times
    $z = z - \lambda \times 2z$;
endrepeat
return z;

LANIA
Laboratorio Nacional de Informática Avanzada A.C.

Derivadas parciales

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))^2,$$

$$\frac{\partial \sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))^2}{\partial b_0} = -2\sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))$$

$$\frac{\partial \sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))^2}{\partial b_1} = -2\sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))x_i.$$

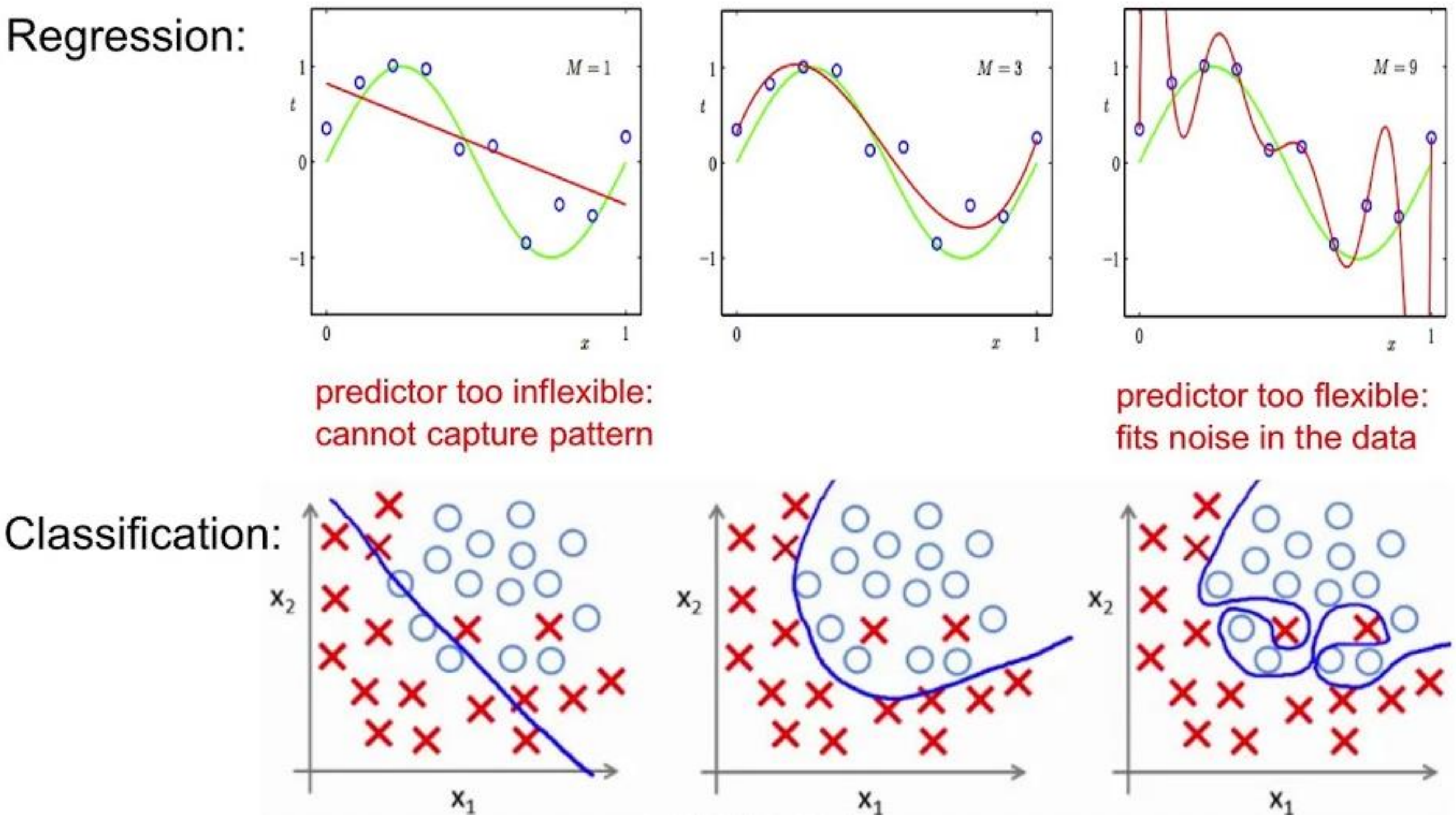
LANIA
Laboratorio Nacional de Informática Avanzada A.C.

Estimación de parámetros en Regresión Lineal

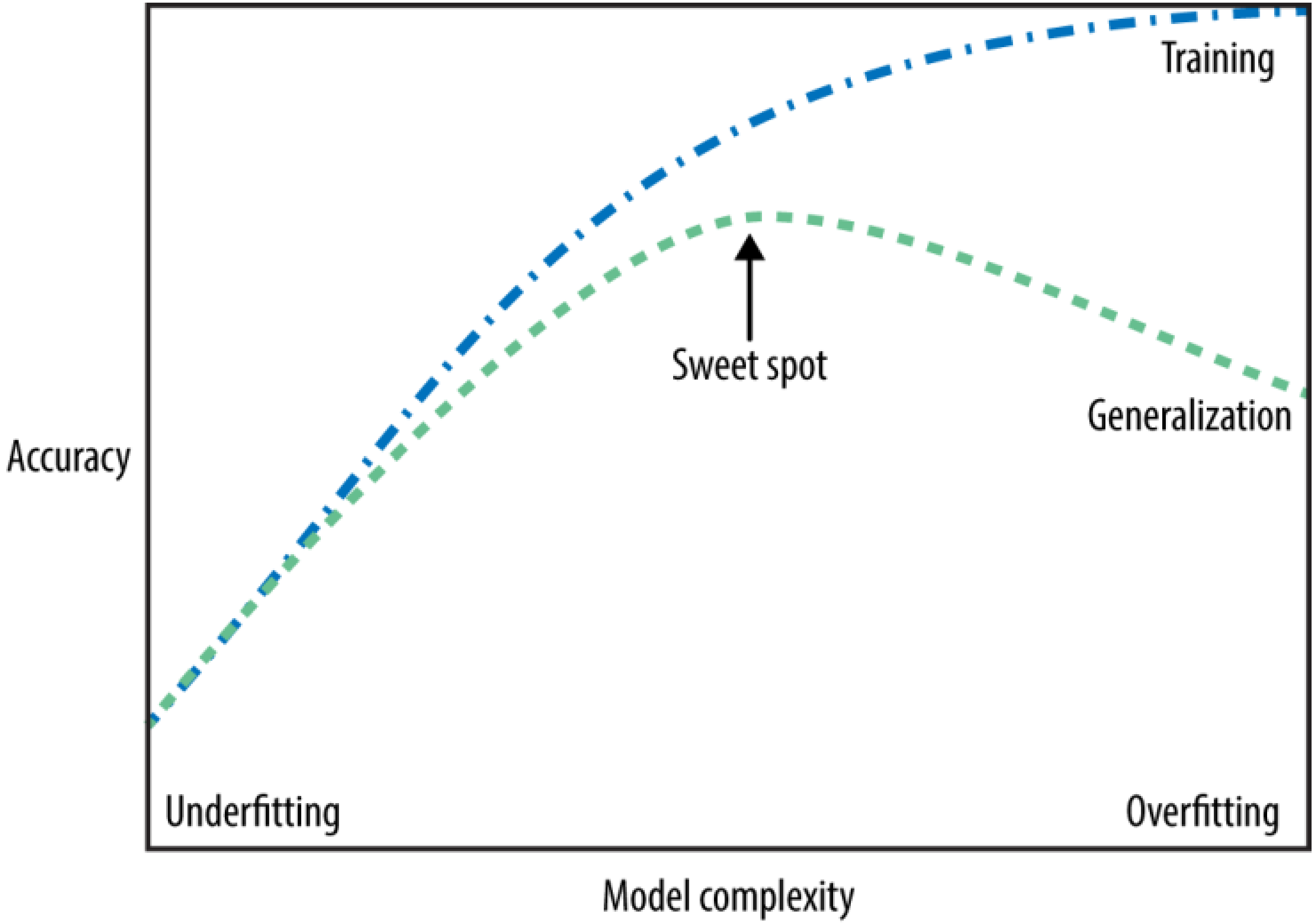
Algorithm 5.2 Gradient_Descent_Simple_Linear_Regression

Input: Set of real data $\{\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$.
Output: Values of b_0 and b_1 that minimize the cost function

Function *Minimizing_Values*;
$b0 = arbitrary\_value\_0$;
$b1 = arbitrary\_value\_1$;
$\lambda = learning\_rate$;
repeat *number_iterations* **times**
$\quad b_0\_gradient = -2\sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))$;
$\quad b_1\_gradient = -2\sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))x_i$;
$\quad b_0 = b_0 - \lambda \times b_0\_gradient$;
$\quad b_1 = b_1 - \lambda \times b_1\_gradient$;
endrepeat
return b_0, b_1;

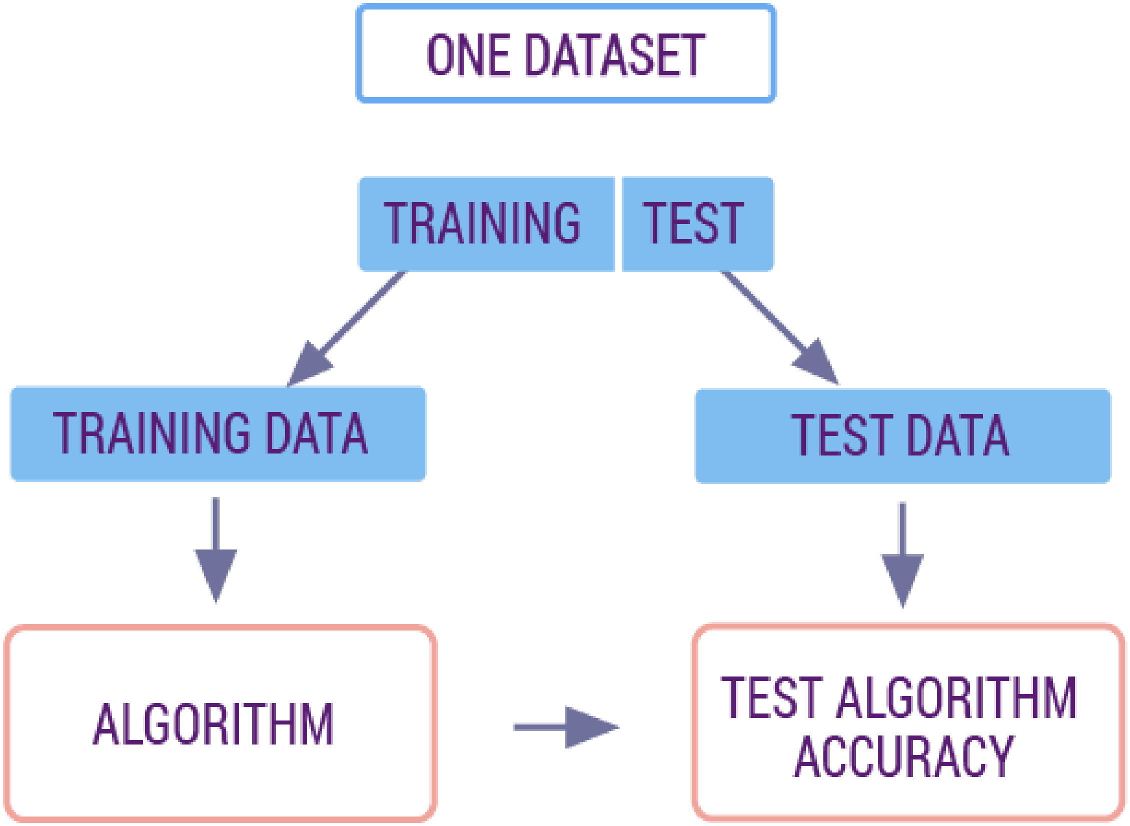


Tomado de: https://i.ytimg.com/vi/dBLZg-RqoLg/maxresdefault.jpg



Compensación de la complejidad del modelo contra la capacitación y la precisión de las pruebas

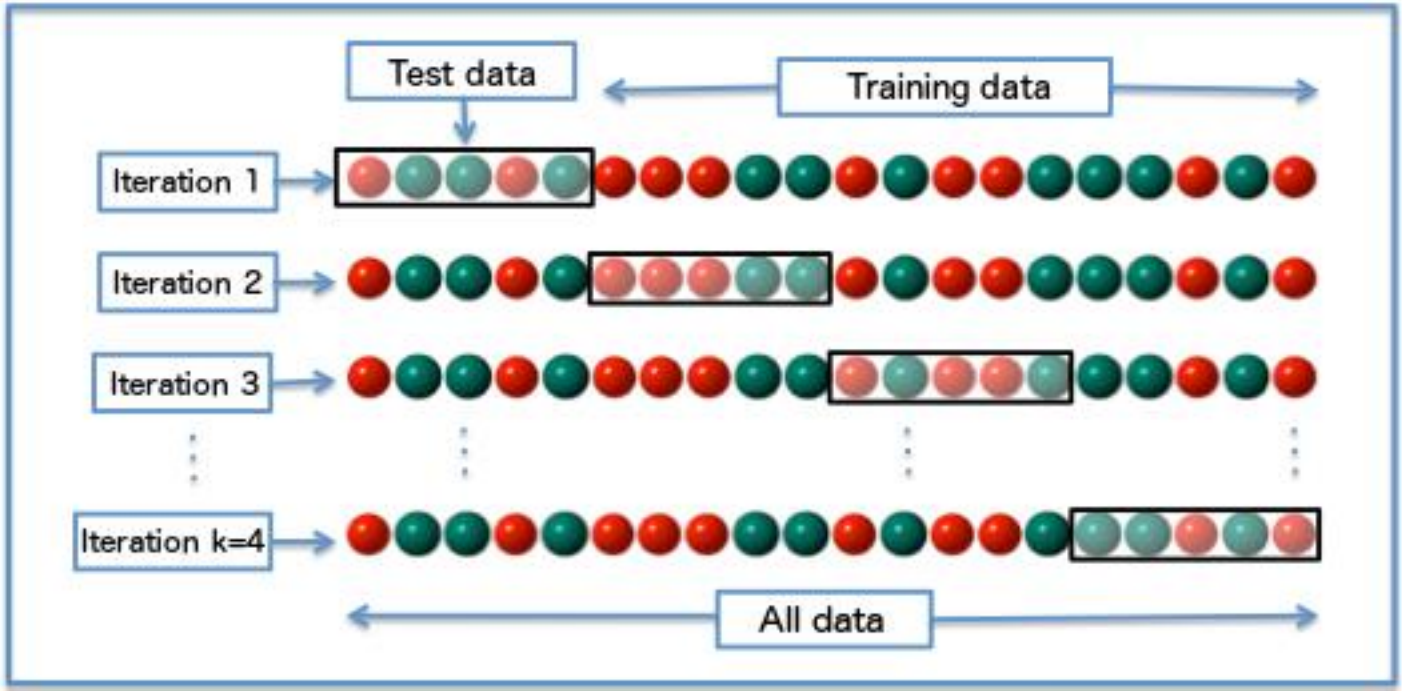
Entrenamiento y prueba



Tomado de: http://www.ikompass.edu.sg/wp-content/uploads/2015/08/training_test_accuracy.png

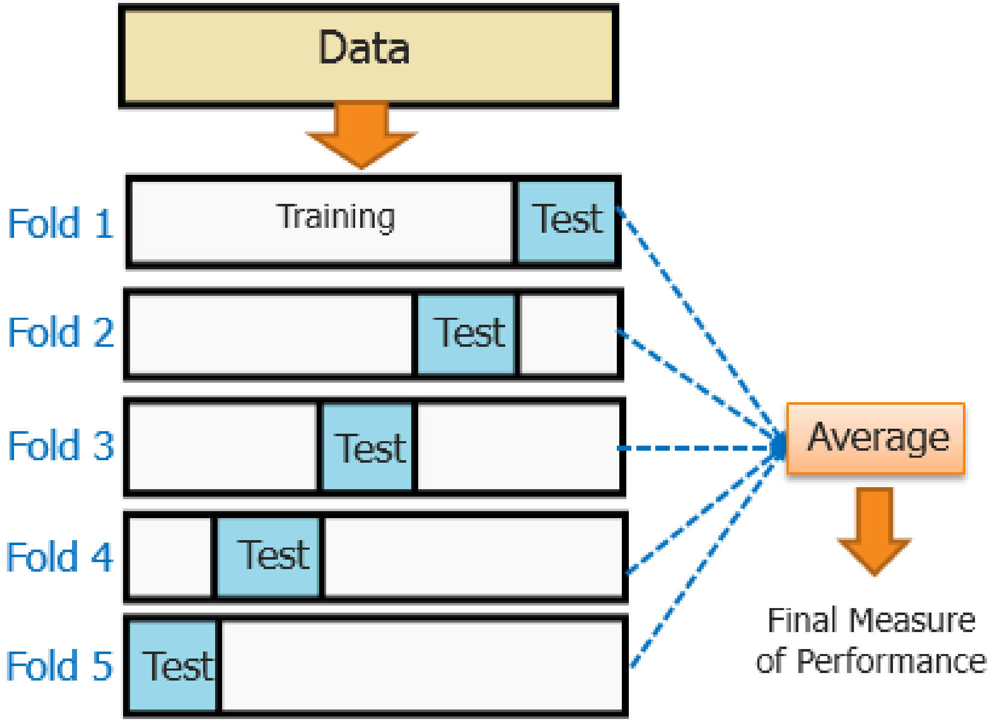
Validación cruzada

La validación cruzada es un procedimiento de remuestreo utilizado para evaluar modelos de aprendizaje automático en una muestra de datos limitada.



Tomada de: https://upload.wikimedia.org/wikipedia/commons/1/1c/K-fold_cross_validation_EN.jpg

K-fold cross validation



La validación cruzada se utiliza principalmente en el aprendizaje automático aplicado para estimar la habilidad de un modelo de aprendizaje automático en datos no vistos. Es decir, usar una muestra limitada para estimar cómo se espera que el modelo funcione en general cuando se usa para hacer predicciones sobre datos que no se usaron durante el entrenamiento del modelo.

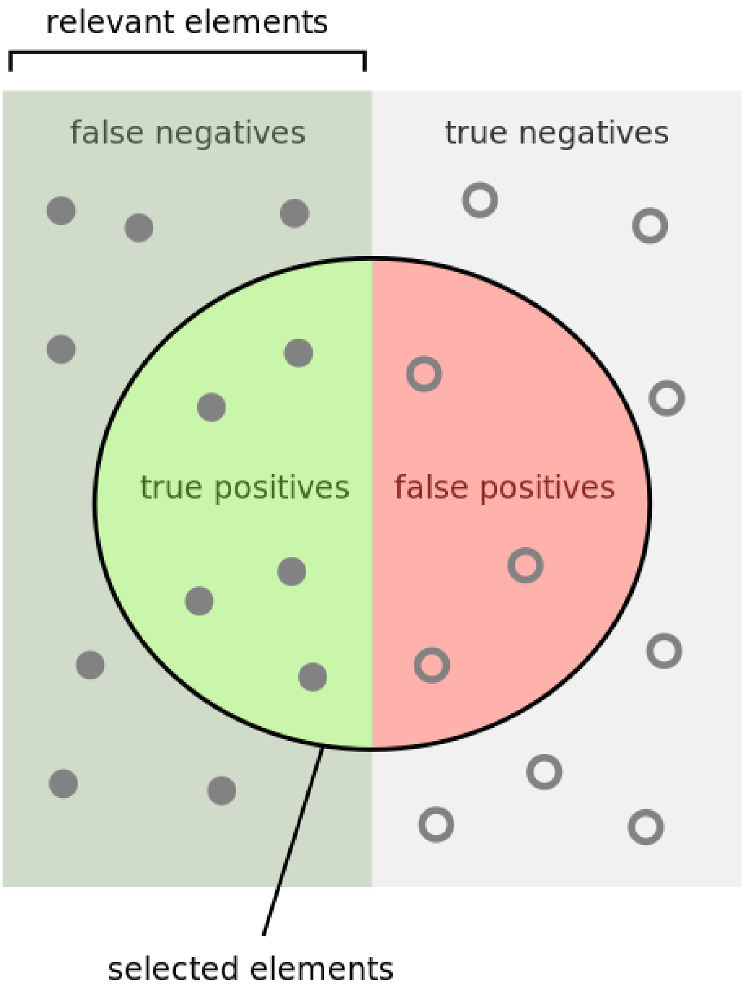
El procedimiento tiene un único parámetro llamado k que se refiere al número de grupos en los que se dividirá una muestra de datos determinada. Como tal, el procedimiento a menudo se llama validación cruzada k-fold. Cuando se elige un valor específico para k, se puede usar en lugar de k en la referencia al modelo, como k = 10 convirtiéndose en una validación cruzada de 10 veces.

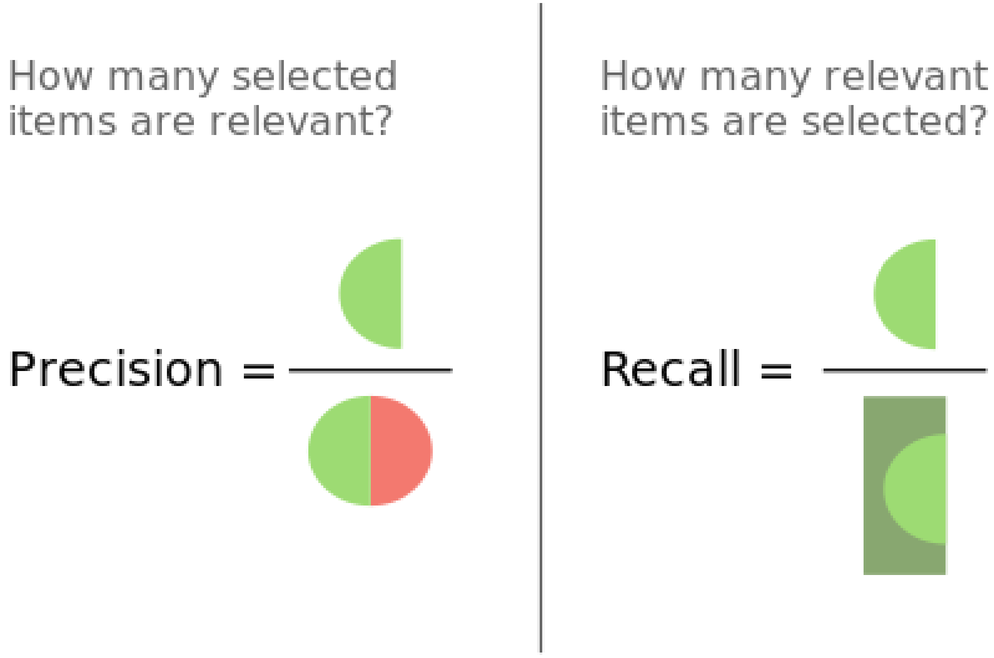
Tomado de: http://tomaszkacmajor.pl/wp-content/uploads/2016/05/cross-validation.png

Consideraciones en la evaluación

- TP Rate: tasa de verdaderos positivos (instancias clasificadas correctamente como una clase dada)
- FP Rate: tasa de falsos positivos (instancias clasificadas falsamente como una clase dada)
- Precisión: proporción de instancias que son verdaderamente de una clase dividida entre el total de instancias clasificadas como esa clase
- Exhaustividad (Recall): proporción de instancias clasificadas como una clase dada dividida entre el total real en esa clase (equivalente a TP Rate)
- Medida F: una medida combinada de precisión y recuperación calculada como

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$





$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Regresión logística

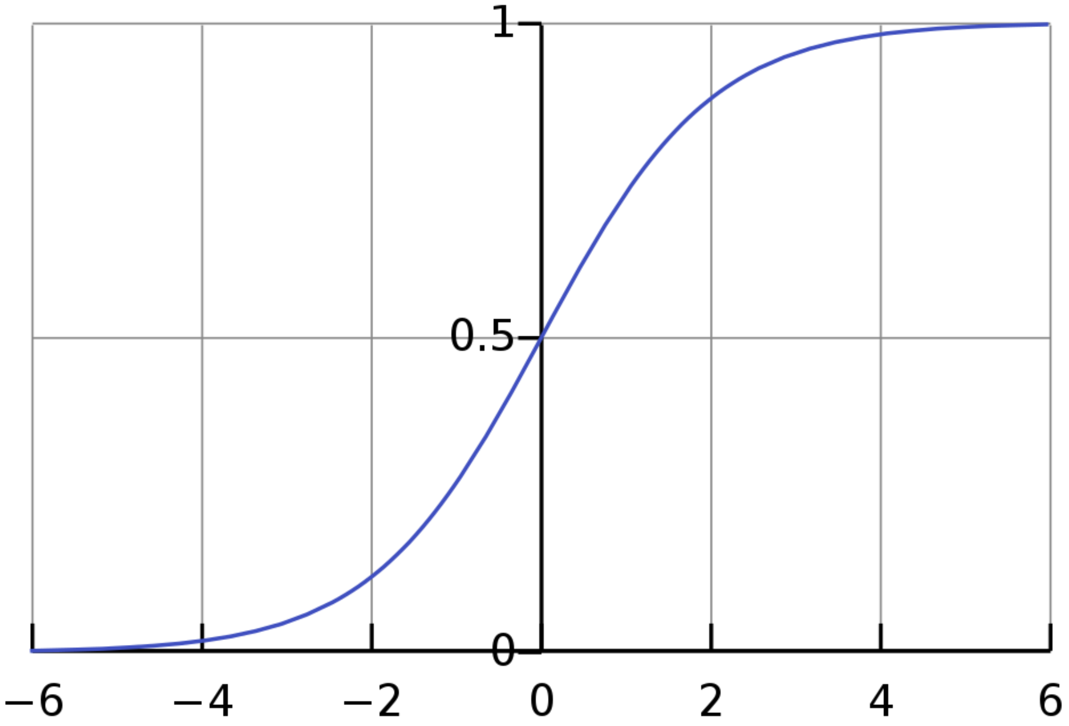
- La regresión logística es muy parecida a la regresión lineal, excepto que asigna una combinación lineal de las variables predictoras a la probabilidad de un resultado binario.

- Por ejemplo, podríamos mapear la edad, la altura, el índice de masa corporal y el nivel de glucosa a la probabilidad de que una persona tenga diabetes

Función sigmoide

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

A menudo la función sigmoide se refiere al caso particular de la función logística



Regresión logística

- Utiliza la función sigmoide para calcular el modelo de predicción

$$f(z) = \frac{\exp(z)}{1 + \exp(z)}.$$

El rango de la función sigmoidea es el intervalo (0,1). Se utiliza en regresión logística para proporcionar la probabilidad de un resultado binario de la siguiente manera:

$$P(Y = 1|x) = \frac{\exp(b_0 + b_1 x)}{1 + \exp(b_0 + b_1 x)}$$

$$P(Y = -1|x) = \frac{1}{1 + \exp(b_0 + b_1 x)}.$$

Regresión logística

Podemos desarrollar funciones de pérdida y costo para una regresión logística simple de la siguiente manera

$$P(Y = y|x) = \frac{\exp\left(y(b_0 + b_1 x)\right)}{1 + \exp(y(b_0 + b_1 x))}.$$

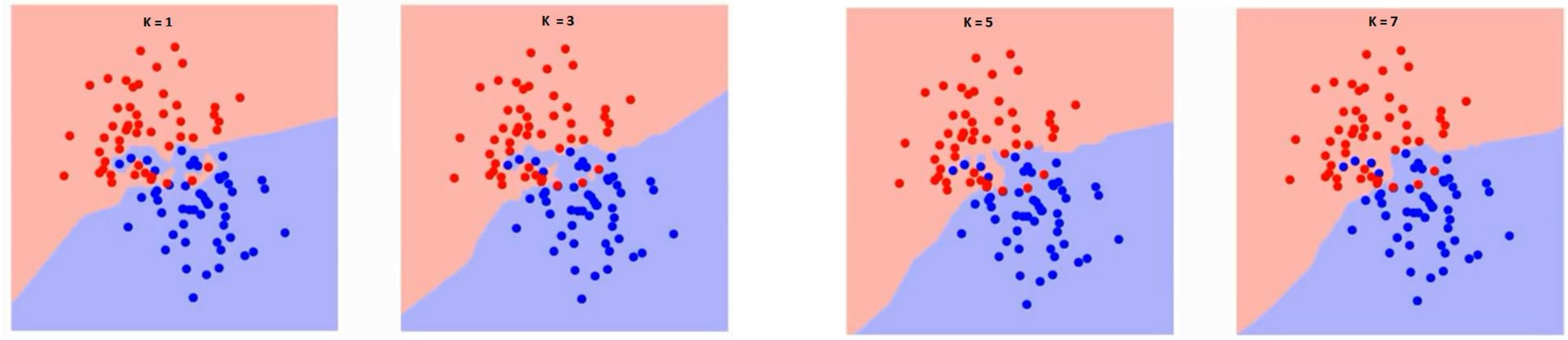
$$\sum_{i=1}^{n} \ln\left(\frac{1 + \exp\left(y_i(b_0 + b_1 x_i)\right)}{\exp(y_i\left(b_0 + b_1 x_i\right))}\right).$$

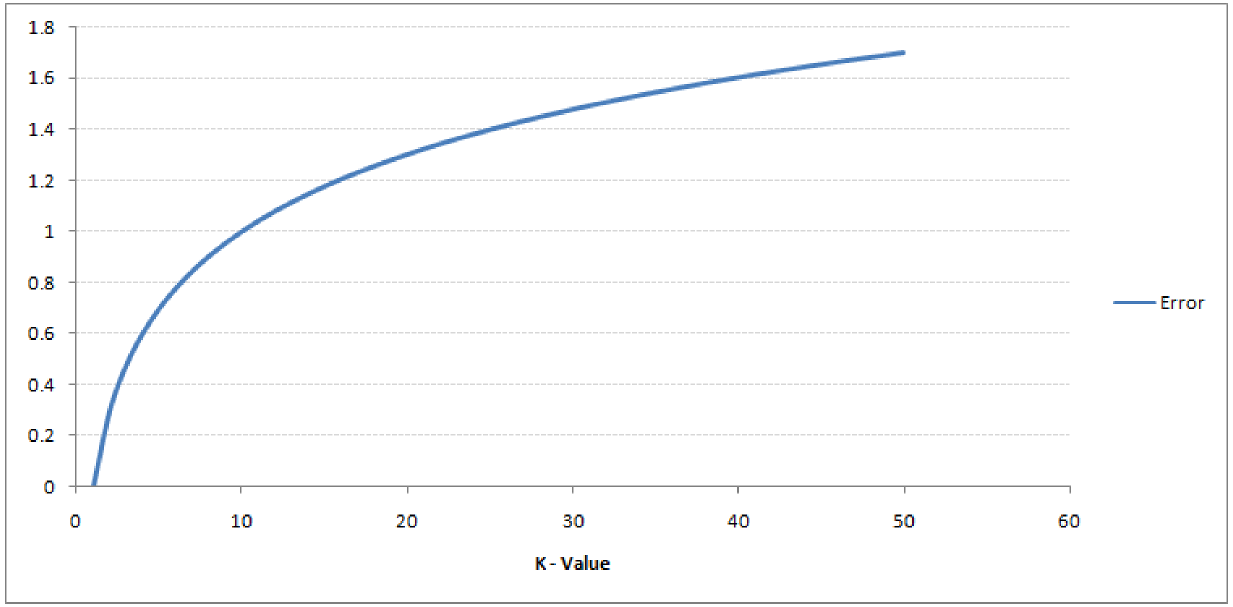
K vecinos más cercanos (k-NN)

- k-NN se basa en el aprendizaje por analogía, es decir, compara una instancia de prueba con instancias de entrenamiento que son similares a ella.
- k-NN representa cada instancia como un punto de datos en un espacio *p*-dimensional, donde *p* es el número de atributos o variables.
- Dada una instancia de prueba, se calcula su proximidad al resto de los puntos de datos en el conjunto de entrenamiento, utilizando alguna medida de distancia o similitud

k-NN

- Se dice que los k vecinos más cercanos de una instancia dada **z** se refiere a los *k* puntos más cercanos a **z**.
- El punto de datos se clasifica de acuerdo a las etiquetas de clase de sus vecinos.
- En el caso donde los vecinos tienen más de una etiqueta, el punto de datos se asigna a la clase mayoritaria de sus vecinos.
- En el caso donde se presenta un empate entre clases se puede elegir aleatoriamente una de ellas para clasificar el punto de datos.
- La asignación de la clase para un punto dado subraya la importancia de elegir adecuadamente el valor para *k*,
  - *Si k* es demasiado pequeño, entonces el clasificador puede ser susceptible al sobreajuste (overfitting).
  - si *k* es demasiado grande, el clasificador puede clasificar erróneamente la instancia de prueba,


K = 1
K = 3
K = 5
K = 7


1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
60
K - Value
Error


60
50
40
30
20
10
0
0
10
20
30
40
50
60
K - Value
Validation error

Laboratorio Nacional de Informática Avanzada A.C.

Algoritmo

Entrada: k número de vecinos, $\boldsymbol{D}_{train}$ conjunto de entrenamiento

1: **for** cada instancia de prueba $\mathbf{z} = (\mathbf{x}', c')$ **do**

2: Calcular $d(\mathbf{x}', \mathbf{x})$, la distancia entre $\mathbf{z}$ y cada instancia $(\mathbf{x}, c) \in \boldsymbol{D}_{train}$

3: Seleccionar $\boldsymbol{D}_z \subseteq \boldsymbol{D}$, el conjunto de k instancias de entrenamiento cercanas a $\mathbf{z}$

4: $c' = \text{máx}_v \sum_{(x_i, c_i) \in D} I(v = c_i)$

5: **end for**

Donde v es una etiqueta de clase, c' es la etiqueta de clase para uno de los vecinos más cercanos e I(.) es un indicador de función que devuelve 1 si su argumento es verdadero y 0 de lo contrario.

$$\text{Votación Mayoritaria: } c' = \underset{v}{\text{máx}} \sum_{(x_i, c_i) \in D_z} I(v = c_i)$$

Algoritmo

Entrada: k número de vecinos, $\boldsymbol{D}_{train}$ conjunto de entrenamiento

1: **for** cada instancia de prueba $\mathbf{z} = (\mathbf{x}', c')$ **do**
2: Calcular $d(\mathbf{x}', \mathbf{x})$, la distancia entre $\mathbf{z}$ y cada instancia $(\mathbf{x}, c) \in \boldsymbol{D}_{train}$
3: Seleccionar $\boldsymbol{D}_z \subseteq \boldsymbol{D}$, el conjunto de k instancias de entrenamiento cercanas a $\mathbf{z}$
4: $c' = \text{máx}_v \sum_{(x_i, c_i) \in D} I(v = c_i)$
5: **end for**

Donde v es una etiqueta de clase, c' es la etiqueta de clase para uno de los vecinos más cercanos e I(.) es un indicador de función que devuelve 1 si su argumento es verdadero y 0 de lo contrario.

$$\text{Votación Mayoritaria: } c' = \underset{v}{\text{máx}} \sum_{(x_i, c_i) \in D_z} I(v = c_i)$$

Árboles de decisión

Los Arboles de Decisión son diagramas con construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema.

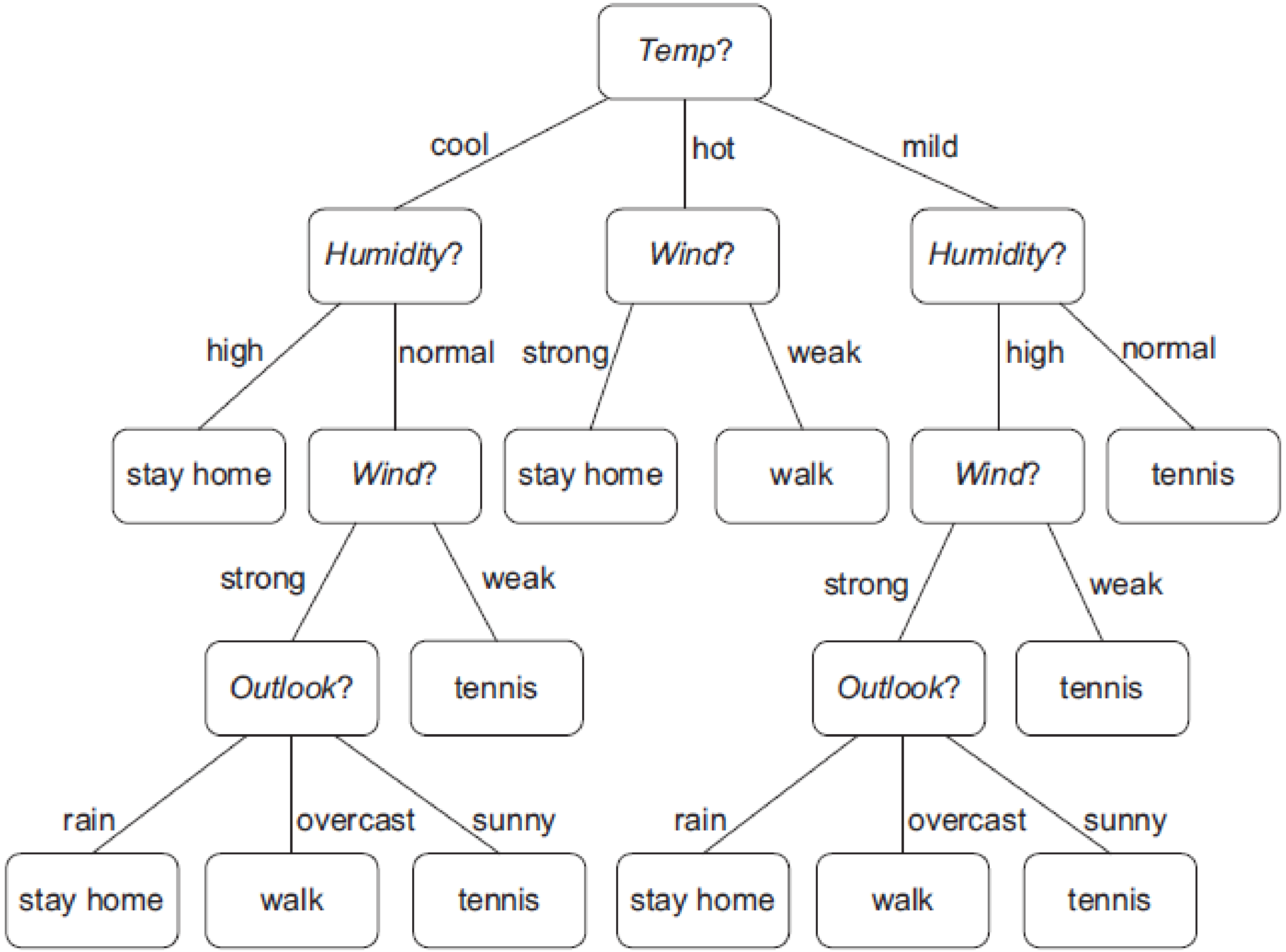
Los Arboles de Decisión están compuestos por nodos interiores, nodos terminales y ramas que emanan de los nodos interiores.

Cada nodo interior en el árbol contiene una prueba de un atributo, y cada rama representa un valor distinto del atributo.

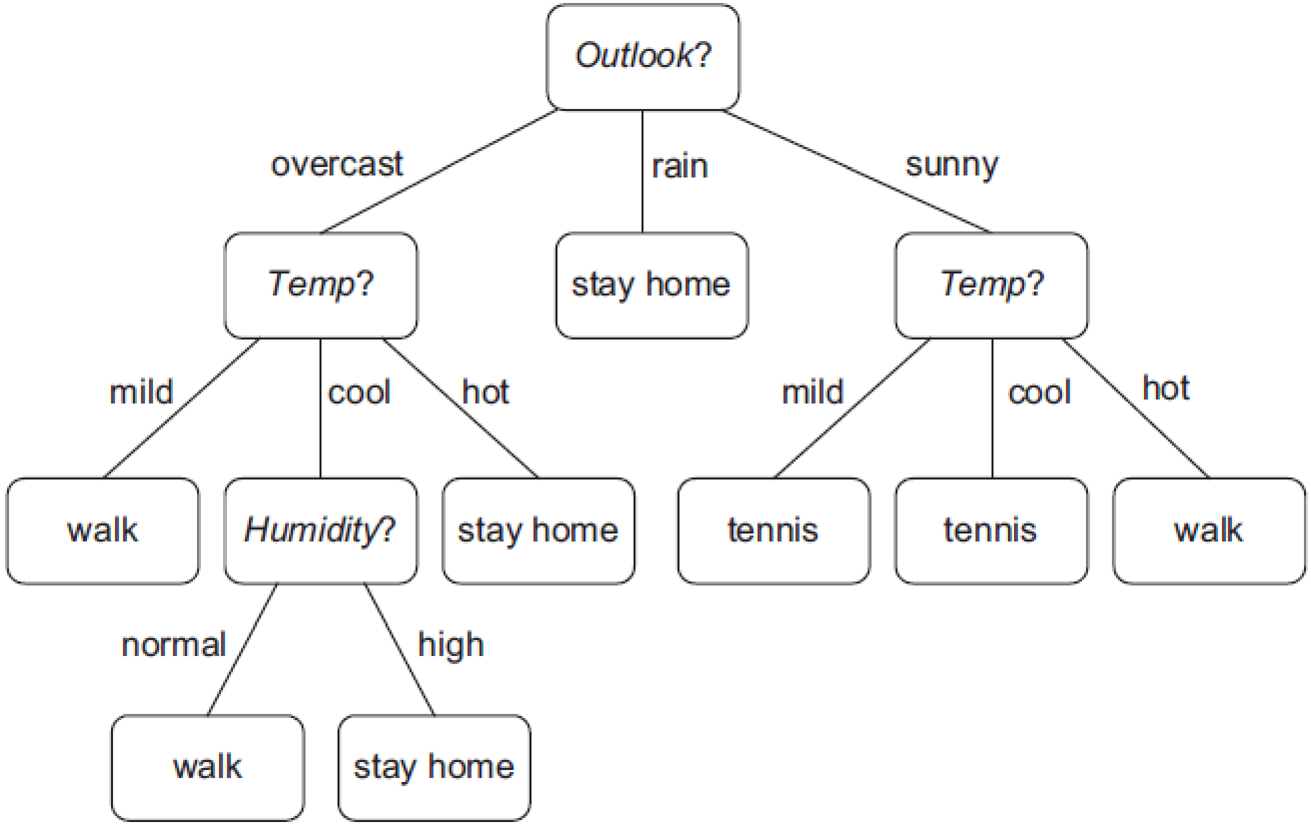
Datos sobre la decisión sobre cómo pasar un sábado por la tarde

Day	Outlook	Temp	Humidity	Wind	Activity
1	rain	hot	high	strong	stay home
2	overcast	cool	high	strong	stay home
3	overcast	cool	normal	strong	walk
4	rain	cool	normal	strong	stay home
5	sunny	cool	normal	strong	tennis
6	sunny	cool	normal	weak	tennis
7	rain	hot	normal	strong	stay home
8	sunny	hot	normal	weak	walk
9	sunny	mild	normal	strong	tennis
10	sunny	mild	high	weak	tennis
11	rain	mild	high	strong	stay home
12	overcast	mild	high	strong	walk
13	sunny	mild	high	strong	tennis
14	overcast	hot	high	strong	stay home

Un árbol de decisión que clasifica las instancias anteriores correctamente



Un árbol de decisión parsimonioso que clasifica las instancias anteriores correctamente



Árboles de decisión para regresión

- Veamos los siguientes datos

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44
14	Rain	Mild	High	Strong	30

Golf players = {25, 30, 46, 45, 52, 23, 43, 35, 38, 46, 48, 52, 44, 30}

Average of golf players = (25 + 30 + 46 + 45 + 52 + 23 + 43 + 35 + 38 + 46 + 48 + 52 + 44 + 30)/14 = 39.78

Standard deviation of golf players = $\sqrt{[((25 - 39.78)^2 + (30 - 39.78)^2 + (46 - 39.78)^2 + \ldots + (30 - 39.78)^2)/14]} = 9.32$

Laboratorio Nacional de Informática Avanzada A.C.

Sunny outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Golf players for sunny outlook = {25, 30, 35, 38, 48}

Average of golf players for sunny outlook = (25+30+35+38+48)/5 = 35.2

Standard deviation of golf players for sunny outlook = $\sqrt{((25 - 35.2)^2 + (30 - 35.2)^2 + \dots)/5)}$ = 7.78

Laboratorio Nacional de Informática Avanzada A.C.

Overcast outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
3	Overcast	Hot	High	Weak	46
7	Overcast	Cool	Normal	Strong	43
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44

Golf players for overcast outlook = {46, 43, 52, 44}

Average of golf players for overcast outlook = (46 + 43 + 52 + 44)/4 = 46.25

Standard deviation of golf players for overcast outlook = $\sqrt{(((46-46.25)^2+(43-46.25)^2+\ldots)} = 3.49$

Rainy outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
10	Rain	Mild	Normal	Weak	46
14	Rain	Mild	High	Strong	30

Golf players for overcast outlook = {45, 52, 23, 46, 30}

Average of golf players for overcast outlook = (45+52+23+46+30)/5 = 39.2

Standard deviation of golf players for rainy outlook = $\sqrt{(((45 - 39.2)^2+(52 - 39.2)^2+\ldots)/5)}$=10.87

Summarizing standard deviations for the outlook feature

Outlook	Stdev of Golf Players	Instances
Overcast	3.49	4
Rain	10.87	5
Sunny	7.78	5

Weighted standard deviation for outlook = (4/14)x3.49 + (5/14)x10.87 + (5/14)x7.78 = 7.66

You might remember that we have calculated the global standard deviation of golf players 9.32 in previous steps. Standard deviation reduction is difference of the global standard deviation and standard deviation for current feature. In this way, maximized standard deviation reduction will be the decision node.

Standard deviation reduction for outlook = 9.32 – 7.66 = 1.66

Hot temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
13	Overcast	Hot	Normal	Weak	44

Golf players for hot temperature = {25, 30, 46, 44}

Standard deviation of golf players for hot temperature = 8.95

Cool temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
9	Sunny	Cool	Normal	Weak	38

Golf players for cool temperature = {52, 23, 43, 38}

Standard deviation of golf players for cool temperature = 10.51

Mild temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
8	Sunny	Mild	High	Weak	35
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
14	Rain	Mild	High	Strong	30

Golf players for mild temperature = {45, 35, 46, 48, 52, 30}

Standard deviation of golf players for mild temperature = 7.65

Summarizing standard deviations for temperature feature

Temperature	Stdev of Golf Players	Instances
Hot	8.95	4
Cool	10.51	4
Mild	7.65	6

Weighted standard deviation for temperature = (4/14)x8.95 + (4/14)x10.51 + (6/14)x7.65 = 8.84

Standard deviation reduction for temperature = 9.32 – 8.84 = 0.47

Laboratorio Nacional de Informática Avanzada A.C.

High humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
8	Sunny	Mild	High	Weak	35
12	Overcast	Mild	High	Strong	52
14	Rain	Mild	High	Strong	30

Golf players for high humidity = {25, 30, 46, 45, 35, 52, 30}

Standard deviation for golf players for high humidity = 9.36

Normal humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Players
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
13	Overcast	Hot	Normal	Weak	44

Golf players for normal humidity = {52, 23, 43, 38, 46, 48, 44}

Standard deviation for golf players for normal humidity = 8.73

Summarizing standard deviations for humidity feature

Humidity	Stdev of Golf Player	Instances
High	9.36	7
Normal	8.73	7

Weighted standard deviation for humidity = (7/14)x9.36 + (7/14)x8.73 = 9.04

Standard deviation reduction for humidity = 9.32 – 9.04 = 0.27

Strong Wind

Day	Outlook	Temp.	Humidity	Wind	Golf Players
2	Sunny	Hot	High	Strong	30
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
14	Rain	Mild	High	Strong	30

Golf players for strong wind= {30, 23, 43, 48, 52, 30}

Standard deviation for golf players for strong wind = 10.59

Weak Wind

1	Sunny	Hot	High	Weak	25
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
13	Overcast	Hot	Normal	Weak	44

Golf players for weakk wind= {25, 46, 45, 52, 35, 38, 46, 44}

Standard deviation for golf players for weak wind = 7.87

Summarizing standard deviations for wind feature

Wind	Stdev of Golf Player	Instances
Strong	10.59	6
Weak	7.87	8

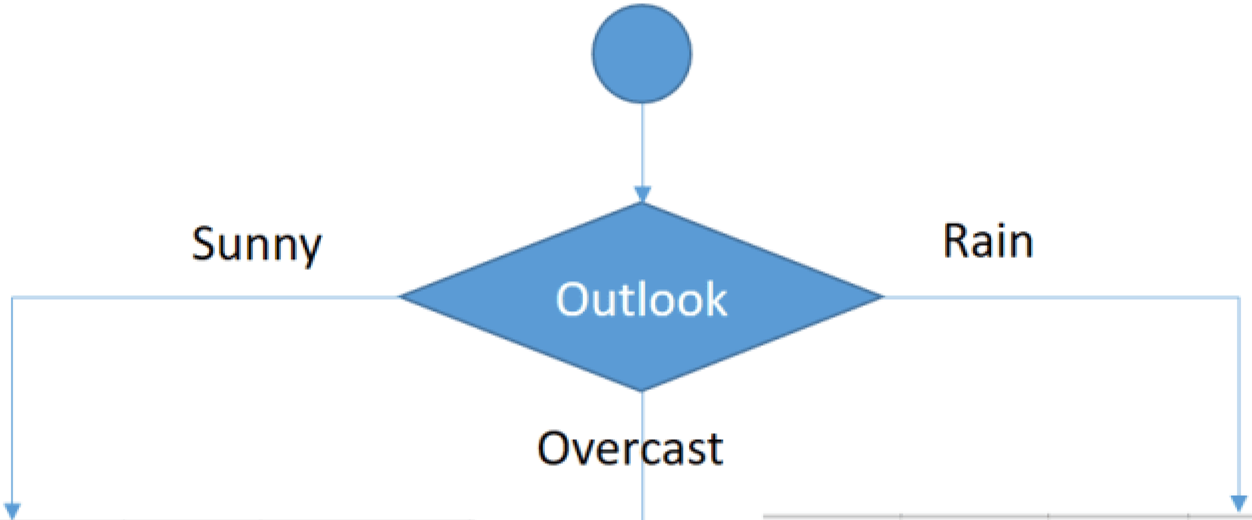
Weighted standard deviation for wind = (6/14)x10.59 + (8/14)x7.87 = 9.03

Standard deviation reduction for wind = 9.32 – 9.03 = 0.29

Por lo tanto, hemos calculado valores de reducción de desviación estándar para todas las funciones. El ganador es “outlook" porque tiene la puntuación más alta.

Feature	Standard Deviation Reduction
Outlook	1.66
Temperature	0.47
Humidity	0.27
Wind	0.29

Técnicas de clasificación/regresión: árboles de decisión



Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
10	Rain	Mild	Normal	Weak	46
14	Rain	Mild	High	Strong	30

Day	Outlook	Temp.	Humidity	Wind	Golf Players
3	Overcast	Hot	High	Weak	46
7	Overcast	Cool	Normal	Strong	43
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44

Sunny Outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Golf players for sunny outlook = {25, 30, 35, 38, 48}

Standard deviation for sunny outlook = 7.78

Notice that we will use this standard deviation value as global standard deviation for this sub data set.

Sunny outlook and Hot Temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30

Standard deviation for sunny outlook and hot temperature = 2.5

Sunny outlook and Cool Temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
9	Sunny	Cool	Normal	Weak	38

Standard deviation for sunny outlook and cool temperature = 0

Sunny outlook and Mild Temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
8	Sunny	Mild	High	Weak	35
11	Sunny	Mild	Normal	Strong	48

Standard deviation for sunny outlook and mild temperature = 6.5

Summary of standard deviations for temperature feature when outlook is sunny

Temperature	Stdev for Golf Players	Instances
Hot	2.5	2
Cool	0	1
Mild	6.5	2

Weighted standard deviation for sunny outlook and temperature = (2/5)x2.5 + (1/5)x0 + (2/5)x6.5 = 3.6

Standard deviation reduction for sunny outlook and temperature = 7.78 – 3.6 = 4.18

Sunny outlook and high humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35

Standard deviation for sunny outlook and high humidity = 4.08

Sunny outlook and normal humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Players
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Standard deviation for sunny outlook and normal humidity = 5

Summarizing standard deviations for humidity feature when outlook is sunny

Humidity	Stdev for Golf Players	Instances
High	4.08	3
Normal	5.00	2

Weighted standard deviations for sunny outlook and humidity = (3/5)x4.08 + (2/5)x5 = 4.45

Standard deviation reduction for sunny outlook and humidity = 7.78 – 4.45 = 3.33

Sunny outlook and Strong Wind

Day	Outlook	Temp.	Humidity	Wind	Golf Players
2	Sunny	Hot	High	Strong	30
11	Sunny	Mild	Normal	Strong	48

Standard deviation for sunny outlook and strong wind = 9

Sunny outlook and Weak Wind

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38

Standard deviation for sunny outlook and weak wind = 5.56

Wind	Stdev for Golf Players	Instances
Strong	9	2
Weak	5.56	3

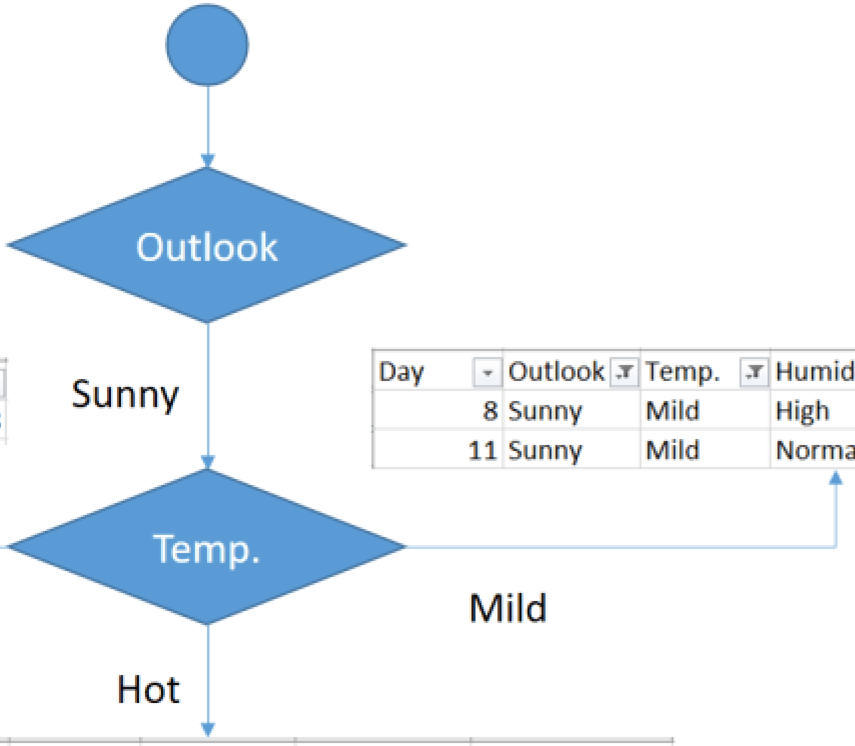
Weighted standard deviations for sunny outlook and wind = (2/5)x9 + (3/5)x5.56 = 6.93

Standard deviation reduction for sunny outlook and wind = 7.78 – 6.93 = 0.85

We've calculated standard deviation reductions for sunny outlook. The winner is temperature.

Feature	**Standard Deviation Reduction**
Temperature	4.18
Humidity	3.33
Wind	0.85

Laboratorio Nacional de Informática Avanzada A.C.



Day	Outlook	Temp.	Humidity	Wind	Golf Players
9	Sunny	Cool	Normal	Weak	38

Day	Outlook	Temp.	Humidity	Wind	Golf Players
8	Sunny	Mild	High	Weak	35
11	Sunny	Mild	Normal	Strong	48

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30

Poda

- La rama Cool tiene una instancia en su conjunto de datos secundarios. Podemos decir que si Outlook es Sunny y la temperatura es fresca, entonces habría 38 jugadores de golf.
- ¿Pero qué hay de la rama Hot? Todavía hay 2 instancias. ¿Deberíamos agregar otra rama para viento débil y viento fuerte?
- No, no deberíamos. Porque esto provoca un ajuste excesivo.
- Deberíamos terminar la construcción de ramas, por ejemplo, si hay menos de cinco instancias en el conjunto de datos secundario. O la desviación estándar del conjunto de datos secundario puede ser inferior al 5% de todo el conjunto de datos.
- La primera opción suele ser más efectiva. Por lo tanto se termina la rama si hay menos de 5 instancias en el conjunto de datos secundario actual. Si se cumple esta condición de terminación, se calcula el promedio del conjunto de datos secundarios. Esta operación se llama poda en los árboles de decisión.

Forma final del árbol de regresión