

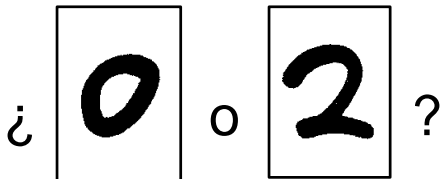
Introduction to ML

Aprendizaje automático

Crear *modelos* capaces de diferenciar elementos de acuerdo a sus características y agruparlos en órdenes o clases

- ▣ De forma automática: sin intervención humana
- ▣ Por inducción: a partir de ejemplos

Problema



Sistemas de predicción eólica

- Energía eólica: más del 18% de la energía total consumida en España.
- Limpia y sostenible.
- Difícil de controlar: ¿cuánto va a soplar el viento hoy?
- Usando modelos matemáticos pueden construirse sistemas para predecir cuánta energía va a generarse en las próximas horas.

Predicciones
meteorológicas



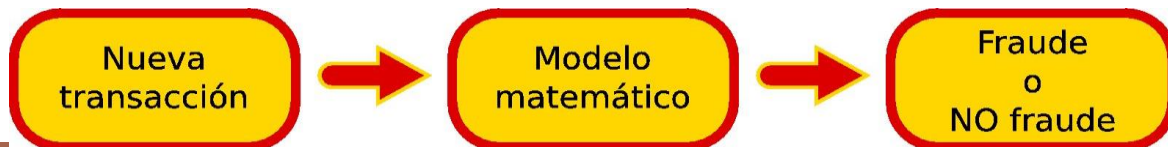
Sistema de
predicción



Predicciones
de energía

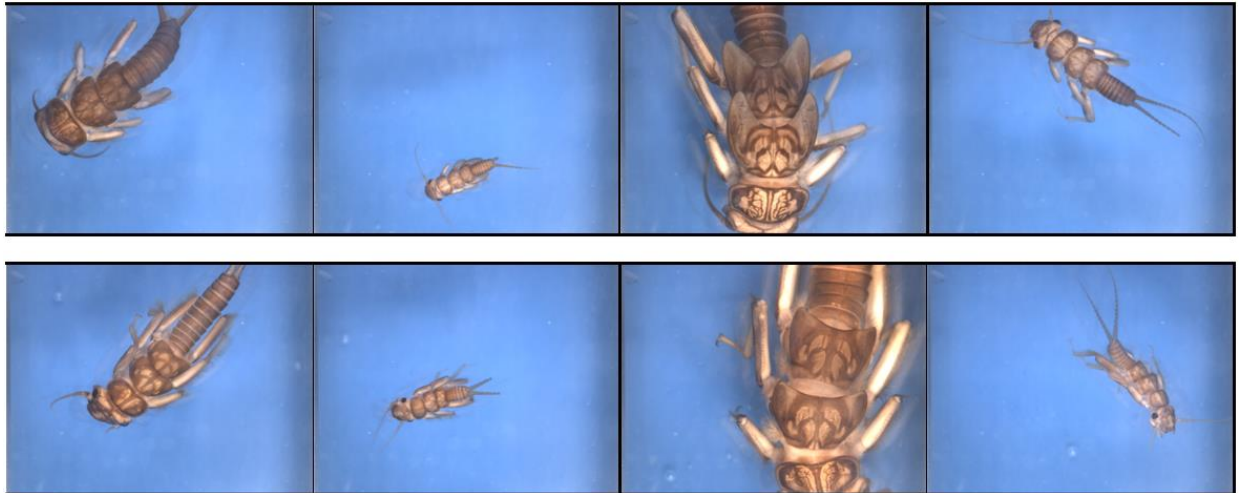
Detección/prevención de fraude en medios de pago

- ❑ Uso extendido de las tarjetas de crédito.
- ❑ Problema: ¿cuándo alguien hace uso de mi tarjeta soy realmente yo o me están suplantando ?
- ❑ Modelos matemáticos “aprenden” el comportamiento del usuario y del defraudador.
- ❑ Sistema implantado en los principales bancos de España.



Procesamiento de imágenes

□ ¿Podrías distinguir las dos especies?



Predicción de marca

STRAVA Panel de control

Llega en forma al día de la carrera. [Empieza un plan de entrenamiento](#)





London Marathon

26 de abril de 2015 | 10:00 | London, UK

Maratón · [Sitio oficial](#) →

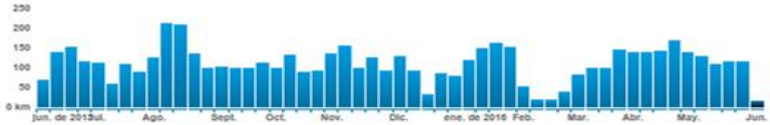
Resultados de la prueba en Strava Entrenamiento

Resultados de la prueba en Strava

Clasificación	Nombre
1	 Alex Brooke-Turner
2	 Ian Kimpton
3	 Christian Kreienbühl
4	 Stuart Spencer

Actividades para 30 de may. de 2016 - 5 de jun. de 2016 1 de jun. de 2015 - 30 de may. de 2016

18,5km | 1h 4min | 65m




0 km Jun. de 2015 Jul. de 2015 Ago. de 2015 Sept. de 2015 Oct. de 2015 Nov. de 2015 Dic. de 2015 ene. de 2016 Feb. de 2016 Mar. de 2016 Abr. de 2016 May. de 2016 Jun. de 2016

Tempo **Distancia** Desnivel positivo Semanal Mensual

martes, 31 de mayo de 2016

Morning Run
Stuart Spencer
6:06 8,4km 4:01/km


15 0



lunes, 30 de mayo de 2016

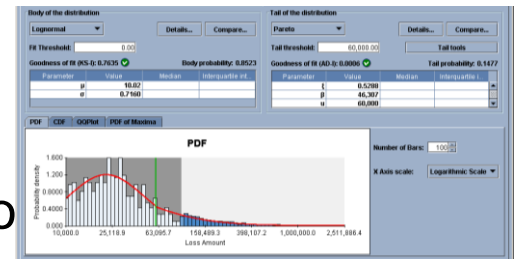
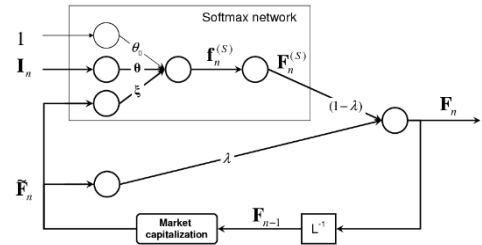
Stuart Spencer ha corrido con Chris Greenwood, Ralph Street y **24 personas más**

STRAVA LABS
[Ver Flybys](#) →



Finanzas

- Optimización avanzada
 - Carteras de inversión
 - Index tracking
- Aprendizaje por refuerzo
 - Gestión dinámica de carteras
- Análisis de riesgos
 - Métodos Monte Carlo
 - Métodos bootstrap
 - Teoría de valores extremo



Recolección de datos

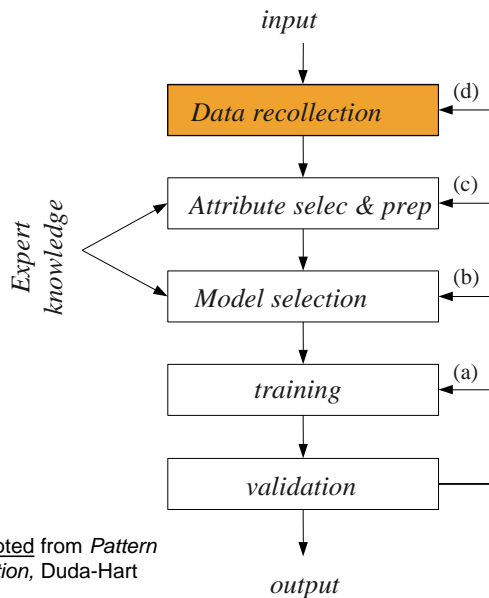
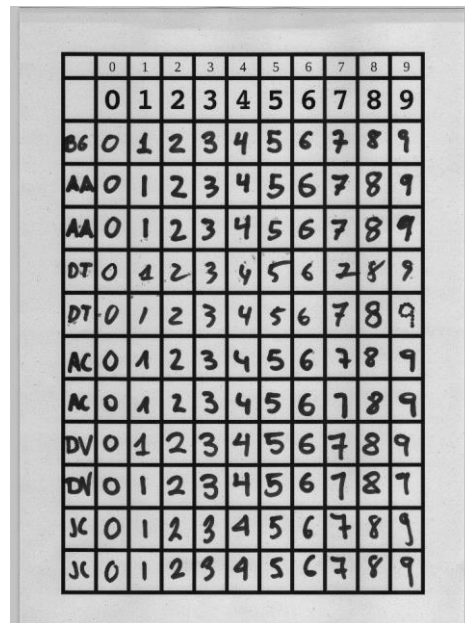


Fig. adapted from *Pattern Recognition*, Duda-Hart



Selección de atributos y preprocesado

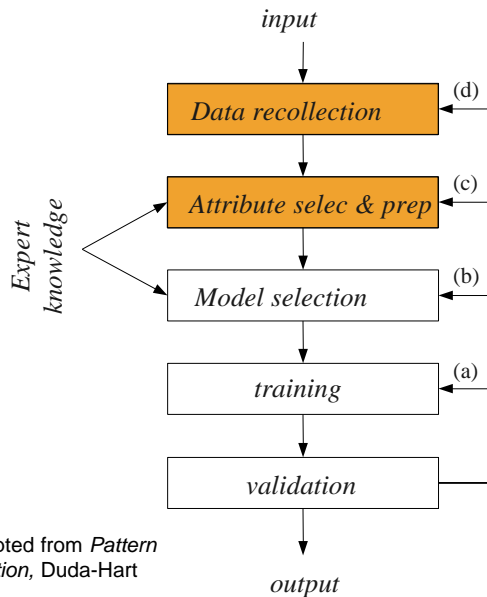
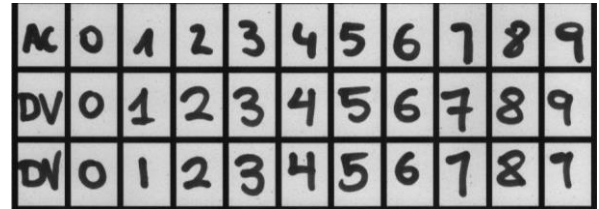


Fig. adapted from *Pattern Recognition*, Duda-Hart

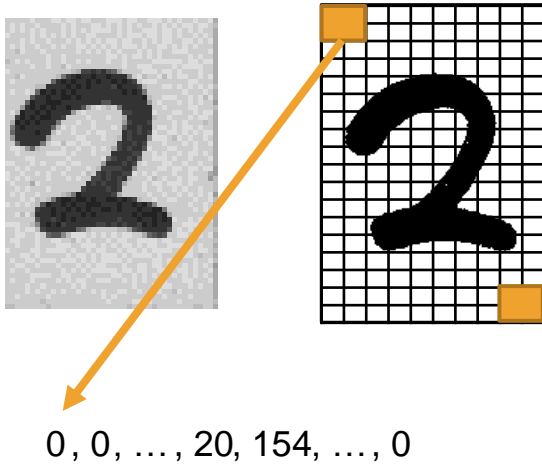


Think about the elements that distinguishes between numbers

Attributes:

- Pixels
- Edges
- ...

Selección de atributos y preprocesado



Applied process:

1. Crop each digit.
2. Binarize image to black and white and center the digit in the image
3. Create a grid (17x9=153 attributes)
4. For each grid node we average the values of the 4 adjacent cells

Selección del modelo

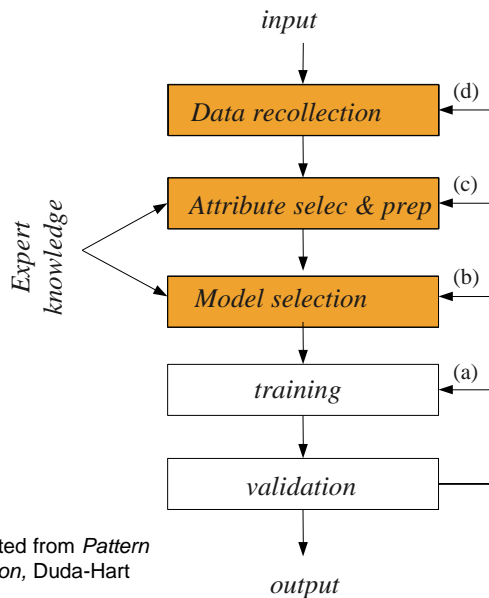
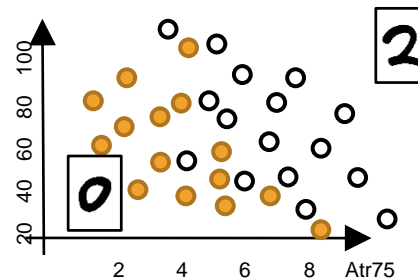


Fig. adapted from *Pattern Recognition*, Duda-Hart

Atr11



Which is the best model?

- Decision tree
- Ensemble
- SVM

Entrenamiento

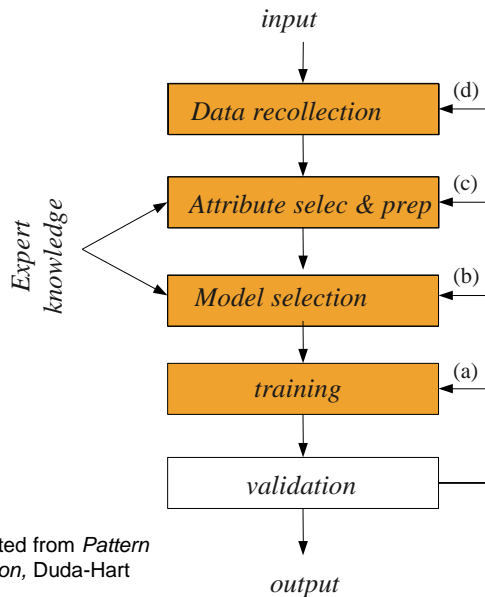
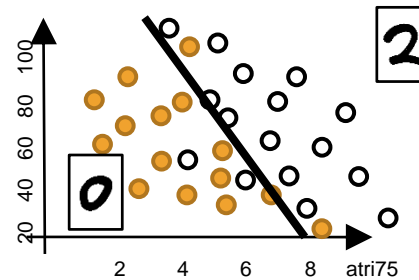


Fig. adapted from *Pattern Recognition*, Duda-Hart

Atr11



Split the data into training and test

The data in training is only used to train the model!!!

Validación

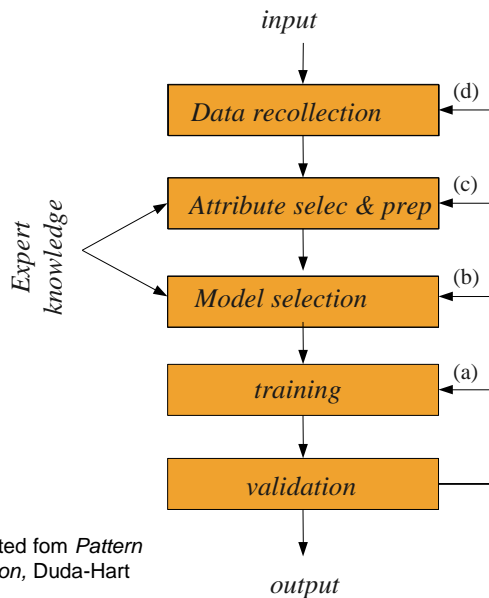
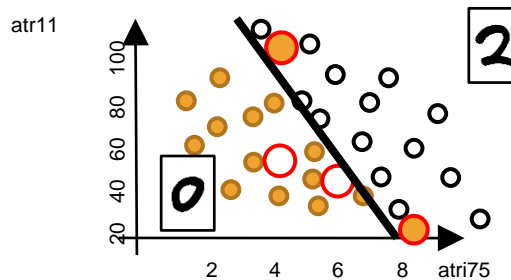


Fig. adapted from *Pattern Recognition*, Duda-Hart



Validate the model using
the left out data.

Test data

Entrenamiento de otro modelo

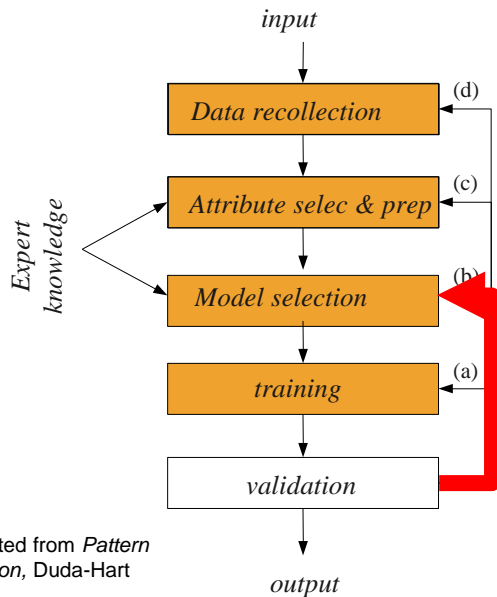
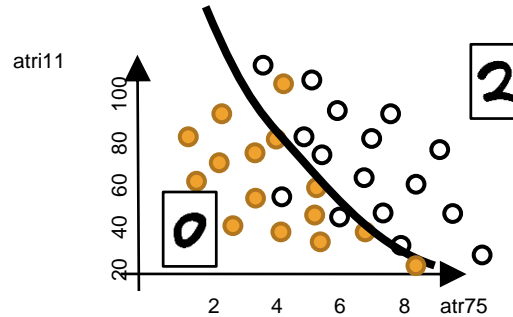


Fig. adapted from *Pattern Recognition*, Duda-Hart



Configure the new model and train it

We have to use the same training data to be able to make a fair comparison!!!

Validación

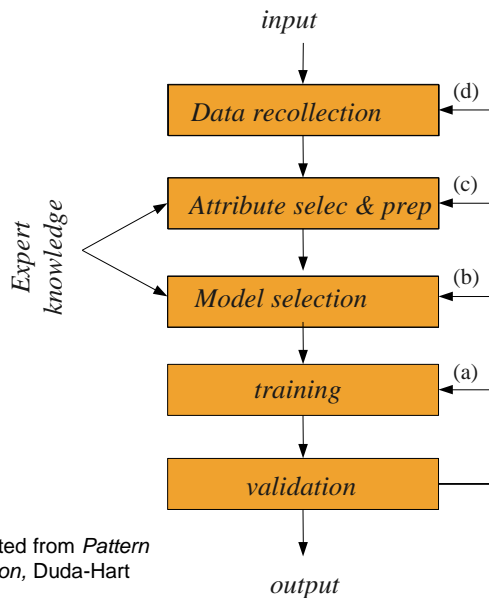
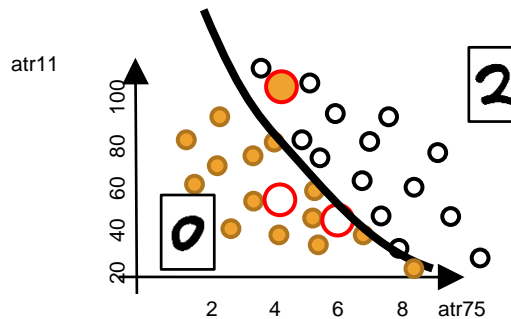


Fig. adapted from *Pattern Recognition*, Duda-Hart



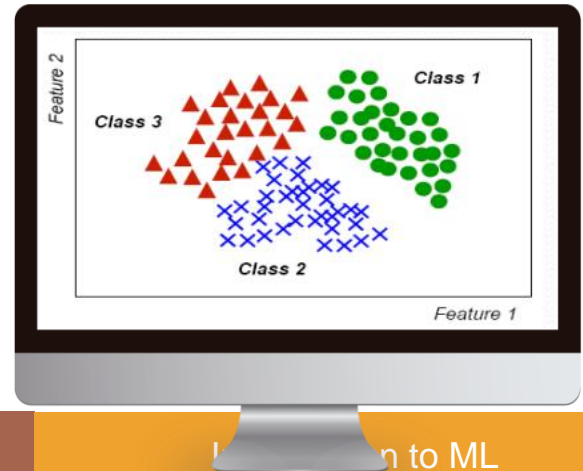
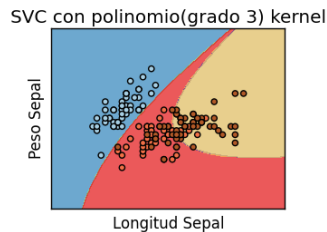
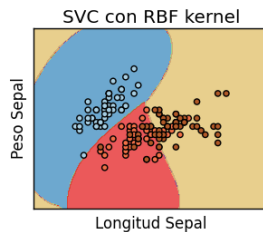
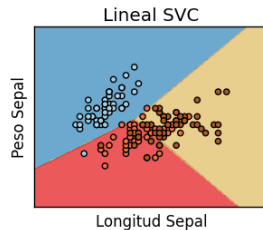
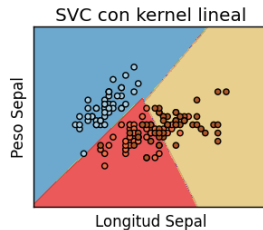
Validate the new model
with the same test data
and
compare results

Tipos de aprendizaje

- Supervidado
 - ▣ Clasificación: La variable de salida es categórica
 - ▣ Regresión: La variable de salida es numérica
- No supervisado
 - ▣ Clústering
 - ▣ Reglas asociativas
- Semi-supervisado
- Aprendizaje por refuerzo

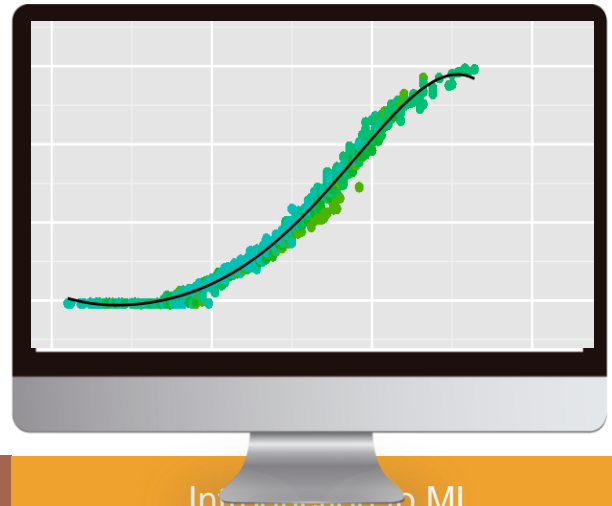
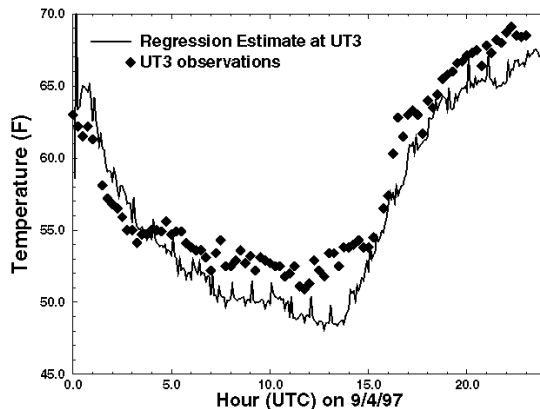
Aprendizaje supervisado - Clasificación

- El objetivo es predecir la categoría de un nuevo objeto/instancia/observación
- La salida del Sistema es la etiqueta de clase
 - Ejemplo: clasificar un product "good" o "bad" in control de



Supervised Learning - Regression

- Generalization of the classification problem
- The system output is a number / real vector
 - Example: to predict the temperature of the next week



Unsupervised Learning – Reglas asociativas

□ Example: Basket analysis

$P(Y|X)$ probability that somebody who buys X also buys Y where X and Y are products/services $\rightarrow P(\text{chips} | \text{beer}) = 0.7$

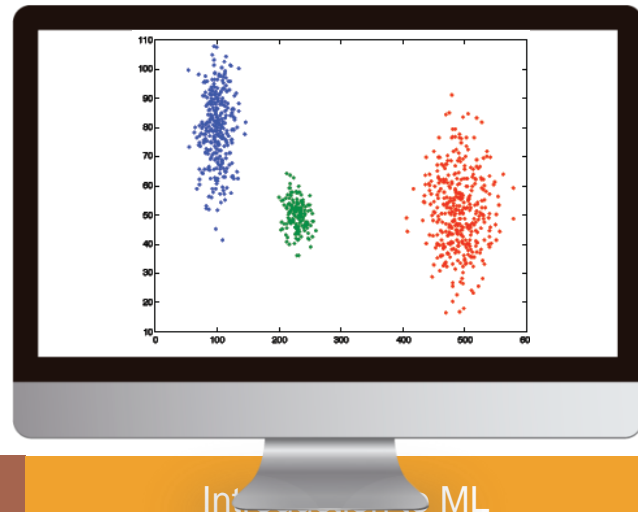
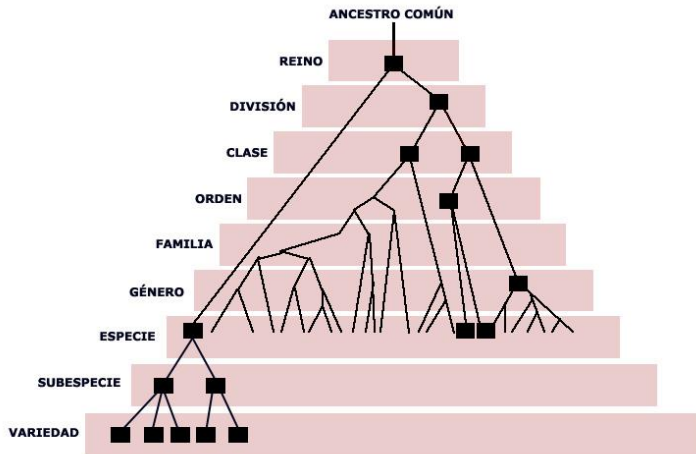


Market-Basket transactions

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Unsupervised Learning - Clustering

- The problem of organizing objects into groups that make sense: similar within cluster and dissimilar between clusters
- The system can organize objects in a hierarchical way
 - ▣ Example: to arrange plants in a taxonomy of species



Semisupervised

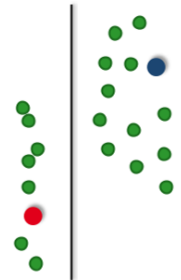
□ People want better performance for free

- unlabeled data is cheap
- labeled data can be hard to get
- human annotation is boring
- labels may require experts
- labels may require special devices

only labeled data



with unlabeled data

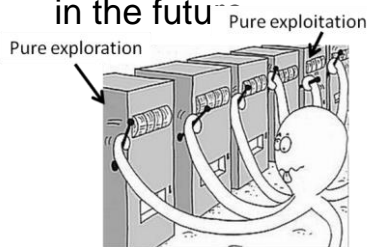


□ Goal

- Using both labeled and unlabeled data to build better learners than using each one alone

Reinforcement Learning

- Reinforcement Learning : the model interacts with the environment seeking ways to maximize the reward. There is a feedback from the environment.
- Objective: get as much rewards as possible.
- Trade-off between exploration and exploitation:
 - The agent has to *exploit* what it already knows in order to obtain reward.
 - The agent also has to *explore* in order to make better action selections in the future



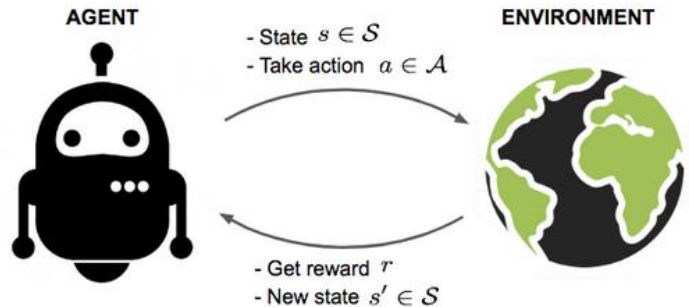
Bandits = heuristics. Image from research.microsoft.com



Reinforcement Learning

- **Agent** takes actions
 - ▣ Drone making a delivery
 - ▣ Estudiante realiza un examen
 - ▣ Autonomous car driving
- **Actions** (A) is the set of all possible moves that an agent can make
 - ▣ Aereal drones: different velocities and accelerations in 3D space

The goal of Reinforcement Learning is to learn a good strategy for the agent from experimental trials and feedback received.



Terminología: Atributos y ejemplos

Atributo = variable = característica = valores de entrada = variables independientes: Columnas de la tabla. Varios tipos

- Nominales o categóricas: P.e. color, ciudad,
- Numéricas: Altura

+

Clase o variable de salida o variable dependiente o etiqueta

=

Ejemplo (o caso o instancia o patrón): Conjunto de atributos etiquetados o no que representan un objeto

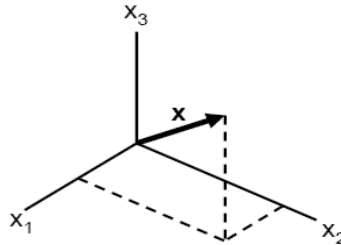
Se supone que las instancias de objetos de una misma clase tienen valores similares

Terminología: Atributos y ejemplos

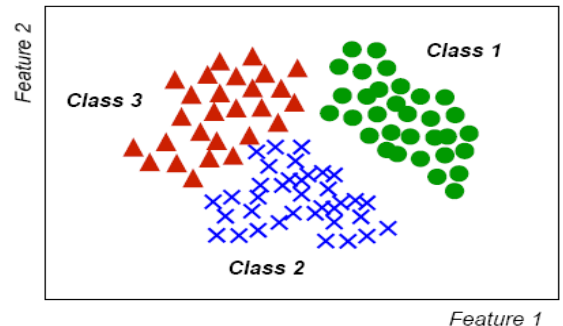
- Cada ejemplo tiene:
 - ▣ un conjunto de atributos (o vector D-dimensional) llamado vector de atributos
 - ▣ Dependiendo del problema una etiqueta continua o discreta
- El espacio D-dimensional definido por este vector es el espacio de atributos con D el número de atributos
- Los ejemplos se representan como puntos en este espacio

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

Vector de
atributos



Espacio de
atributos



Datos

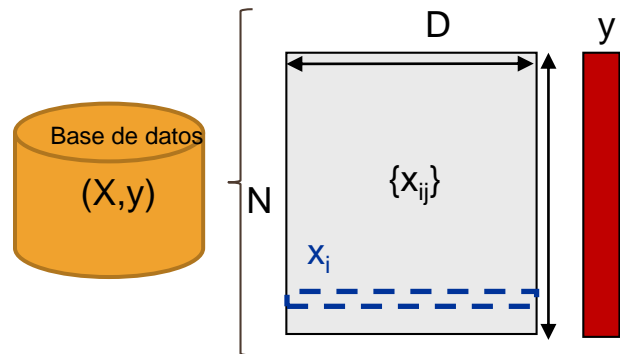
$$\mathcal{D} = \{(\bar{\mathbf{x}}_i, y_i), i = 1, 2, \dots, N\}$$

\mathcal{D} : Conjunto de datos de entrenamiento

\mathbf{x}_i : Vector D-dimensinal de atributos del ejemplo i

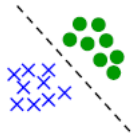
y_i : Etiqueta de clase del ejemplo i

N : Número de ejemplos



Atributos y ejemplos

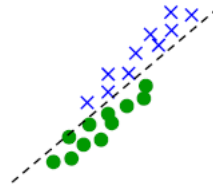
Conceptos relacionados con los atributos



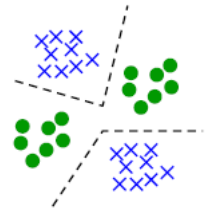
Lineal Separability



Non lineal Separability



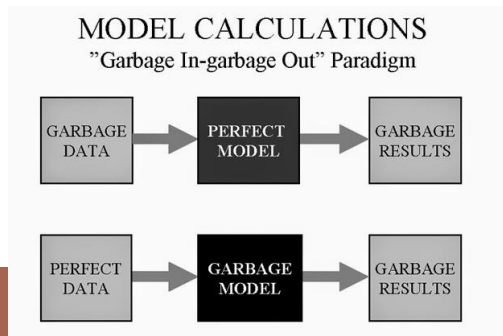
Highly
Correlated attributes



Multi-modal

Proceso de aprendizaje automático (parte i)

- Recolección de datos
- Selección de atributos y preprocesado
 - Limpieza de datos: valores omitidos (missing values), gestión de anomalías (outliers), corrección de ruido, etc.
 - Integración de datos: Provenientes de distintas fuentes
 - Transformación: Construcción de atributos, PCA, equilibrado de clases, normalización
 - Reducción: reducción de dimensionalidad con PCA, etc.
 - Selección de atributos: Es un elemento crítico

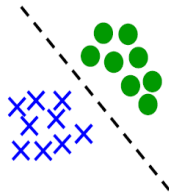


Proceso de aprendizaje automático (parte i)

What is a "good" vector of attributes?

The quality of a vector of attributes is related to its ability to discriminate examples from different classes:

- The attributes of instances of the same class should have similar values
- The attributes of instances from different classes should have different values



"Good" features



"Bad" features

Proceso de aprendizaje automático (parte i)

- Recolección de datos
- Preprocesado Relacionado con el modelo elegido
 - Missing values
 - Detección de anomalías
 - Equilibrado de clases
 - Normalización de atributos
 - Conversión de atributos categóricos a continuos: One-hot encoding
 - Conversión de atributos continuos a categóricos: discretización

Proceso de aprendizaje automático (parte ii)

- Selección del modelo
- Entrenamiento
- Evaluación

Proceso de aprendizaje automático (parte ii)

➤ Selección del modelo:

➤ Diversos modelos

- Naive Bayes
- Vecinos próximos
- Modelos lineales
- SVM
- Conjuntos de clasificadores
- Redes neuronales
- Etc.

➤ No solo es seleccionar el modelo sino también sus hiperparámetros

¿Diferencia entre parámetros e hiperparámetros de un modelo?

Proceso de aprendizaje automático (parte ii)

➤ Selección del modelo:

- Diversos modelos **con algunos de sus hiperparámetros**
 - Naive Bayes
 - Vecinos próximos: número de vecinos
 - Modelos lineales: constante de aprendizaje
 - SVM: tipo de kernel, C, gamma
 - Conjuntos de clasificadores: profundidad árboles, etc.
 - Redes neuronales: Número de capas y sus tipos, solver, etc.
 - Etc.
- No solo es seleccionar el modelo sino también sus hiperparámetros

¿Diferencia entre parámetros e hiperparámetros de un modelo?

¿Diferencia entre métodos paramétricos y no paramétricos?

Proceso de aprendizaje automático (parte ii)

➤ Selección del modelo:

➤ Diversos modelos ¿Son paramétricos?

- Naive Bayes: Sí
- Vecinos próximos: No
- Modelos lineales: Sí
- SVM: No
- Conjuntos de clasificadores: No
- Redes neuronales: Sí
- Etc.

➤ No solo es seleccionar el modelo sino también sus hiperparámetros

¿Diferencia entre parámetros e hiperparámetros de un modelo?

¿Diferencia entre métodos paramétricos y no paramétricos?

Proceso de aprendizaje automático (parte ii)

➤ Entrenamiento/Validación

- Se usan los datos para entrenar muchos modelos y se elige el mejor modelo+configuración de hyperparámetros.
 - Hay que definir qué es mejor -> métrica
 - Hay que elegir el proceso de validación

Selección de modelos

Menos complejos

Menos flexibles

Más robustos

Tendencia a sub ajustar

Más complejos

Más flexibles

Menos robustos

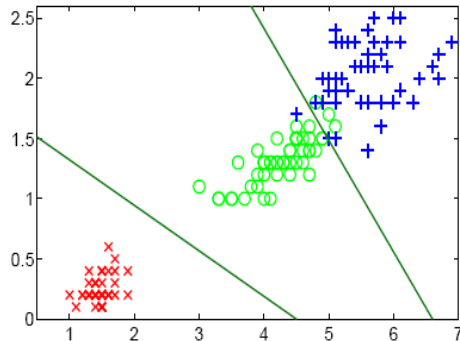
Tendencia a sobre ajustar



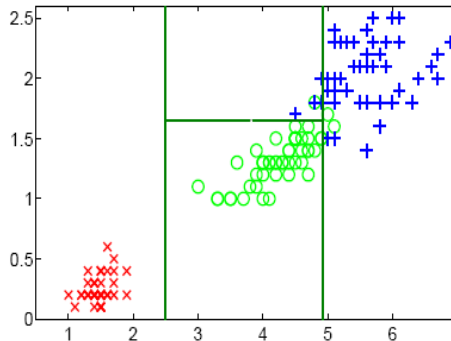
Modelos

Steps to develop a prediction model: **select models**

Discriminante Lineal



Árboles de decisión

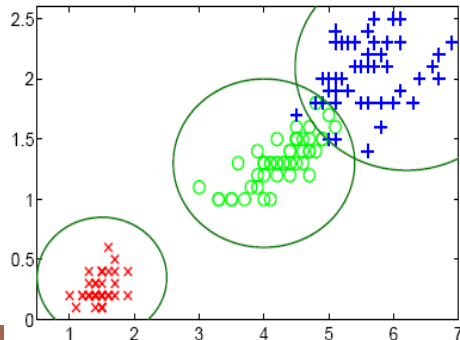


Representación:
Iris Data Set
(Fisher, 1936)

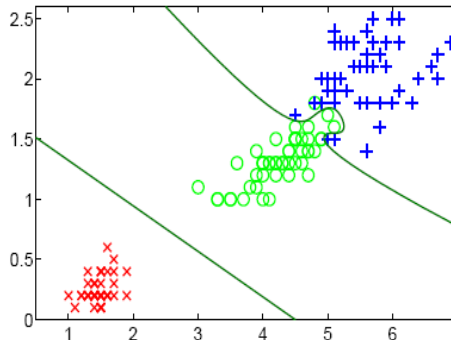
setosa

versicolor

virginica



Mezclas de Gausianas



Kernel method (SVM)

roduction to ML

Proceso de aprendizaje automático (parte ii)

➤ Métrica: define qué método es mejor

➤ Clasificación:

- Error de generalización
- Área bajo la curva (AUC)
- Precisión/recall/F1

➤ Regresión

- Error cuadrático medio (MSE)
- Error absoluto medio (MAE)

➤ Clústering

- AIC, BIC

Proceso de aprendizaje automático (parte ii)

➤ Métricas: visualización

➤ Clasificación:

- Matriz de confusión
- Curva ROC

➤ Regresión

- Gráfico de dispersión predicción vs. real

Matriz confusión y métricas

	True condition		
Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{Condition positive}}{\Sigma \text{Total population}}$
Predicted condition positive	True positive	False positive (Type I error)	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{True positive}}{\Sigma \text{Test outcome positive}}$
Predicted condition negative	False negative (Type II error)	True negative	False omission rate (FOR) = $\frac{\Sigma \text{False negative}}{\Sigma \text{Test outcome negative}}$
Accuracy (ACC) = $\frac{\Sigma \text{True positive} + \Sigma \text{True negative}}{\Sigma \text{Total population}}$	True positive rate (TPR), Sensitivity, Recall = $\frac{\Sigma \text{True positive}}{\Sigma \text{Condition positive}}$	False positive rate (FPR), Fall-out = $\frac{\Sigma \text{False positive}}{\Sigma \text{Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$
	False negative rate (FNR), Miss rate = $\frac{\Sigma \text{False negative}}{\Sigma \text{Condition positive}}$	True negative rate (TNR), Specificity (SPC) = $\frac{\Sigma \text{True negative}}{\Sigma \text{Condition negative}}$	Negative likelihood ratio (LR-) = $\frac{\text{FNR}}{\text{TNR}}$

Steps to develop a prediction model: **select models**

- Sensibilidad (“sensitivity”), Recall :

- ▣ De la clase (+) ¿qué fracción reconozco como (+)? $TPR_{Sens} = \frac{TP}{TP + FN}$

- ▣ Sobre los aciertos, ¿cuántos son (+)?

- ▣ Un valor alto de “Recall” significa que el algoritmo recupera más casos (+) que (-).

- Especificidad (“specificity”):

$$Spec = \frac{TN}{TN + FP}$$

- ▣ De la clase (-) ¿qué fracción reconozco como (-)?

- Exactitud (“accuracy”)

- ▣ Del total de la muestra ¿qué fracción clasifico bien, tanto (+) como (-)? $acc = \frac{TN + TP}{P + N}$

- Precisión (“precision”):

- ▣ Sobre lo clasificado como (+), ¿qué fracción es realmente (+)?

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

Steps to develop a prediction model: **select models**

Ejemplo: (+) Estar enfermo, (-) Estar sano

- La sensibilidad nos indica la capacidad de nuestro estimador para dar como casos positivos:

- los casos realmente enfermos;
- proporción de enfermos correctamente identificados.

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

Es decir, la sensibilidad caracteriza la capacidad de la prueba para detectar la enfermedad en sujetos enfermos.

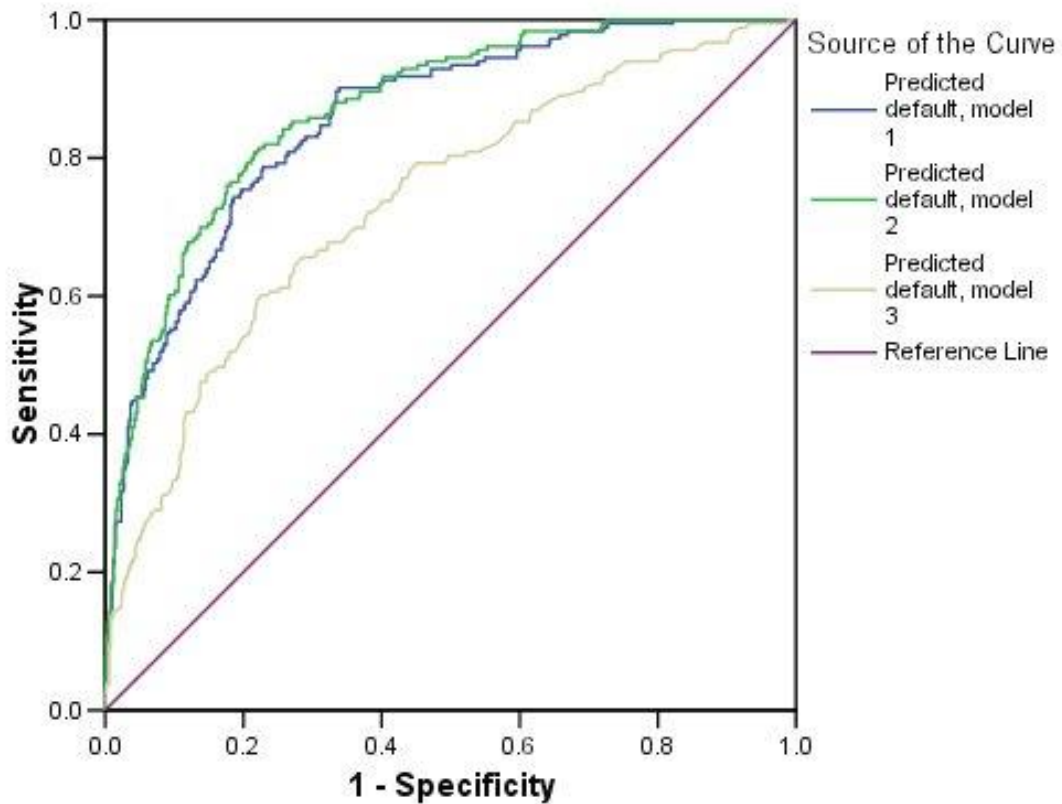
- La especificidad nos indica la capacidad de nuestro estimador para dar como casos negativos los casos realmente sanos;

- proporción de sanos correctamente identificados.

Es decir, la especificidad caracteriza la capacidad de la prueba para detectar la ausencia de la enfermedad en sujetos sanos.

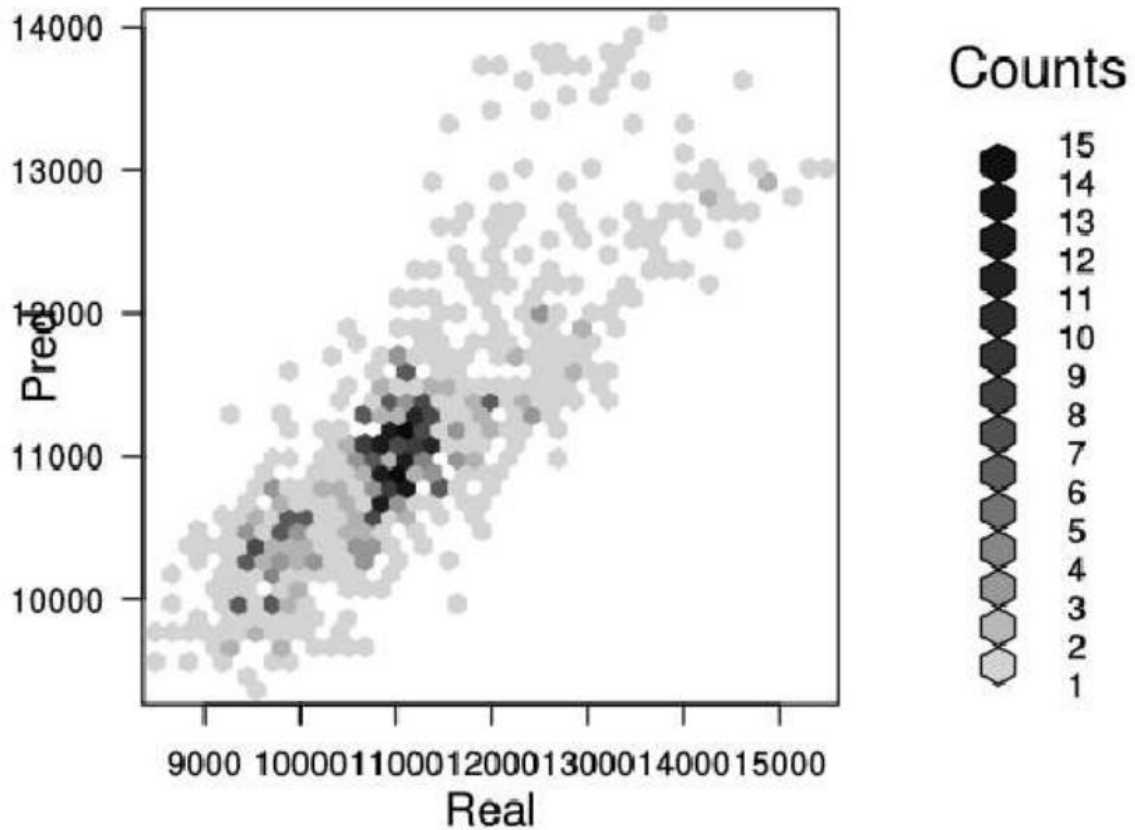
$$Spec = \frac{TN}{TN + FP}$$

Curva ROC



Diagonal segments are produced by ties.

Dispersión (regresión)

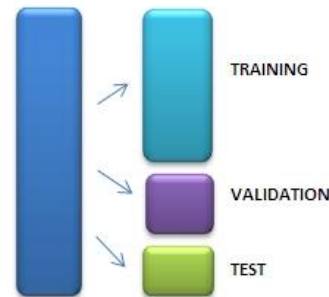


Proceso de aprendizaje automático (parte ii)

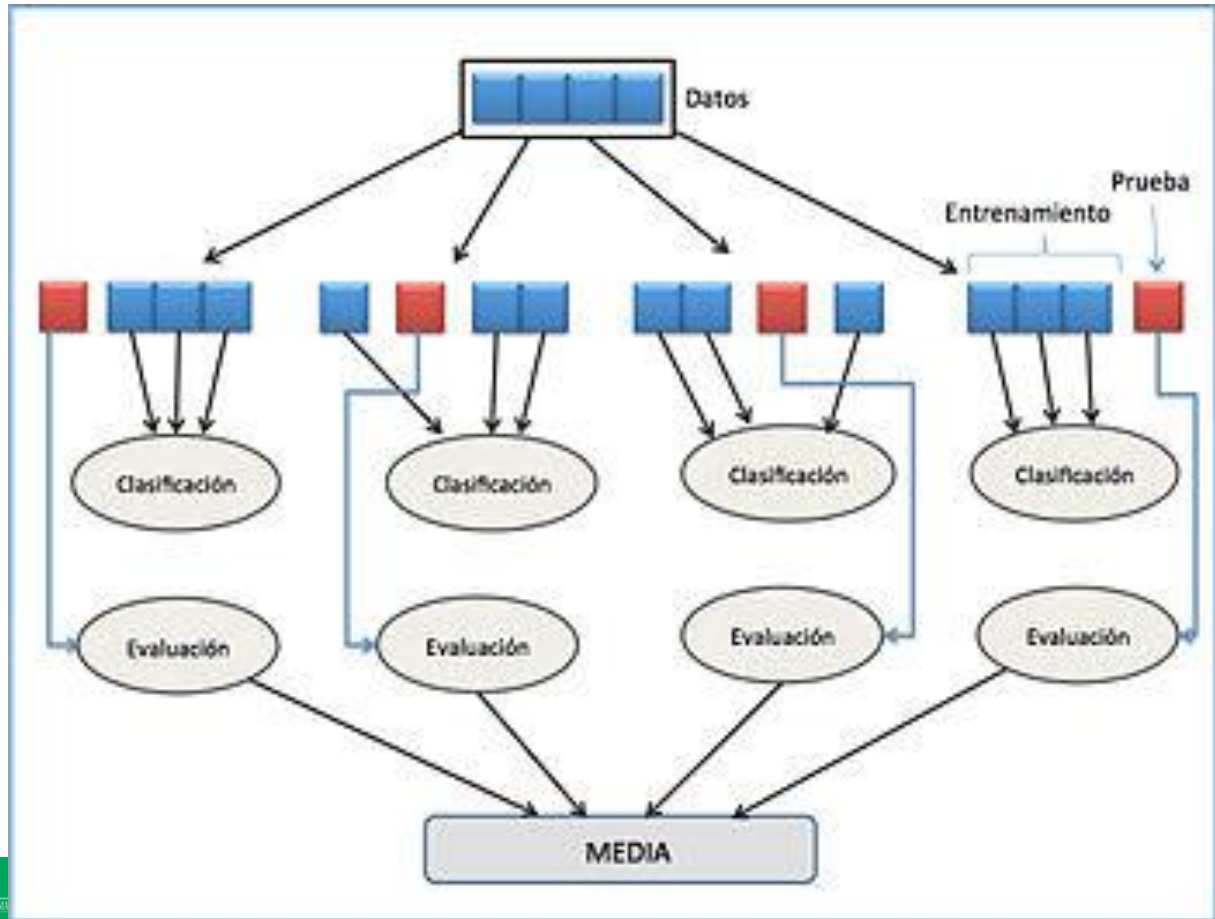
➤ Proceso de validación

- Partición aleatoria simple de datos en train/validación/test: **Solo si hay muchos datos**
- Múltiples particiones aleatorias en partición de datos train/validación/test
- Validación cruzada en K-pliegues (K-fold cross validation)
- Leave-out-out : **Solo si hay muy pocos datos**

Pueden ser estratificados en clasificación



K-fold cross-validation para $K=4$



Validación anidada

- ¡¡¡Un buen proceso de validación debe ser un proceso anidado con dos niveles de validación!!!
- Veamos un ejemplo: Queremos comparar:
 - SVM: con kernel RBF y $C=\{1, 1000\}$ y $\gamma=\{0.001, 1\}$
 - Árbol de decisión con profundidad máxima= $\{5, -1\}$ y criterio= $\{\text{gini}, \text{entropy}\}$
 - 3-fold cross-validation con 3-fold cross-validation en entrenamiento

Validación anidada

- Dividimos datos en 3 (1)+(2)+(3)
 - Train=(1)+(2) y Test (3)
 - Train=(1)+(3) y Test (2)
 - Train=(2)+(3) y Test (1)
- En cada test solo se va a probar la mejor combinación de SVM con sus hiperparámetros y la mejor de DT con los suyos
 - Estos mejores hiperparámetros se obtendrán en train con otra validación cruzada

Validación anidada

- Para cada Train dividimos los datos en 3 $(1')+(2')+(3')$
 - $\text{Train}'=(1')+(2')$ y Validación $(3')$
 - Usamos train' para entrenar todas las posibles combinaciones de parámetros de SVM y DT
 - Las validamos en Validación con la métrica elegida
 - $\text{Train}'=(1')+(3')$ y Validación $(2')$
 - Idem
 - $\text{Train}'=(2')+(3')$ y Validación $(1')$
 - Idem
- Seleccionamos la SVM y el DT con mejor resultado medio en Val
- Esas combinaciones se usan para generar un modelo final usando todo el train $(1')+(2')+(3')$ y se valida en Test

Validación anidada

- Esto se denomina búsqueda en rejilla
- Mañana vamos a implementar esto en python con sklearn. Miraos
 - Kfold y StratifiedKFold
 - GridSearchCV
 - scoring
 - Pipeline
 - Grid
 - kfold