



# Machine Learning y analítica predictiva

Profesor: César Hormazábal

LinkedIn: [linkedin.com/in/hormazabalcesar](https://www.linkedin.com/in/hormazabalcesar)

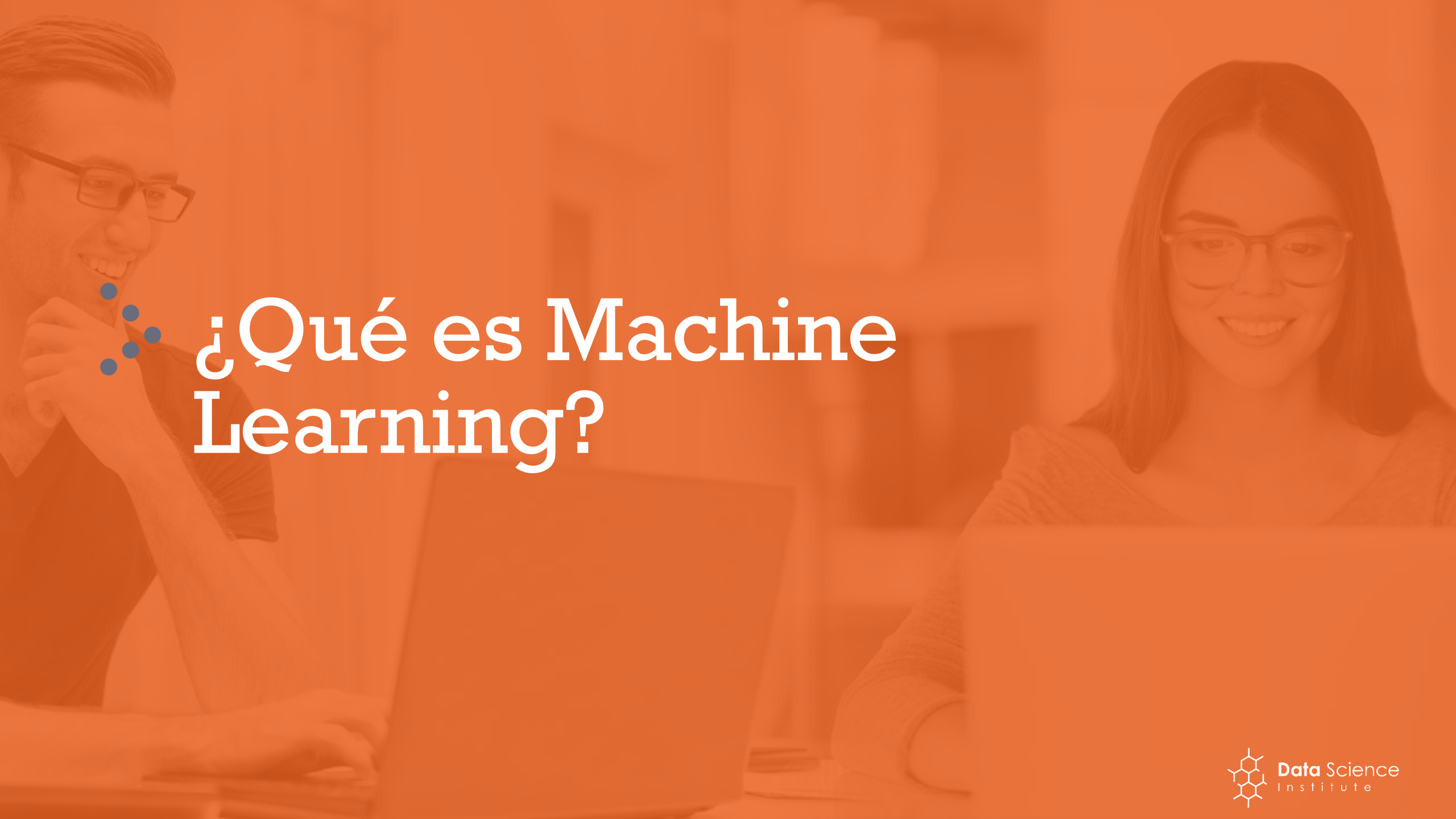
Email: [cesar.hormazabal@ihlab.cl](mailto:cesar.hormazabal@ihlab.cl)

Fecha: 6 de septiembre de 2018



# Agenda

1. Introducción
2. Tipos de modelos en Machine Learning y Aplicaciones.
3. Regresiones y sus diferencias
4. Modelos de Clasificación.
5. Técnicas Extra
6. Como llevar un proyecto de machine learning exitosamente.



# ¿Qué es Machine Learning?

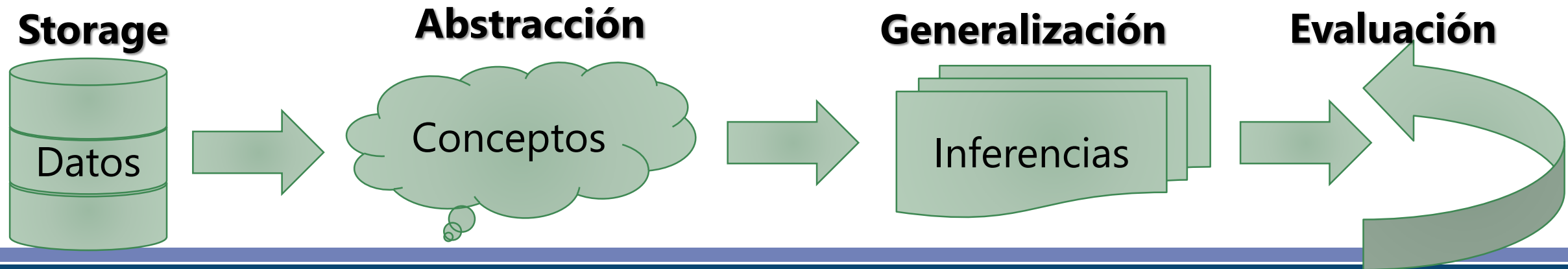
# Machine Learning



¿Cómo aprenden las máquinas?

*"A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$  if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ ."*

Tom M.  
Mitchell (1997)



# Supongamos que estamos buscando un enfoque para encontrar fraude crediticio.

---

Nombre	Monto	Fraude
Juan	2500	No
Pedro	1300	Sí
Pablo	5000	Sí
Andrea	7000	No

---

¿Existe algún patrón que  
podamos encontrar en la  
data?

# Ahora examinemos el siguiente ejemplo:

---

Nombre	Monto	De dónde es la cuenta	Dónde fue usado	Edad	Fraude
Juan	2500	Chile	Brasil	30	No
Pedro	1300	Rusia	España	22	Sí
Pablo	5000	Rusia	Chile	26	Sí
Andrea	7000	Colombia	Argentina	40	No

---

Con más data empezamos a encontrar más patrones.

¿Qué pasa si la data empieza a crecer a millones?

No se pueden encontrar patrones visualmente, por lo cual se hace uso de técnicas estadísticas para encontrar aquellos patrones.



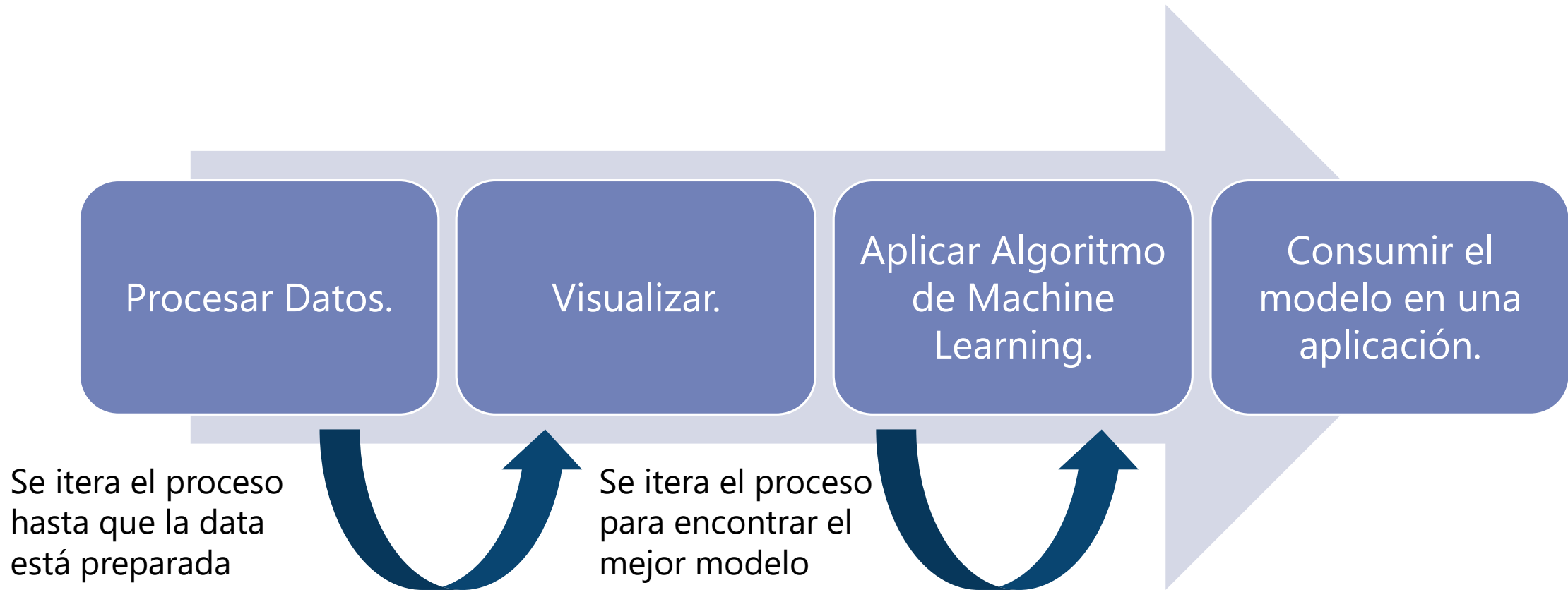
# Machine Learning

---

Lo que hace un proceso de machine learning es:

- Aplicar técnicas Estadísticas para encontrar patrones.
- Generar una implementación que reconoce aquel patrón. (Es decir, un código).
- Este código es conocido como un **modelo** y puede ser llamado por las aplicaciones que quieran resolver el problema.

# Proceso de Machine Learning



# Procesar Datos

---

¿Qué datos son importantes para el modelo?

¿Tienen sentido los datos para el formato del problema?

¿Existe alguna transformación de los datos que haga que tengan sentido?

# Visualizar Datos

---

¿Puedo  
descubrir  
relaciones entre  
los datos?

Debido a las  
dificultades de nuestra  
visión, se recomienda  
buscar las relaciones de  
a pares

Se visualizan  
tendencias,  
dispersiones.

# Aplicar Modelo de Machine Learning.

---

Al aplicarlo  
tendremos cierto  
nivel de certeza  
del modelo.

Es el momento de  
compararlo con  
otros modelos.

Se compara en base al  
éxito del modelo y el  
costo de cálculo que  
suponga, además de que  
tan comprensible es.

# Aplicaciones de Machine Learning

---

Anti-Fraude.

Trading algorítmico.

Fondos de inversión manejados por IA

Anti-Lavado de dinero.

Análisis de transacciones en tiempo real.

Credit-Scoring.

Inversión automática.

Modelos de Fuga y Propensión de Clientes.

# Aplicaciones de Machine Learning

---

JPMorgan -> Algoritmo COIN (Análisis automático de documentos).

-> Emerging Opportunities Engine

Wells Fargo->Chatbot para la mensajería de la compañía.

Citibank ->FeedzAI (Detector de fraudes)

ChileCompra->Detector de Fraude en compra públicas.

Comparador de precios automático

# Discusión de Primer Caso

---

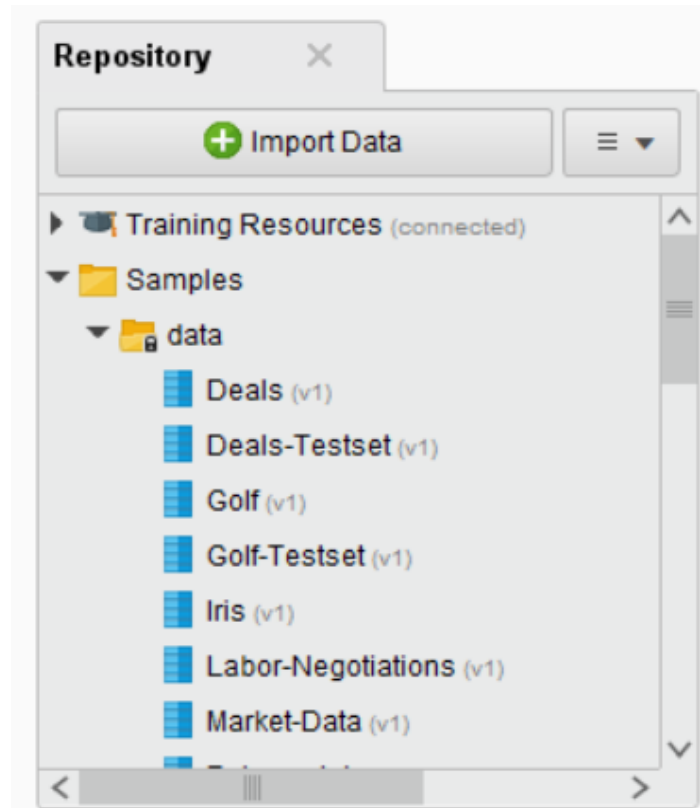
Piense 5 situaciones en su compañía donde la velocidad de reacción a una situación ha sido más lenta de lo necesario, discuta si sería posible pasarla a un modelo de machine learning.

Discuta 5 situaciones donde el volumen de datos es más grande de lo que puede procesar una persona. ¿Cómo lo automatizaría, qué variables serían las importantes?



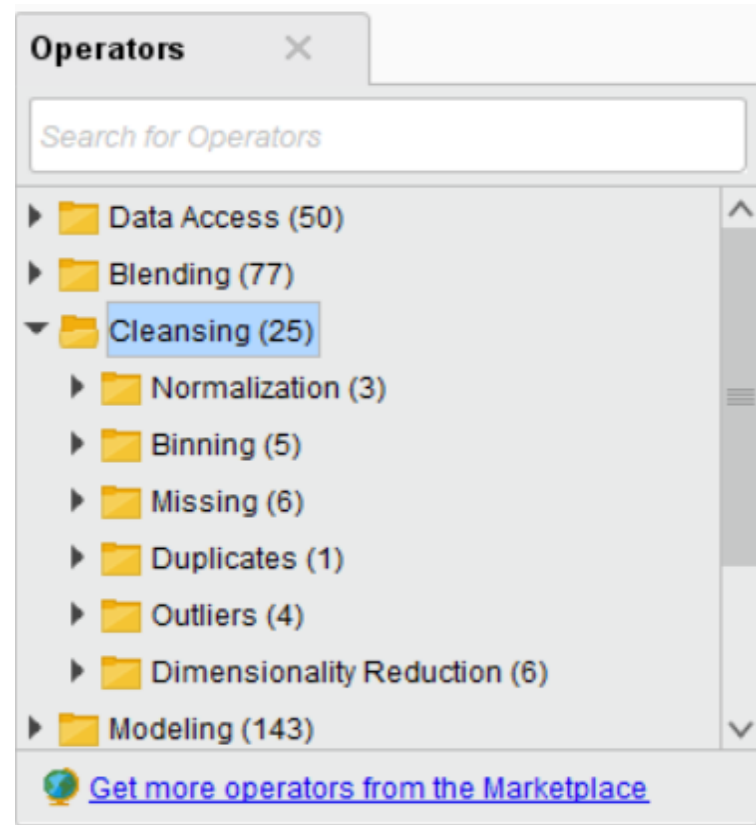
# Ejemplo en Rapid-Miner

Carga de dato en repositorio



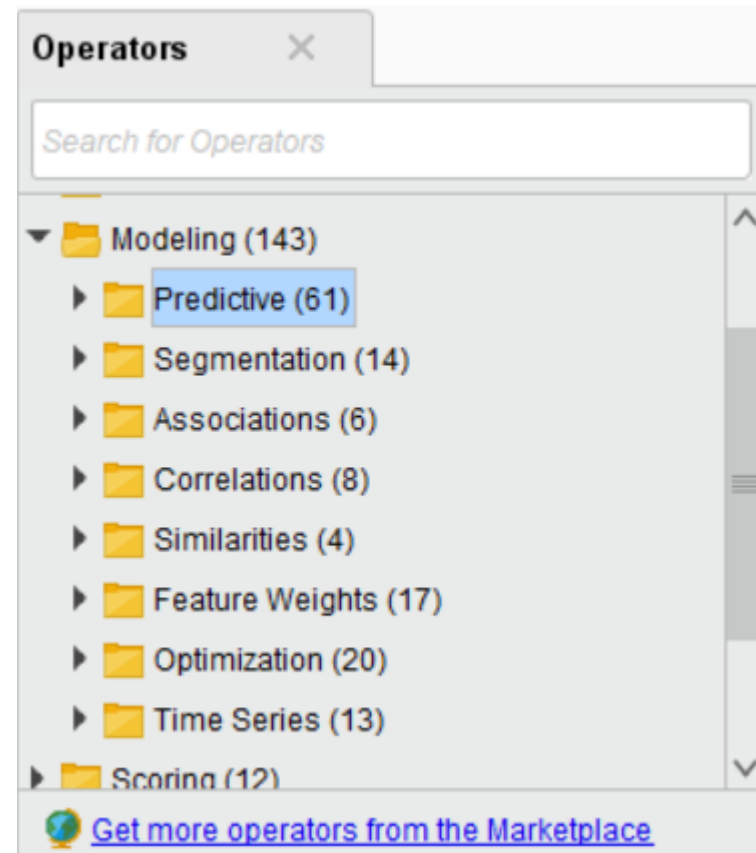
# Ejemplo en Rapid-Miner

Limpieza de Data



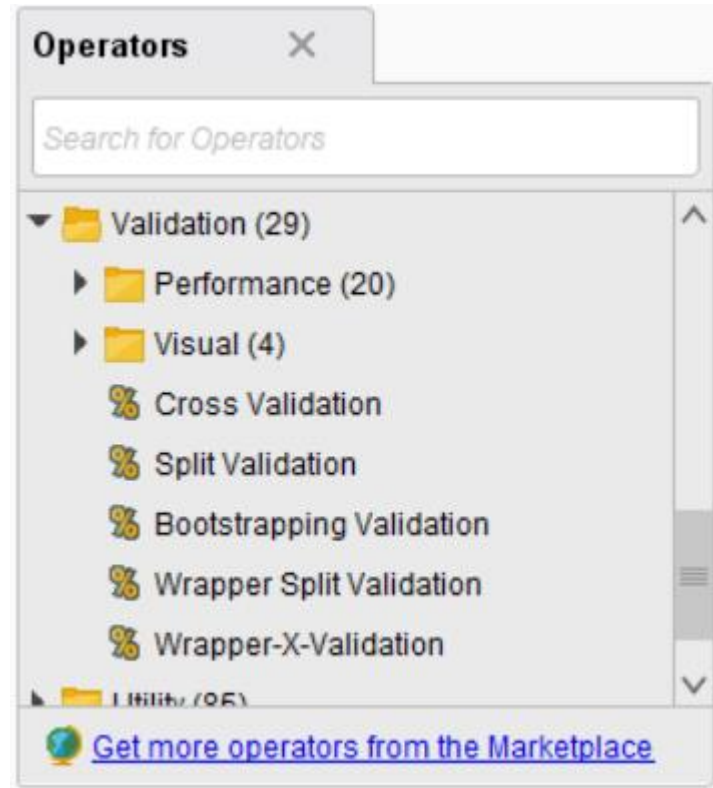
# Ejemplo en Rapid-Miner

Creación de  
Modelo



# Ejemplo en Rapid-Miner

Validación de  
Resultados



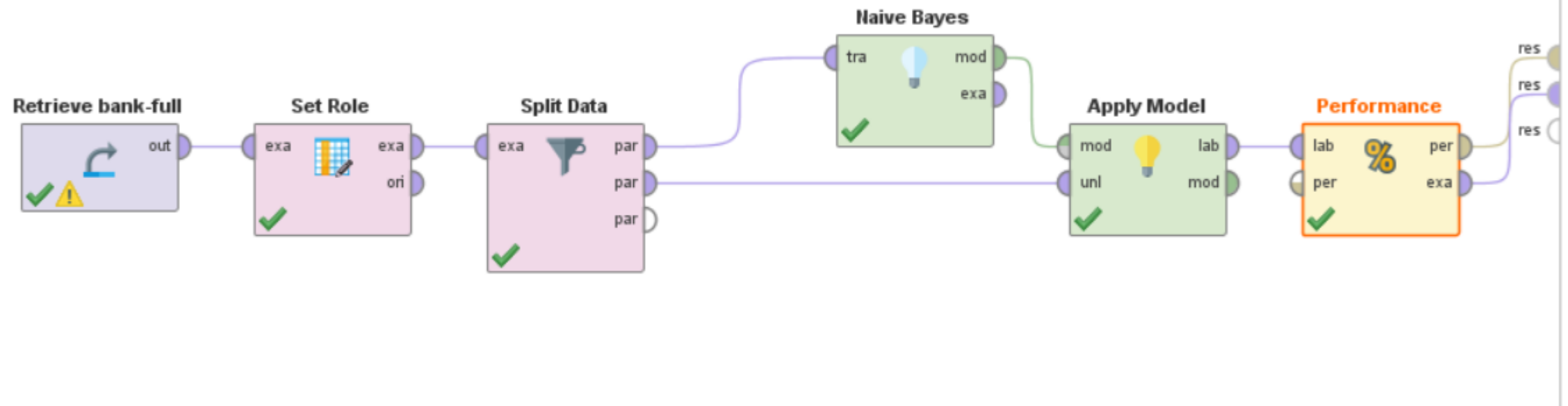
# Creación de un modelo en RM

---

1. Cargue la data de banco.
2. Asigne roles y tipos pertinentes.
3. Use un modelo de árbol para generar un modelo predictivo.
4. Valide su nivel de error.

Process

p



accuracy: 95.73%

	true no	true yes	class precision
pred. no	2819	52	98.19%
pred. yes	76	53	41.09%
class recall	97.37%	50.48%	

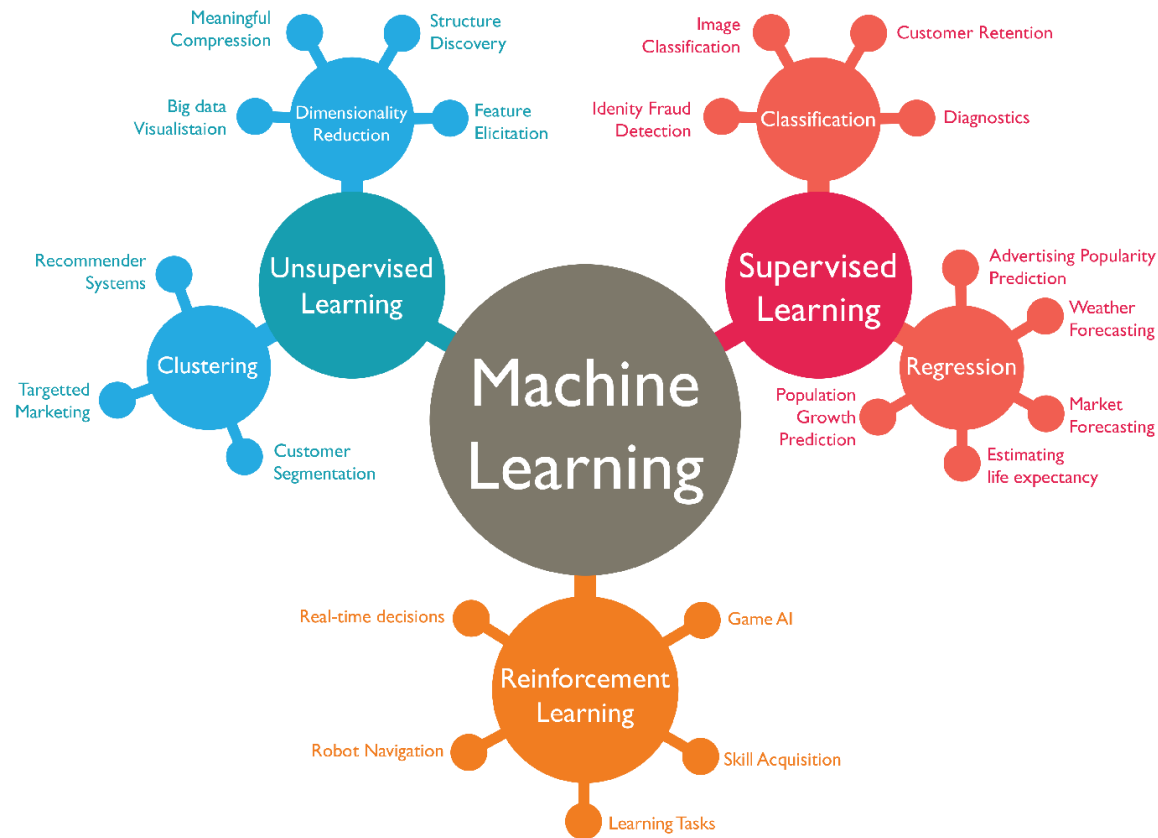


# Modelos ML



# Machine Learning

¿Qué tipo de aprendizajes existen?



# Aprendizaje Supervisado vs No Supervisado

---

El objetivo del **aprendizaje supervisado**, es mapear la relación entre un conjunto de variables y su respuesta.

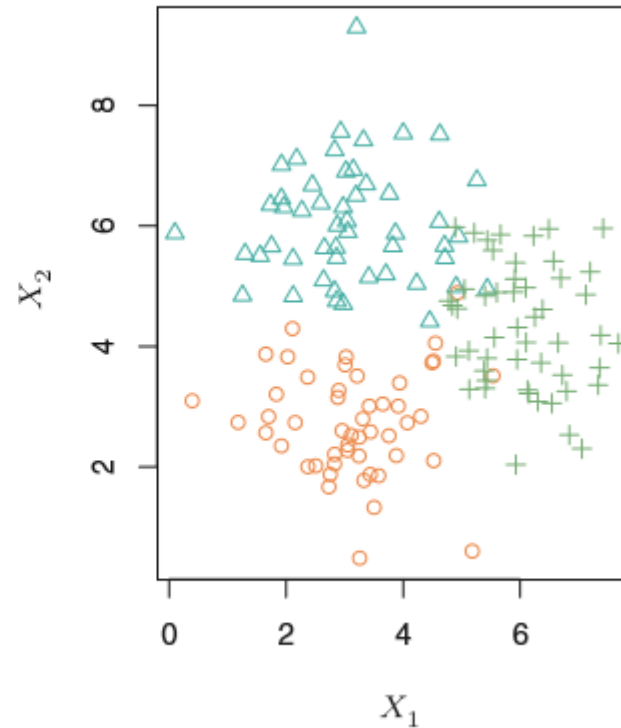
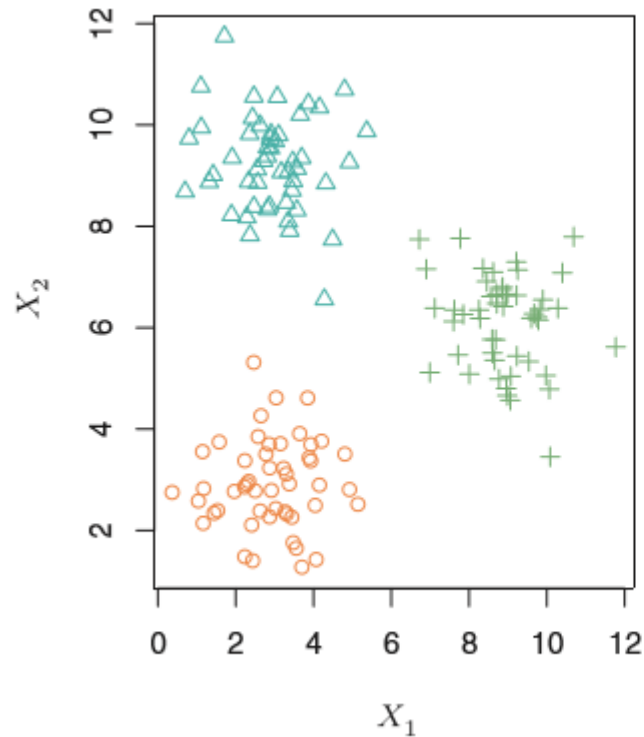
Esta relación no es necesariamente una función, es decir para un mismo conjunto de variables, pueden haber observaciones que tengan valores distintos.

La relación encontrada o establecida entre las relaciones de las variables es lo que llamamos un modelo.

En el aprendizaje **no supervisado**, no tenemos nuestra variable de respuesta, así que dado las variables buscamos relaciones o patrones entre ellas.

# Aprendizaje Supervisado vs No Supervisado

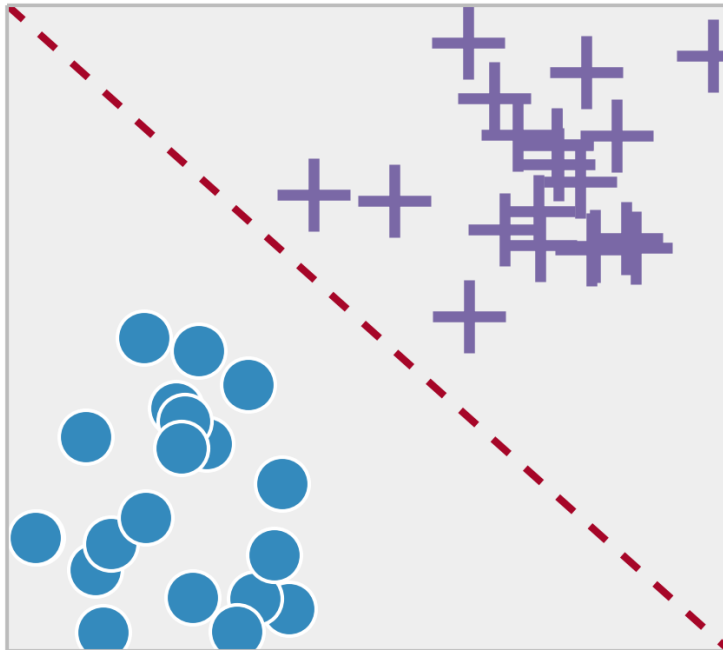
---



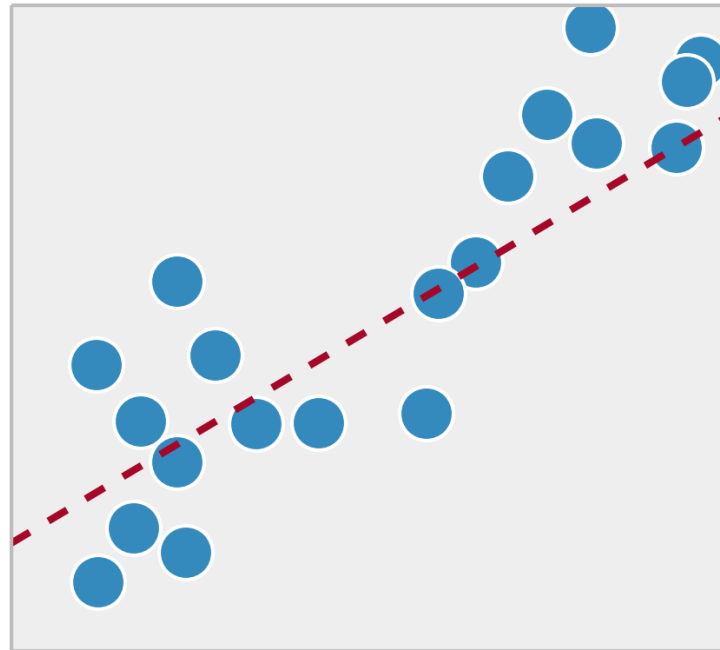
# Regresión vs Clasificación.

---

Clasificación



Regresión



# Aprendizaje supervisado

---

¿Qué es ?

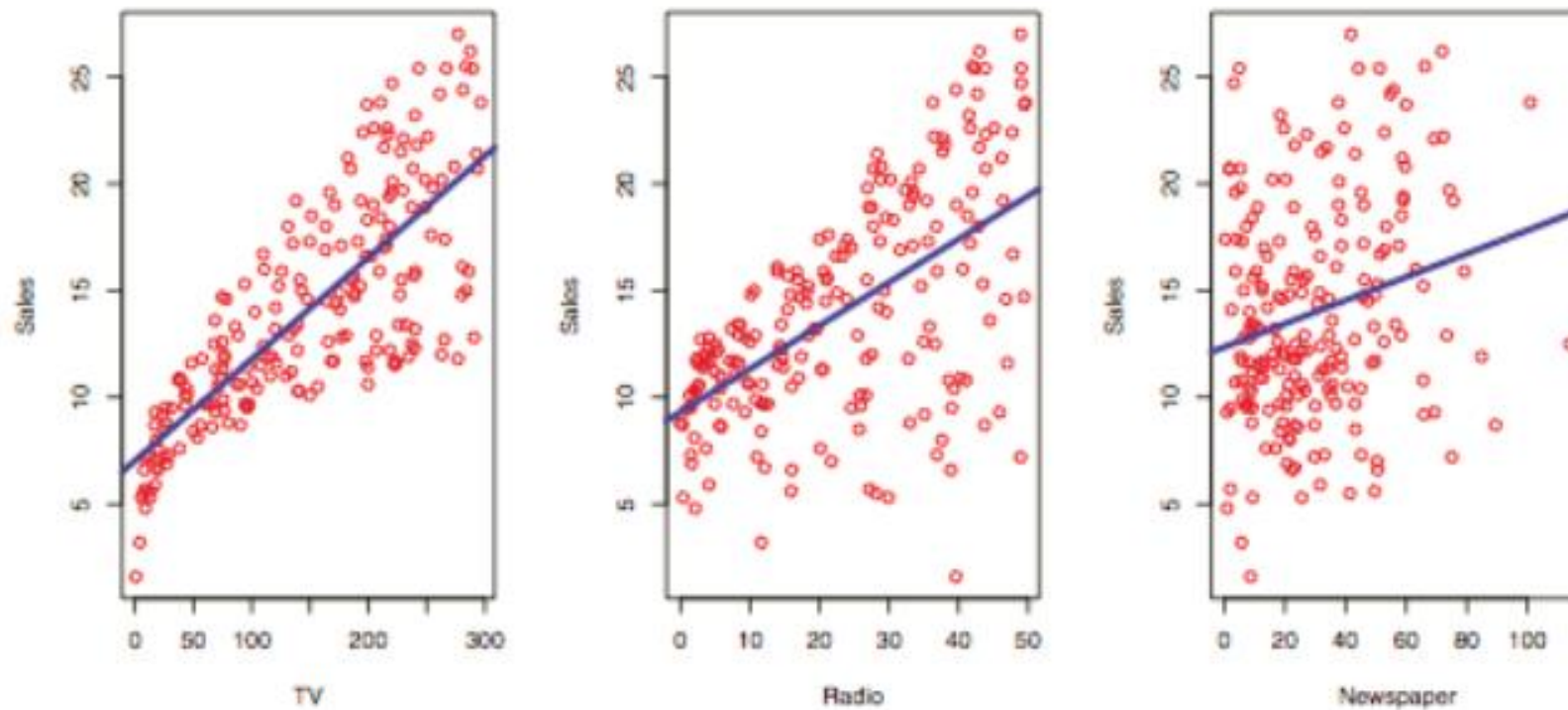
Consideremos la siguiente situación:

- Data: Registro de ventas de 200 productos
- Inversión en campañas para cada producto en TV, Radio y Diario

Necesidad: Ante el lanzamiento de un nuevo producto, ¿Cómo debiera invertir el dinero en campañas para generar la mayor cantidad de ventas? ¿Cuánto dinero y en qué medio?

# Aprendizaje supervisado

---



# Aprendizaje supervisado

---

- ¿Qué elementos definen a este problema (inputs)?
- ¿Qué buscamos finalmente (output)?
- ¿Cómo?



# Aprendizaje supervisado

---

Fórmula

$$\text{sales} = f(\text{TV}, \text{radio}, \text{newspaper}) + \text{noise}$$

Queremos estimar la  
función  $f$ . En general

$$X = (X_1, X_2, \dots, X_p)^T$$

inputs

$Y$

output

$$Y = f(X) + \epsilon$$

relationship



# Aprendizaje supervisado

---

$$\begin{array}{ll} X = (X_1, X_2, \dots, X_p)^T & \text{inputs} \\ Y & \text{output} \\ Y = f(X) + \epsilon & \text{relationship} \end{array}$$

Nos interesa estimar la función  $f$  en dos contextos: regresión y clasificación.

# Aprendizaje supervisado

---

Situación actual:

Tenemos una necesidad definida, nuestra data y observaciones ... *¿cómo estimamos  $f$ ?*

- Modelos paramétricos
- Modelos no paramétricos

# Aprendizaje supervisado

- Contexto supervisado:
  - Existe una variable dependiente que es de interés (usualmente "Y").
  - Se busca explicar ésta última a través predictores (usualmente ("X").
  - Se supone un modelo matemático (paramétrico, no paramétrico, semi paramétrico)
  - Ej: Regresión,  $Y = f(X) + \epsilon$ , (supuestos distribucionales en  $\epsilon$ ).
- Algunos modelos usuales
  - Regresiones (lineales, GLM, no paramétricas, series de tiempo, etc.)
  - KNN, SVM, Redes Neuronales, Árboles de decisión (y sus derivados)



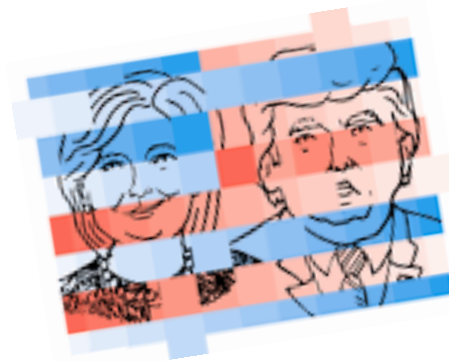
# Aprendizaje supervisado

---

En el contexto supervisado, la variable dependiente (**Y**) puede ser numérica o categórica.

- **Categórica:** Problema de clasificación.
- **Numérica:** Problema de regresión.

Clasificación



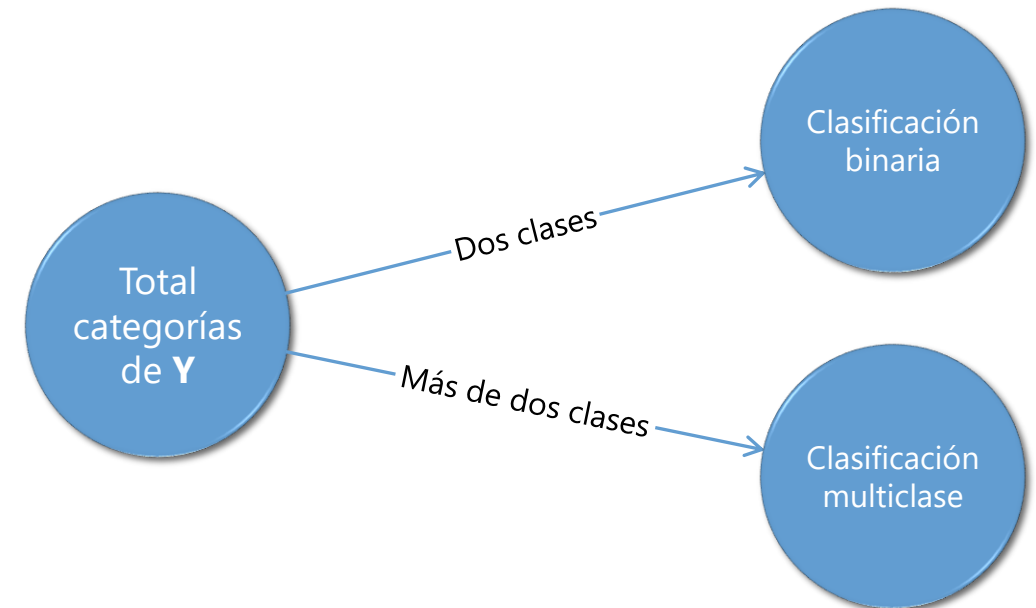
Regresión



# Modelos de Clasificación

Algunos modelos de clasificación

- KNN (K-nearest neighbor).
- SVM (Support Vector Machine).
- Redes Neuronales.
- Árboles de decisión (y sus derivados).
- Regresión logística.
- Clasificador de Bayes.
- LDA (Linear discriminant analysis).
- QDA (Quadratic discriminant analysis).
- Entre otros...

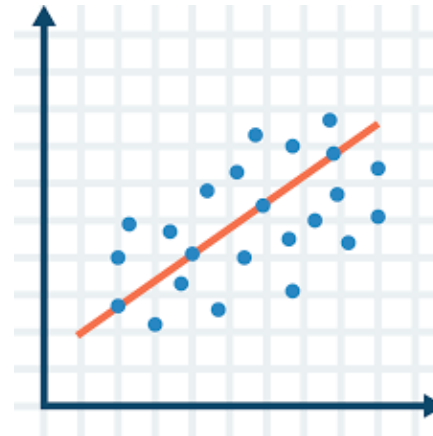


# Modelos de regresión

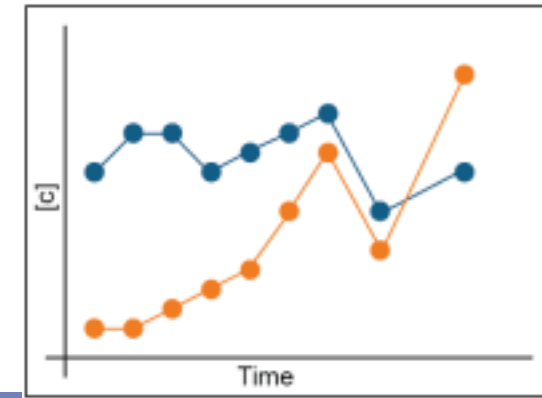
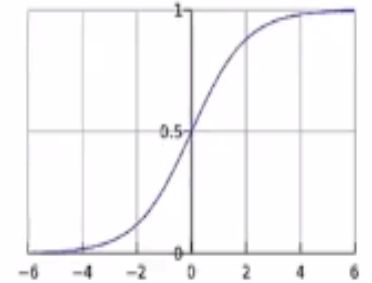
Sólo por mencionar algunos modelos:

- Regresión lineal.
- Regresión logística.
- Regresión Poisson.
- Modelos mixtos
- Modelos aditivos generalizados
- Regresión quantil.
- Regresión no paramétrica.
- ...
- Series temporales.
- Árboles de regresión.
- Regresión KNN.
- Etc...

} GLM



$$g(x) = \ln \frac{\pi(x)}{1 - \pi(x)} = \beta_0 + \beta_1 x$$



# Resumen de modelos de ML

Algoritmo	Tipo de Problema	Nivel de Interpretabilidad	Precisión de los resultados	Velocidad de entrenamiento
KNN	Ambos	Alto	Bajo	Rápido
Regresión Lineal	Regresión	Alto	Bajo	Rápido
Regresión Logística	Clasificación	Medio	Bajo	Rápido
Naive Bayes	Clasificación	Medio	Bajo	Rápido
Decision Tree	Ambos	Medio	Bajo	Rápido
Random Forest	Ambos	Bajo	Alto	Lento
AdaBoost	Ambos	Bajo	Alto	Lento
Red Neuronal	Ambos	Nulo	Alto	Lento

# Resumen de modelos de ML

Algoritmo	Velocidad de Predicción	Nivel de ajuste de parámetros a realizar	Funciona con un nivel mínimo de datos	Separa bien la señal del ruido
KNN	Rápido	Mínimo	No	No
Regresión Lineal	Rápido	Nulo	Sí	No
Regresión Logística	Rápido	Nulo	Sí	No
Naive Bayes	Rápido	Algo	Sí	Sí
Decision Tree	Rápido	Algo	No	No
Random Forest	Moderada	Algo	No	Sí
AdaBoost	Rápido	Algo	No	Sí
Red Neuronal	Rápido	Mucho	No	Sí



# Resumen de modelos de ML

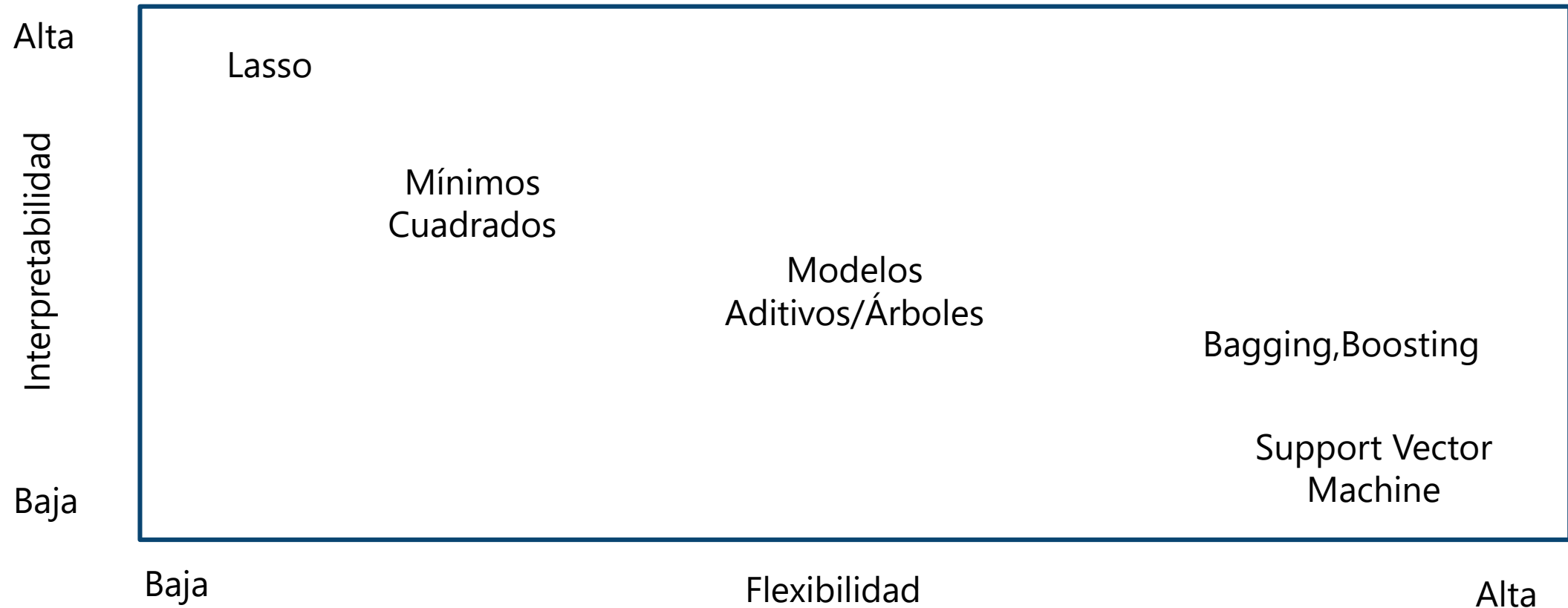
Algoritmo	Modela la interacción entre variables	Entrega la probabilidad de pertenecer a una clase	¿Paramétrico?	Necesita escalar la data	Método o librería asociada en R
KNN	No	Sí	No	Sí	knn
Regresión Lineal	No	N/A	Sí	No	lm
Regresión Logística	No	Sí	Si	No	glm
Naive Bayes	No	No	Sí	No	e1071::naiveBayes
Decision Tree	Sí	Posiblemente	No	No	rpart
Random Forest	Sí	Posiblemente	No	No	ranger
AdaBoost	Sí	Posiblemente	No	No	adaboost
Red Neuronal	Sí	Posiblemente	No	Sí	neuralnet



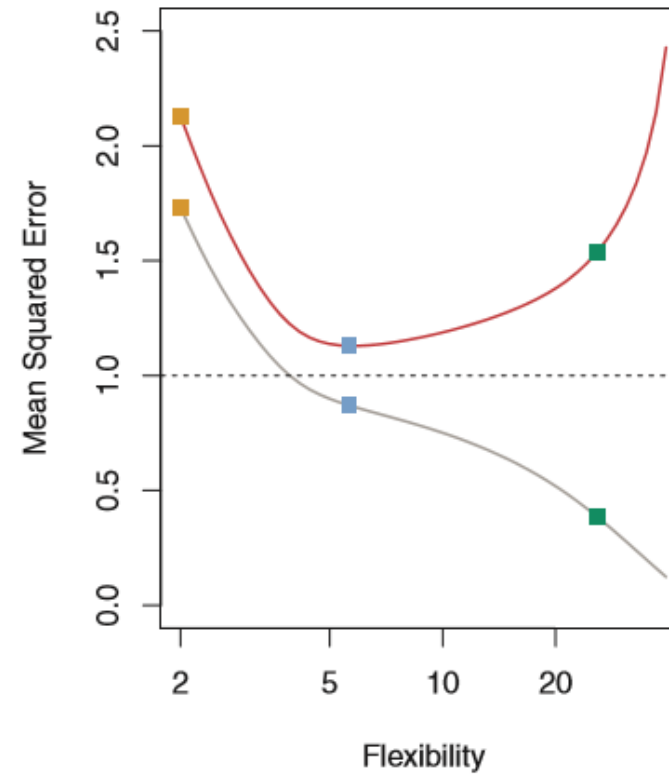
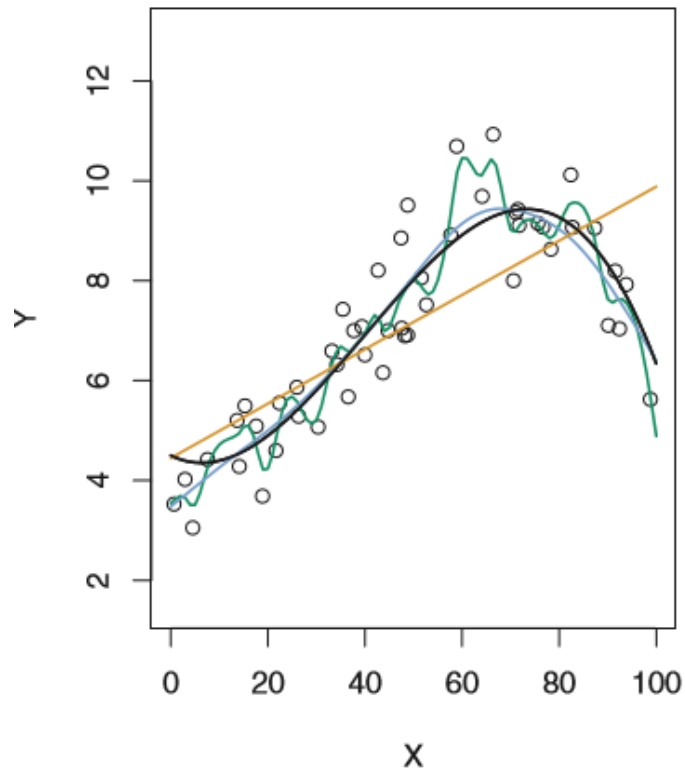
# ••• Evaluación de modelos de ML

Cómo saber si voy por un buen camino

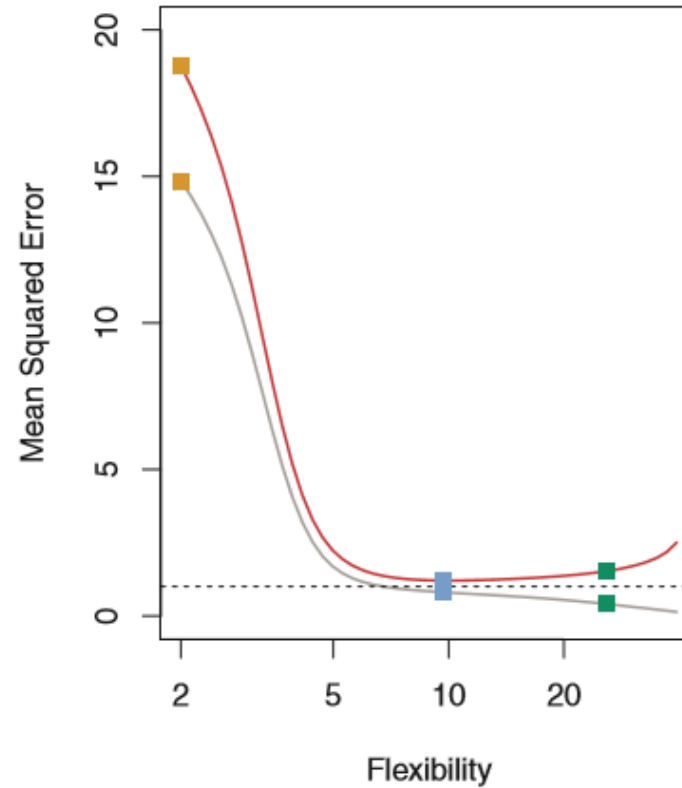
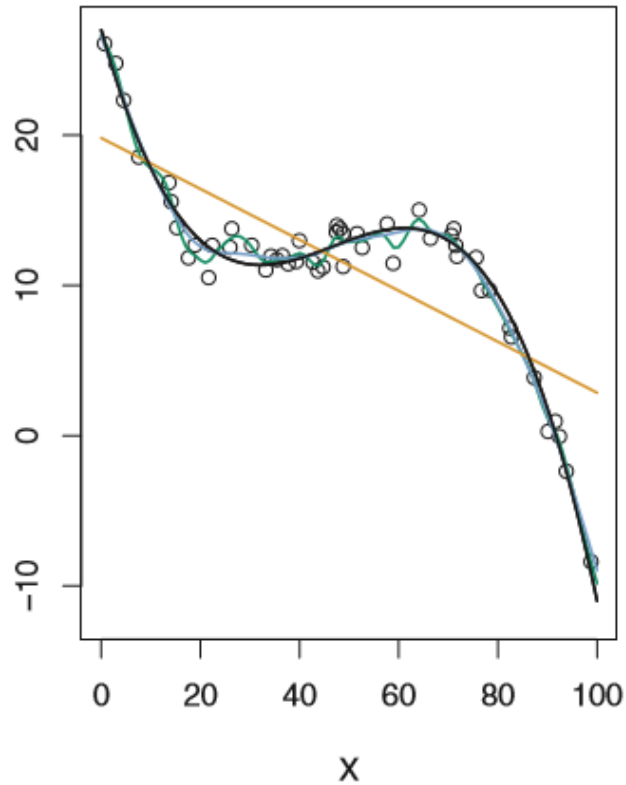
# La compensación entre la flexibilidad e interpretabilidad del modelo.



# Midiendo la calidad del modelo



# El *trade off* entre Sesgo y Varianza.



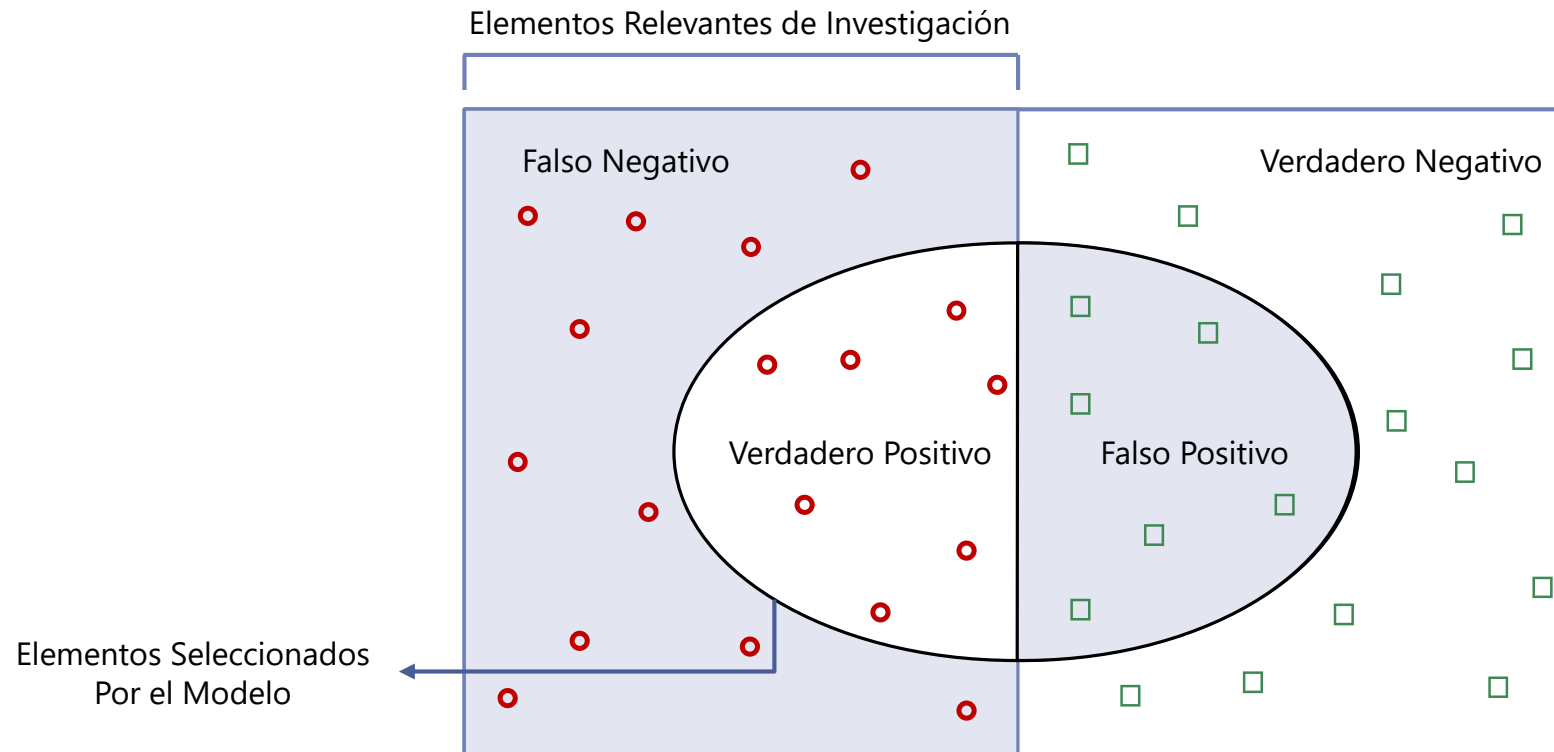
# El *trade off* entre Sesgo y Varianza.

---

$$E \left( y_0 - \hat{f}(x_0) \right)^2 = \underbrace{\text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2}_{\text{Sesgo y Varianza}} + \text{Var}(\epsilon).$$

Minimizar la Suma, de Sesgo,  
Varianza, y Varianza del Error.

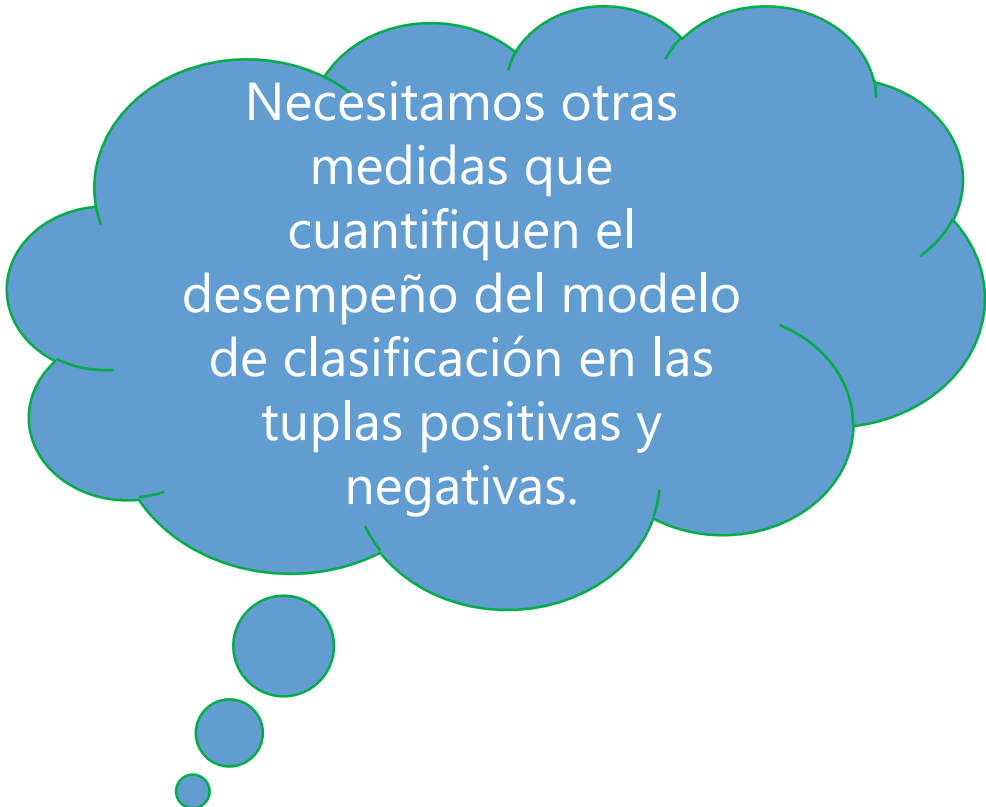
# Medidas de Calidad para un Modelo



# Evaluación de modelos de clasificación

---

- ¿Qué ocurre si la clase de interés es rara o **poco frecuente**?
- Ejemplo: detección de fraude, cáncer, fuga de clientes, etc
- Una tasa de reconocimiento del 97% (tasa de error de un 3%) puede ser muy alta, pero **engañosa** si sólo el 3% de las tuplas tenían cáncer.
- Puede ocurrir que el clasificador sea muy bueno reconociendo los registros libres de cáncer, pero incapaz de detectar las que sí lo padecen.



Necesitamos otras medidas que cuantifiquen el desempeño del modelo de clasificación en las tuplas positivas y negativas.



# Medidas de Calidad para un Modelo

## Matriz de Confusión

La matriz de confusión es el resultado final del entrenamiento, después de seleccionar un Threshold o Punto de Corte. Y Se utiliza para evaluar el poder predictivo del modelo dado una muestra de testeo.

		Predicción del Modelo	
		Predicción Fugados (Predicción de Positivos)	Predicción No Fugados (Predicción de Negativos)
Valor Real	Fugados	Verdadero Positivo (True Positive)	Falso Negativo (False Negative)
	No Fugados	Falso Positivo (False Positive)	Verdadero Negativo (True Negative)

# Medidas de Calidad para un Modelo

Medidas útiles en un Matriz de Confusión

Verdadero Positivo (VP)	Falso Negativo (FN)
Falso Positivo (FP)	Verdadero Negativo (TN)

$$Accuracy = \frac{(VP + VN)}{(VP + VN + FN + FP)}$$

Accuracy, es útil para evaluar un modelo en términos generales, se puede utilizar en modelos que predicen dos clases, es recomendable para clases que son **balanceadas**.

$$Recall = \frac{VP}{(VP + FN)}$$

Recall, a partir de los Fugados Reales cual sería la tasa de casos que logro predecir el modelo.

$$Precisión = \frac{VP}{(VP + FP)}$$

Precisión, a partir de los Fugados predichos por el modelo calcula la tasa de Fugados Reales que tiene.

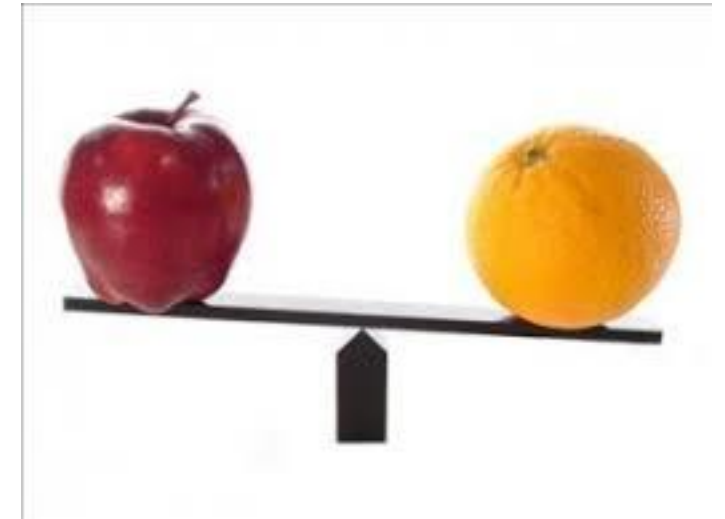
$$F1\ Score = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

F1 Score, esta medida resume las medidas de Recall y Precisión.

# Evaluación de modelos de clasificación

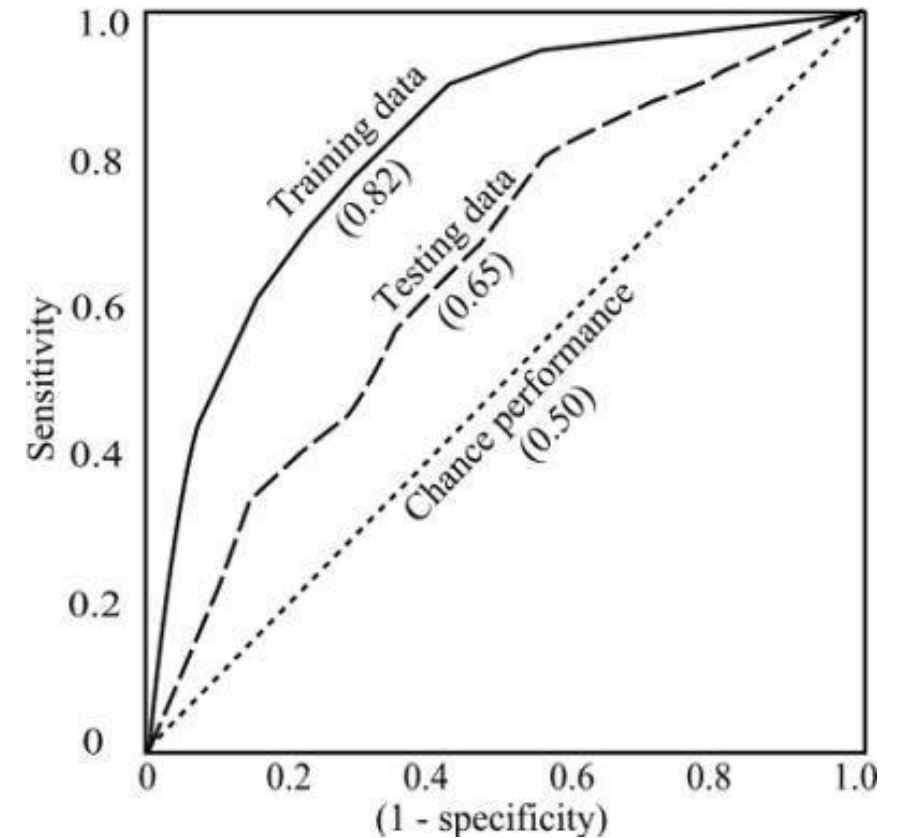
---

- Cómo comparar dos clasificadores
  - TP, TN, FP, FN también son útiles para cuantificar los costos y beneficios asociados con un modelo de clasificación.
  - Los costos de un FN (predecir que está sano un paciente con cáncer) es mucho mayor que el de un FP (predecir que está con cáncer alguien sano).
  - Se puede pesar más uno tipo de error que el otro, asignándole un costo más alto.
  - Estos costos consideran el riesgo de la mala clasificación, costos económicos, etc.



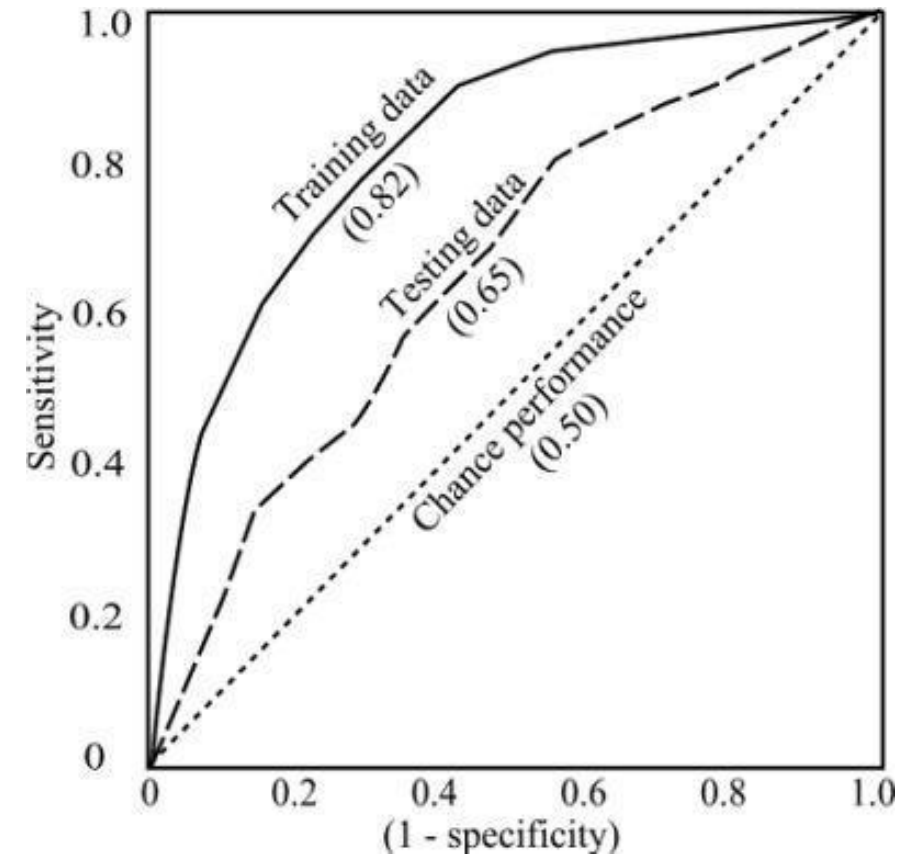
# Evaluación de modelos de clasificación, curva ROC

- Para un problema con 2 clases, la curva ROC permite visualizar el compromiso entre la tasa en que el modelo puede reconocer con precisión los casos positivos versus la tasa de falsos positivos para diferentes porciones del conjunto donde se está validando el modelo.
- Eje Y: Tasa de verdaderos positivos (TPR o sensibilidad)
- Eje X: Tasa de falsos positivos (FPR o 1-Especificidad)



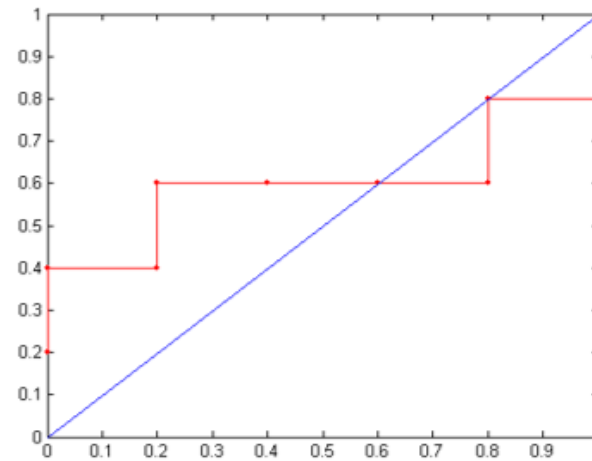
# Apéndice: evaluación de modelos de clasificación, curva ROC

- Permiten comparar visualmente distintos modelos de clasificación.
- El área que queda bajo la curva es una medida de la precisión del clasificador (AUC)
  - Más cerca de la diagonal (área = 0.5), menos preciso será el modelo
  - Por tanto, un modelo perfecto tendrá área = 1.



# Clasificación binaria: Curva ROC

Ejemplo	P(+E)	Clase
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+



Clase	+	-	+	-	-	-	+	-	+	+	
Probabilidad	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

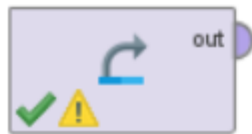
# Ejemplo Rapid Miner

---

## Process

inp

### Retrieve bank-full



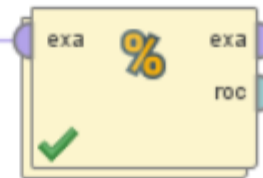
### Set Role



### Split Data



### Compare ROCs

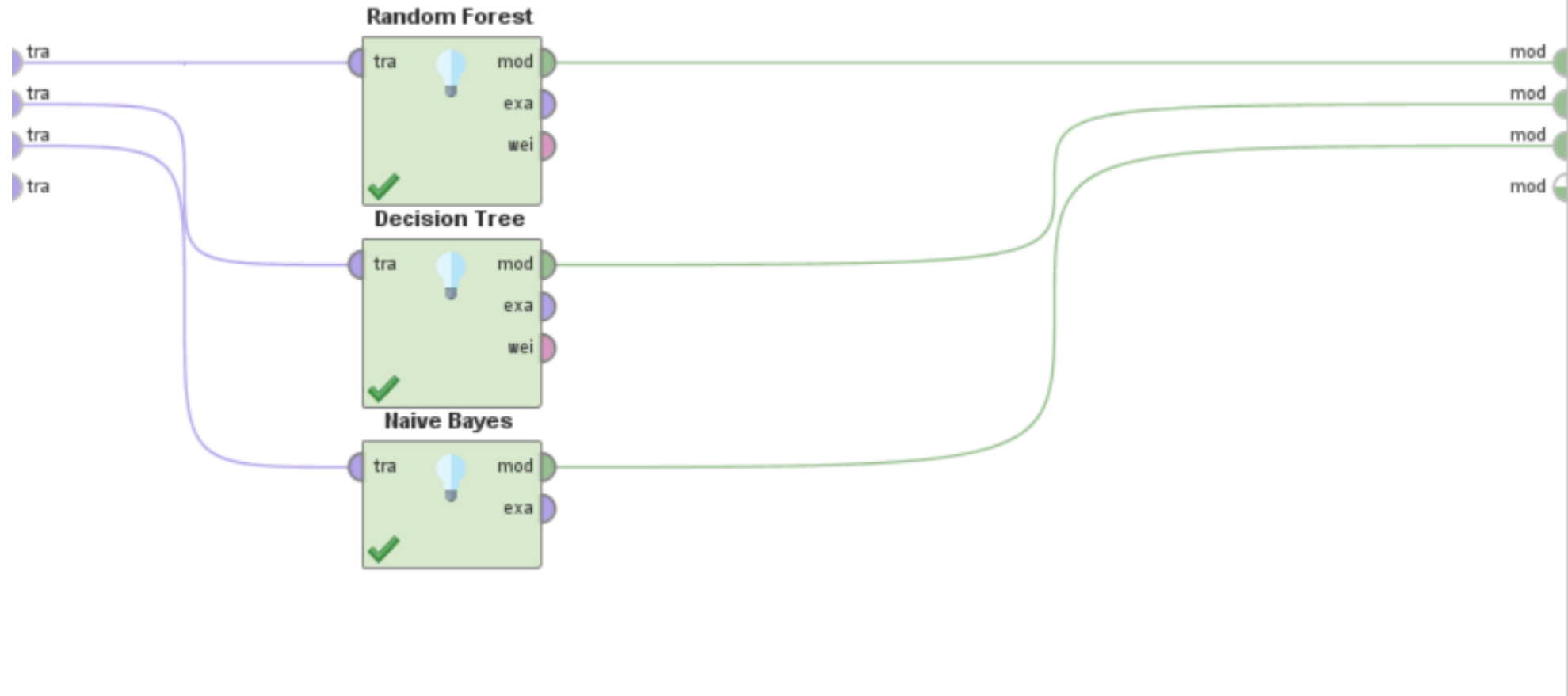


res

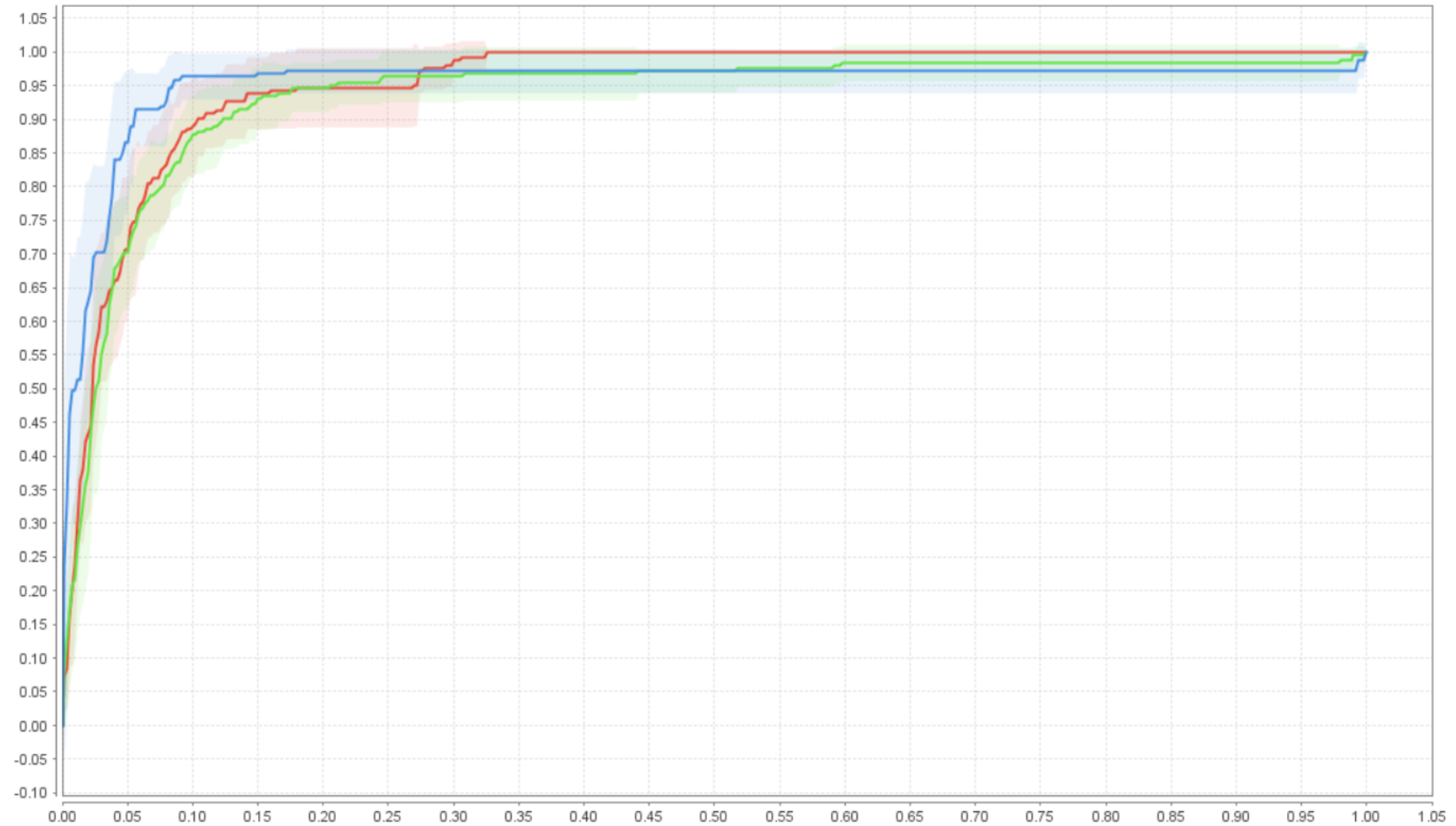
res



## Compare ROCs



Decision Tree Naive Bayes Random Forest





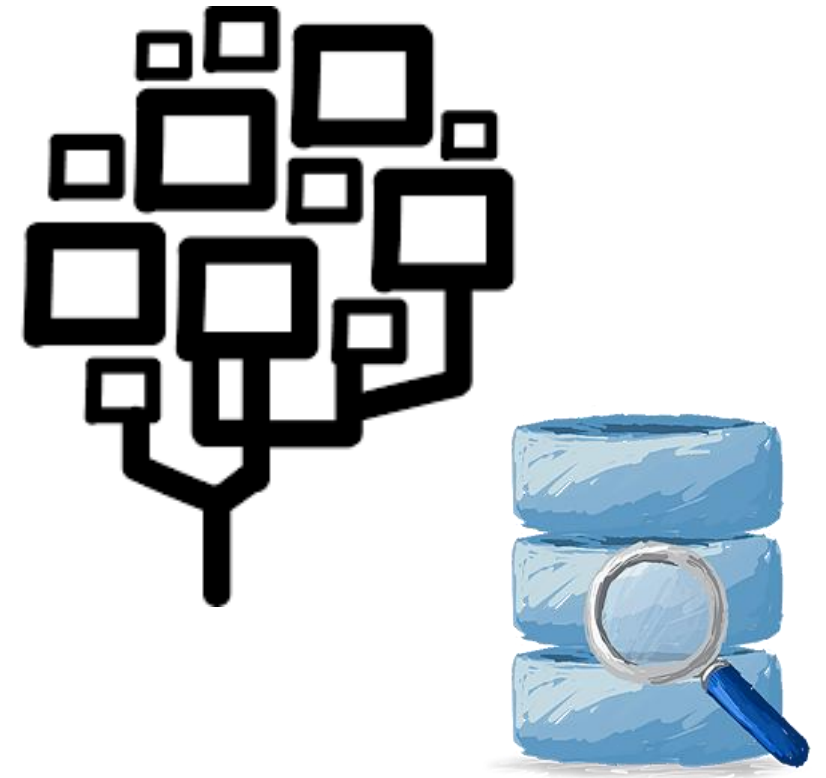
# Clustering

¿Qué información nos da la data?

# Aprendizaje no supervisado

---

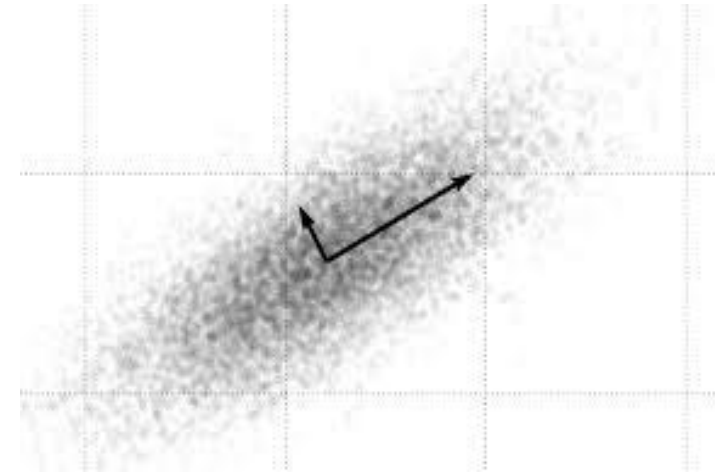
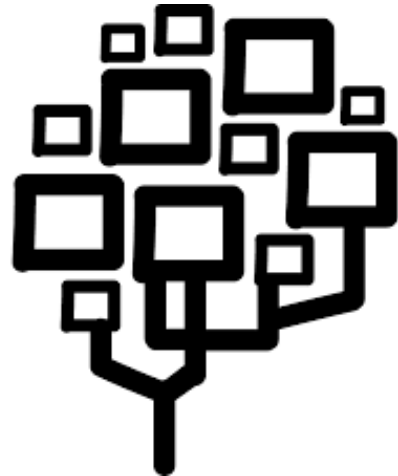
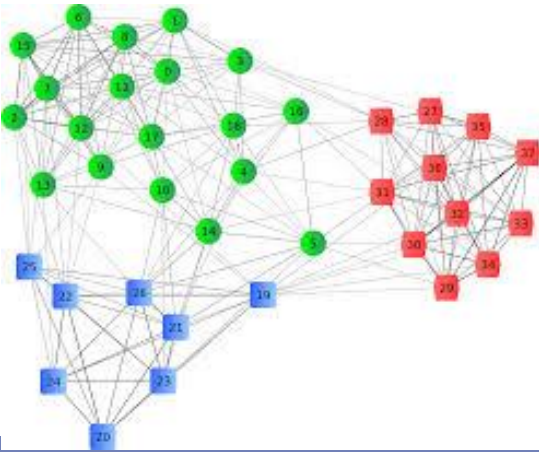
- Data no supervisada no contiene etiquetas y no existe una variable respuesta.
- Existe interes en encontrar una estructura escondida (que nunca es completamente observada).
- Validacion de resultados compleja.



# Aprendizaje no supervisado

Algunas tareas usuales

- Generación de Clústers
- Reglas de asociación.
- Reducción de dimensionalidad

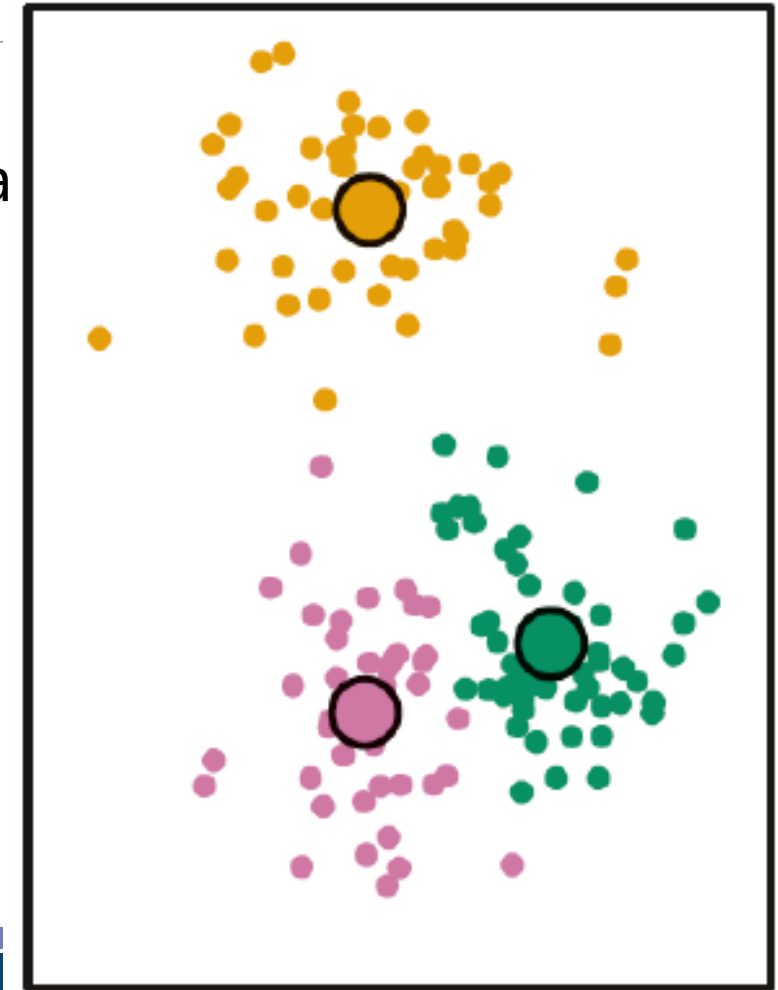


# k-means

**Objetivo:** dado un conjunto de datos **numéricos** (multidimensional), se quiere identificar grupos en la data

1. Idea del algoritmo :
2. Suponer (por ejemplo),  $K=3$  grupos y asignar aleatoriamente las obs. a ellos.
3. Definir los centroides.
4. Reasignar la data a al centroide más cercano.
5. Redefinir centroides
6. Repetir proceso hasta que no haya cambio en la asignación

Final Results



# k-means

---

¿Cómo determinar el número óptimo de clústers?

- Minimizando la suma de cuadrados totales de todos los grupos (total wss)

$$\underset{C_1, \dots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^K W(C_k) \right\}$$

$$W(C_k) = \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2,$$

# K-medoids

---

- Los medoides son objetos representativos de un conjunto de datos o un grupo cuya disimilaridad promedio a todos los objetos en el grupo, es mínima.
- Es un concepto similar a la media o centroide, pero los medoides siempre son miembros del conjunto de datos.
- Puede ocurrir que exista más de un medoide, como es el caso de la mediana



# K-medoids

---

- Se utiliza el criterio de error absoluto, definido como:

$$E = \sum_{i=1}^k \sum_{p \in C_i} \text{dist}(\mathbf{p}, \mathbf{o}_i)$$

Donde  $\mathbf{o}_i$  es el objeto representativo de  $C_i$ .

- La agrupación se realiza minimizando este error absoluto, donde en cada iteración se escoge un nuevo representante.



# ••• Selección de variables y reducción de dimensionalidad

Usar solo lo relevante

# Tipos de reducciones

---

Reducción por contexto

Reducción a categorías

Reducción por correlación

Reducción por dimensionalidad

# Principal component analysis (PCA)

Idea principal:

- Analizar un set de datos de alta dimensionalidad en un espacio de menor dimensión manteniendo la estructura de variabilidad de los datos originales.
- Feature extraction != Feature selection

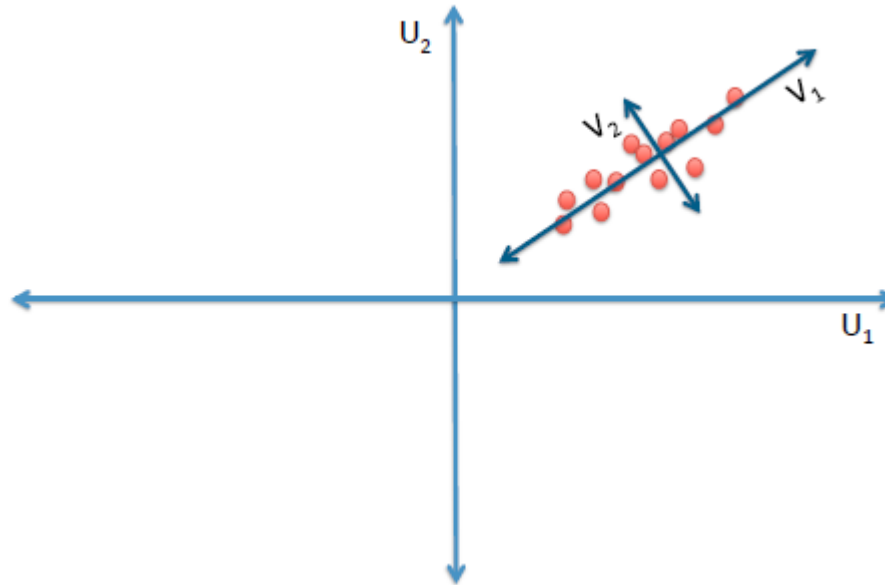
- Selección de variables :  $[X_1, X_2, \dots, X_p] \rightarrow [X_1, X_2, \dots, X_\kappa]$   
 $p$  variables  $\kappa$  variables

- Extracción de características :  $[X_1, X_2, \dots, X_p] \rightarrow [f_1(X_1, X_2, \dots, X_p), \dots, f_\kappa(X_1, X_2, \dots, X_p)]$   
 $= [Y_1, Y_2, \dots, Y_\kappa]$

# Principal component analysis (PCA)

---

Idea básica: Determinar una rotación que describa los datos del espacio  $U$  en  $V$ . (Buscamos un espacio ortogonal)



# Principal component analysis (PCA)

Enfoque:

$$\begin{bmatrix} X_1^\top \\ X_2^\top \\ \vdots \\ X_j^\top \\ \vdots \\ X_p^\top \end{bmatrix}_{p \times n} \rightarrow \begin{bmatrix} Y_1^\top \\ Y_2^\top \\ \vdots \\ Y_\kappa^\top \end{bmatrix}_{\kappa \times n} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1j} & \dots & w_{1p} \\ w_{21} & w_{22} & \dots & w_{2j} & \dots & w_{2p} \\ \vdots & \vdots & & \vdots & & \vdots \\ w_{\kappa 1} & w_{\kappa 2} & \dots & w_{\kappa j} & \dots & w_{\kappa p} \end{bmatrix}_{\kappa \times p} \begin{bmatrix} X_1^\top \\ X_2^\top \\ \vdots \\ X_j^\top \\ \vdots \\ X_p^\top \end{bmatrix}_{p \times n}$$

$$\mathbf{Y}^\top = \mathbf{W}^\top \mathbf{X}^\top \quad \text{or} \quad \mathbf{Y} = \mathbf{X} \mathbf{W}$$

$\kappa \times n \quad \kappa \times p \quad p \times n \qquad n \times \kappa \quad n \times p \quad p \times \kappa$

Terminology:

- ▶  $Y_1, \dots, Y_\kappa$  are the *scores*,
- ▶  $w_1, \dots, w_\kappa$  are the *loadings*.

# Principal component analysis (PCA)

---

- Se puede probar que  $\mathbf{W}_{p \times \kappa}$  corresponde a los vectores propios de  $\mathbf{X}^\top \mathbf{X}$  asociados al mayor valor propio número  $\kappa$ .
- La varianza explicada por la  $j$ -ésima componente corresponde al  $j$ -ésimo valor propio de  $\mathbf{X}^\top \mathbf{X}$ .

$$\mathbf{Y}_{n \times \kappa} = \mathbf{X}_{n \times p} \mathbf{W}_{p \times \kappa}$$

$$\mathbf{W}^\top \mathbf{W} = \mathbf{I}.$$

# Principal component analysis (PCA)

---

Ejercicio



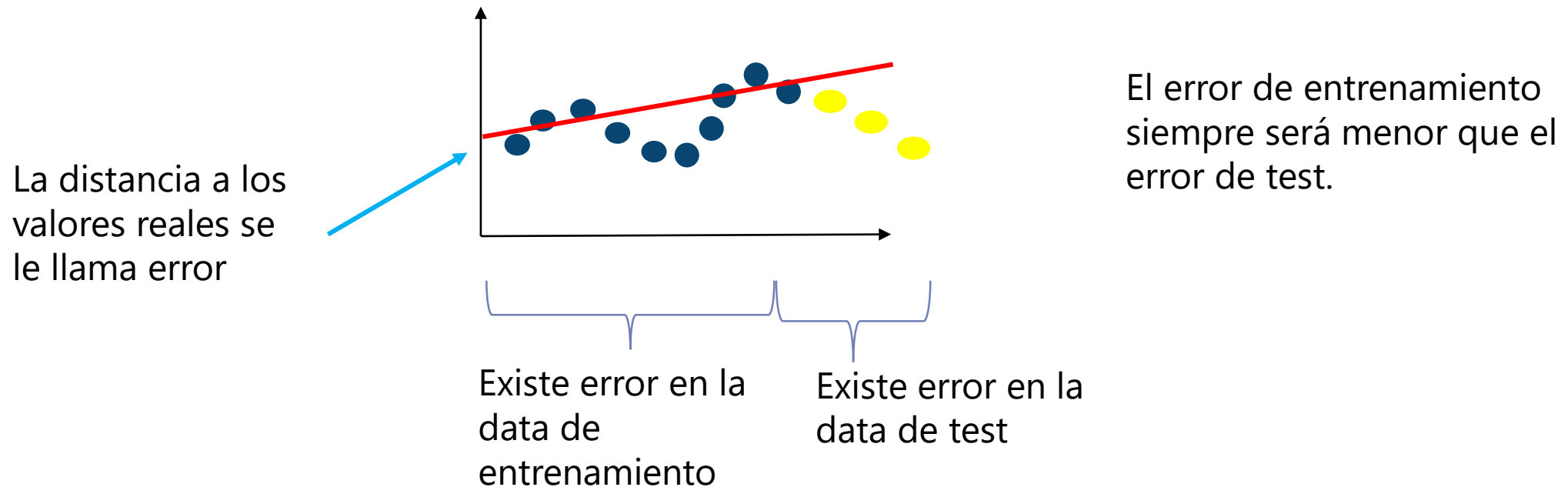
# Recordar

---

- Siempre se elige entre flexibilidad de un modelo e interpretabilidad.
  - Por ejemplo, en una regresión lineal vs una red neuronal
- Si es que tengo una variable de respuesta en mis datos, puedo entrenar un modelo para esa respuesta a esto se le llama, **modelo supervisado**.
  - Dentro de los modelos supervisados, si la variable es categórica (estados), el problema se le dice problema de clasificación.
  - Si la variable es continua, al problema se le dice problema de regresión.
- Cuando no tengo la variable de respuesta en mis datos, busco patrones, a esto se le llama modelo no supervisado.

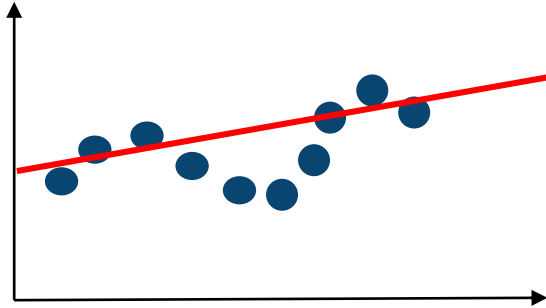
# Recordar

- Así en resumen, para problemas supervisados:

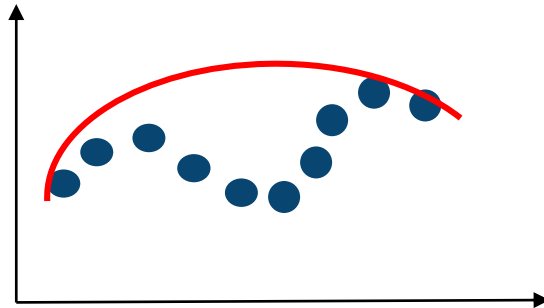


# Recordar

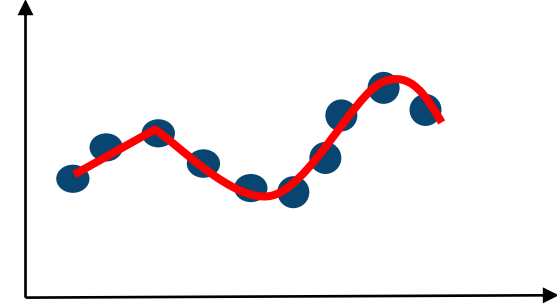
- Hay que tener cuidado con el trade off entre sesgo y varianza



Alto Sesgo  
Baja Varianza



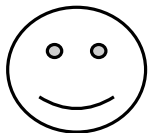
Sesgo Medio  
Varianza Media



Sesgo Bajo  
Varianza Alta

# Recordar

- Para los problemas de clasificación:

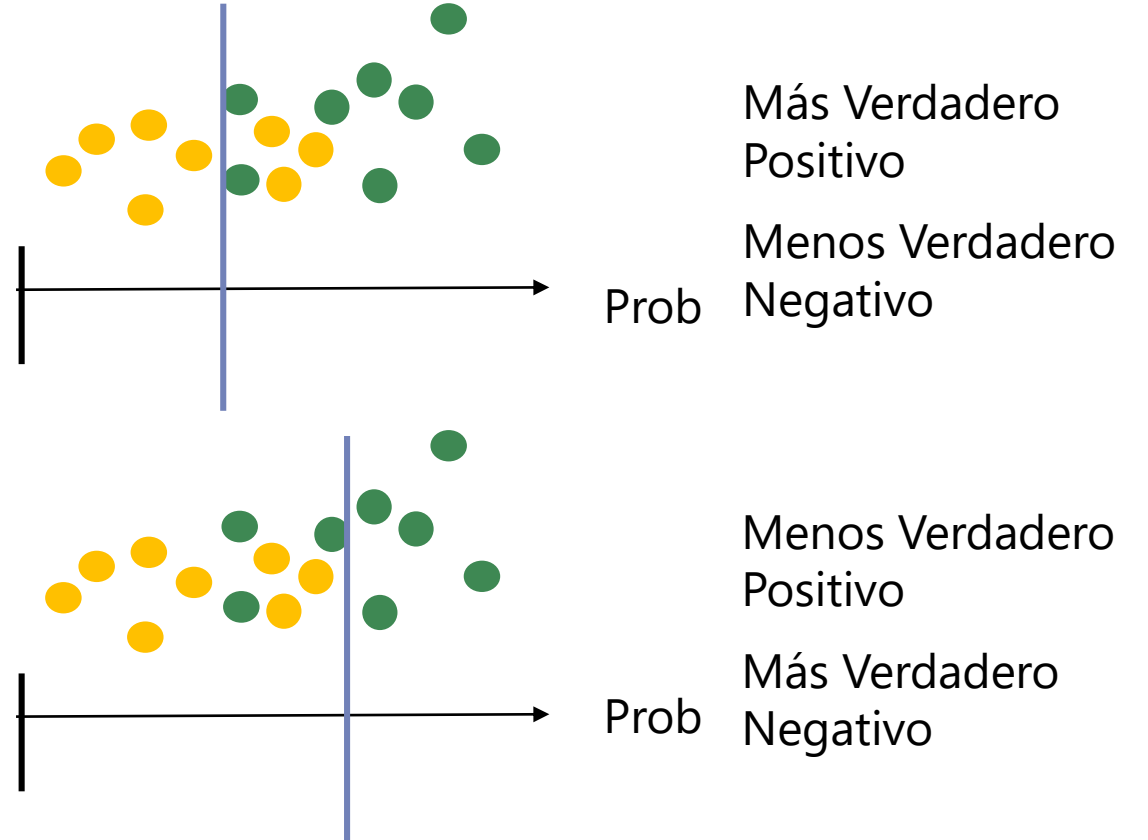
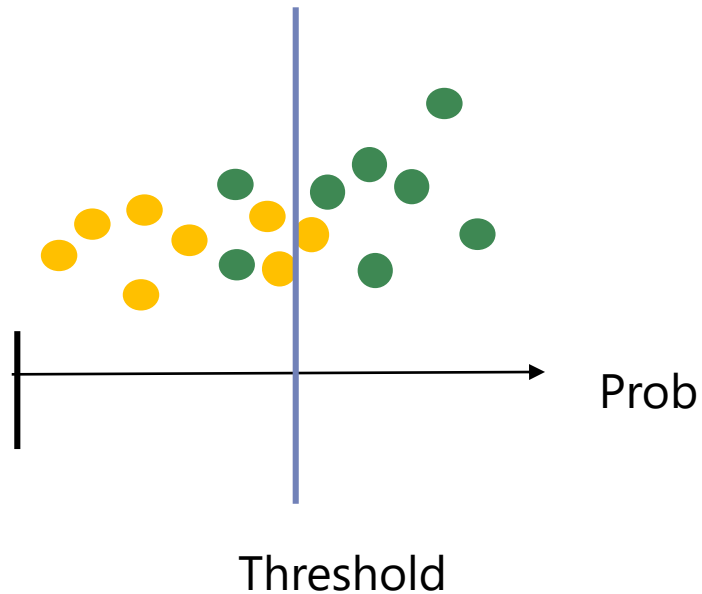


Verdadero Positivo ● ● ● ● ● ● ● ●	Verdadero Negativo ● ● ● ● ● ● ● ●
Falso Positivo ● ● ● ● ●	Falso Negativo ● ● ● ● ● ● ●



# Recordar

- Para los problemas de clasificación:



# Segundo caso

---

Forme grupos y decida entre las problemáticas que encontró en la actividad anterior cuales corresponden a un problema de Clasificación, cuales a uno de Regresión, y cuáles a un problema de Clustering.

Para sus problemas de clasificación revisen si es un problema balanceado o no, ¿Qué problemas pueden surgir en las métricas de desempeño?

Genere dos métricas de desempeño por cada problema propuesto, converse los pros y contras de las métricas definidas.

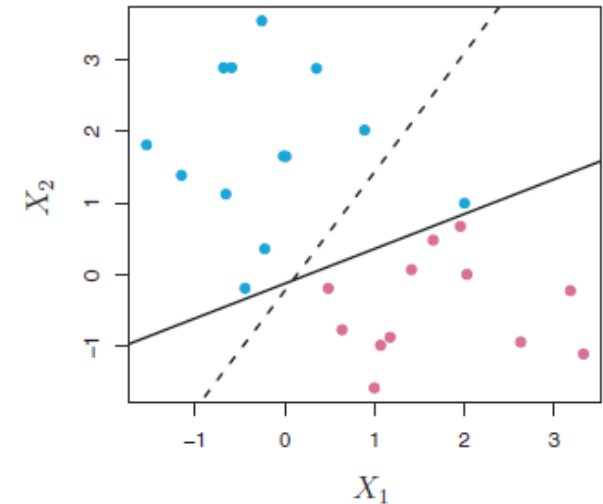
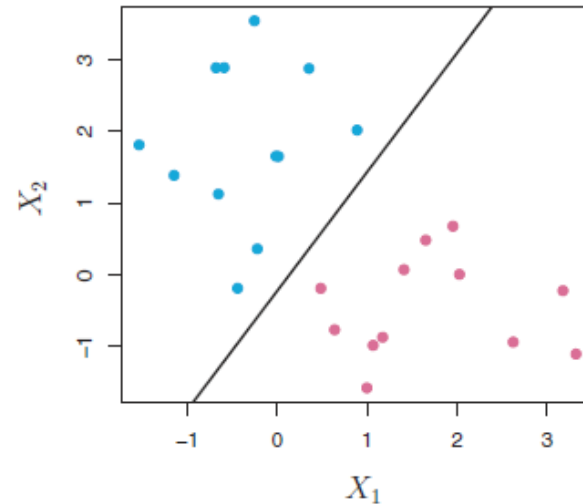


# ••• Otros modelos de clasificación y regresión

# Support vector machines (SVM)

- Separación por hiperplanos. (maximal margin).
- Caso no separable. (Soft margin)

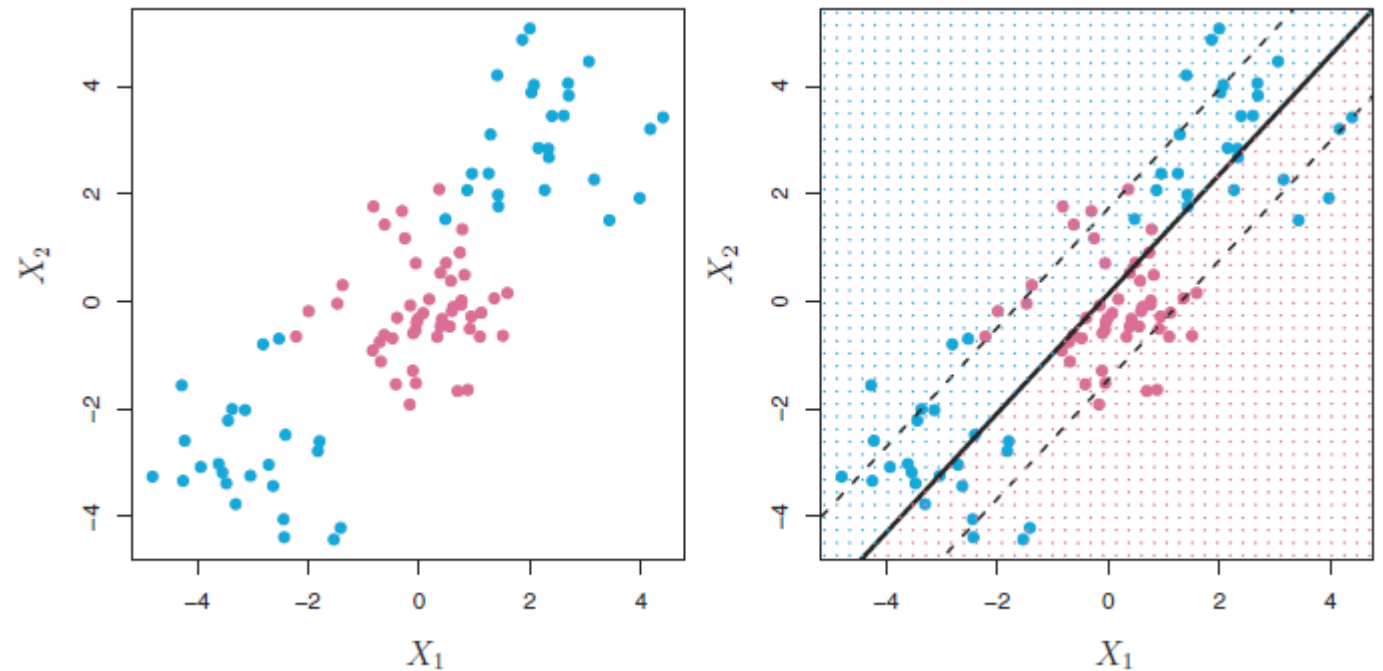
$$\begin{cases} \max_{\beta, \beta_0, \|\beta\|=1} M \\ \text{sujeto a } \xi \geq 0, \sum_{i=1}^N \xi_i \leq K, y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i), i = 1, \dots, N \end{cases}$$





# Support vector machines (SVM)

- Fronteras de decisión no lineales.
- Se introduce flexibilidad mediante la utilización de Kernels.



# Support vector machines (SVM)

---

Considérese el set de entrenamiento  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathcal{X}$ , con  $N \in \mathbb{N}$  y  $\mathcal{X} \subset \mathbb{R}^n$  compacto. De acuerdo a la formulación tradicional de un clasificador SVM (véase por ejemplo [1]), supóngase un kernel del tipo

$$k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle,$$

donde  $\phi$  es una función que mapea  $\mathcal{X} \rightarrow \mathcal{H}$ , con  $\mathcal{H}$  un espacio completo y con un producto interno  $\langle \cdot, \cdot \rangle$  definido. Para efectos de esta aplicación  $\mathcal{H} = \mathbb{R}^p, p \in \mathbb{N}$ , parámetro que variará con cada uno de los modelos a implementar.

El Kernel a utilizar en este modelo es el radial, dado por

$$k(\mathbf{x}, \mathbf{y}) := \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$$

- [1] B. Schölkopf and A.J. Smola, *Learning with Kernels*. MIT Press, 2002.

# Support vector machines (SVM)

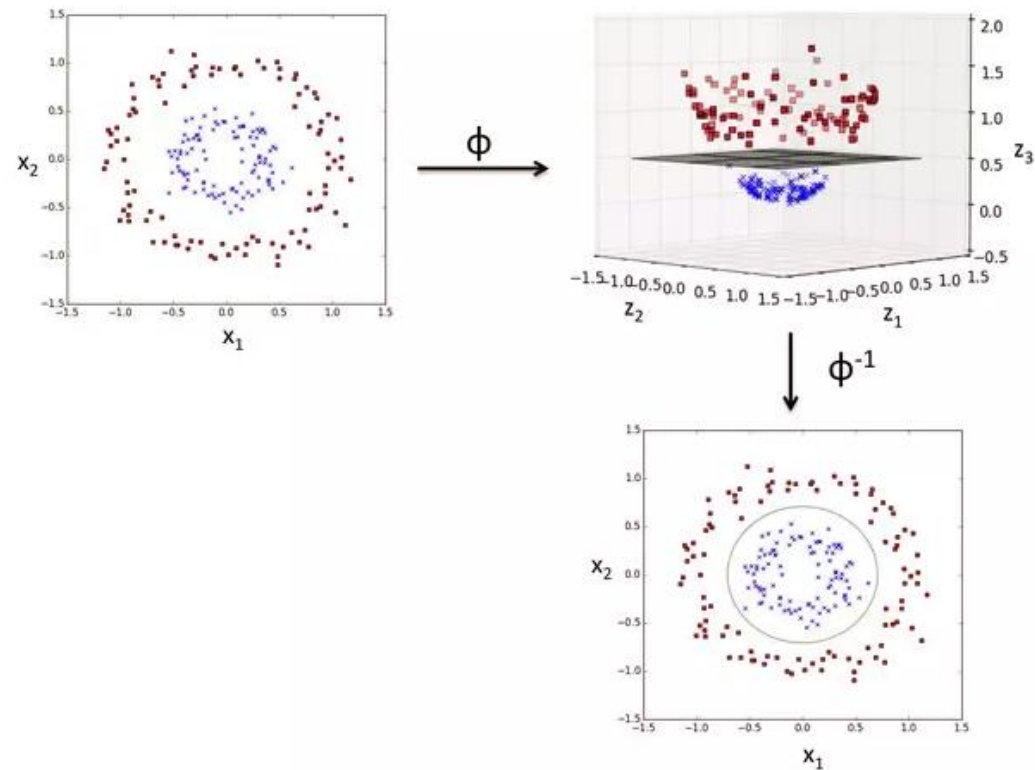
---

$$\left\{ \begin{array}{l} \max_{\beta, \beta_0, ||\beta||=1} M \\ \text{sujeto a } \xi \geq 0, \sum_{i=1}^N \xi_i \leq K, y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i), i = 1, \dots, N \end{array} \right.$$



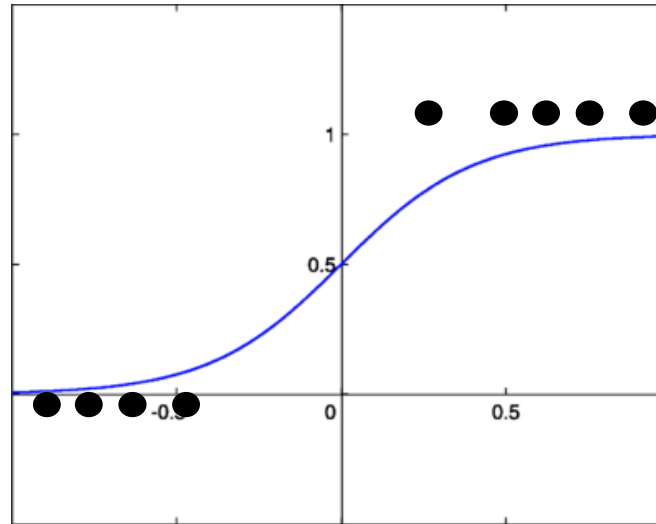
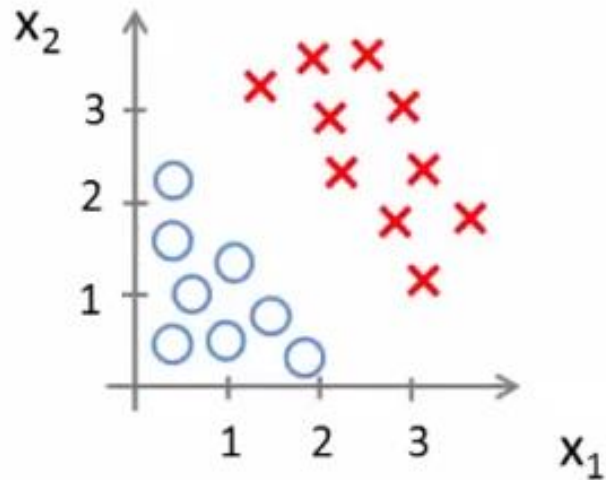
$$\left\{ \begin{array}{l} \min_{\beta, \beta_0} \frac{1}{2} ||\beta||^2 + C \sum_{i=1}^N \xi_i \\ \text{sujeto a } \xi \geq 0, y_i(\langle \phi(x_i), \beta \rangle + \beta_0) \geq (1 - \xi_i), i = 1, \dots, N \end{array} \right.$$

# Support vector machines (SVM)



# Regresión Logística

¿Qué es la regresión logística?



Se separan los datos mediante una función logit.

$$P = \frac{e^{a+bX}}{1 + e^{a+bX}}$$

# Multiclass Logistic Regression

---

La búsqueda de los parámetros se desarrolla usando Limited Memory BFGS.

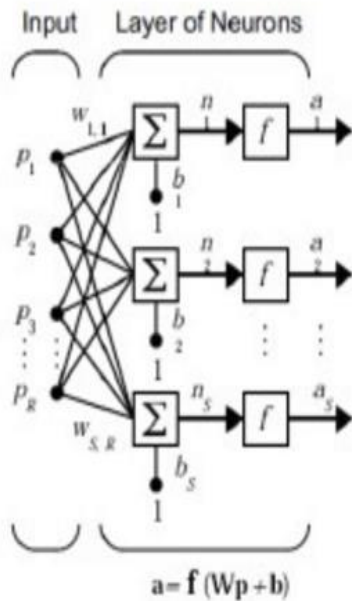
Que una aproximación del hessiano usado en el método de newton.

**Optimization Tolerance**, la tolerancia de la aproximación al hessiano.

Memory Size for L-BFGS, especifica el total de memoria en MB usado para el LBFGS optimizer.

# Red Neuronal (ANN)

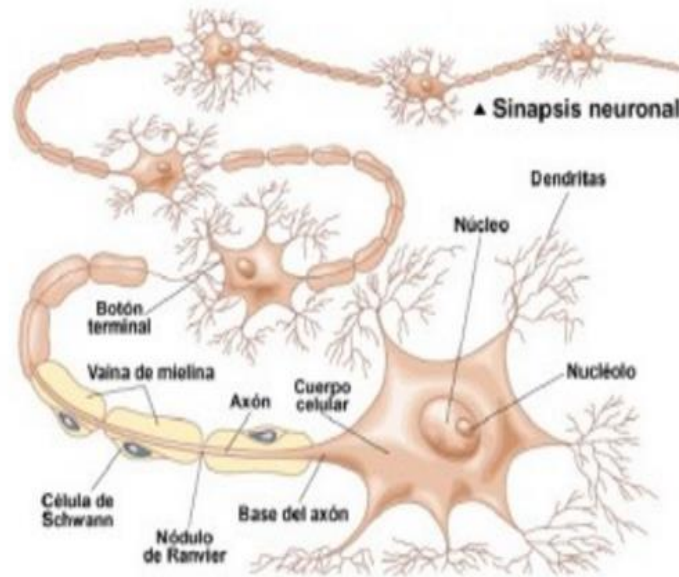
## Analogía Redes Neuronales



Where...

$R$  = number of elements in input vector

$S$  = number of neurons in layer

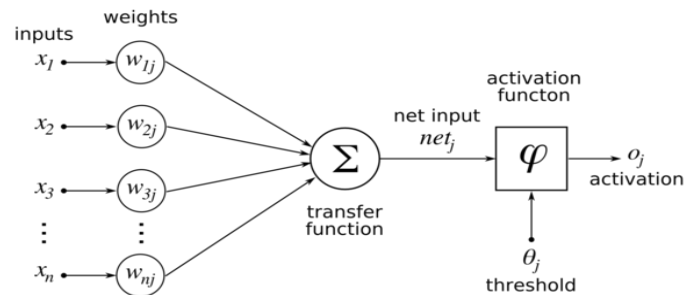


## Características Relevantes

- ✓ Aprendizaje Adaptativo (dinámico autoadaptativo)
- ✓ Auto-organización (implica la propiedad de generalización)
- ✓ Tolerancia a Fallos (forma de almacenamiento)
- ✓ Operaciones en tiempo real (implementación paralela)

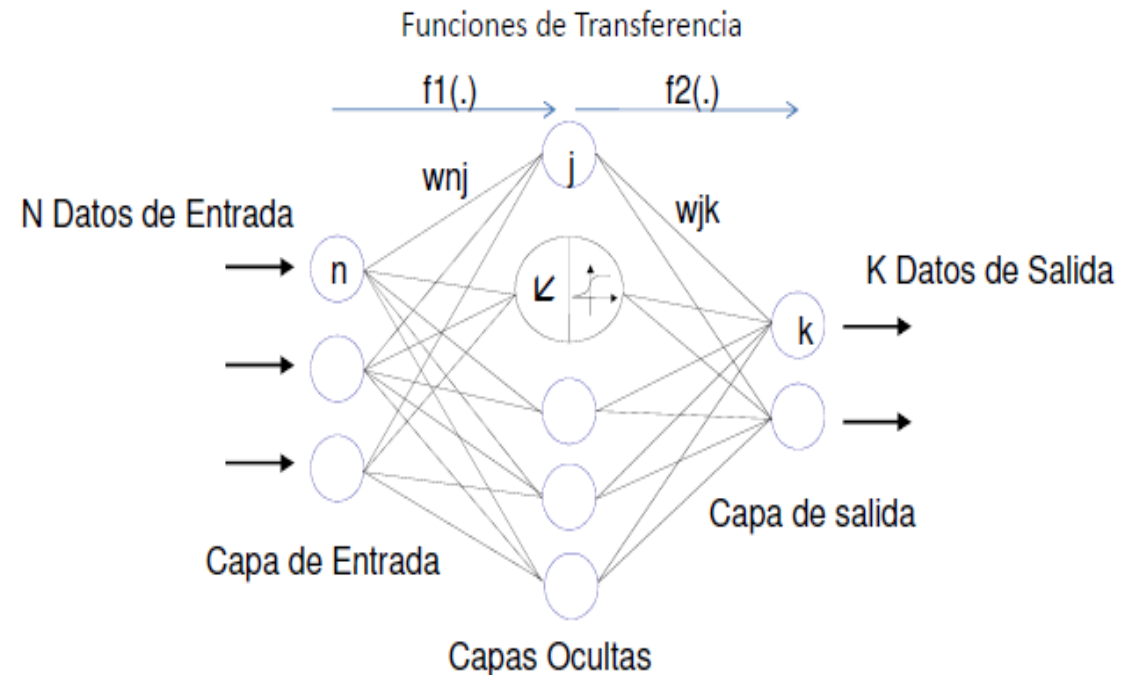
# Red Neuronal (ANN)

## Funcionamiento



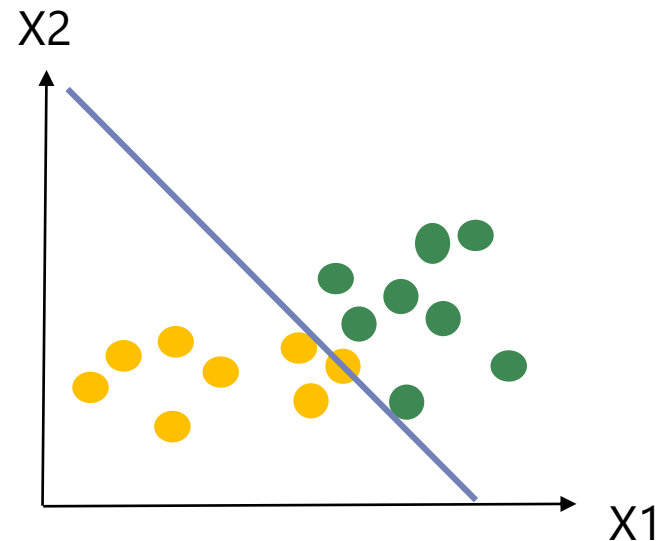
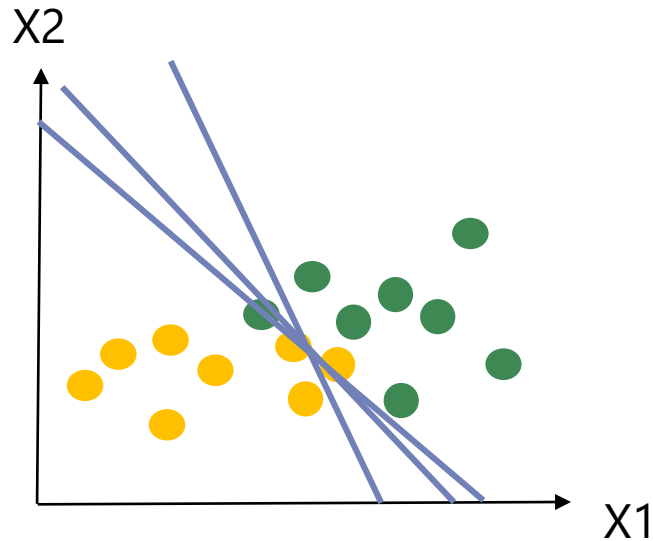
Esta constituida por 4 componentes:

- ✓ Capa de Entrada: se encargan de recibir las señales o patrones que proceden del exterior.
- ✓ Capas Ocultas: tienen la misión de realizar el procesamiento no lineal de los patrones recibidos.
- ✓ Capa de Salida: actúa como salida de la red, proporcionando la respuesta de la red al exterior.
- ✓ Función de Transferencia: Corresponde a la "sinapsis" y es el factor principal de la técnica.





# Support Vector Machine



Mejor hiperplano de separación.  
Maximiza la separación entre los conjuntos

# Regresión Lineal

Regresión lineal tiene la forma de

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, i = 1, \dots, n$$

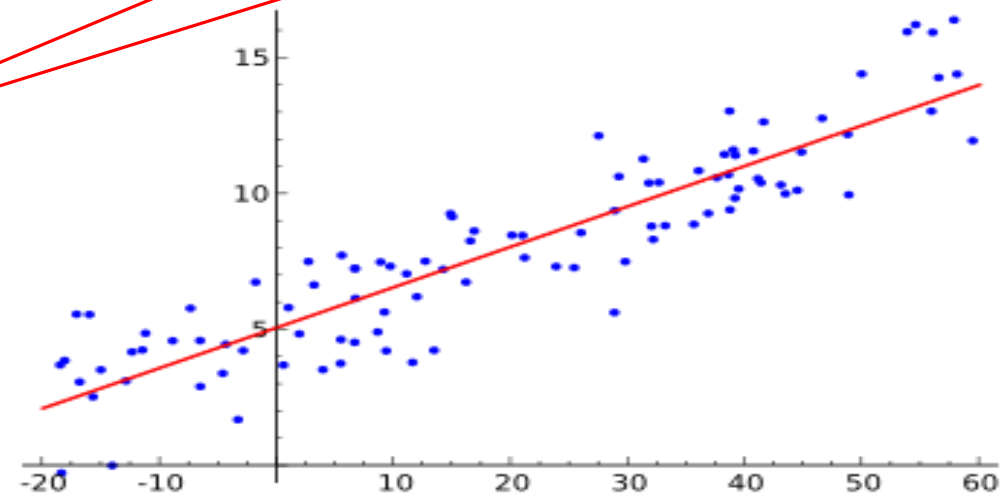
Las cantidades  $\beta_0$  y  $\beta_1$  son la ordenada al origen y la tendencia de la regresión, que son asumidas fijas pero desconocidas.  $\varepsilon_i$  es una variable aleatoria que representa el ruido pero se sume que  $E(\varepsilon_i) = 0$ , por lo que:

$$E(Y_i|x_i) = \beta_0 + \beta_1 x_i$$

La linealidad de la regresión se interpreta en términos de los parámetros.

Es decir:  $E(Y_i|x_i) = \beta_0 + \beta_1^2 x_i$  no es regresión lineal.

Generalmente se utiliza el método de máxima verosimilitud para encontrar los valores de  $\beta$



# Regresión Lineal Bayesiana

A partir de la Regresión lineal que tiene la forma de:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, i = 1, \dots, n$$

Tenemos que en su forma clásica los coeficientes son encontrados generalmente con la función de máximo verosimilitud. En el caso Bayesiano, los coeficientes se encuentran utilizando el siguiente método:

1. Se define la distribución a priori para los coeficientes  $\beta_0$  y  $\beta_1$
2. Luego se determina la verosimilitud de los datos para actualizar los coeficientes y así considere la incertidumbre de estos.
3. Aplicar el teorema de Bayes para actualizar la distribución a priori.

Teorema de Bayes

$$P(A_i/B) = \frac{P(A_i) \cdot P(B/A_i)}{P(B)}$$

Donde:

$P(A_i)$  = Probabilidad a priori

$P(B/A_i)$  = Probabilidad condicional

$P(B)$  = Probabilidad Total

$P(A_i/B)$  = Probabilidad a posteriori

# Regresión de Poisson

En el caso de la regresión de Poisson la variable dependiente se ajusta bien a una distribución de Poisson. (Variables no continuas)

Características:

- ✓ Esta distribución modela bien casos de conteo (número de personas que tienen infarto al corazón, número de llamados de teléfono a una central telefónica, etc).
- ✓ Por eso tiene valores enteros no negativos
- ✓ Su varianza y esperanza es la misma.
- ✓ Tiene la particularidad que mientras más grande es el valor esperado mayor es la dispersión de los valores de la variable.

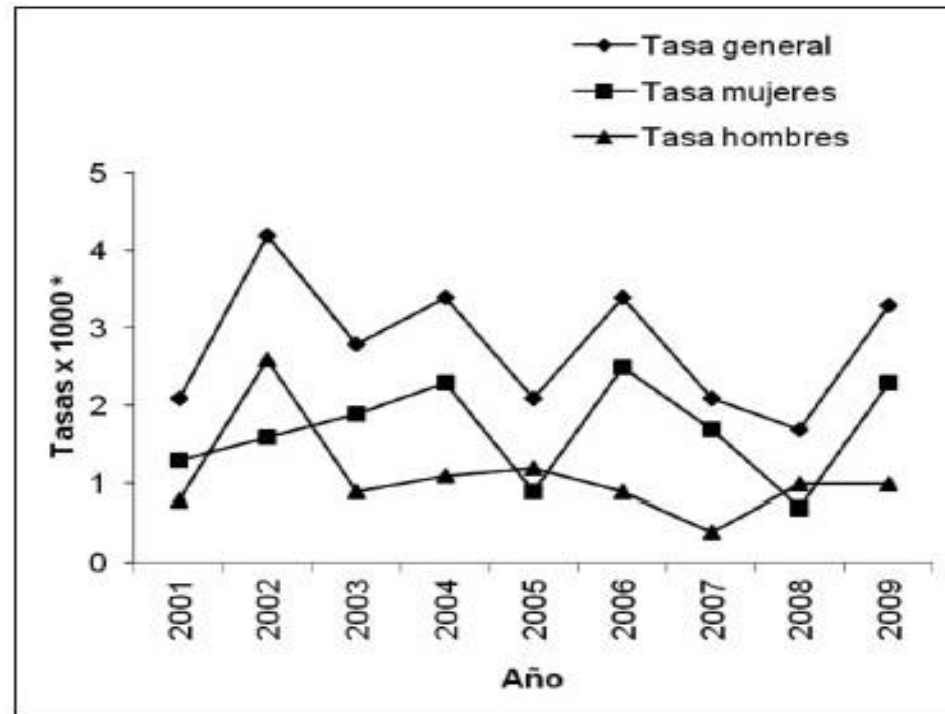
Distribución de Poisson

$$P(X = x) = \frac{e^{-\lambda} \lambda^x}{x!}$$

Solamente posee un parámetro el cual es  $\lambda^x$

# Regresión de Poisson

Ejemplo Regresión de Poisson:



Anestesia en nonagenarios (\*tasas x 1.000 anestесias en adultos/año).

# Discusión de Tercer Caso

---

Piense en 5 casos distintos que podrían modelarse con una regresión. Entre las variables respuesta posibles se encuentran:

- Precio de un bono.
- Cantidad de clientes nuevos mensuales.
- Costos totales asociados a las transacciones.
- Número de clientes endeudados.
- Etc.

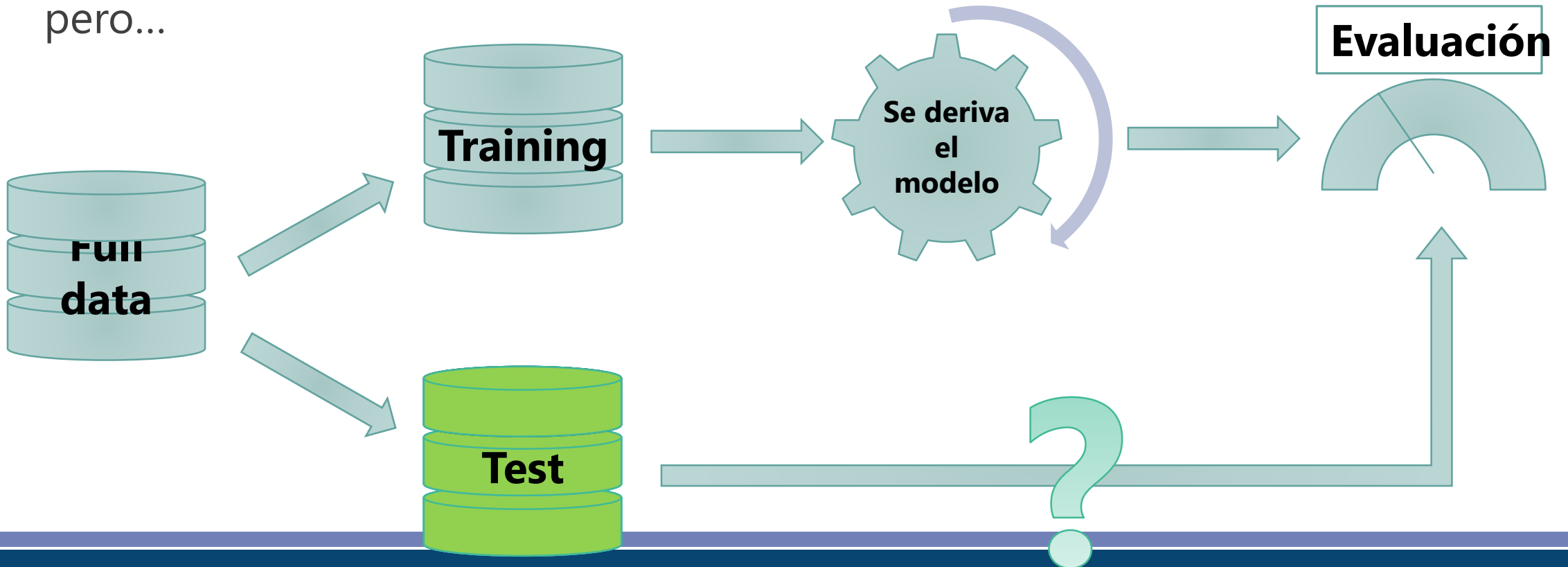
En cada uno de los casos planteados determine qué tipo de regresión sería apropiado usar.



# Mejorando las métricas de error

# Validación CRUZADA

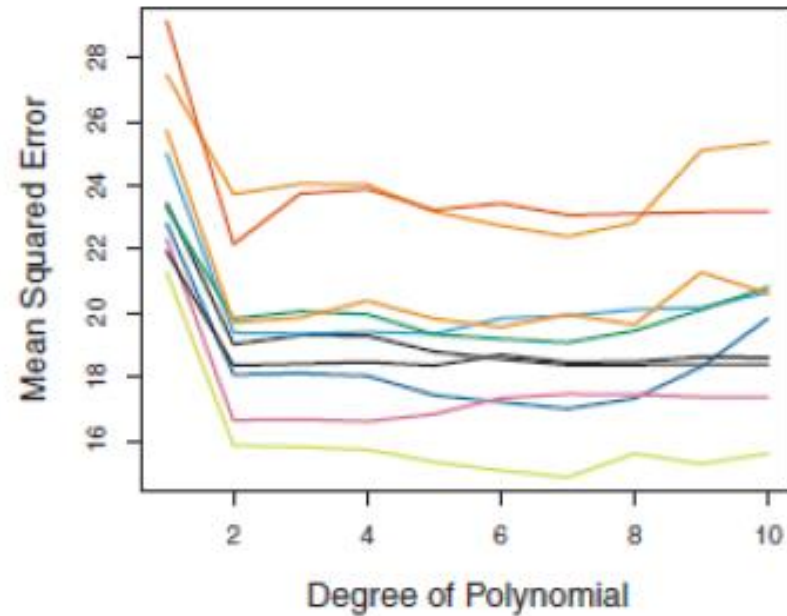
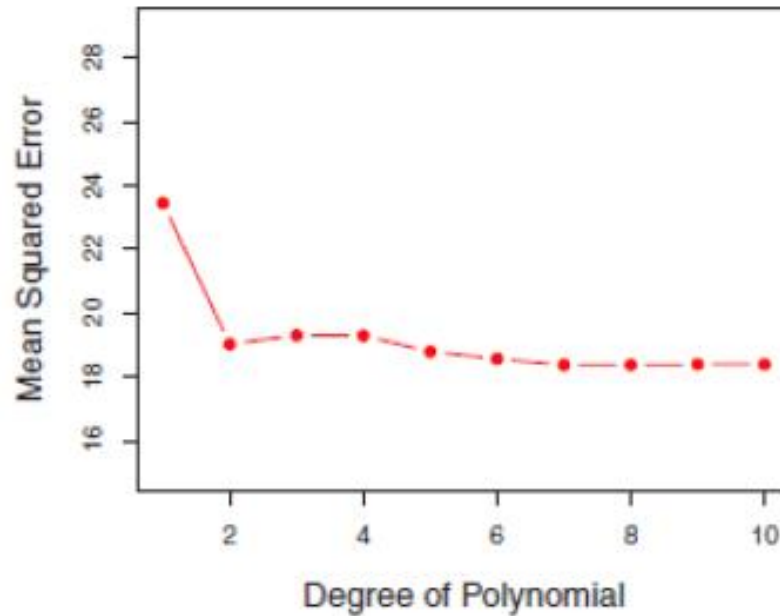
Hasta ahora hemos seguido este enfoque,  
pero...



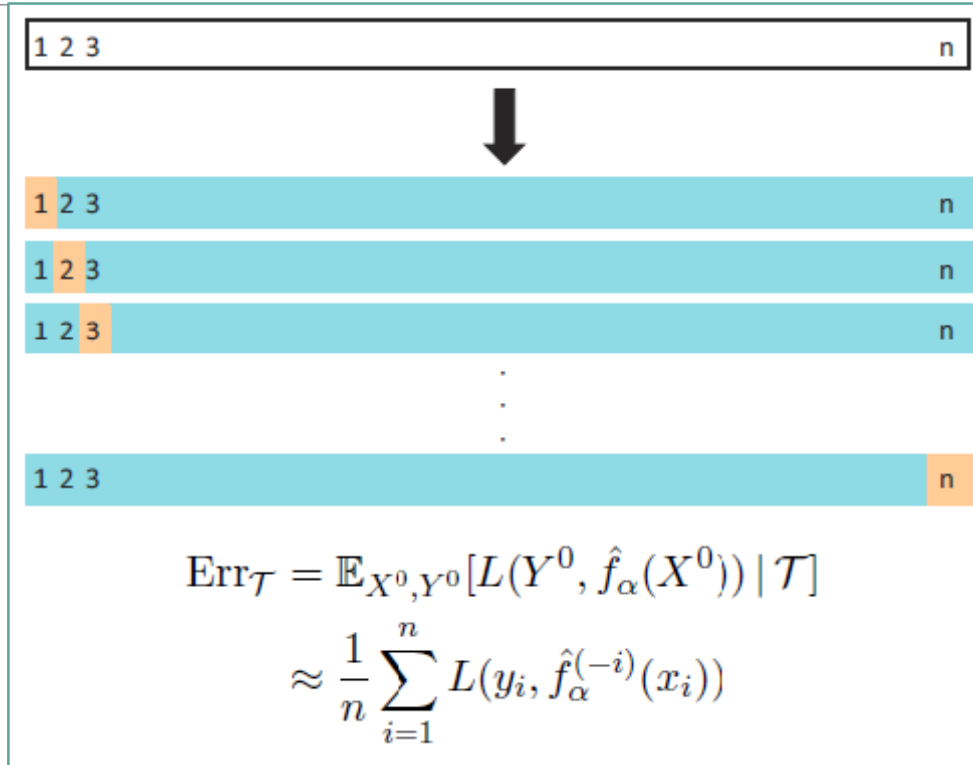


# Validación CRUZADA

Auto data set. izquierda: tasa de error para una sola data de test. Derecha: tasa de error repetida 10 veces (diferentes divisiones aleatorias)



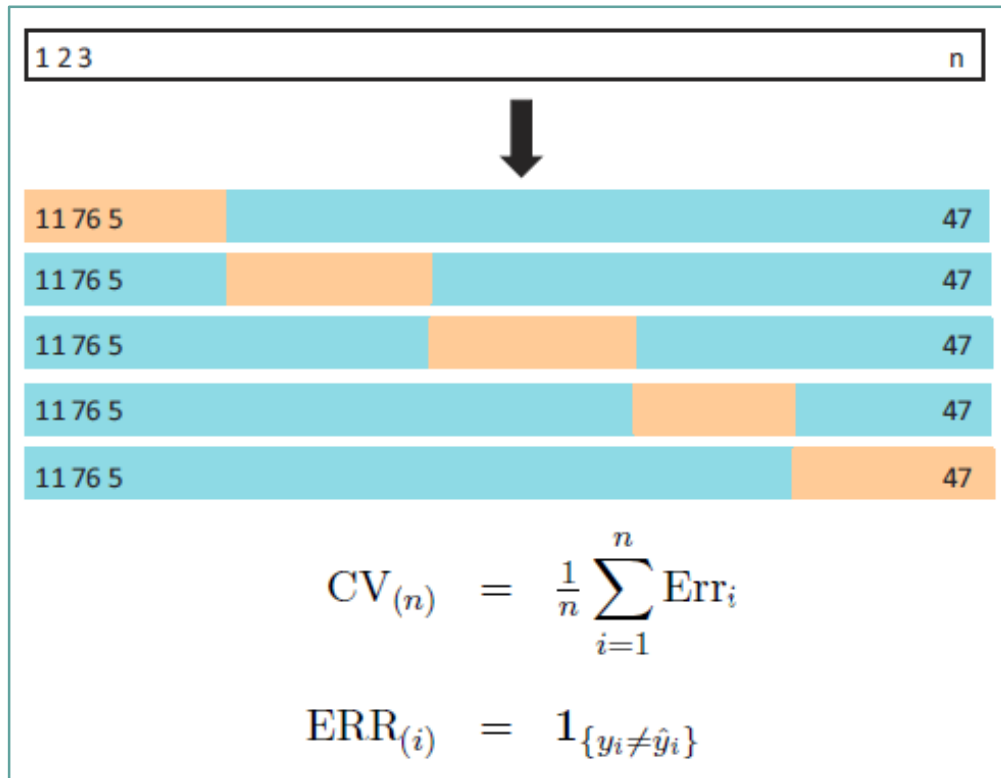
# Validación cruzada: LOOCV



- Intuitivamente
  - ¿Qué inconveniente es evidente?

# Validación cruzada: K-fold

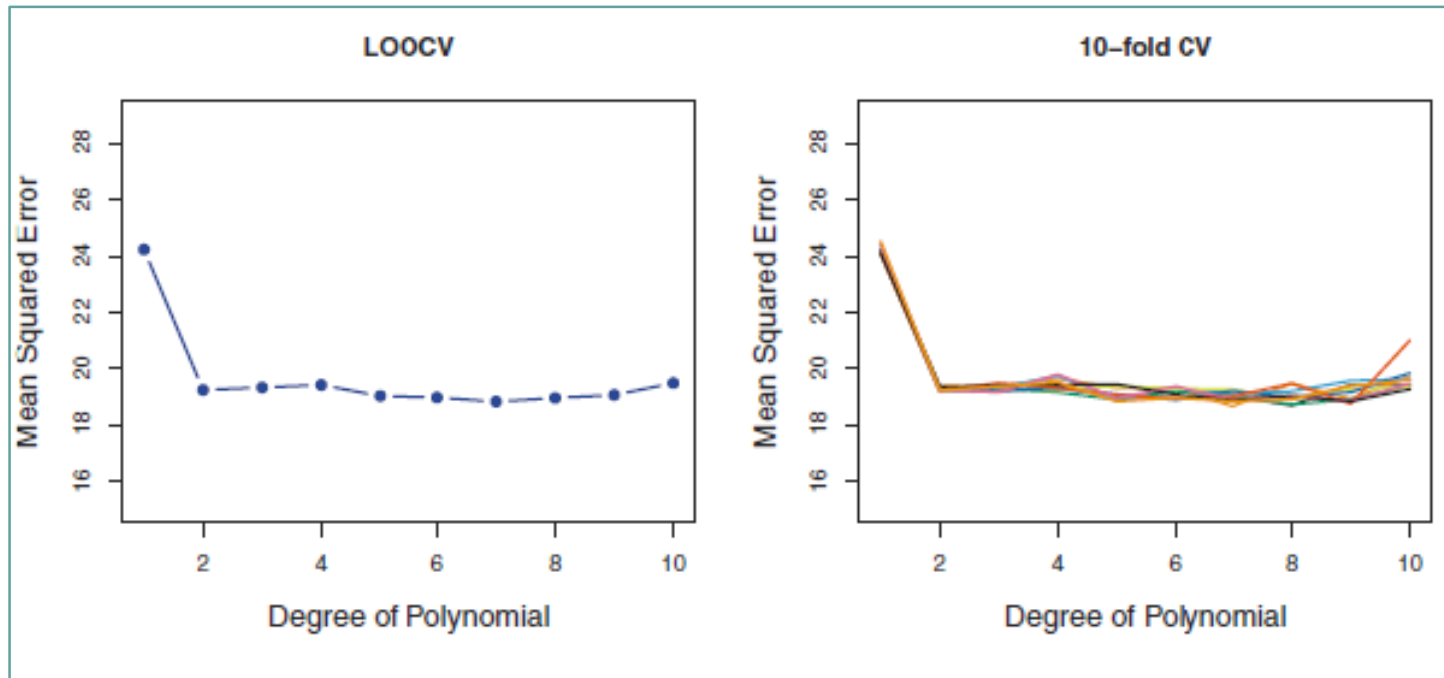
## CV



- Similar a LOOCV pero de menor costo computacional

# Validación cruzada: K-fold cv

Se puede demostrar que con  $k=10$ , se obtiene una estimación similar a la obtenida a través de LOOCV del error de test.

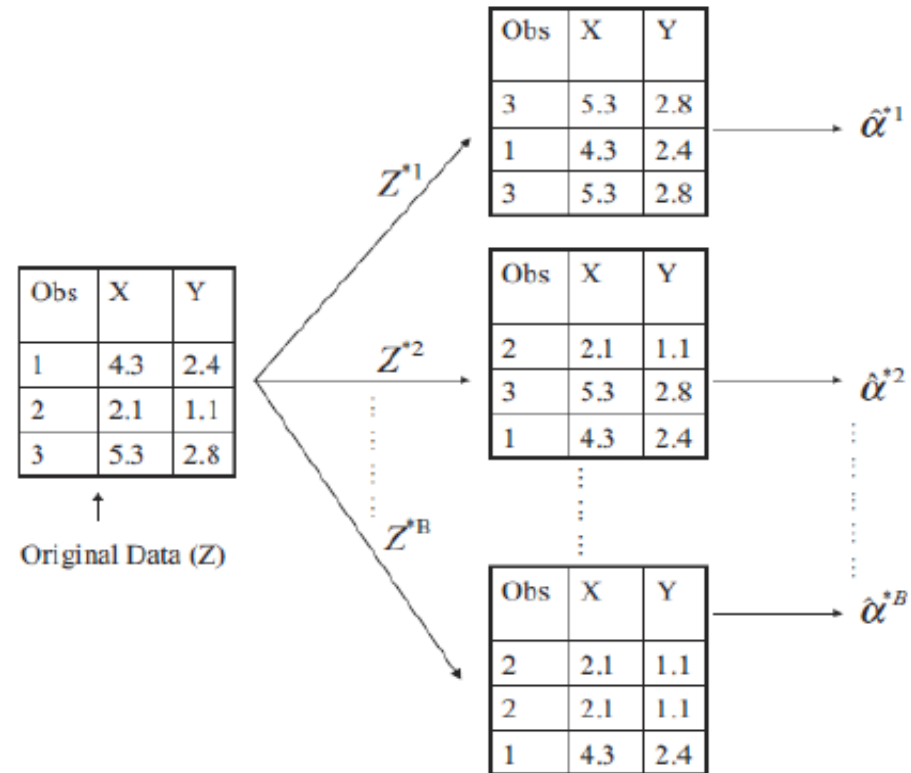




# Mejorando los modelos más simples

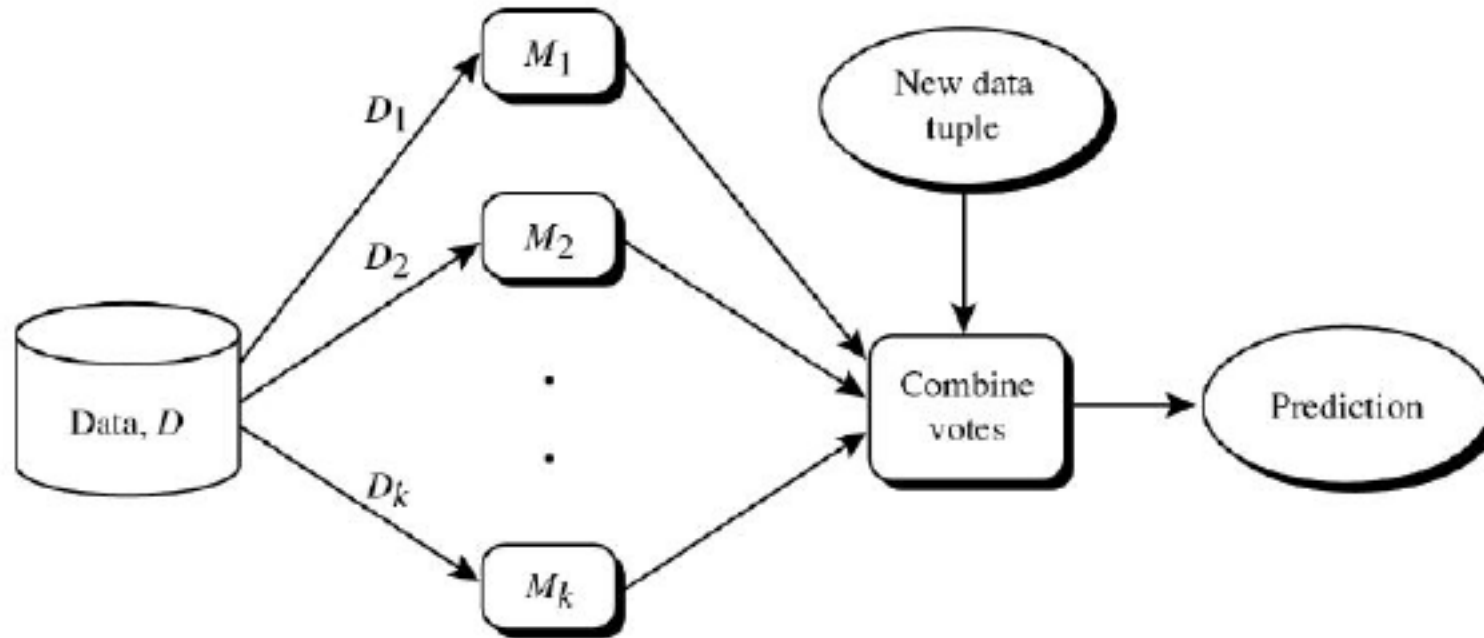
# Bootstrap

- Intuición...



# Bagging

- Con la idea de Bootstrap podemos generar métodos ensamblados de la siguiente



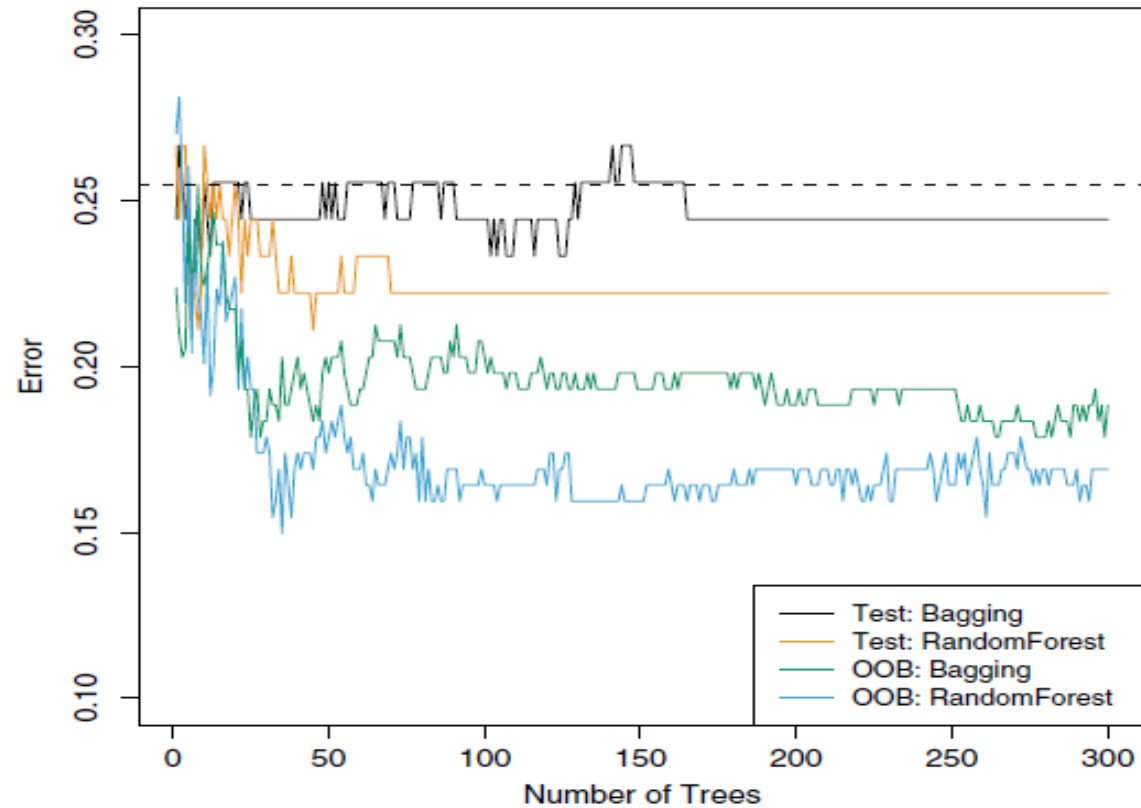
# OOB Error y LOOCV Error

---

- Al generar muestras bootstrap y ensamblar un método de clasificación (o regresión) mediante Bagging, es posible estimar el error de test sin la necesidad de llevar a cabo validación.
- Aproximadamente un 63% de los datos serán parte de cada muestra bootstrap.
- Se puede predecir y promediar el error en el “tercio” restante y estimar el error.
- Se puede demostrar que para un alto número de muestras bootstrap OOB converge a LOOCV error.



# Métodos ensamblados



# Métodos ensamblados

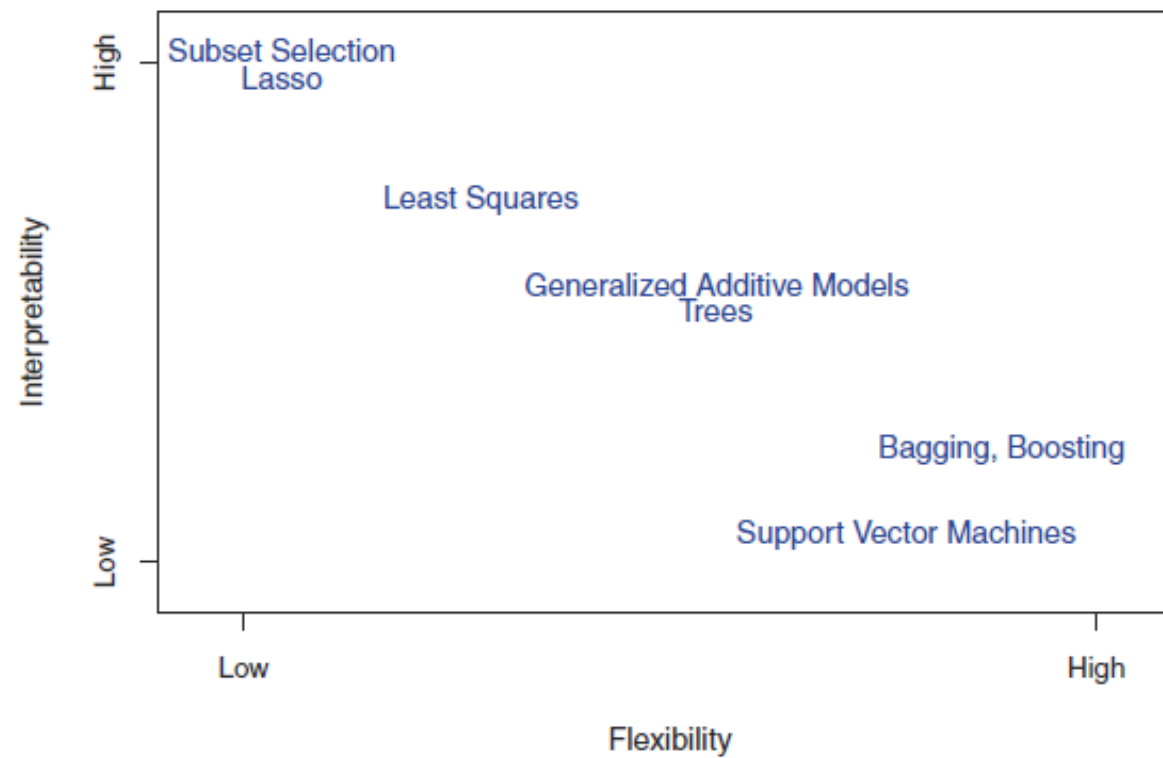
---

- Algunos de los algoritmos ensamblados más populares, son modelos basados en Árboles de decisión:
- Bagging
- Random Forest
- Adaboost

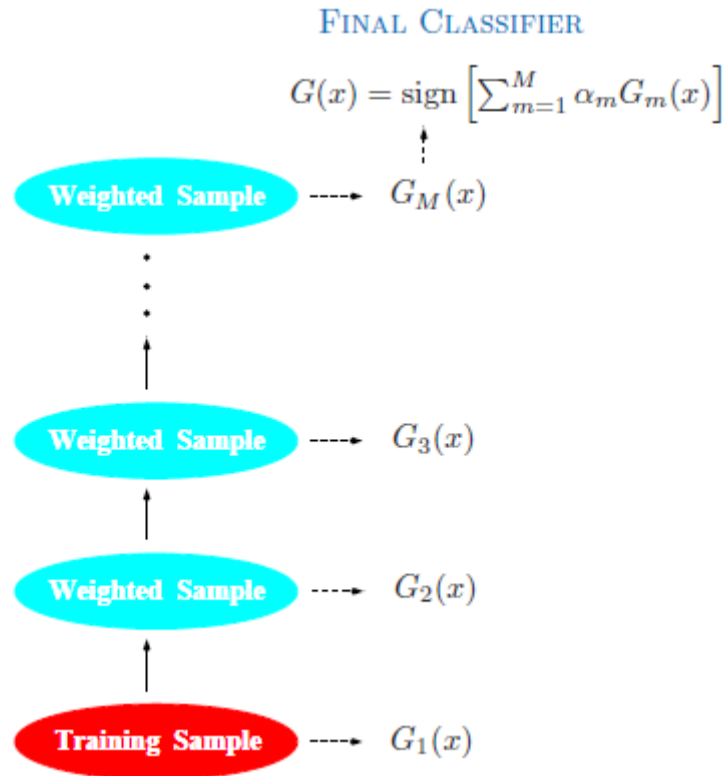


# Métodos ensamblados

---



# Adaptative Boosting (Adaboost)



---

## Algorithm 10.1 *AdaBoost.M1*.

---

1. Initialize the observation weights  $w_i = 1/N$ ,  $i = 1, 2, \dots, N$ .
  2. For  $m = 1$  to  $M$ :
    - (a) Fit a classifier  $G_m(x)$  to the training data using weights  $w_i$ .
    - (b) Compute
$$\text{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}.$$
    - (c) Compute  $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$ .
    - (d) Set  $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$ ,  $i = 1, 2, \dots, N$ .
  3. Output  $G(x) = \text{sign} \left[ \sum_{m=1}^M \alpha_m G_m(x) \right]$ .
-