

# Curso de Especialización de Machine Learning con Python



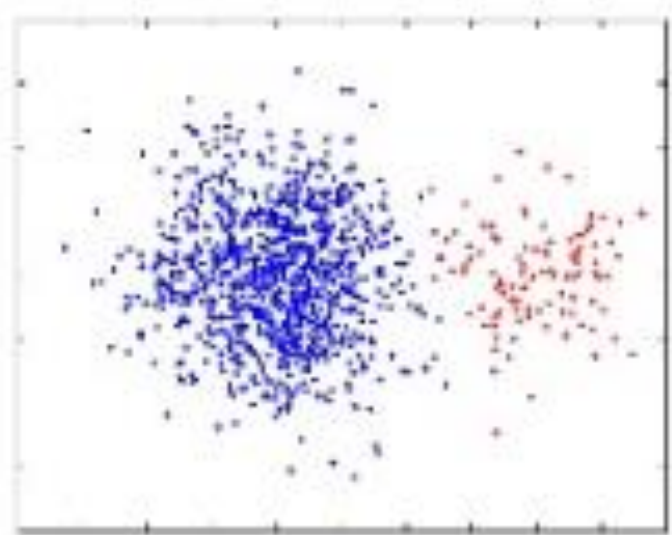


# TÉCNICAS DE BALANCEO DE DATOS

# ¿Qué es data desbalanceada?

Comúnmente hace referencia a la variable “Target”.

*Def:* Se da cuando la frecuencia de clases de la variable Target son muy distintas o muy desiguales.



*Data desbalanceada*

# ¿Qué consecuencias puede traer?

Al momento de entrenar un algoritmo de ML con el dataset desbalanceado, se puede originar un sesgo hacia una clase en particular de la variable target. Hacia la clase mayoritaria.

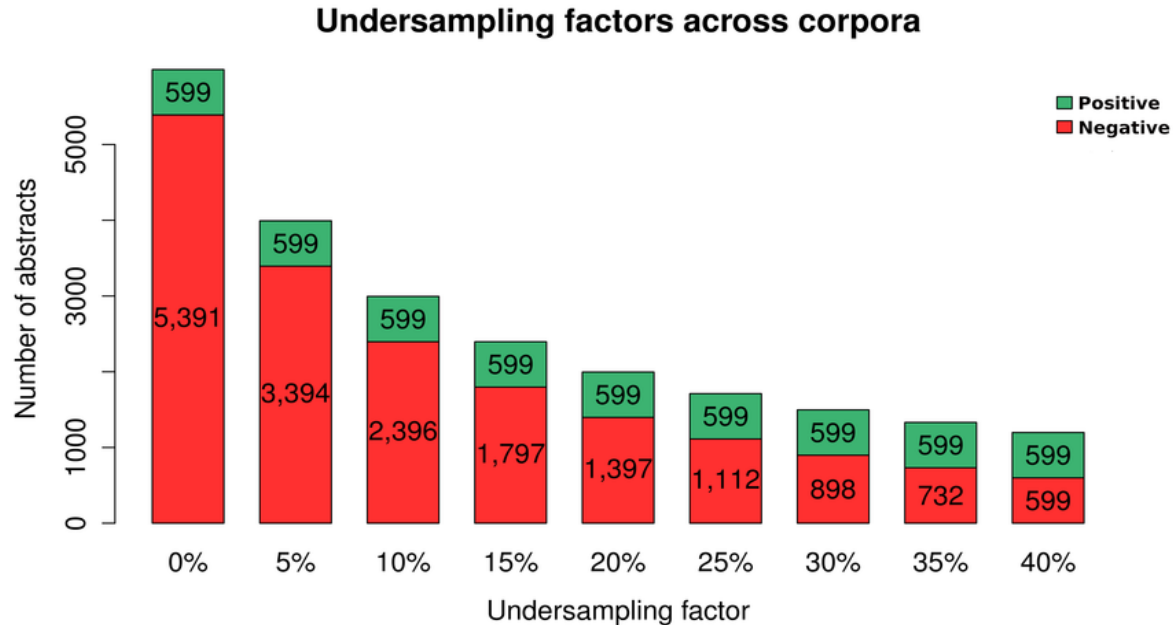
## ¿Cómo lo solucionamos?

Balanceando la data.

- Técnicas: Undersampling, Oversampling, SMOTE, a criterio propio, otras

# UNDERSAMPLING

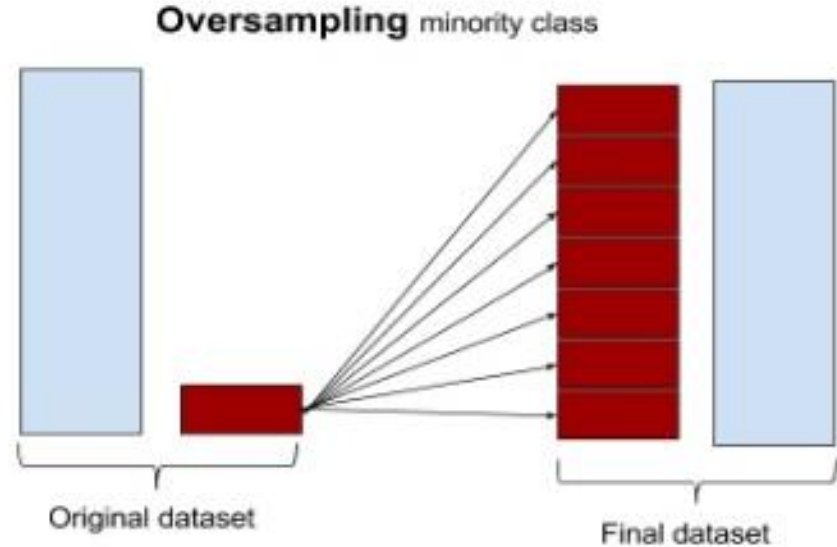
En esta técnica se busca reducir la cantidad de registros de la clase mayoritaria a la cantidad de registros de la clase minoritaria.



Mayor a menor → Undersampling

# OVERSAMPLING

En esta técnica se busca incrementar la cantidad de registros de la clase minoritaria a la cantidad de registros de la clase mayoritaria.



**Menor a mayor → Oversampling**

# RESUMEN

## Undersampling



## Oversampling



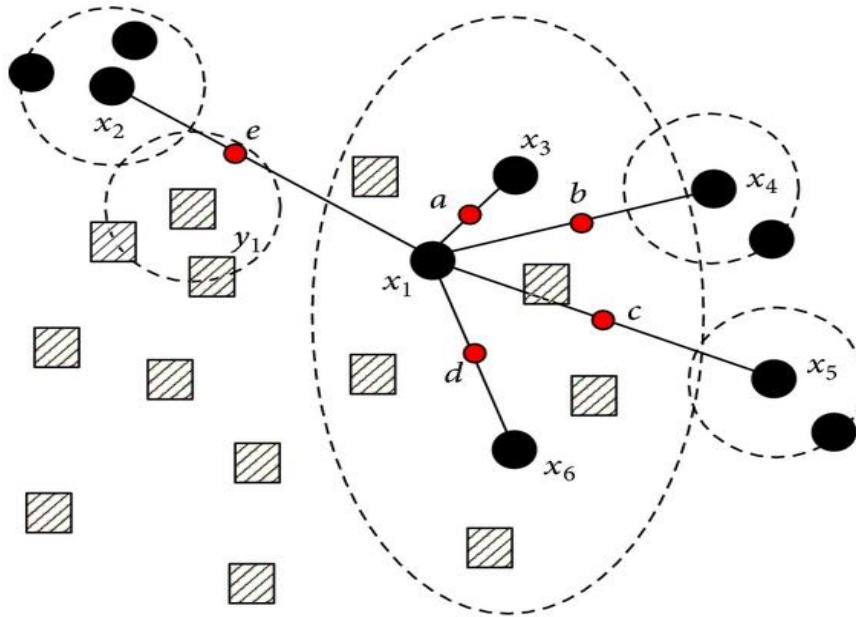
# SMOTE

Synthetic Minority Over-sampling Technique

Es una técnica de oversampling.

El objetivo es crear puntos sintéticos a partir de la data de la clase minoritaria.

Ejemplo:  $k = 5$



- Majority class samples
- Minority class samples
- Synthetic samples

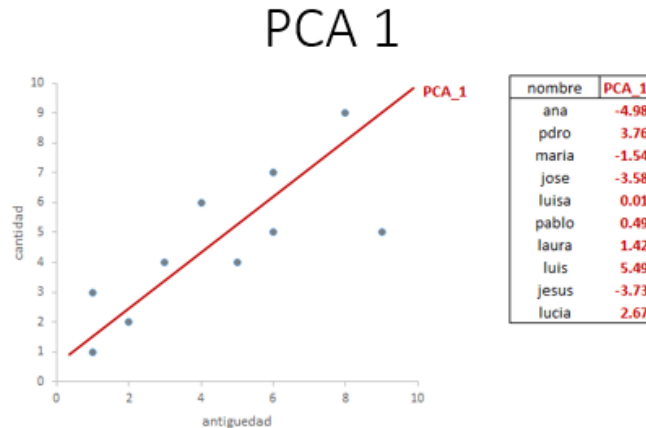


# Análisis de Componentes Principales (PCA)

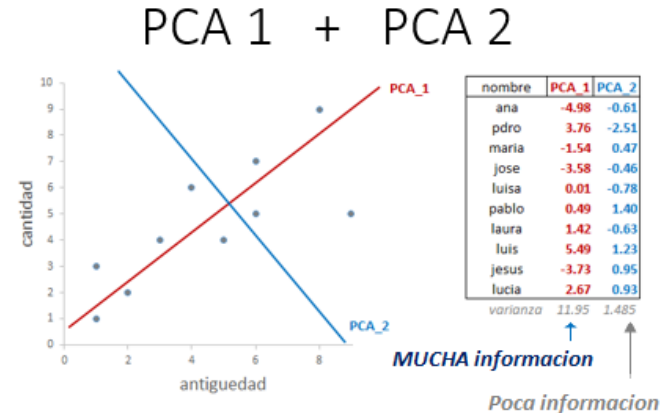
Técnica utilizada para describir un set de datos en términos de nuevas variables ("componentes") no correlacionadas.

## Objetivo

- Reducir la dimensionalidad de un conjunto de datos
- Convertir un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables sin correlación



En este ejemplo el PCA\_1 tiene el **89%** de la varianza

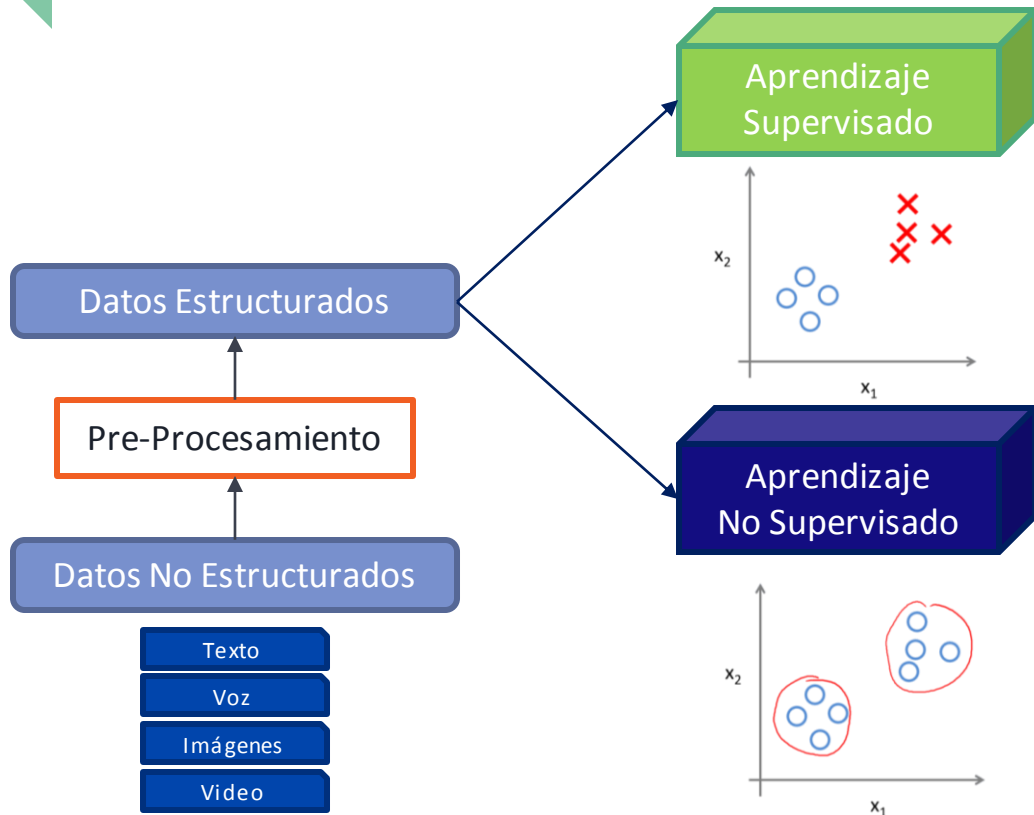


En este ejemplo el PCA\_2 tiene el **11%** de la varianza ( $1.485/(11.95+1.48)$ )

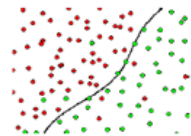
# Módulo 5:

# Construcción y Evaluación de Modelos

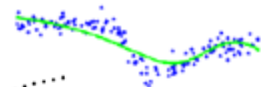
# Tipos de Modelos



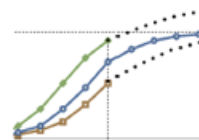
☐ Clasificación



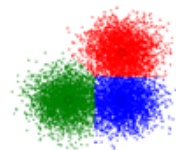
☐ Regresión



☐ Pronóstico



☐ Segmentación



☐ Asociación



☐ Secuenciación



# Regresión Lineal

Modelo matemático para estimar los valores de una **variable continua** (*dependiente*) en función de otra(s) variable(s) (*independientes*).

Modelo:  $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon$

**Y = Variable dependiente**

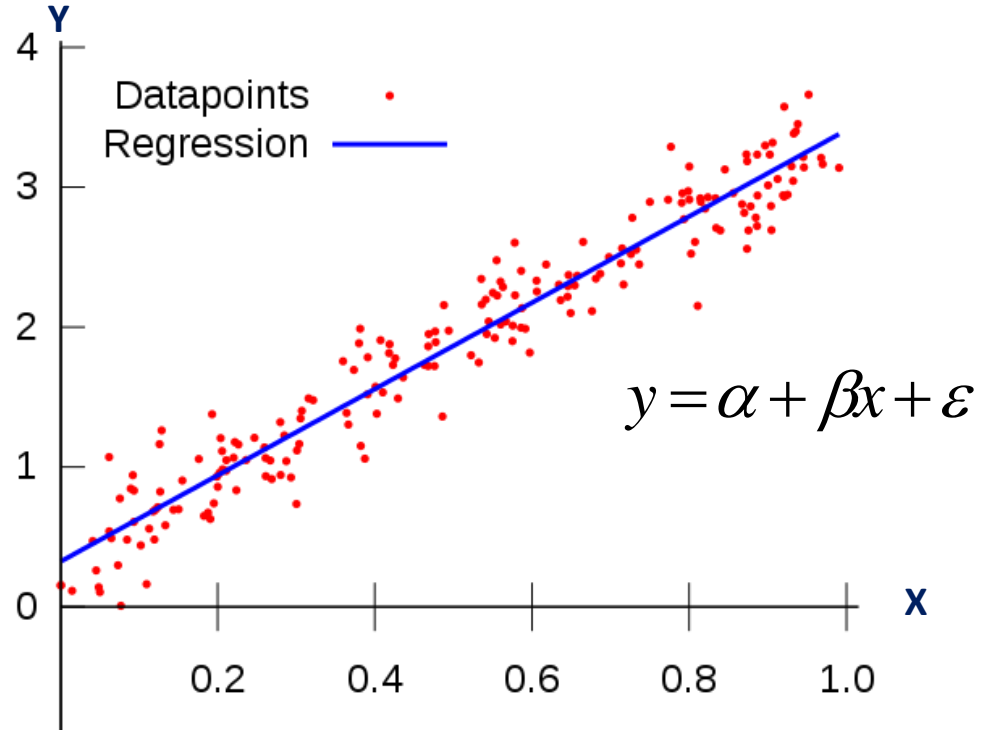
(Predicha o Explicada)

**X = Vector de Variables Independientes**

(Predictora o Explicativa)

**$\beta$  = Vector de Coeficientes**

Se llama **regresión lineal simple** cuando solo existe una variable independiente, o **múltiple** si existen varias variables independientes.



# Regresión Logística

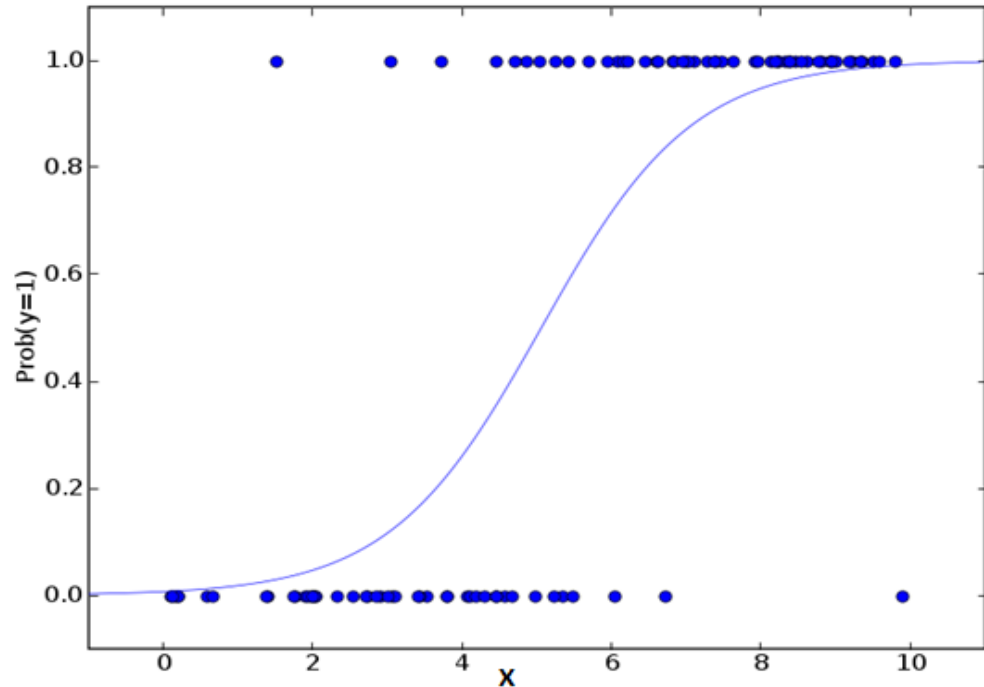
Modelo matemático para estimar el valor de una **variable categórica o discreta** (*dependiente*) en función de otra(s) variable(s) (*independientes*). Predice la probabilidad de ocurrencia de un evento ajustando los datos a una **función logit**.

$p$  = Probabilidad de ocurrencia de un evento

$1 - p$  = Probabilidad de no ocurrencia de un evento

Para poder predecir la variable binaria, se transforma la regresión lineal en una regresión logística, convirtiendo "**y**" en " **$\ln(p/(1-p))$** " y luego se aplica una regresión lineal sobre esta transformación.

$$\ln\left(\frac{p}{1-p}\right) = z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + u$$



$$\text{Prob}(Y_i = 1) = \frac{1}{1 + e^{-(\alpha + \beta_k X_{ki})}} = \frac{e^{\alpha + \beta_k X_{ki}}}{1 + e^{\alpha + \beta_k X_{ki}}}$$

# Regresión Logística

Datos

| y | x  |
|---|----|
| 0 | 25 |
| 1 | 54 |
| 1 | 67 |
| 0 | 6  |
| 1 | 77 |
| 0 | 49 |
| 1 | 81 |

$$y = \beta_0 + \beta_1 x_1 + u$$

## Paso 1

Se transforma **y** en el logaritmo de la probabilidad de **y**

A la transformación  $P/(1-P)$  también le dicen **Odds Ratio**, donde  $P$ =probabilidad de **y**

$$\ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x_1 + u$$

## Paso 2

Se calcula la regresión lineal para predecir el logaritmo de los Odds Ratio

## Paso 3

Transformar el resultado de la regresión lineal en la probabilidad final

$$\frac{e^{\beta_0 + \beta_1 x_1 + u}}{1 + e^{\beta_0 + \beta_1 x_1 + u}}$$

Donde **e** es una constante = 2.718281828

$$\ln\left(\frac{p}{1-p}\right) = z$$

$$e^{\ln\left(\frac{p}{1-p}\right)} = e^z$$

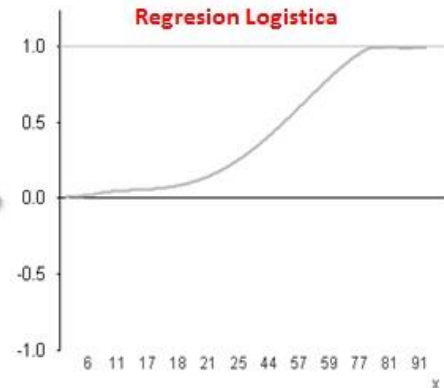
$$\frac{p}{1-p} = e^z$$

$$p = e^z - pe^z$$

$$p(1 + e^z) = e^z$$

$$(\text{prob final}) p = \frac{e^z}{1 + e^z}$$

Regresión Logística



# Knn (K vecinos más cercanos)

Es un simple algoritmo que almacena todos los casos disponibles en el “entrenamiento” y clasifica los nuevos casos por el voto mayoritario de sus  $k$  vecinos más cercanos (según una función de distancia). Se puede usar para problemas de **clasificación** y **regresión**.

Funciones de Distancia:

Euclidean  $\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$

Manhattan  $\sum_{i=1}^k |x_i - y_i|$

Minkowski  $\left( \sum_{i=1}^k (|x_i - y_i|^q) \right)^{1/q}$

Vars Continuas

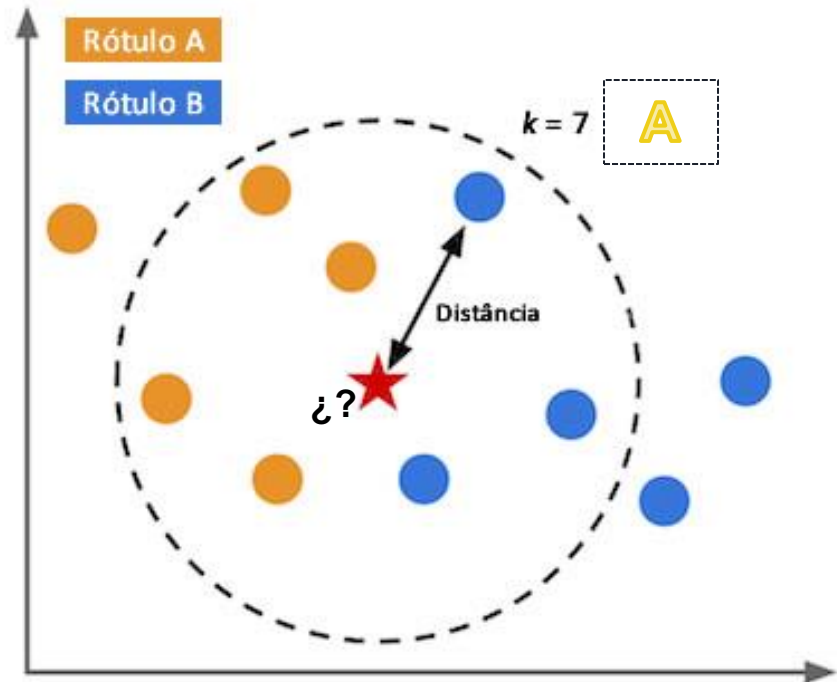
Hamming Distance

$$D_H = \sum_{i=1}^k |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

Vars Categóricas



# Naive Bayes

Los métodos de Naive Bayes son un conjunto de algoritmos de aprendizaje supervisado basados en la aplicación del teorema de Bayes con la suposición de independencia "ingenua" (Naives) entre cada par de características.

Modelo de probabilidad para un clasificador

$$p(C|F_1, \dots, F_n)$$

Según teorema de Bayes  $p(C|F_1, \dots, F_n) = \frac{p(C) p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)}$ .

El objetivo es hallar  $p(C) p(F_1, \dots, F_n|C)$  (numerador)  $\rightarrow \mathbf{Z}$

Es igual a

$$\begin{aligned} &= p(C) p(F_1, \dots, F_n|C) \\ &= p(C) p(F_1|C) p(F_2, \dots, F_n|C, F_1) \\ &= p(C) p(F_1|C) p(F_2|C, F_1) p(F_3, \dots, F_n|C, F_1, F_2) \\ &= p(C) p(F_1|C) p(F_2|C, F_1) p(F_3|C, F_1, F_2) p(F_4, \dots, F_n|C, F_1, F_2, F_3) \end{aligned}$$

Según el supuesto "Naives" de independencia condicional

$$p(F_i|C, F_j) = p(F_i|C)$$

Entonces la formula anterior queda así

$$\begin{aligned} p(C) p(F_1, \dots, F_n|C) &= p(C) p(F_1|C) p(F_2|C) p(F_3|C) \dots \\ &= p(C) \prod_{i=1}^n p(F_i|C). \end{aligned}$$

Y por lo tanto

$$p(C|F_1, \dots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^n p(F_i|C)$$

Algoritmos:

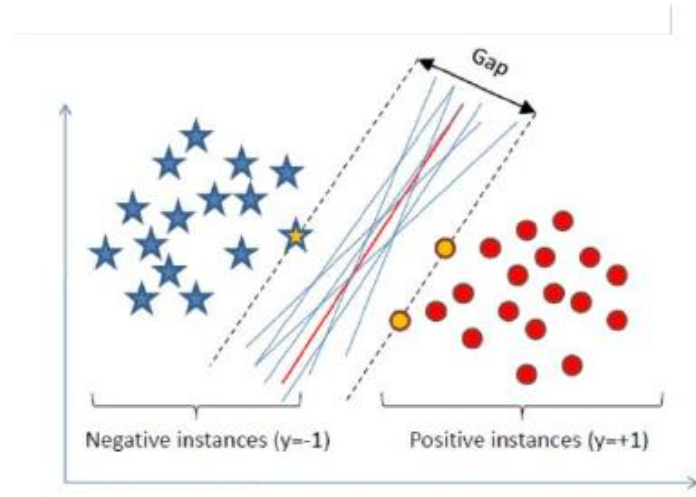
- Gaussian Naive Bayes
- Multinomial Naive Bayes
- Bernoulli Naive Bayes



# Support Vector Machine (SVM)

Es un algoritmo de aprendizaje supervisado cuyo objetivo es encontrar un hiperplano canónico que maximice el margen del conjunto de datos de entrenamiento, esto nos garantiza una buena capacidad de generalización

**Maquina de  
Aprendizaje Lineal**



## Problema con SVM

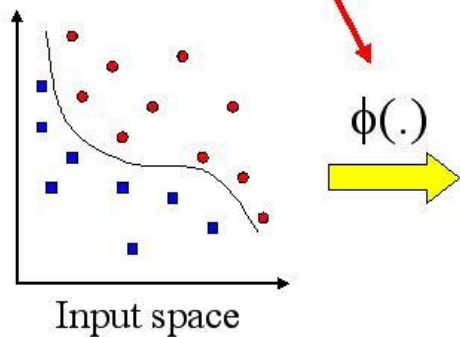
En los dataset de aplicación comunes existen:

- a) Más de dos variables predictoras
- b) Curvas no lineales de separación
- c) Casos donde los conjuntos de datos no pueden ser completamente separados
- d) Clasificaciones en más de dos categorías.

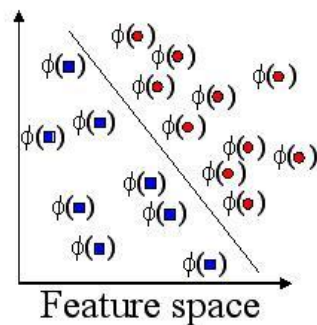
Para solucionarlo utilizamos los KERNEL.

# Kernel

*Kernel Function*

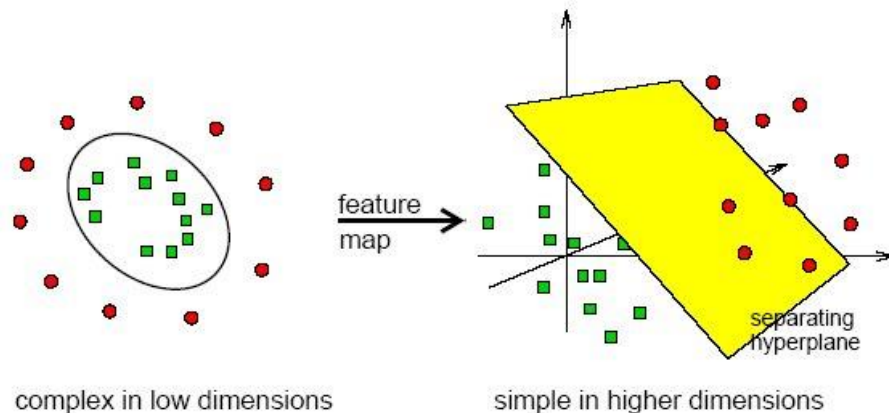


*Non-linearly separable*



*Linearly separable*

Separation may be easier in higher dimensions



# Árbol de Decisión

Técnica de Aprendizaje Supervisado no-paramétrica que puede ser usado tanto para predecir una variable categórica (clasificación) y continua (regresión). Los árboles responden preguntas secuenciales que nos envían a cierta ruta del árbol dada la respuesta. El modelo se comporta usando condiciones de "si esto se cumple entonces" que finalmente produce un resultado específico.

## Funciones para división:

Índice de Gini

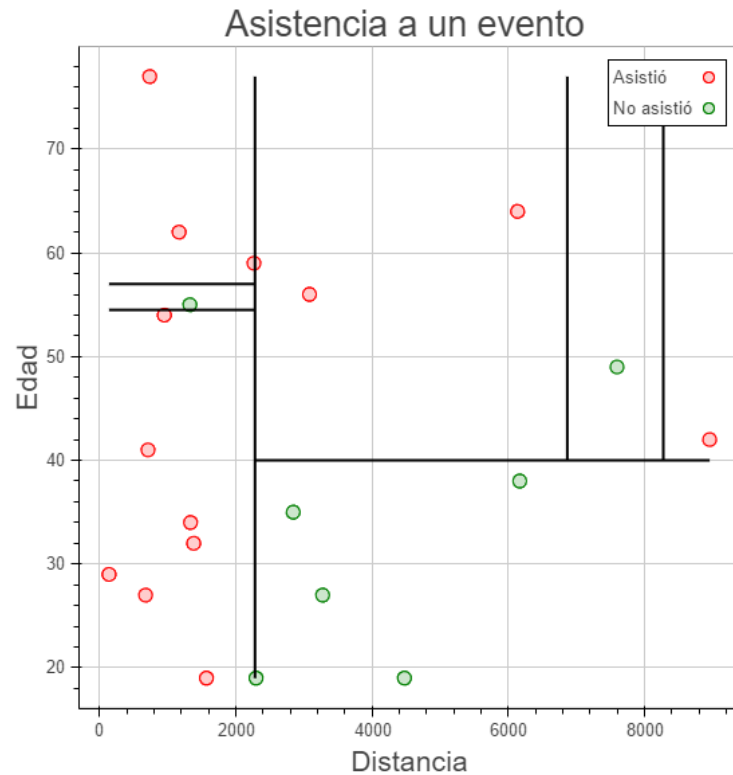
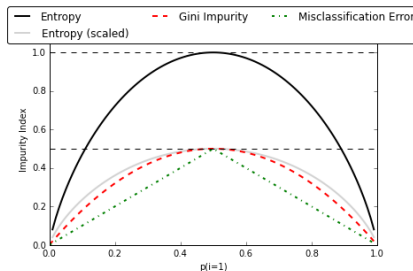
$$1 - \sum_j p_j^2$$

Entropía

$$-\sum_j p_j \log_2(p_j)$$

Error de Clasificación

$$1 - \max(p_j)$$



# Métricas de Evaluación de Clasificación

## Matriz de Confusión

|            |       | Real                      |                           |
|------------|-------|---------------------------|---------------------------|
|            |       | (+) 1                     | (-) 0                     |
| Predicción | (+) 1 | Verdaderos Positivos (VP) | Falsos Positivos (FP)     |
|            | (-) 0 | Falsos Negativos (FN)     | Verdaderos Negativos (VN) |

$$\text{tasa de aciertos} = \frac{VP + VN}{\text{total}} \quad (\text{accuracy})$$

$$\text{tasa de error} = \frac{FN + FP}{\text{total}}$$

$$\text{precisión} = \frac{VP}{VP + FP}$$

|            |       | Real                      |       |
|------------|-------|---------------------------|-------|
|            |       | (+) 1                     | (-) 0 |
| Predicción | (+) 1 | Verdaderos Positivos (VP) |       |
|            | (-) 0 | Falsos Negativos (FN)     |       |

$$\text{especificidad} = \frac{VN}{VN + FP}$$

|            |       | Real                      |                       |
|------------|-------|---------------------------|-----------------------|
|            |       | (+) 1                     | (-) 0                 |
| Predicción | (+) 1 | Verdaderos Positivos (VP) | Falsos Positivos (FP) |
|            | (-) 0 |                           |                       |

$$\text{sensibilidad} = \frac{VP}{VP + FN}$$

|            |       | Real  |                           |
|------------|-------|-------|---------------------------|
|            |       | (+) 1 | (-) 0                     |
| Predicción | (+) 1 |       | Falsos Positivos (FP)     |
|            | (-) 0 |       | Verdaderos Negativos (VN) |

# Métricas de Evaluación de Clasificación

## Matriz de Confusión

|            |       | Real  |       |
|------------|-------|-------|-------|
|            |       | (+) 1 | (-) 0 |
| Predicción | (+) 1 | 70    | 10    |
|            | (-) 0 | 20    | 50    |

Calcular:

$$tasa\ de\ aciertos = \frac{70 + 50}{150} = 0.800$$

$$precisión = \frac{70}{70 + 10} = 0.875$$

$$sensibilidad = \frac{70}{70 + 20} = 0.778$$

$$especificidad = \frac{50}{50 + 10} = 0.833$$

# Métricas de Evaluación de Clasificación

## Matriz de Confusión

|            |       | Real  |       |
|------------|-------|-------|-------|
|            |       | (+) 1 | (-) 0 |
| Predicción | (+) 1 | 30    | 25    |
|            | (-) 0 | 20    | 925   |

**Target altamente desbalanceado.**

Clase (+) representa el 5% de toda la base

**Calcular:**

$$\text{especificidad} = \frac{925}{925 + 25} = 0.974$$

$$\text{sensibilidad} = \frac{30}{30 + 20} = 0.600$$

$$\text{precisión} = \frac{30}{30 + 25} = 0.545$$

$$\text{tasa de aciertos} = \frac{30 + 925}{1000} = 0.955$$

Modelo excelente?

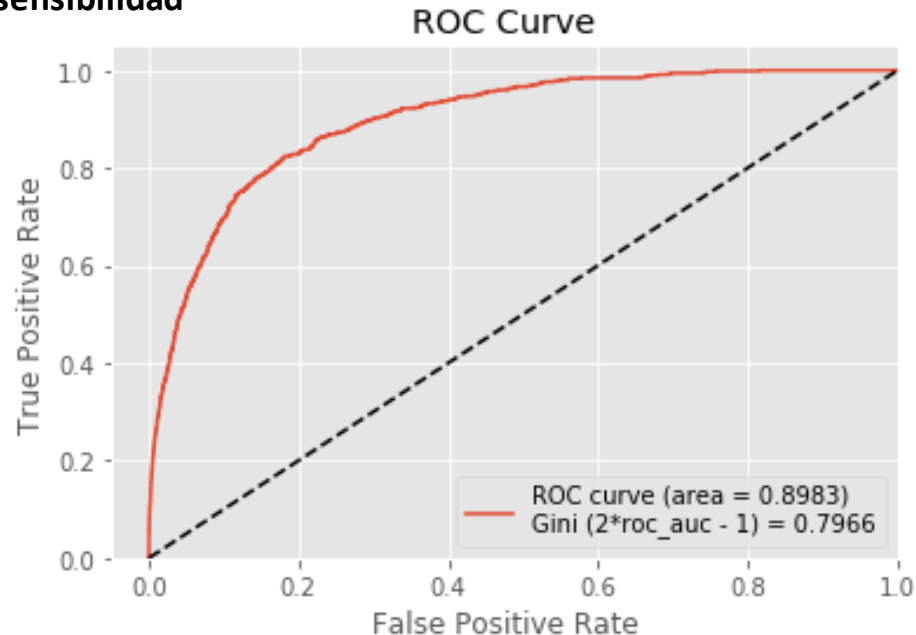
# Métricas de Evaluación de Clasificación

## ROC - AUC

La Curva ROC proporciona un índice de la capacidad de un modelo para discriminar entre estados alternativos de la clase target.

Es útil para comparar modelos y seleccionar umbrales de decisión ( puntos de corte entre (+) y (-) )

sensibilidad

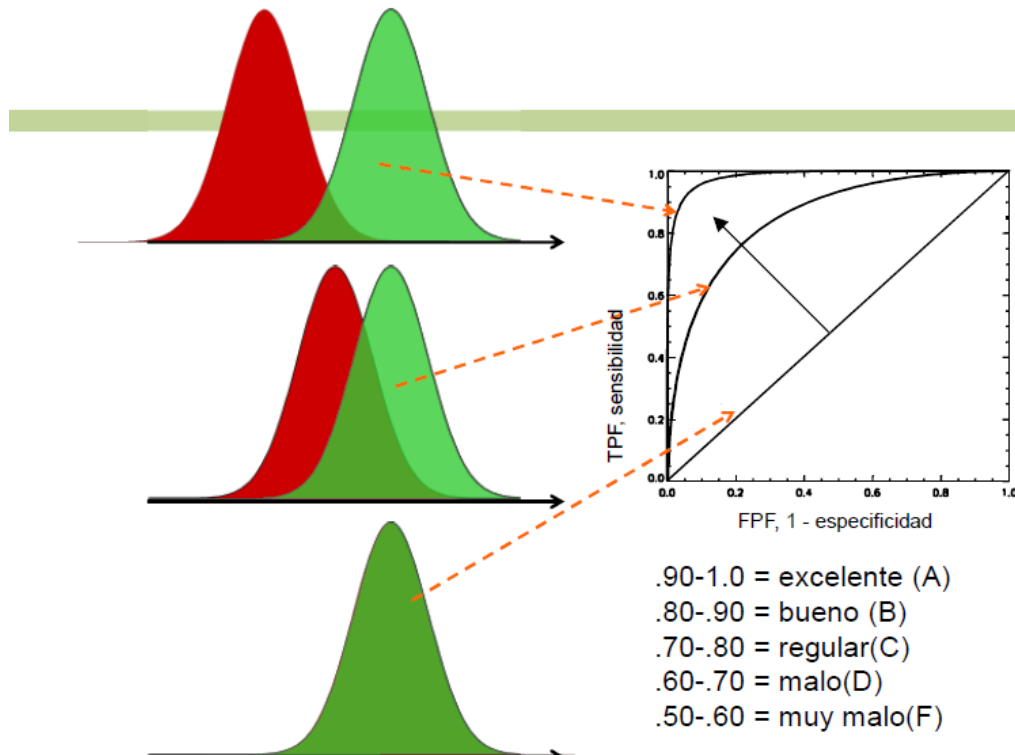


1 - especificidad



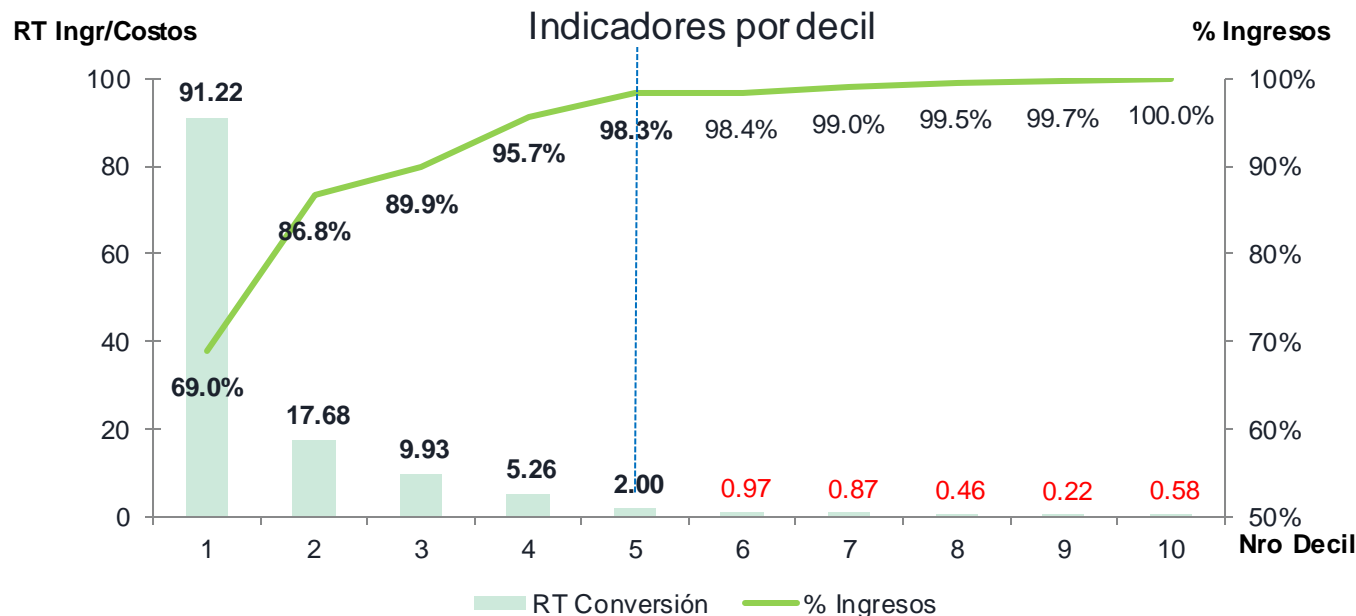
# Métricas de Evaluación de Clasificación

## ROC - AUC



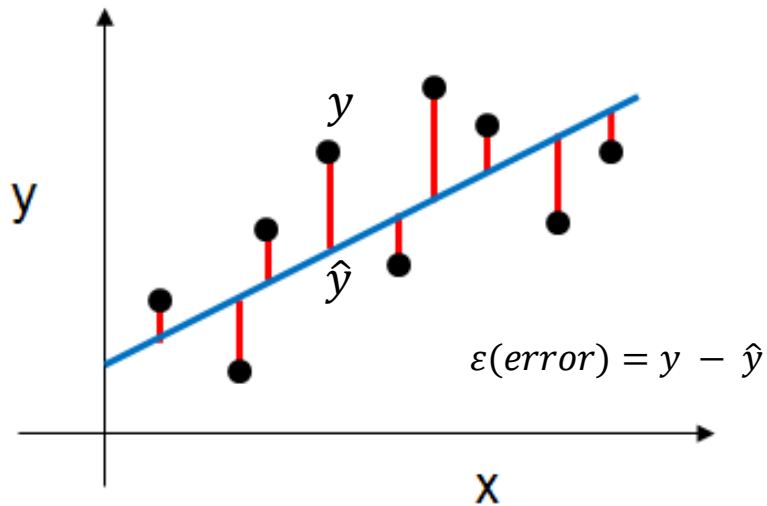
# Métricas de Evaluación de Clasificación

## Resultados Modelo de Propensión de Compra



Se dividió la base por deciles según los resultados del modelo. Los primeros 5 grupos generaron el 98% de los ingresos totales de la campaña feb-17.

# Métricas de Evaluación de Regresión



**Error Absoluto Medio:**

$$MAE = \frac{\sum_{t=1}^n |\hat{y}_t - y_t|}{n}$$

**Error Cuadrático Medio:**

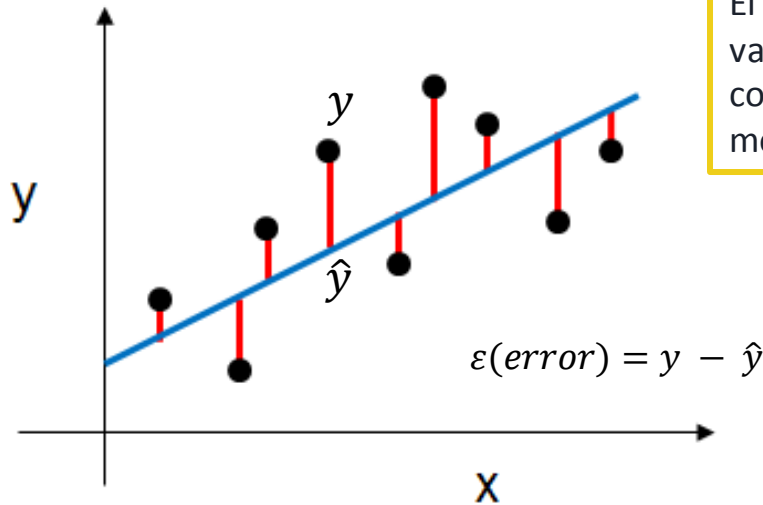
$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2$$

**Raíz Cuadrada del Error Cuadrático Medio:**

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}}$$

# Métricas de Evaluación de Regresión

## Coefficiente de Determinación, R<sup>2</sup>



El R Cuadrado se define como la proporción de la varianza total de la variable explicada por la regresión. El R Cuadrado, también llamado coeficiente de determinación, refleja la bondad del ajuste de un modelo a la variable que pretender explicar.

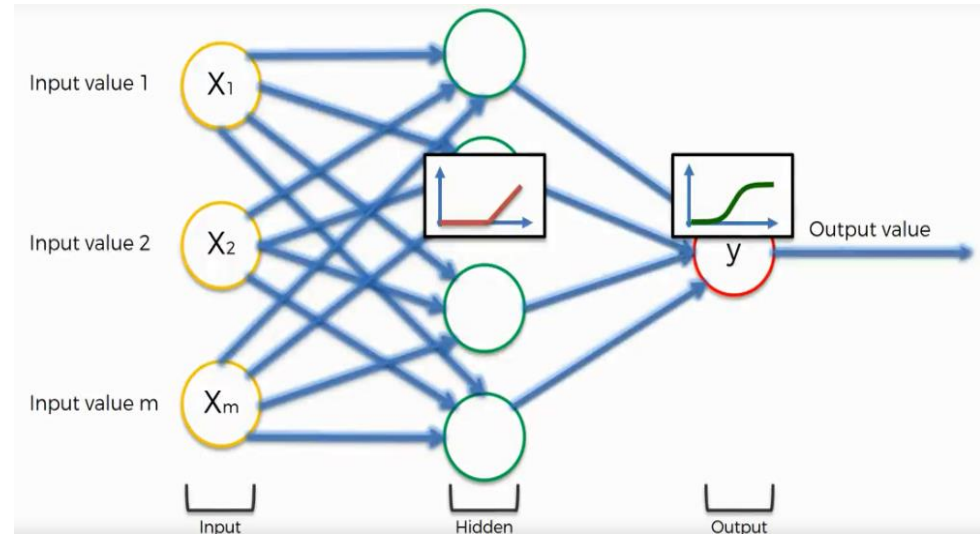
El cociente  $R^2 = \text{SCM}/\text{SCT} = 1 - \text{SCR}/\text{SCT}$  es la proporción de variación de las respuestas explicadas por la regresión; se conoce como **coeficiente de determinación**.

Ej: si  $R^2 = 0.85$ , la variable x explica un 85% de la variación de la variable y

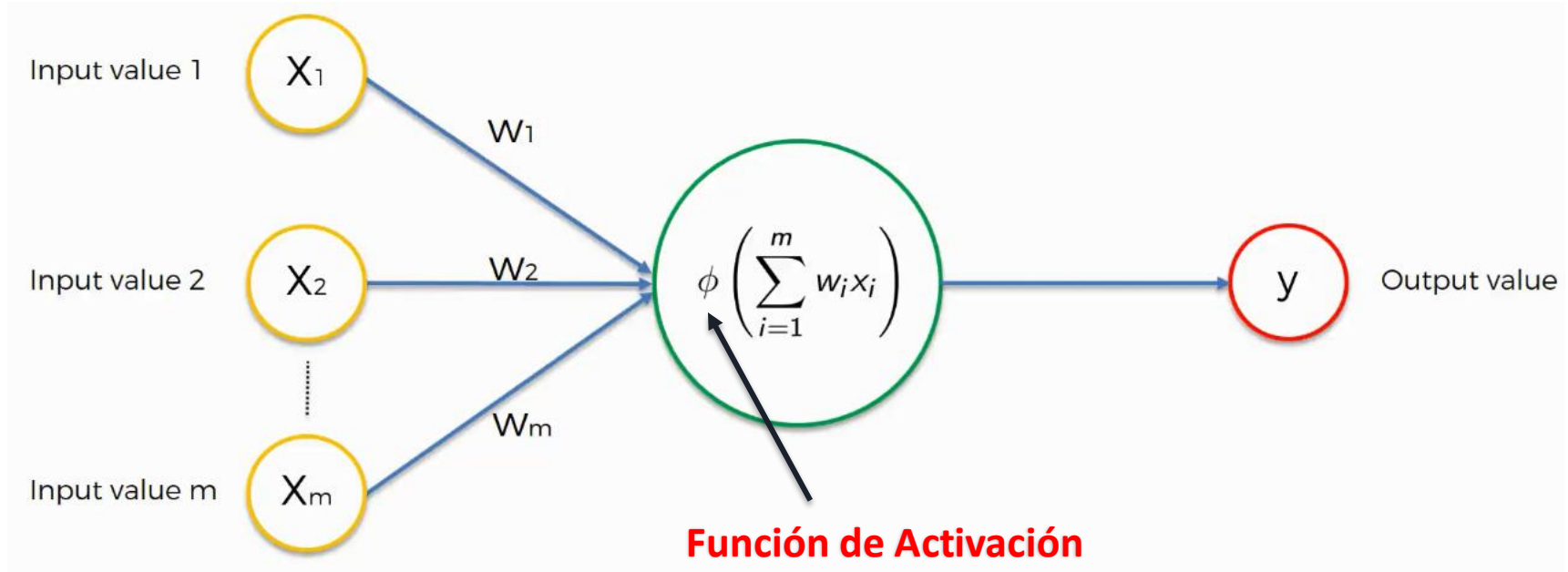
# Redes Neuronales

Es una técnica de Aprendizaje Supervisado para **variables continuas** que emula el modo en que el cerebro humano procesa la información, funciona simulando un número elevado de unidades de procesamiento interconectadas que parecen versiones abstractas de neuronas.

Las unidades de procesamiento se organizan en capas. Hay tres partes normalmente en una red neuronal : una **capa de entrada**, con unidades que representan los campos de entrada; una o varias **capas ocultas**; y una **capa de salida**, con una unidad o unidades que representa el campo o los campos de destino.

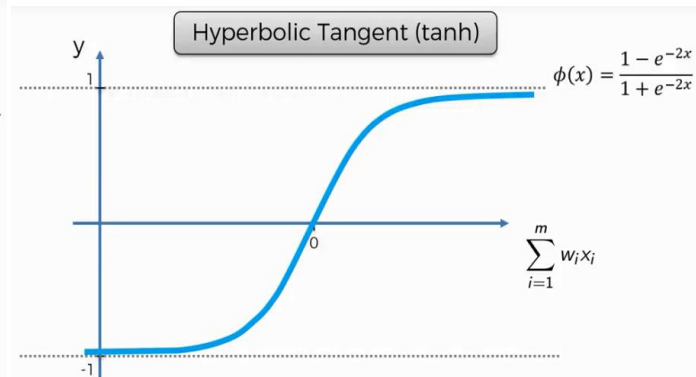
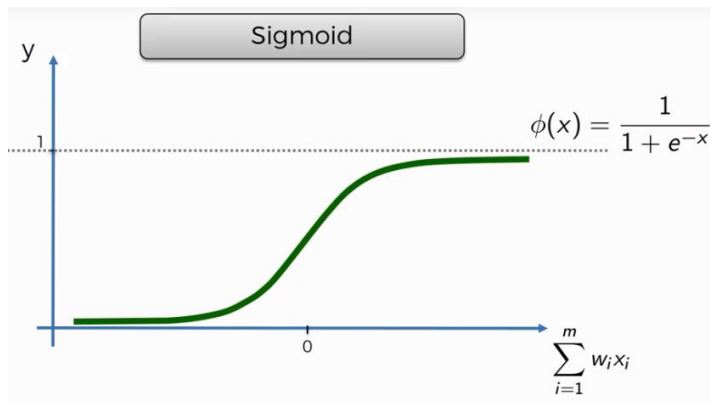
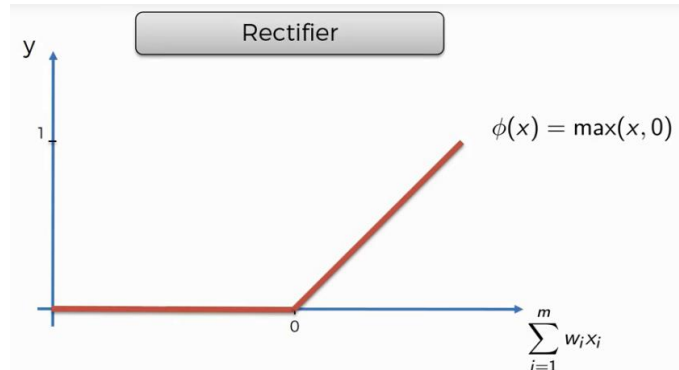
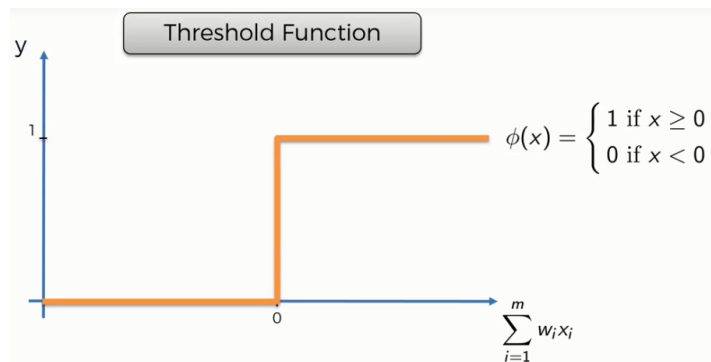


# Perceptrón

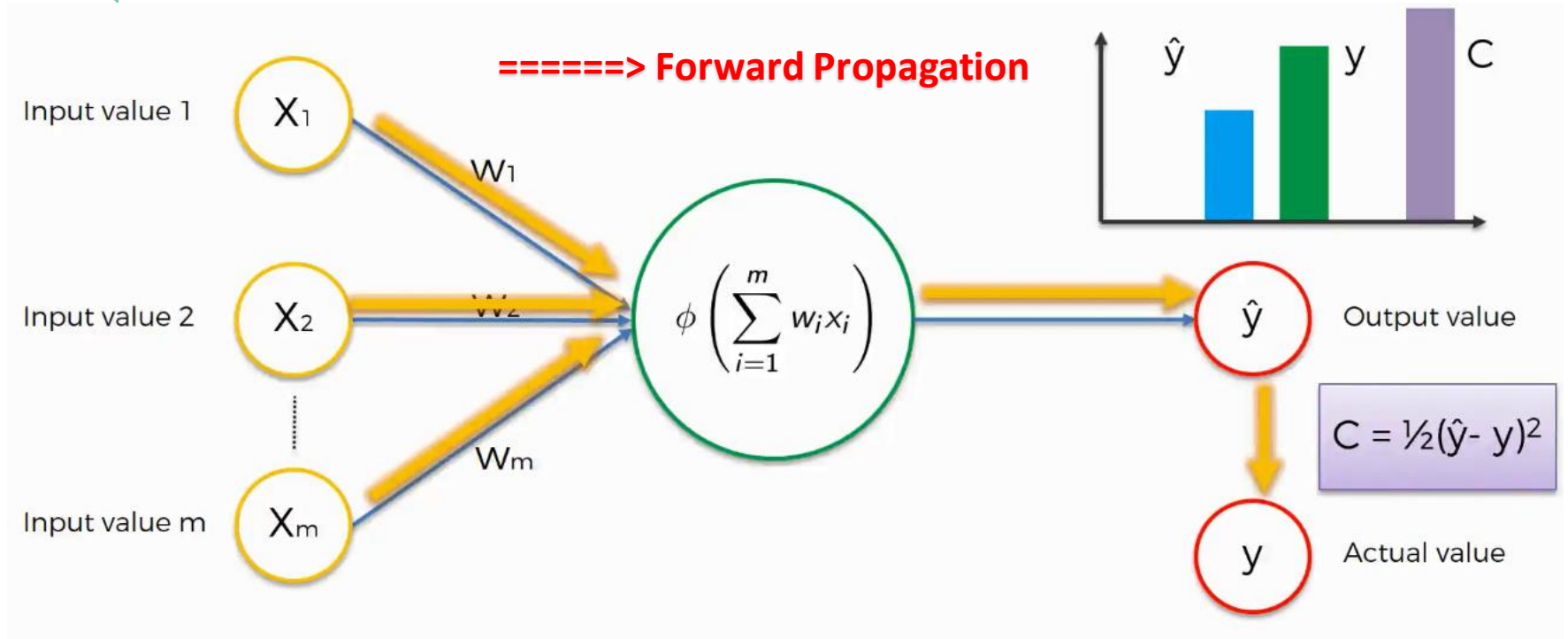


# Función de Activación $\Phi$

Es la función que activa la emisión de una señal, transforma la información neta suministrada por otras neuronas en un valor. Las funciones de activación más comunes son:

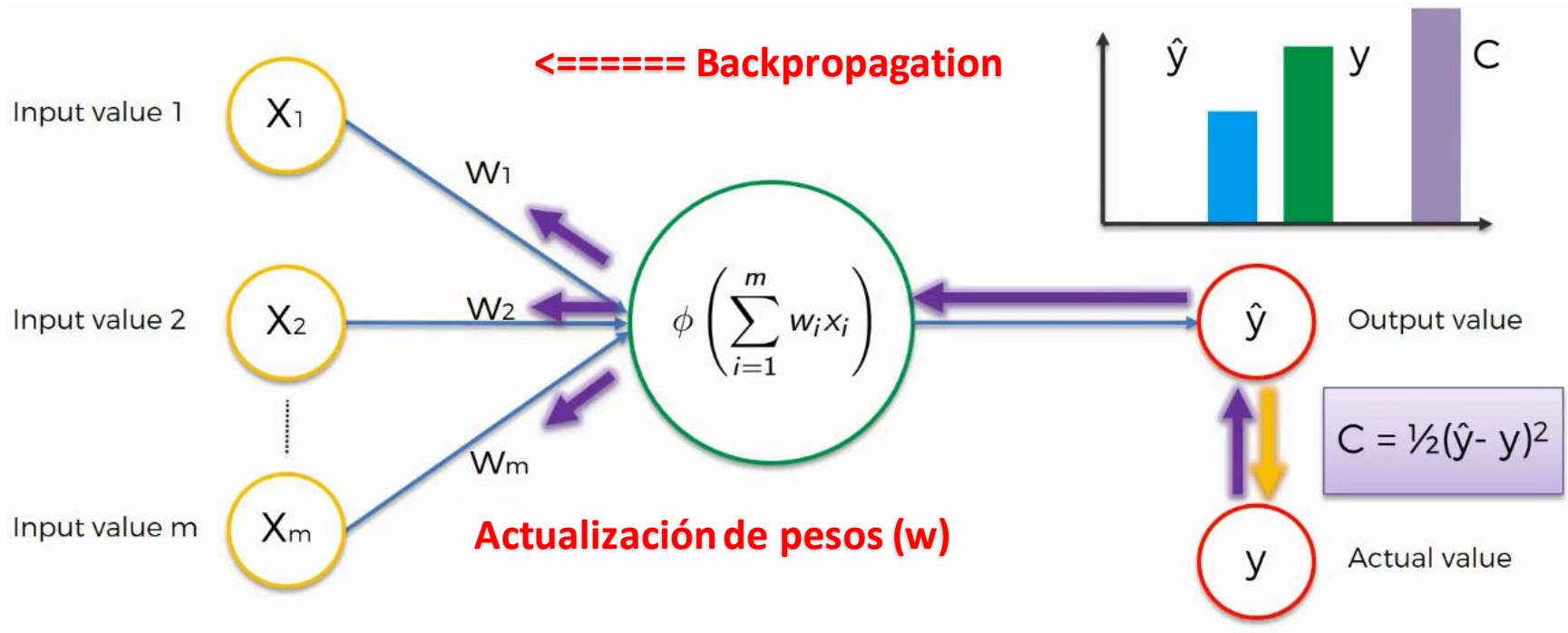


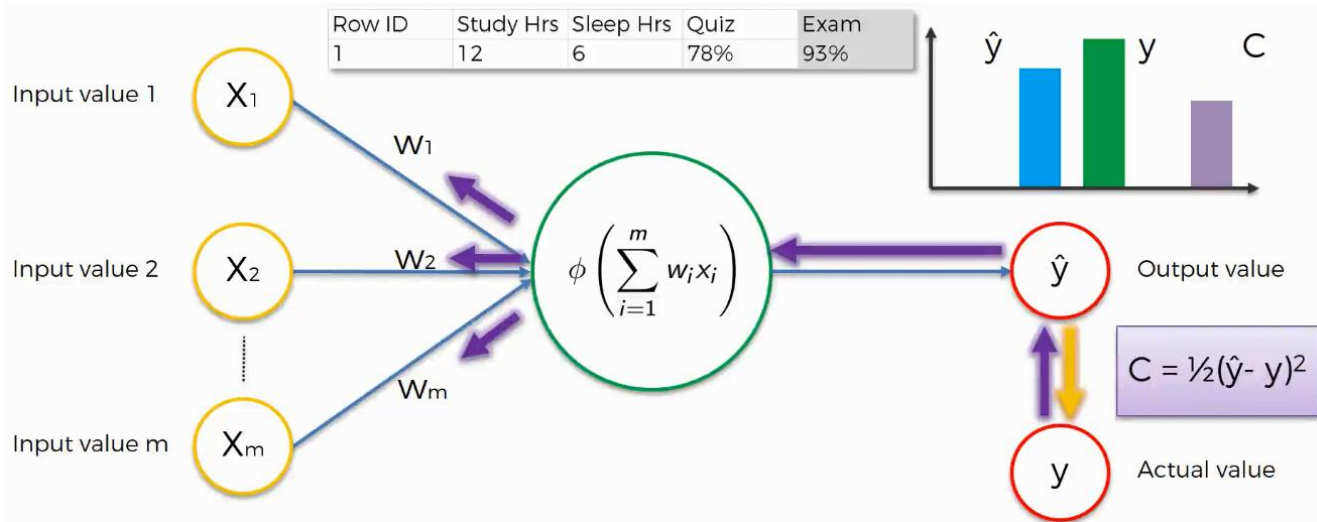
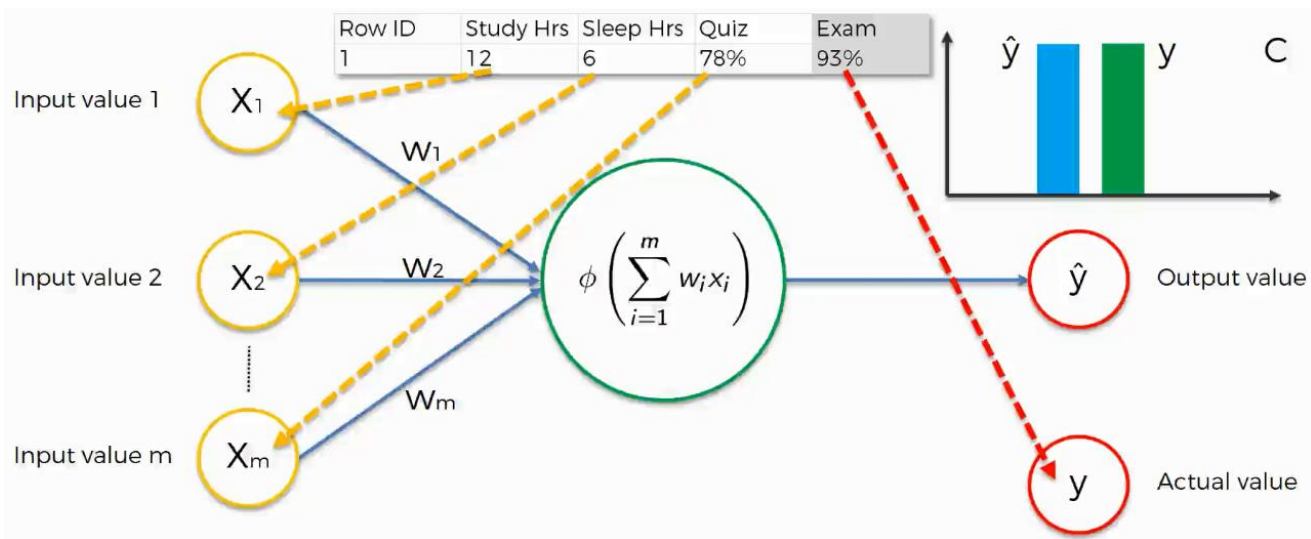
# ¿Cómo aprende el perceptrón (NN)?

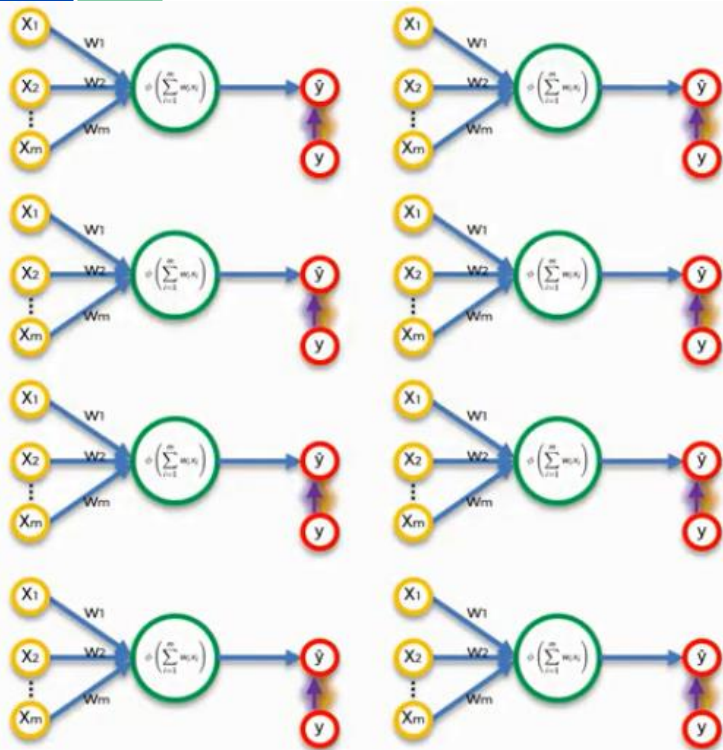




# ¿Cómo aprende el perceptrón (NN)?

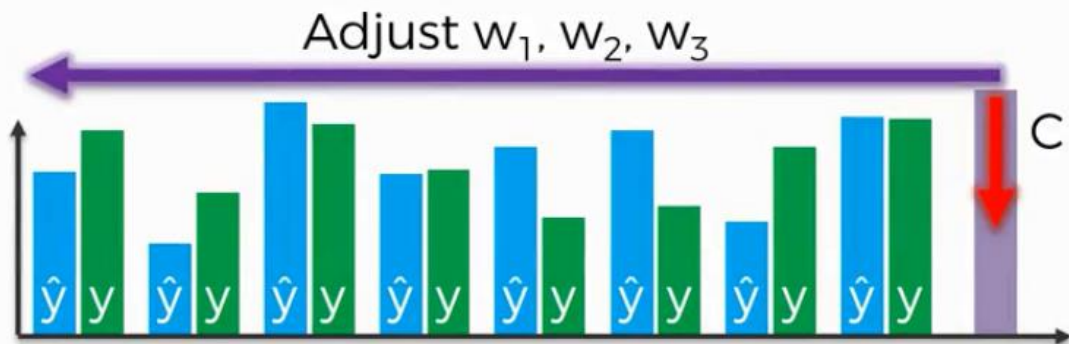






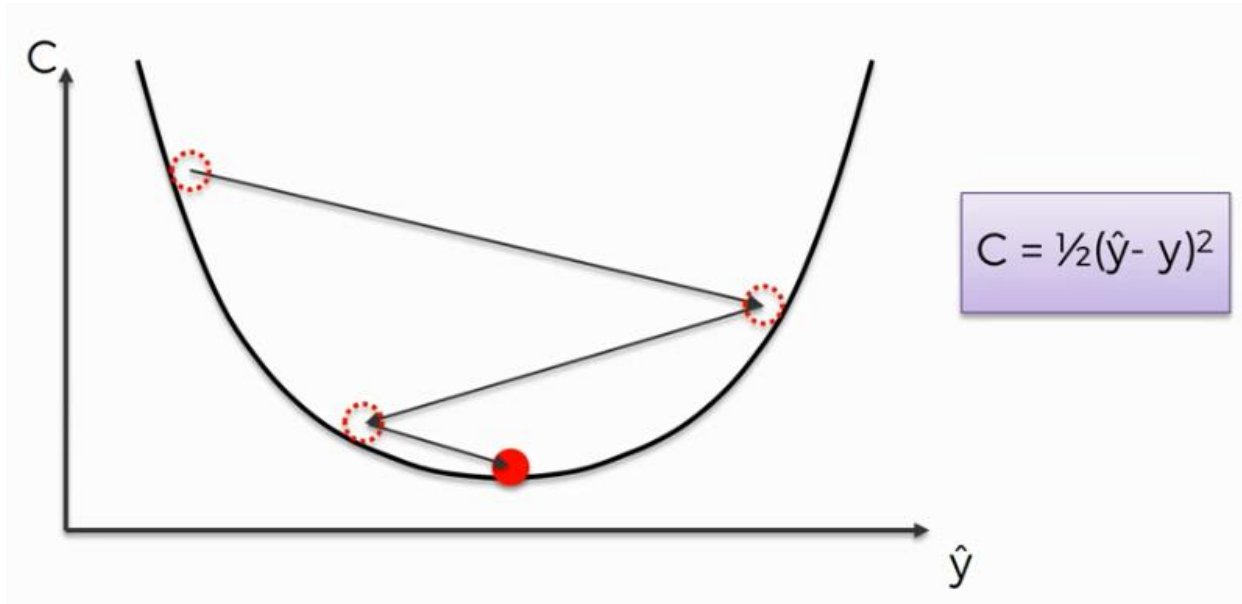
| Row ID | Study Hrs | Sleep Hrs | Quiz | Exam |
|--------|-----------|-----------|------|------|
| 1      | 12        | 6         | 78%  | 93%  |
| 2      | 22        | 6.5       | 24%  | 68%  |
| 3      | 115       | 4         | 100% | 95%  |
| 4      | 31        | 9         | 67%  | 75%  |
| 5      | 0         | 10        | 58%  | 51%  |
| 6      | 5         | 8         | 78%  | 60%  |
| 7      | 92        | 6         | 82%  | 89%  |
| 8      | 57        | 8         | 91%  | 97%  |

$$C = \sum \frac{1}{2} (\hat{y} - y)^2$$



# Gradient Descent

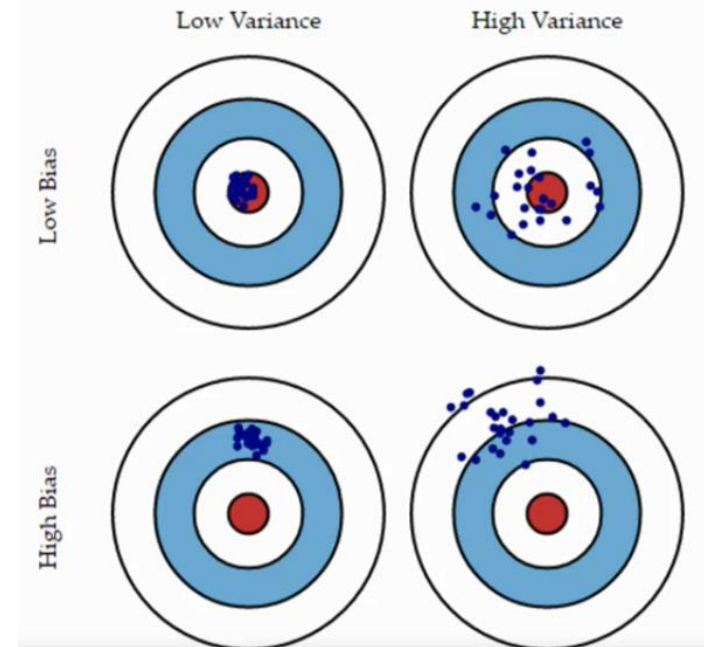
Algoritmo de optimización iterativo

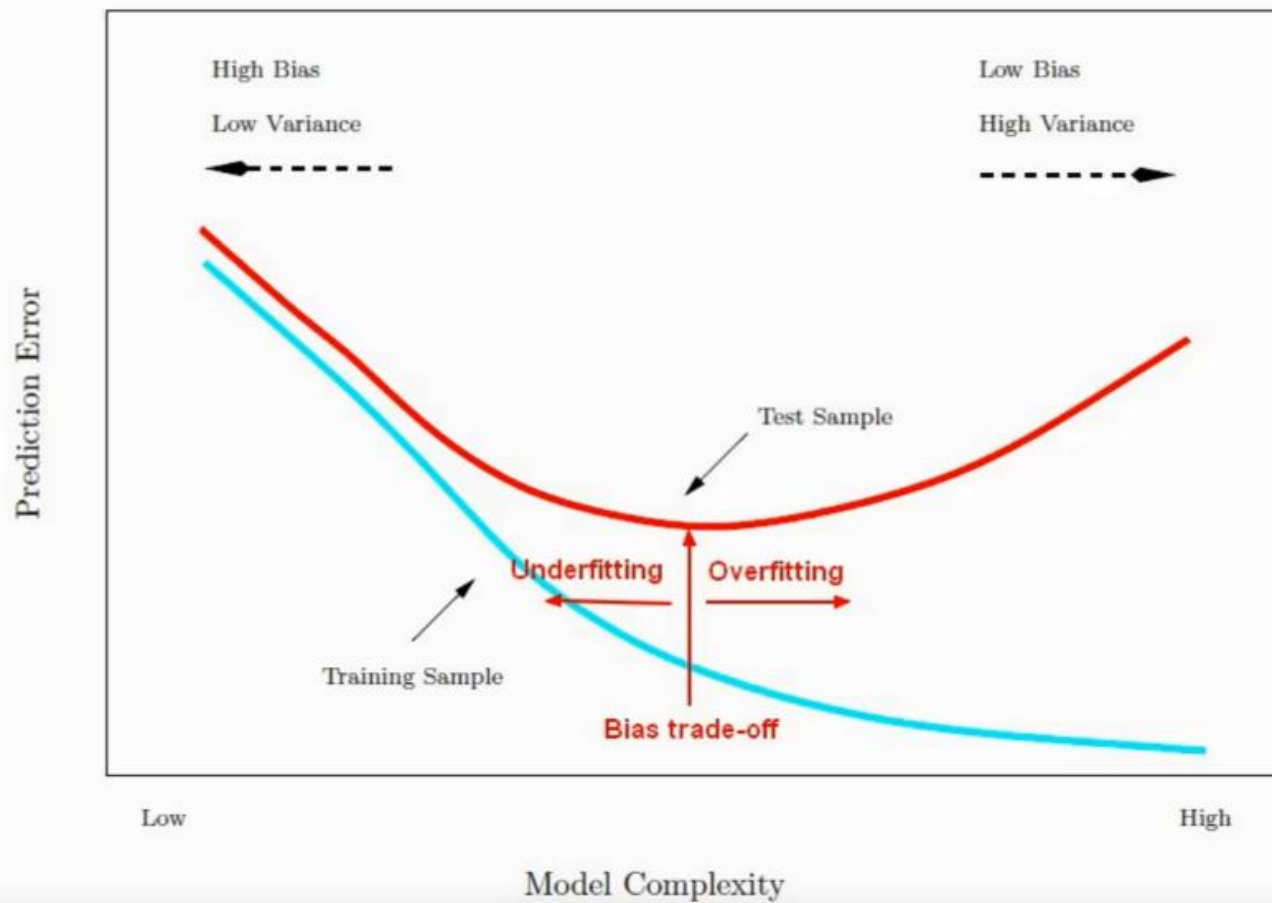
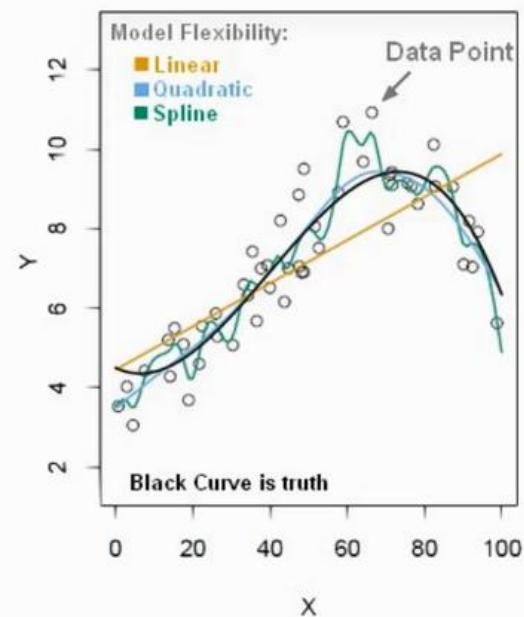


# Bias – Variance Trade off

1. La disyuntiva de sesgo-varianza es cuando agregamos ruido agregando complejidad del modelo (flexibilidad)
2. El error de entrenamiento disminuye, pero el error de prueba empieza a subir
3. El modelo después del trade-off del sesgo comienza a sobreajustarse.

[Figura] El centro es el objetivo es un modelo que predice perfectamente los valores correctos. A medida que nos alejamos del objetivo, nuestras predicciones empeoran cada vez más.

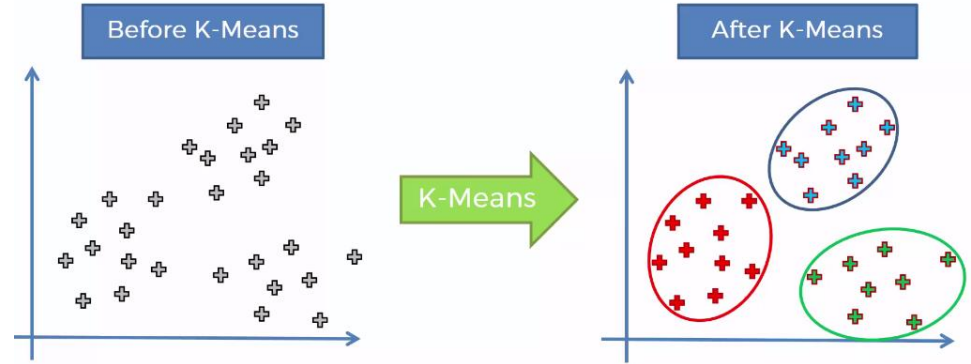




# Modelos No Supervisados

# K-Means

Es una técnica de Aprendizaje No Supervisado para **variables continuas**. K-Means tiene como objetivo dividir  $n$  objetos en  $k$  clústeres en los que cada objeto pertenece al clúster con la media más cercana. El objetivo de la agrupación de K-Means es minimizar la varianza intra-cluster total, o la función de error cuadrática.



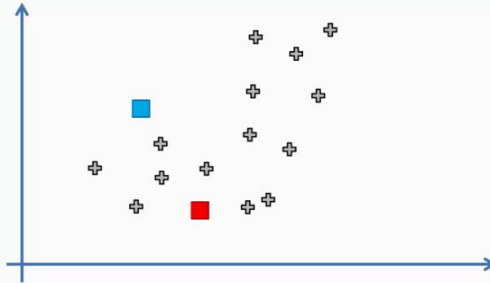
## Pasos del Algoritmo

1. Se agrupa los datos en  $k$  grupos donde  $k$  está predefinido.
2. Seleccionar  $k$  puntos al azar como centros de clúster.
3. Asigne objetos a su centro de clúster más cercano según la función de distancia euclidiana.
4. Calcule el centroide o la media de todos los objetos en cada grupo.
5. Repita los pasos 2, 3 y 4 hasta que se asignen los mismos puntos a cada grupo en rondas consecutivas.

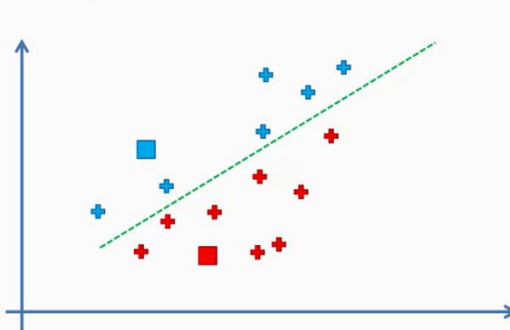
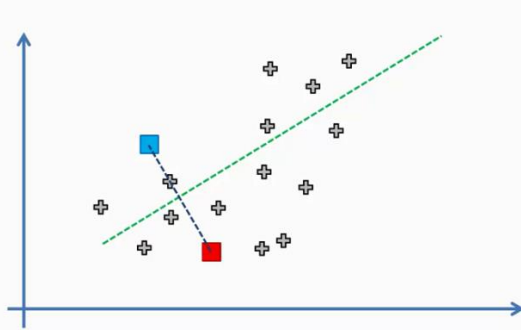


STEP 1: Choose the number K of clusters:  $K = 2$

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)

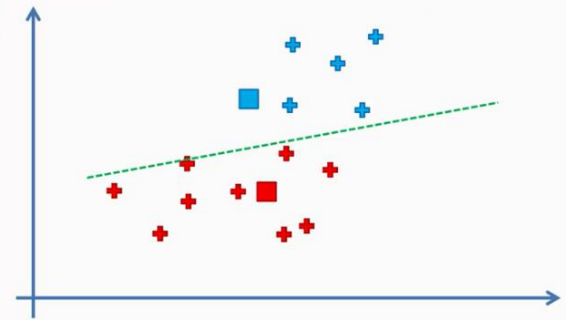
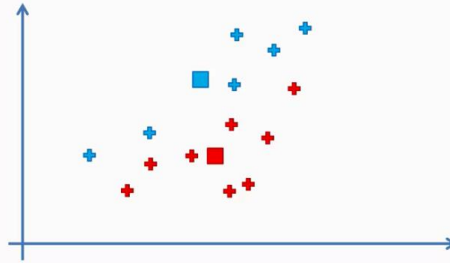
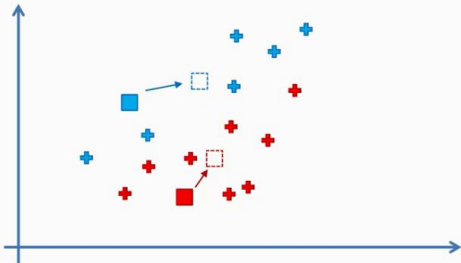


STEP 3: Assign each data point to the closest centroid → That forms K clusters



STEP 4: Compute and place the new centroid of each cluster

STEP 5: Reassign each data point to the new closest centroid. If any reassignment took place, go to STEP 4, otherwise go to FIN.



# K - Modes

Es una técnica de Aprendizaje No Supervisado para agrupamiento de datos que solo tienen **variables categóricas**. Define los clústeres según el número de categorías coincidentes entre los puntos de datos. La distancia entre los puntos de datos se calcula por la cantidad de no coincidencias de sus variables.

- Se reemplaza el concepto de clusters por modes.
- Se usa el método basado en frecuencia para encontrar los modes.

- Dissimilarity measure

$$d_1(X, Y) = \sum_{j=1}^m \delta(x_j, y_j)$$

where

$$\delta(x_j, y_j) = \begin{cases} 0 & (x_j = y_j) \\ 1 & (x_j \neq y_j) \end{cases}$$

- Mode of a set

A mode of  $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$  is a vector  $\mathbf{Q} = [q_1, q_2, \dots, q_m]$  minimise

$$D(\mathbf{X}, \mathbf{Q}) = \sum_{i=1}^n d_1(X_i, \mathbf{Q})$$

# K – Prototypes

Es una técnica de Aprendizaje No Supervisado para agrupamiento de **variables continuas y categóricas**. El algoritmo k-prototypes combina los algoritmos k-modes y k-means para poder agrupar data mixta (numérica y categórica).

$$d_2(X, Y) = \underbrace{\sum_{j=1}^p (x_j - y_j)^2}_{\text{Distancia Euclidiana (K-Means)}} + \gamma \underbrace{\sum_{j=p+1}^m \delta(x_j, y_j)}_{\text{Medida de No Matching (K-Modes)}}$$

Distancia Euclidiana (K-Means)      Medida de No Matching (K-Modes)

## Función de Costo

Minimizar  $\implies P(W, Q) = \sum_{l=1}^k \left( \sum_{i=1}^n w_{i,l} \sum_{j=1}^p (x_{i,j} - q_{l,j})^2 + \gamma \sum_{i=1}^n w_{i,l} \sum_{j=p+1}^m \delta(x_{i,j}, q_{l,j}) \right)$



**Y SI COMBINAMOS  
VARIOS MODELOS?...**

# Métodos Ensamblados

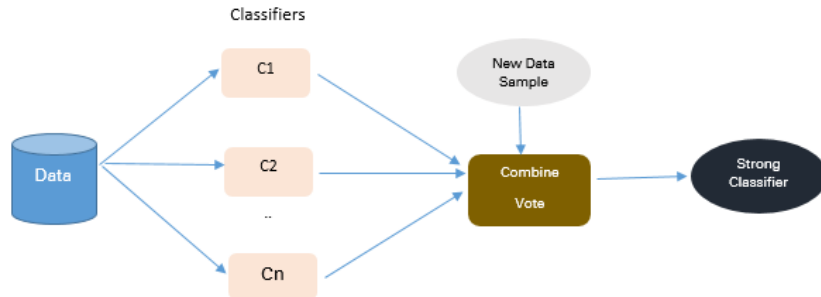
El aprendizaje ensamblado (o "conjunto") es el proceso de combinar varios modelos predictivos para producir un modelo combinado que es más preciso que cualquier modelo individual.

- Regresión: tomar el promedio de las predicciones.
- Clasificación: vote y use la predicción más común, o tome el promedio de las probabilidades.

**Bagging:** Votación por mayoría. Varios clasificadores diferentes votan para decidir la clase de un caso de prueba (usa clase de un caso de prueba (usa bootstrapping bootstrapping)).

**Boosting:** Votación ponderada. Los clasificadores tienen distintos pesos en la votación (en función de su precisión).

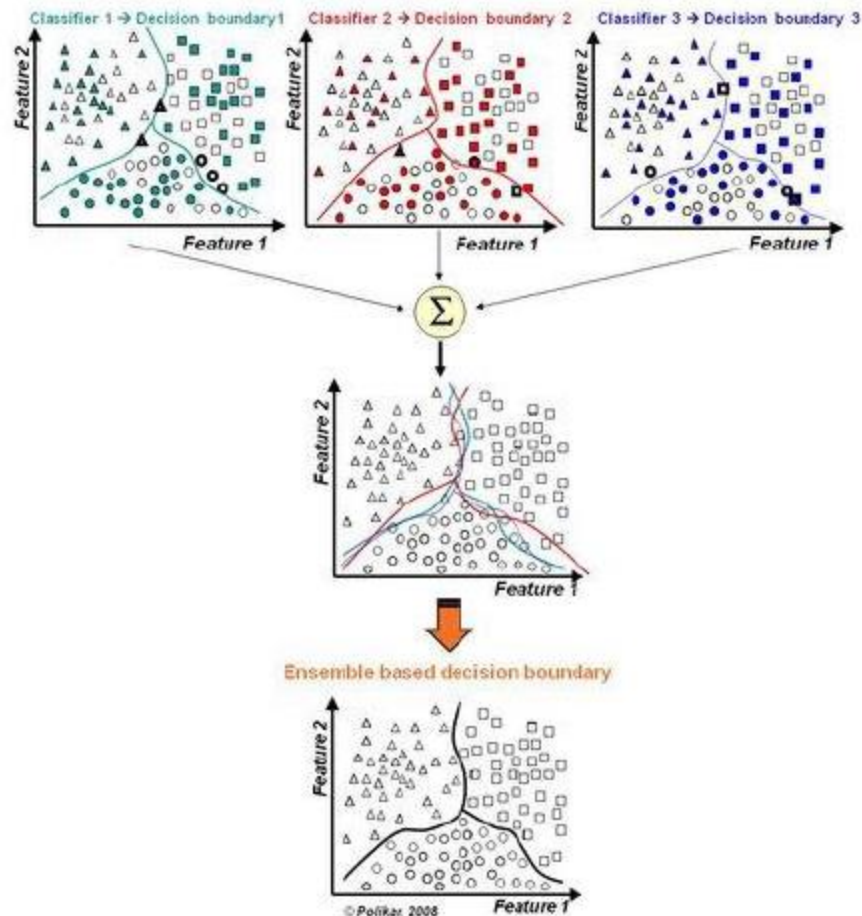
**Stacking:** Formamos un nuevo dataset con las predicciones de los modelos a ensamblar para entrenar un nuevo modelo que generará las predicciones finales.



*"Si tienes todas las opiniones de un comité de expertos, considéralas todas para tomar una decisión"*



# Métodos Ensamblados



# Métodos Ensamblados (Bagging)

**Bagging** definido a partir de “bootstrap aggregating”.

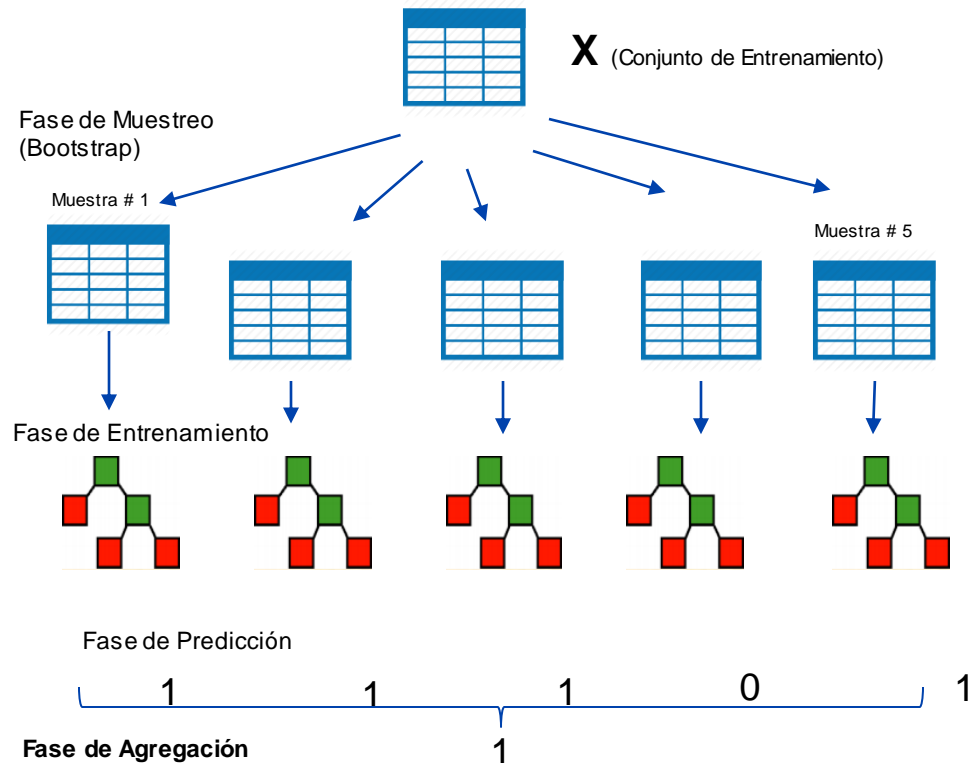
**Bagging.**[Breiman,94] Repeat for  $t = 1, \dots, T$ :

- Select, at random *with replacement*,  $N$  training examples.
- Train learner on selected samples to generate  $h_t$

Final hypothesis is simple vote:

$$H(x) = MAJ(h_1(x), \dots, h_T(x))$$

“Si tienes todas las opiniones de un comité de expertos, considéralas todas para tomar una decisión”



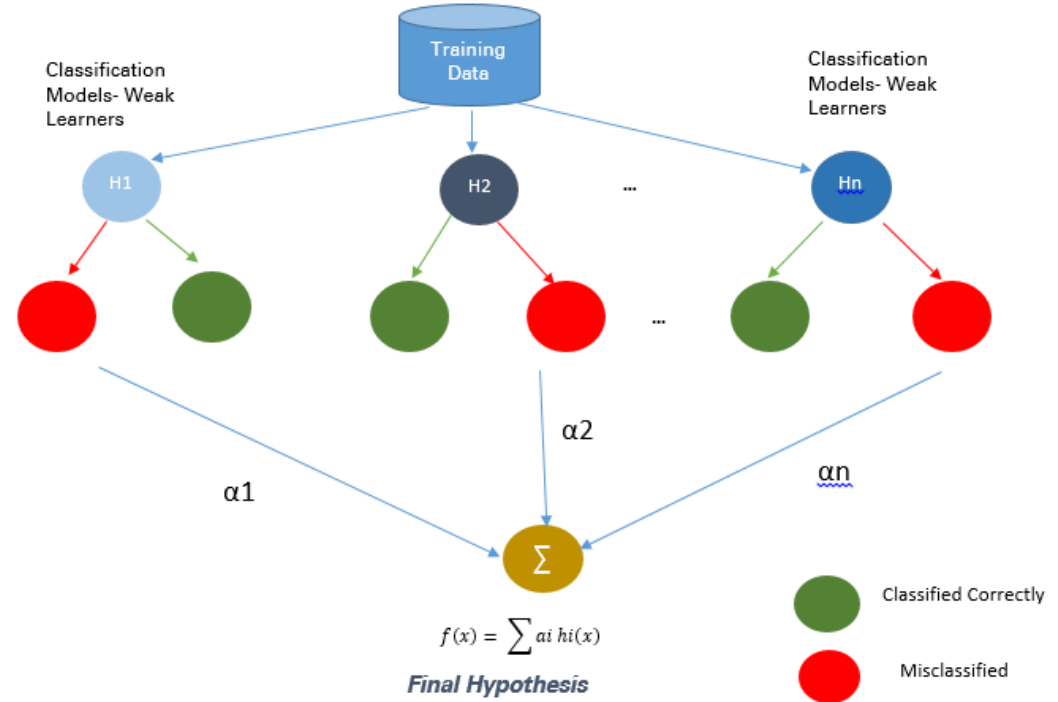
# Métodos Ensamblados (Boosting)

## Muestreo ponderado:

- En lugar de hacer un muestreo aleatorio de los datos de entrenamiento, se ponderan las muestras para concentrar el aprendizaje en los ejemplos más difíciles.
- Intuitivamente, los ejemplos más cercanos a la frontera de decisión son más difíciles de clasificar, y recibirán pesos más altos.

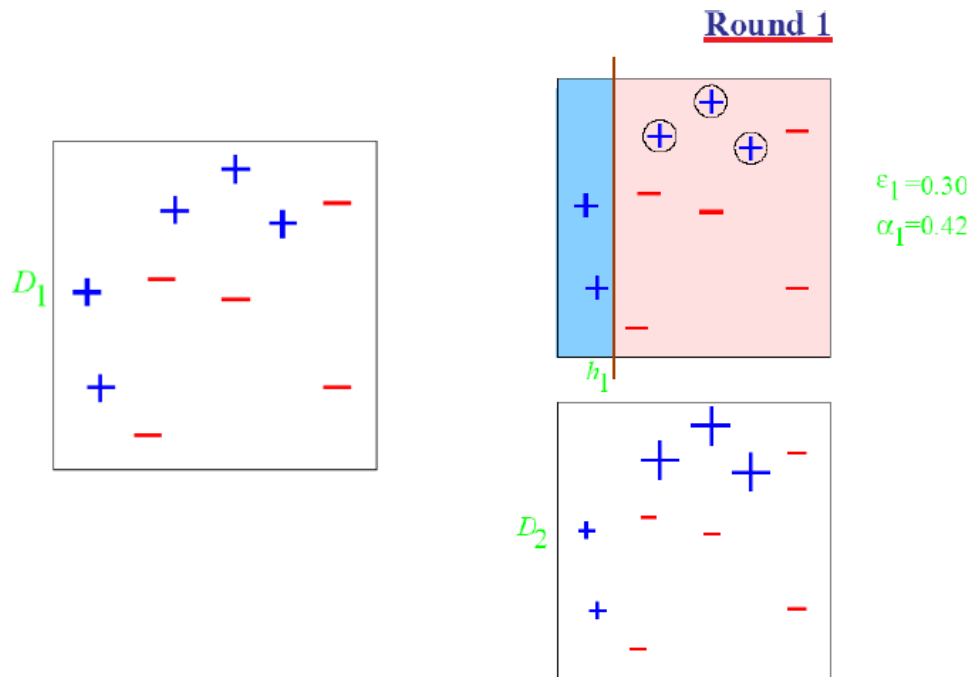
## Votos ponderados (clasificadores):

- En lugar de combinar los clasificadores con el mismo peso en el voto, se usa un voto ponderado.
- Esta es la regla de combinación para el conjunto de clasificadores débiles.
- En conjunción con la estrategia de muestreo anterior, esto produce un clasificador más fuerte.



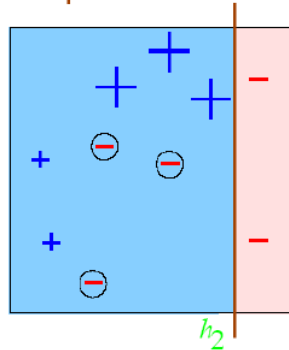
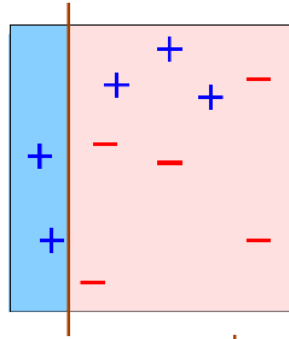


# Métodos Ensamblados (Boosting)



# Métodos Ensamblados (Boosting)

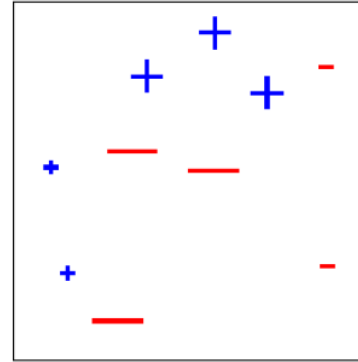
Round 2



$$\varepsilon_2 = 0.21$$

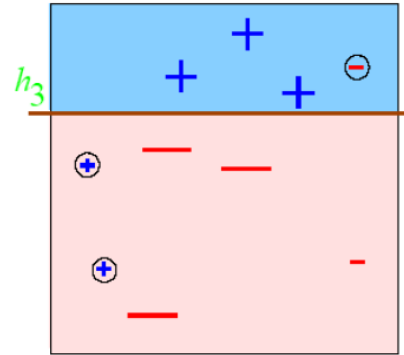
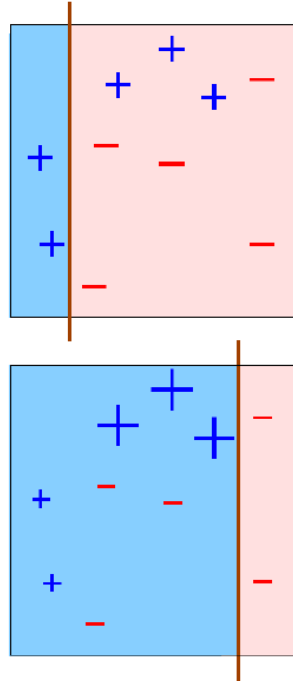
$$\alpha_2 = 0.65$$

$D_3$



# Métodos Ensamblados (Boosting)

Round 3



$$\epsilon_3 = 0.14$$

$$\alpha_3 = 0.92$$

# Métodos Ensamblados (Boosting)

## Final Hypothesis

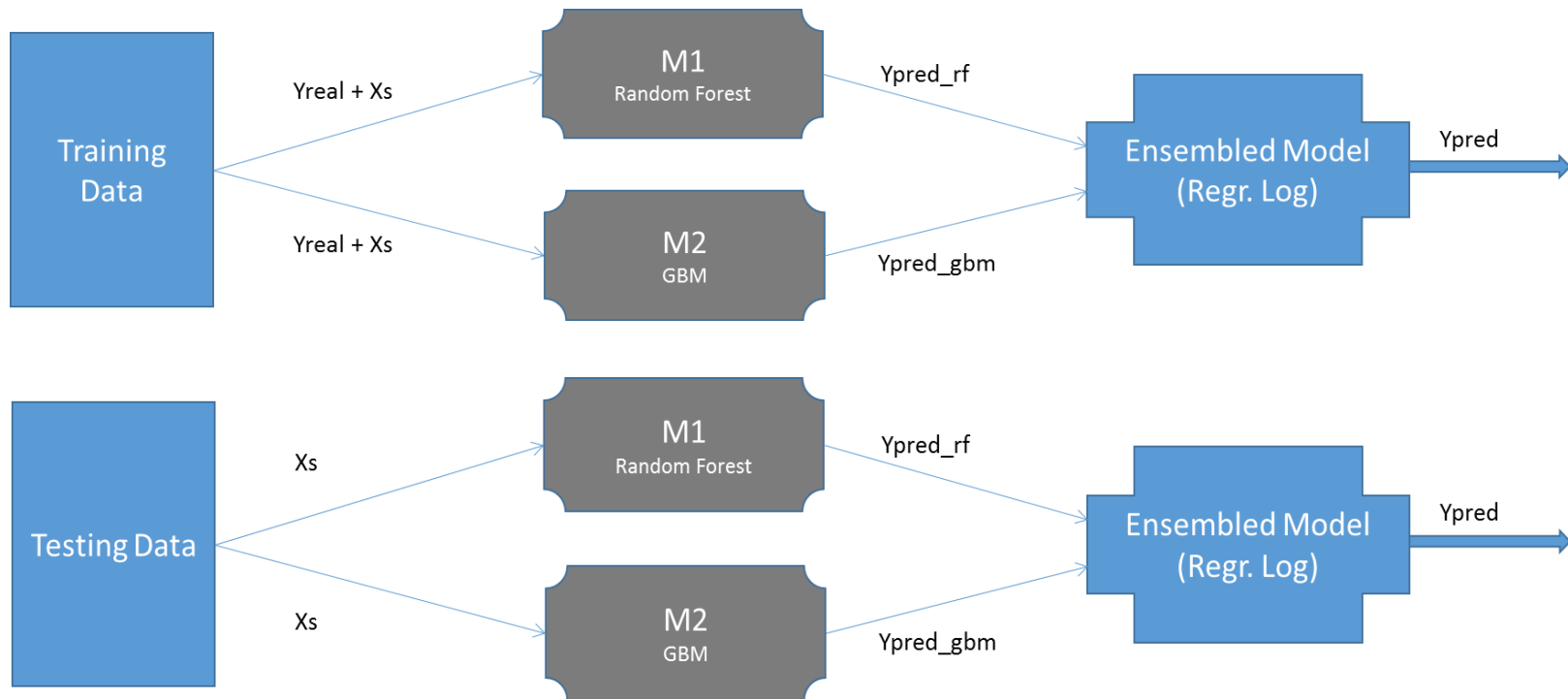
$$H_{\text{final}} = \text{sign} \left( 0.42 \begin{array}{|c|c|} \hline \text{blue} & \text{red} \\ \hline \end{array} + 0.65 \begin{array}{|c|c|} \hline \text{blue} & \text{red} \\ \hline \end{array} + 0.92 \begin{array}{|c|c|} \hline \text{blue} & \text{red} \\ \hline \end{array} \right)$$
  

$$= \begin{array}{|c|c|c|} \hline \text{blue} & \text{blue} & \text{red} \\ \hline \text{blue} & \text{red} & \text{red} \\ \hline \end{array}$$

The diagram illustrates the final hypothesis  $H_{\text{final}}$  as a weighted sum of three weak hypotheses. Each hypothesis is represented by a square divided into two regions (blue and red) by a vertical line. The weights are 0.42, 0.65, and 0.92. The final hypothesis is the result of combining these, shown as a square divided into four regions (blue and red) by two vertical lines. The regions are labeled with '+' for blue and '-' for red.

# Métodos Ensamblados (Stacking)

Formamos un nuevo dataset con las predicciones de los modelos a ensamblar para entrenar un nuevo modelo que generará las predicciones finales.



# Enfoque de la Solución

## Ensamblado – Promedio Ponderado

LightGBM\_10



CV: 0.972421

PUB: 0.97447

LB PRIV: 0.97384

$\times 5/8$

XGB\_19



CV: 0.972767

PUB: 0.97440

LB PRIV: 0.97414

$\times 1/8$

$\times 2/8$

LightGBM\_26



CV: 0.973760

PUB: 0.97437

LB PRIV: 0.97444

$f_x$

| test_id | is_female_x | is_female_y | is_female_z | is_female |
|---------|-------------|-------------|-------------|-----------|
| 0       | 0           | 0.996615    | 0.994650    | 0.994433  |
| 1       | 1           | 0.061414    | 0.097544    | 0.057922  |
| 2       | 2           | 0.986808    | 0.984562    | 0.978972  |
| 3       | 3           | 0.986533    | 0.978258    | 0.986918  |
| 4       | 4           | 0.467452    | 0.589511    | 0.637860  |

CV: 0.973982

PB: 0.97488

PRIV: 0.97460

|     | a | b | c | metric   |
|-----|---|---|---|----------|
| 0   | 0 | 0 | 1 | 0.972421 |
| 10  | 0 | 1 | 0 | 0.972767 |
| 120 | 1 | 0 | 0 | 0.973760 |
| 132 | 1 | 1 | 1 | 0.973839 |

|   | a | b | c | metric   |
|---|---|---|---|----------|
| 0 | 5 | 1 | 2 | 0.973982 |