

# Aprendizaje automático

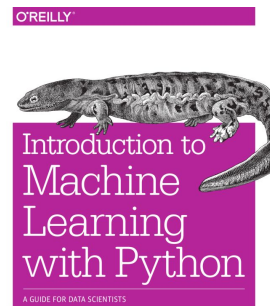
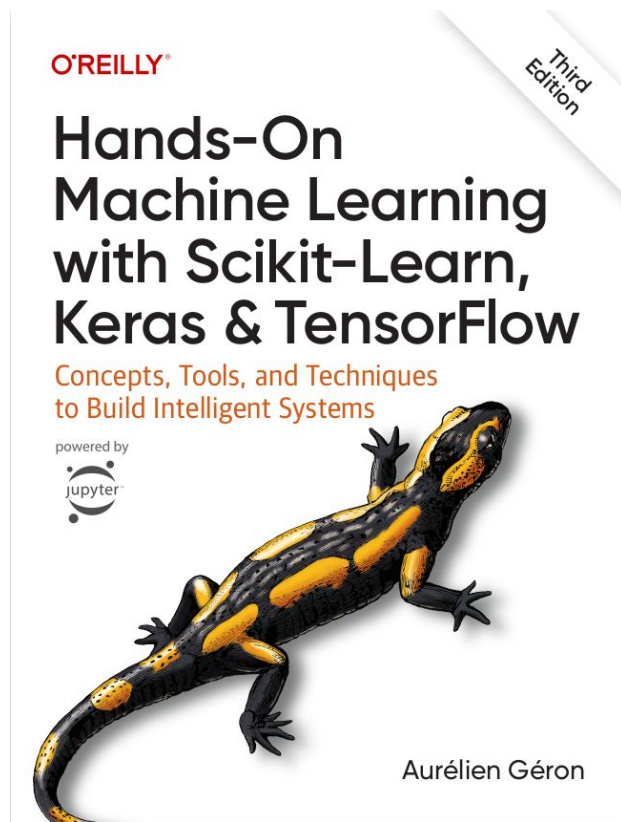


FACULTAD DE  
INGENIERÍA

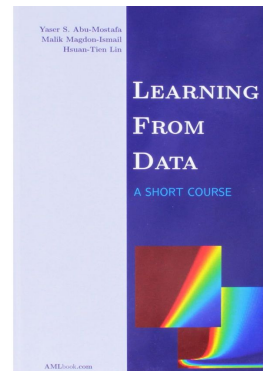


UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY

# Bibliografía



Andreas C. Müller & Sarah Guido



# Algunos cursos de grado y posgrado

- Fundamentos de Aprendizaje Automático  
Curso de grado de Ingeniería Eléctrica
  - Material: <https://eva.fing.edu.uy/course/view.php?id=1252>
  - Videos: <https://open.fing.edu.uy/courses/fuaa/>
- Taller de Aprendizaje Automático  
Curso de grado de Ingeniería Eléctrica
  - Material: <https://eva.fing.edu.uy/course/view.php?id=1492>
  - Videos: <https://open.fing.edu.uy/courses/taa/>
- DLVis (Deep Learning para Visión Artificial)
  - Material: <https://eva.fing.edu.uy/course/view.php?id=1046>
  - Videos: <https://www.youtube.com/@DLVISFing>

# ¿Qué es el aprendizaje automático?



FACULTAD DE  
INGENIERÍA



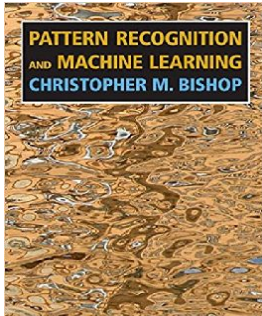
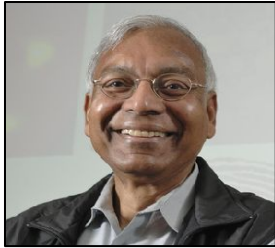
UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY



# Aprendizaje automático

*El estudio de cómo las máquinas pueden, observando el ambiente, aprender a distinguir patrones de interés de un fondo y a tomar decisiones razonables sobre categorías de los mismos.*

*El Reconocimiento de Patrones es una disciplina que estudia métodos que permiten identificar objetos, tipos de*



*regularidades en los datos usando algoritmos en computadora y a partir de esas regularidades tomar acciones.*

aprendizaje: mejorar el desempeño de una tarea con la experiencia.

medida de desempeño

datos

aplicación



## Algunas definiciones más

El aprendizaje automático es la ciencia (y el arte) de programar computadoras para que puedan aprender de los datos.

—Yaser Abu-Mostafa \*

El aprendizaje automático es el campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programadas explícitamente.

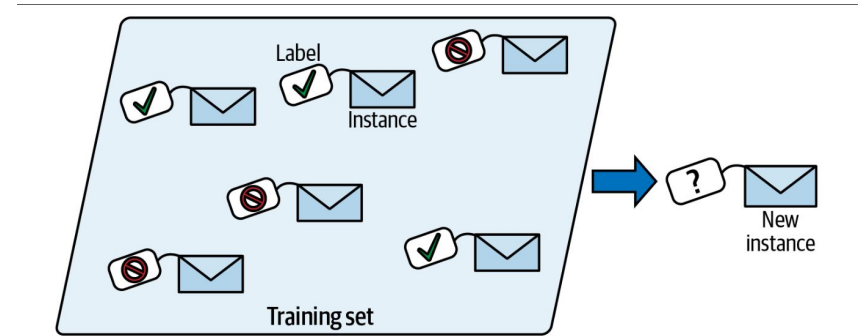
—Arthur Samuel, 1959

Una computadora aprende de la experiencia  $E$  con respecto a alguna tarea  $T$  y alguna medida de desempeño  $P$ , si su desempeño en  $T$ , medido por  $P$ , mejora con la experiencia  $E$ .

—Tom Mitchell, 1997

# Ejemplo - Filtro de spam

- Tarea T
  - Identificar si un correo es o no spam
- Experiencia E
  - Conjunto de correos de ejemplo que son spam y que no son spam identificados por los usuarios
- Desempeño o performance P
  - Medido por ejemplo como la tasa de aciertos

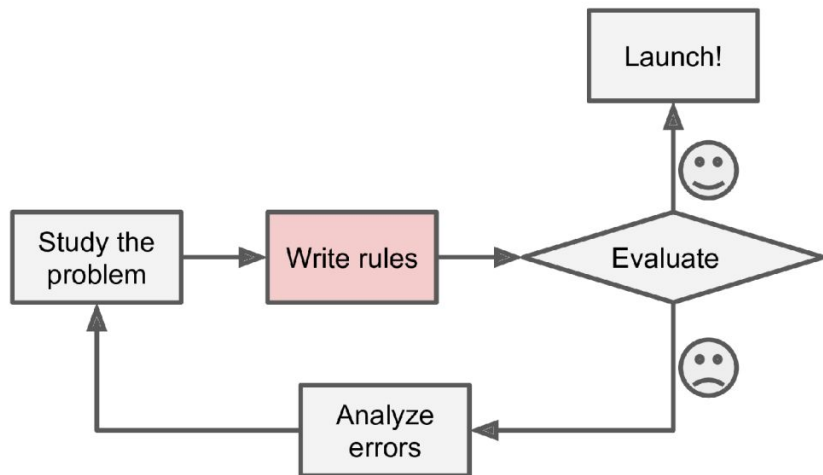




# Esquema tradicional sin AA

1. Analizar el problema. Encontrar algunos patrones en los correos que son spam
  - Algunas palabras que aparecen frecuentemente en el asunto
    - gratis, felicitaciones, has ganado, dinero, extra, sexo, etc
  - Algunos patrones en el nombre del remitente y el cuerpo del correo
2. Programar un detector para cada uno de los patrones encontrados en el paso 1.
3. Testear el programa e iterar los pasos 1 y 2 hasta que el filtro tenga un desempeño adecuado para ponerlo en línea.

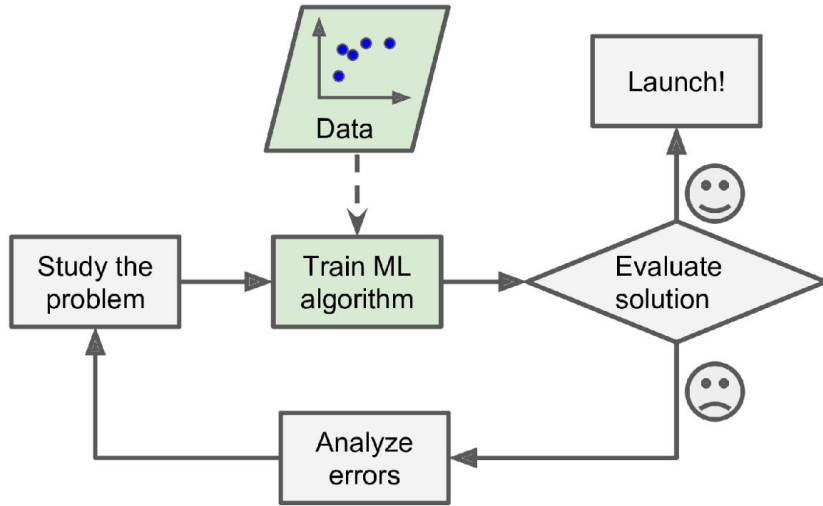
# Esquema tradicional sin AA



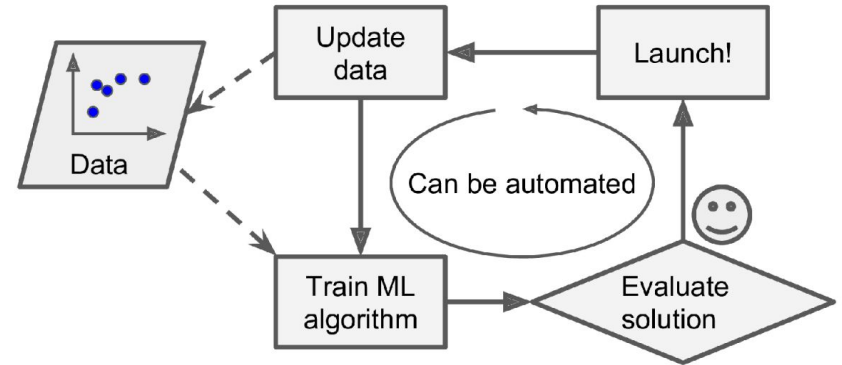
## Dificultades del encare::

- Tenemos un problema complejo
- Solución es una larga lista de reglas complejas
- Habrá que descubrir constantemente nuevos patrones y actualizar reglas
  - Ej. los spammers empiezan a escribir GR4TI\$ en lugar de gratis
- El problema no escala bien
- El mantenimiento se hace muy difícil

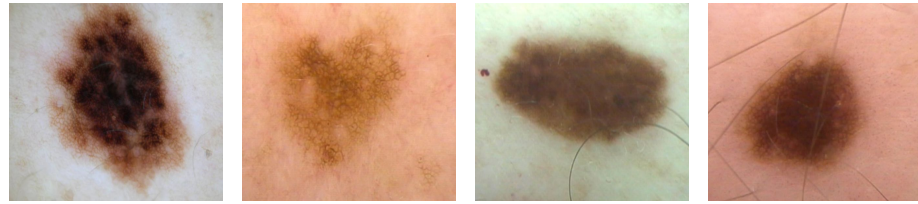
# Esquema de AA



Se puede automatizar



# Ejemplo: Lesiones de piel



**Table 1**

ABCD rule. Points in the third column are multiplied by its corresponding weight factor. The TDS is computed as the sum of the sub-scores of each feature. Lesion classification: benign ( $TDS < 4.75$ ), CDL ( $4.75 \leq TDS \leq 5.45$ ), malignant ( $TDS > 5.45$ ).

Feature	Description	Points	Weight factor	Sub-score range
Asymmetry	One point per asymmetry w.r.t. each axis	0–2	1.3	0–2.6
Border	Eight segments, one point for abrupt pigment cutoff	0–8	0.1	0–0.8
Color	One point per color: white, red, black light brown, dark brown, blue-gray	1–6	0.5	0.5–3
Dermoscopic structures	One point per structure: pigment network, structureless area, dots, globules, branched streaks	1–5	0.5	0.5–2.5
Total score range				1.0–8.9

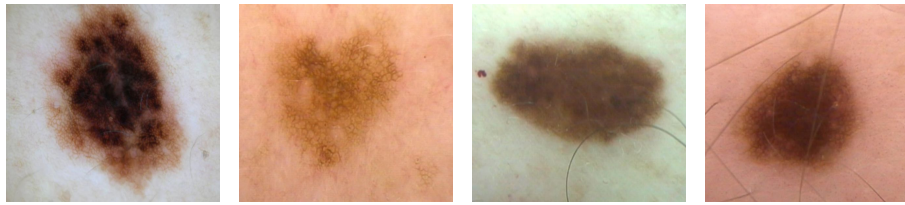
**Asymmetry:** The lesion is bisected by its two principal axes. Symmetry takes into account the contour, colors, and structures within the lesion.

**Border:** The lesion is divided into octants, by its principal axes and two supplementary axes. Next, one counts the number of segments that have an abrupt border cutoff.

**Color:** One point per each color listed in Table 1.

**Dermoscopic structures:** One point per each structure listed in Table 1.

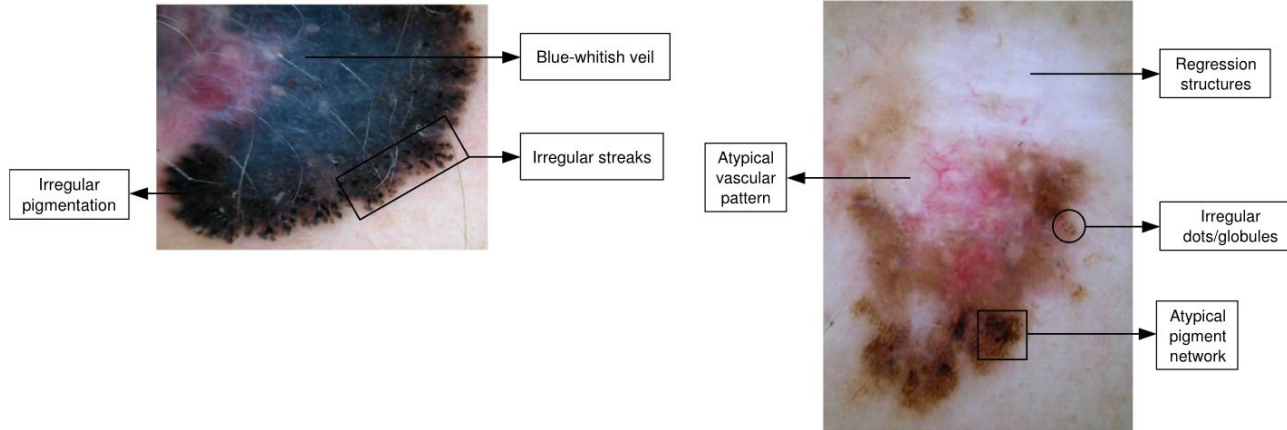
# Lesiones de piel



**Table 2**

7 Points checklist. The total score is computed by weighting each structure by its corresponding score, and summing them up. Lesion classification: benign (total score < 3), malignant (total score  $\geq$  3).

Major criteria	Score	Minor criteria	Score
Atypical pigment network	2	Irregular streaks	1
Blue-whitish veil	2	Irregular pigmentation	1
Atypical vascular pattern	2	Irregular dots/globules	1
		Regression structures	1



# Lesiones de piel

- Esquema:
  - Se toma imagen con una cierta cámara e iluminación
  - Se realizan observaciones sobre la imagen y calculan características numéricas (features)
  - Se clasifica en base a una regla de decisión
- Positivo
  - Reglas basadas en experiencia acumulada
- Negativo
  - Implementación depende del operador
  - Características están fijas, podrían haber otras mejores que no consideramos
  - Umbrales fijos. La regla de decisión no se actualiza con nueva evidencia

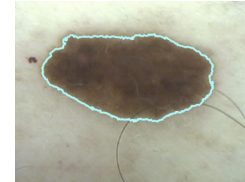
# Lesiones de piel - Aproximación con AA

- Base de datos original:
  - Imágenes + etiquetas (benigno, maligno)
  - Etiquetas = "Ground truth", en este caso son el resultado del estudio histopatológico

# Lesiones de piel - Aproximación con AA

- Base de datos original:
  - Imágenes + etiquetas (benigno, maligno)
  - Etiquetas = "Ground truth", en este caso son el resultado del estudio histopatológico

- Procesamiento de imágenes:
  - Mejoramiento
  - Segmentación de la región





# Lesiones de piel - Aproximación con AA

- Base de datos original:
  - Imágenes + etiquetas (benigno, maligno)
  - Etiquetas = "Ground truth", en este caso son el resultado del estudio histopatológico

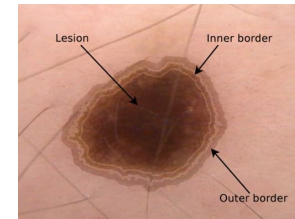
- Procesamiento de imágenes:

- Mejoramiento
- Segmentación de la región



- Extracción de características

- Valores numéricos extraídos de la imagen
  - En la región y en el borde de la lesión
  - Características de color, textura, rugosidad del borde, etc



# Lesiones de piel - Aproximación con AA

- Base de datos original:
  - Imágenes + etiquetas (benigno, maligno)
  - Etiquetas = "Ground truth", en este caso son el resultado del estudio histopatológico

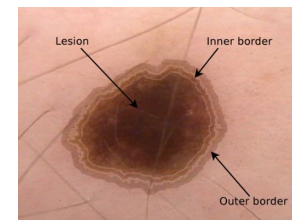
- Procesamiento de imágenes:

- Mejoramiento
- Segmentación de la región



- Extracción de características

- Valores numéricos extraídos de la imagen
  - En la región y en el borde de la lesión
  - Características de color, textura, rugosidad del borde, etc



- Base de trabajo:

var_color	int_color	rugosidad_borde	grado_textura	simetria	etiqueta
0.784308	0.228888	0.732274	0.685757	0.258122	1
0.079096	0.368308	0.456033	0.671064	0.333669	-1
0.581140	0.236488	0.433978	0.916911	0.550707	1
0.668902	0.260335	0.822786	0.625535	0.083581	-1
0.290028	0.929439	0.949531	0.197930	0.511425	1
0.701532	0.288410	0.243439	0.627376	0.265766	1
0.949927	0.300188	0.833955	0.051482	0.949068	1
0.889260	0.135078	0.743078	0.844928	0.129343	1
0.834925	0.774843	0.045272	0.430064	0.509264	-1
0.714861	0.701171	0.909670	0.968240	0.307349	1

# Esquema de AA



Clasificador ideal

y

Resultado  
Maligno / Benigno  
Positivo (+1) / Negativo (-1)

X

var_color	int_color	rugosidad_borde	grado_textura	simetria
0.853362	0.084435	0.078379	0.804044	0.391882

Clasificador real

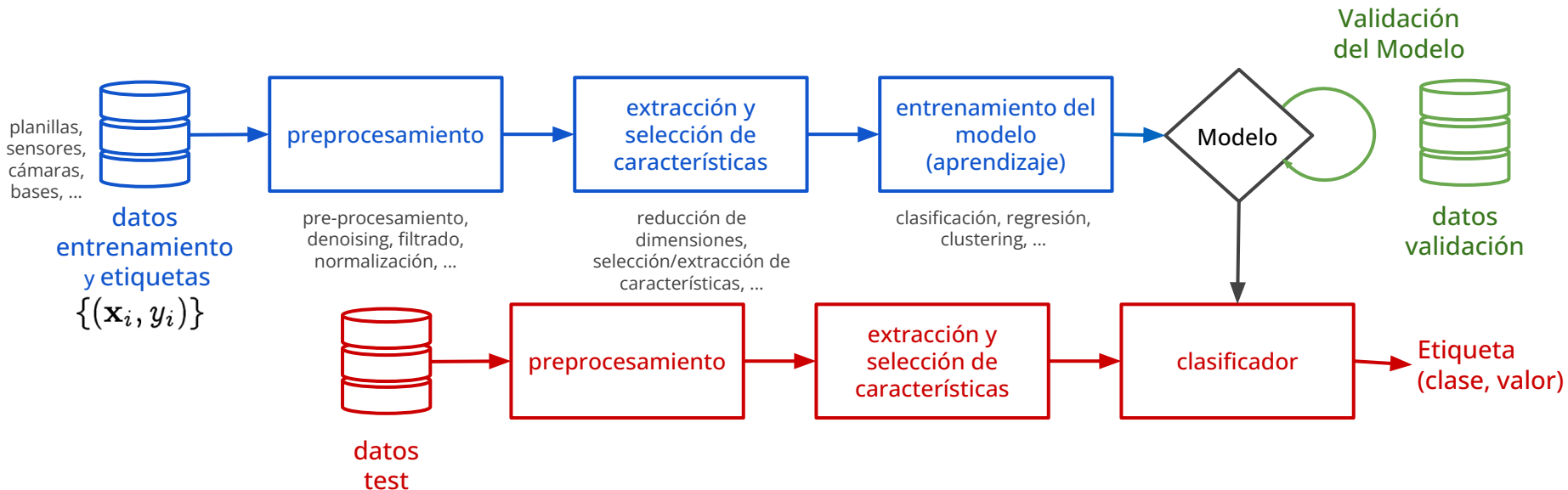
$\hat{y}$

# Esquema de AA

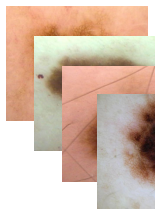
- El set de datos es una muestra del universo del problema
  - es representativo ?
  - es suficiente ?
- Con el set de datos que disponemos quisiéramos:
  - entrenar el clasificador
    - Mostrarle un conjunto de pares (x,y) al clasificador
    - Ajustar parámetros del clasificador para minimizar la diferencia ( $y - \hat{y}$ ) sobre el conjunto
  - elegir un buen modelo de clasificador
  - estimar cuál va a ser el desempeño cuando lo ponga a funcionar con nuevos datos del universo

var_color	int_color	rugosidad_borde	grado_textura	simetria	etiqueta
0.784308	0.228888	0.732274	0.685757	0.258122	1
0.079096	0.368308	0.456033	0.671064	0.333669	-1
0.581140	0.236488	0.433978	0.916911	0.550707	1
0.668902	0.260335	0.822786	0.625535	0.083581	-1
0.290028	0.929439	0.949531	0.197930	0.511425	1
0.701532	0.288410	0.243439	0.627376	0.265766	1
0.949927	0.300188	0.833955	0.051482	0.949068	1
0.889260	0.135078	0.743078	0.844928	0.129343	1
0.834925	0.774843	0.045272	0.430064	0.509264	-1
0.714861	0.701171	0.909670	0.968240	0.307349	1

# Aprendizaje automático



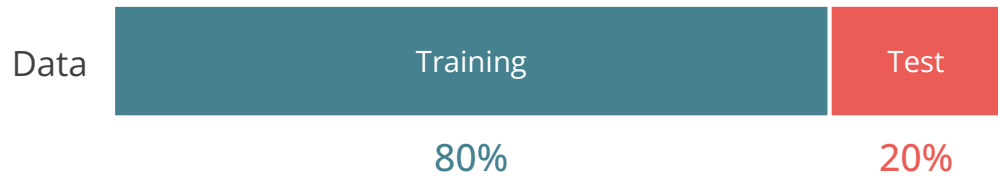
Datos



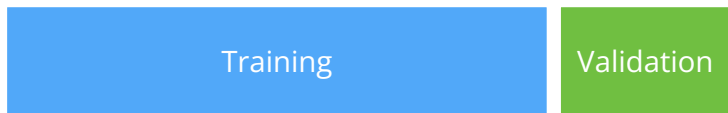
var_color	int_color	rugosidad_borde	grado_textura	simetria	etiqueta
0.784308	0.228888	0.732274	0.685757	0.258122	1
0.079096	0.368308	0.456033	0.671064	0.333669	-1
0.591140	0.236488	0.433978	0.916911	0.550707	1
0.1902	0.260335	0.822786	0.625535	0.083581	-1
0.929439	0.949531	0.197930	0.511425	0.265766	1
0.288410	0.243439	0.627376	0.265766	0.949068	1
0.300188	0.833955	0.051482	0.949068	0.129343	1
0.135078	0.743078	0.844928	0.129343	0.509264	-1
0.834925	0.774843	0.045272	0.430064	0.509264	-1
0.714861	0.701171	0.909670	0.968240	0.307349	1



# Cross-validation: k-fold



k = 4



20%

# Lo esencial de un problema de AA

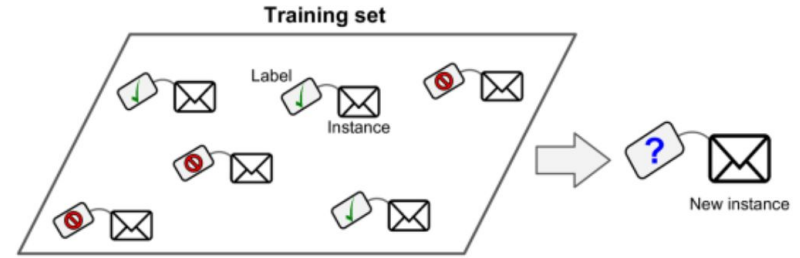
- Debe existir un patrón subyacente
- El problema es complejo y no sabemos o no podemos describir analíticamente las relaciones
- Tenemos datos

# Tipos de aprendizaje

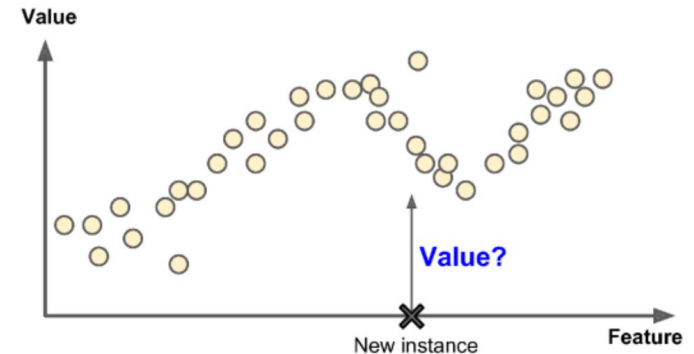
## Aprendizaje supervisado

- Tenemos pares de datos de entrada-salida generados por humanos
- Según el tipo de salida dos posibles problemas:
  - salida categórica: clasificación
  - salida numérica: regresión

## Problema de Clasificación



## Problema de Regresión



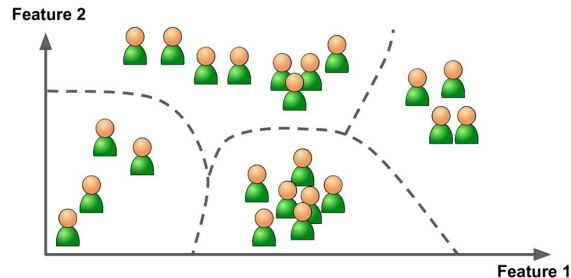
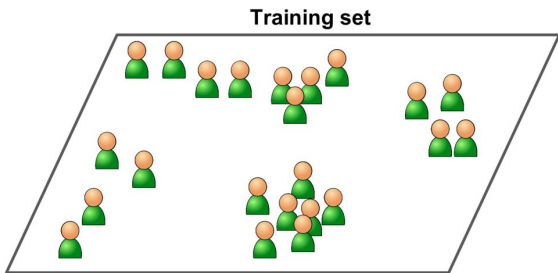


# Tipos de aprendizaje

## Aprendizaje no supervisado

La mayor parte de los datos disponibles no están etiquetados. Su potencial etiquetado requiere intervención humana (muchas veces expertos especializados) y su costo es muy alto.

- **Clustering:** descubrir estructura dentro de un conjunto de datos, agrupándolos en subconjuntos (clusters) que muestren una cierta coherencia o similitud interna.



# Tipos de aprendizaje

## Aprendizaje no supervisado

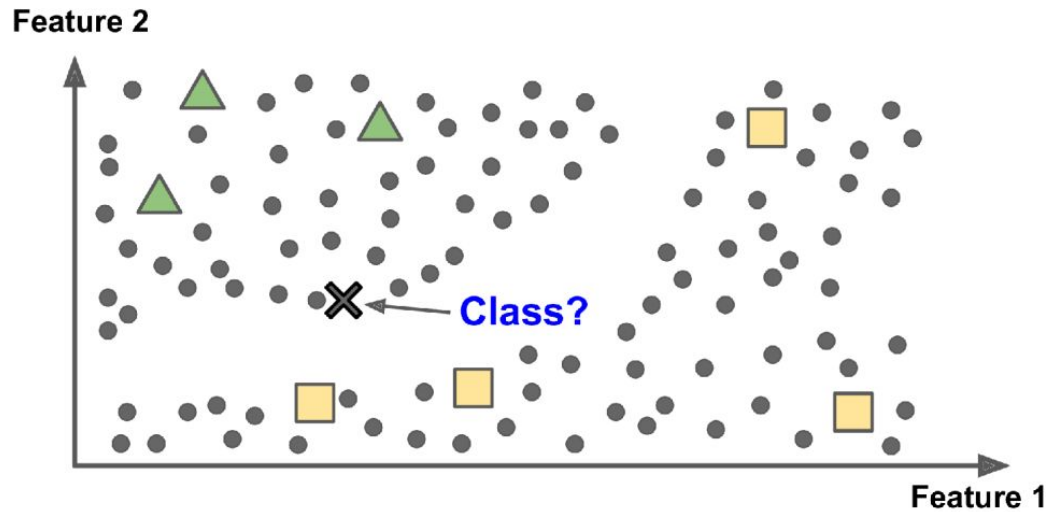
- **Detección de anomalías:** detectar aquellas muestras con baja probabilidad de haber sido generadas por el modelo de normalidad que explica los datos (muestras alejadas de los clusters, o muestras poco probables de acuerdo a las densidades estimadas).



# Tipos de aprendizaje

## Aprendizaje semi - supervisado

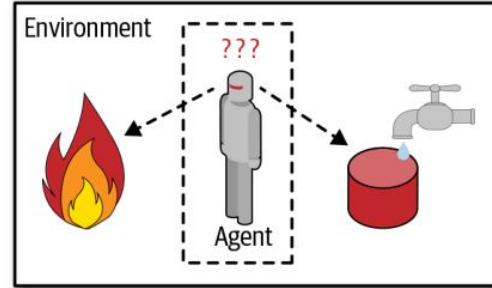
- Tenemos sólo algunos datos etiquetados y muchos sin etiquetar



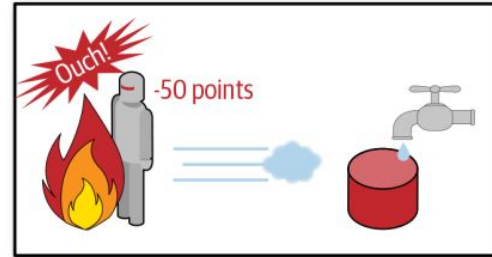
# Tipos de aprendizaje

## Aprendizaje por refuerzo (Reinforcement learning)

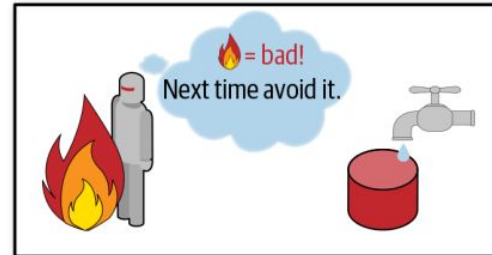
- Un agente actúa en un ambiente y obtiene recompensas ante cada acción
- Las acciones se toman con una cierta política
- El objetivo es obtener una política que maximice la recompensa acumulada



- 1 Observe
- 2 Select action using policy



- 3 Action!
- 4 Get reward or penalty



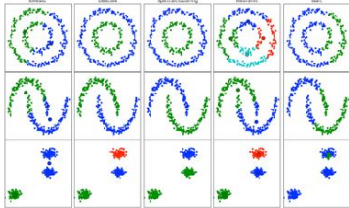
- 5 Update policy (learning step)
- 6 Iterate until an optimal policy is found

# Categorías de los sistemas de AA

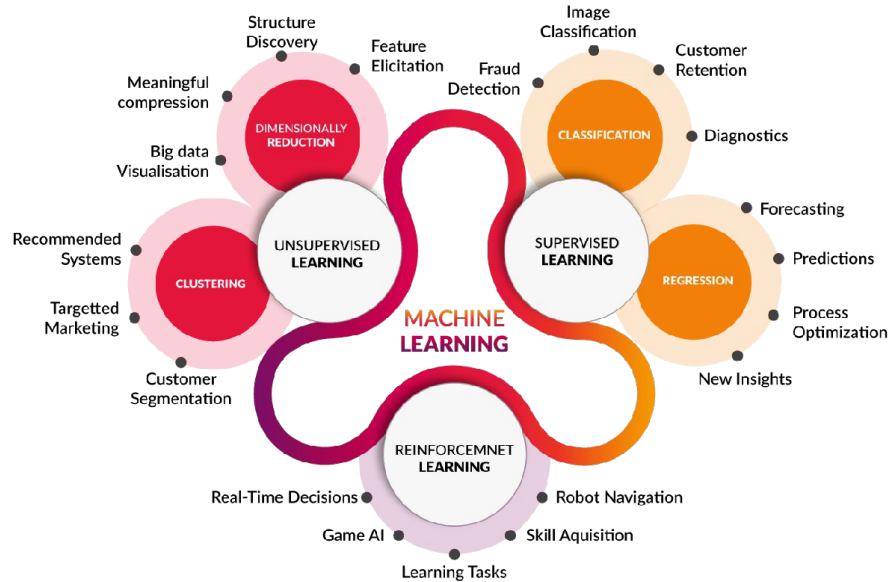
**unsupervised:** the labels (classes) of the data are not known, they are organized on the basis of the features.

- pattern/structure identification
- clustering: clustering
- parametric or not
- # classes unknown

k-means, Fuzzy C-means, Hierarchical Clustering, Spectral Clustering, Gaussian Mixtures, Hidden Markov Model, Neural Networks, PCA, Isomap, MDS, Diffusion Maps, ...



**semi-supervised:** the labels (classes) of some data are known.



**supervised:** training data with labels (classes) or output values to predict new data.

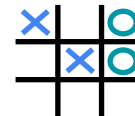
- classification and regression
- explicit learning
- direct assessment
- class/value prediction
- parametric or not

Nearest Neighbor, Support Vector Machines (SVM), Decision Trees, Random Forest, Discriminant Analysis, Naive Bayes, Neural Networks, Linear Regression, SVR, ...



**reinforcement learning:** feedback of the task result: reward or penalty

- approach to AI
- define strategies for events
- maximize reward



# Clasificación por vecinos más cercanos (k-NN)



FACULTAD DE  
INGENIERÍA



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY

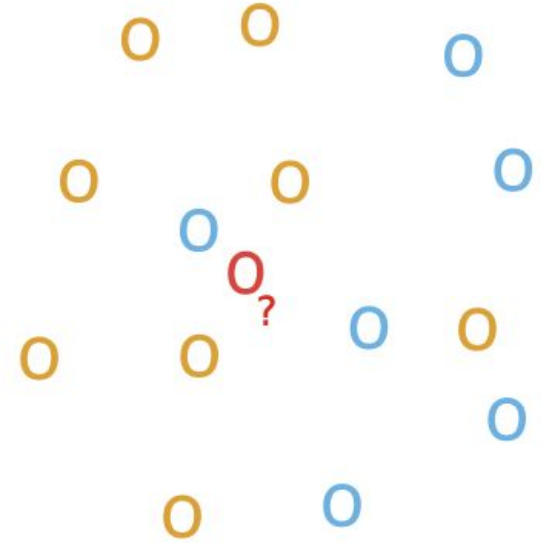
# Clasificación por vecinos más cercanos (k-NN)

- **Basado en instancias**: no hay una fase de entrenamiento explícita: memoriza los datos de entrenamiento como conocimiento para la fase de predicción.
- Idea principal: puntos de **datos similares** tienden a tener **etiquetas similares**

# Clasificación por vecinos más cercanos (k-NN)

Se asigna la **clase dominante** dentro de los **k** puntos más cercanos del conjunto de entrenamiento.

Dado un nuevo punto **x** (conjunto de test):



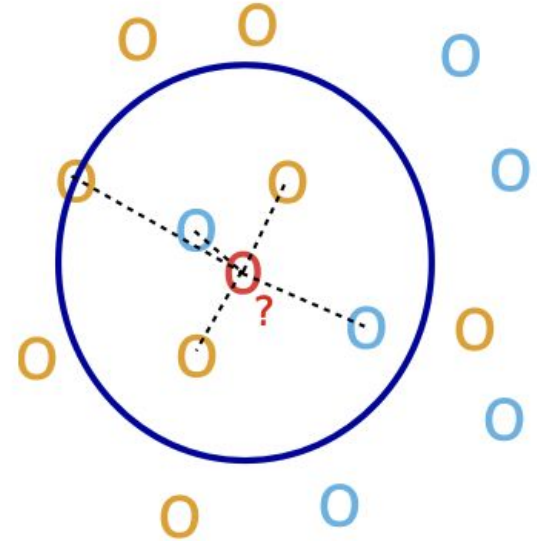


# Clasificación por vecinos más cercanos (k-NN)

Se asigna la **clase dominante** dentro de los **k** puntos más cercanos del conjunto de entrenamiento.

Dado un nuevo punto **x** (conjunto de test):

1. Se buscan los **k** vecinos más cercanos en el conjunto de entrenamiento según cierta **distancia**

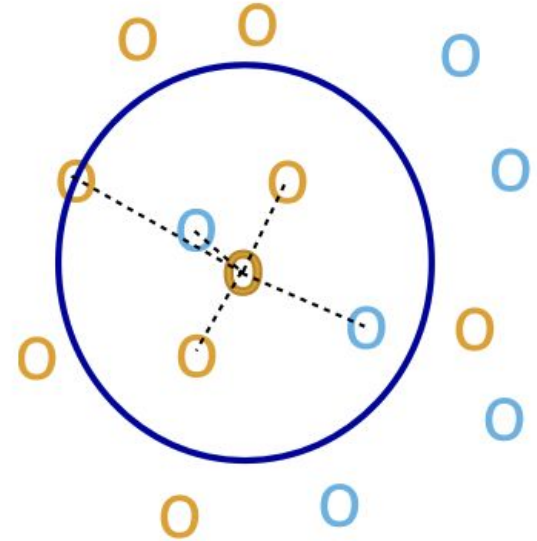


# Clasificación por vecinos más cercanos (k-NN)

Se asigna la **clase dominante** dentro de los **k** puntos más cercanos del conjunto de entrenamiento.

Dado un nuevo punto **x** (conjunto de test):

1. Se buscan los **k** vecinos más cercanos en el conjunto de entrenamiento según cierta **distancia**
2. Se clasifica a **x** con la etiqueta dominante entre los **k** vecinos

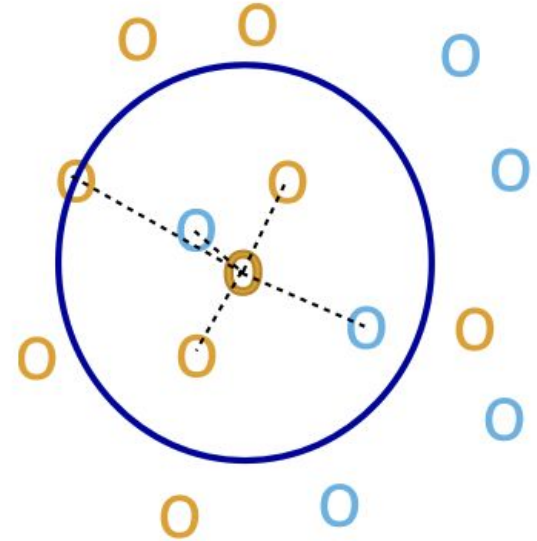


# Clasificación por vecinos más cercanos (k-NN)

Se asigna la **clase dominante** dentro de los **k** puntos más cercanos del conjunto de entrenamiento.

Dado un nuevo punto **x** (conjunto de test):

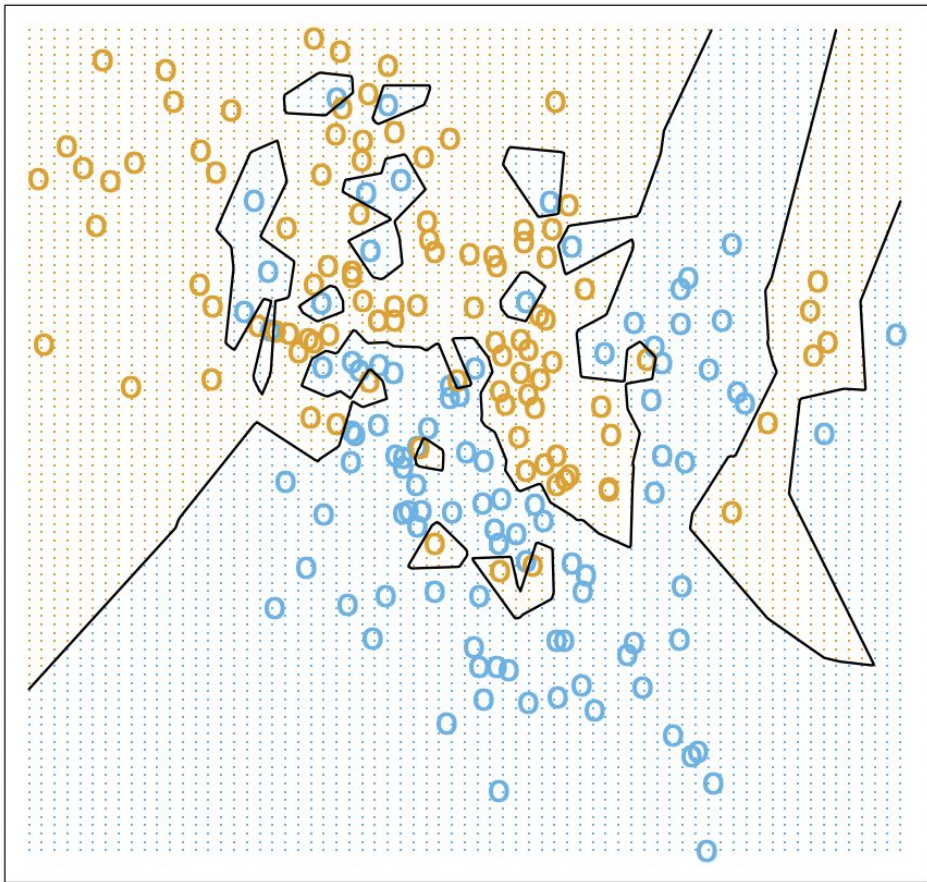
1. Se buscan los **k** vecinos más cercanos en el conjunto de entrenamiento según cierta **distancia**
2. Se clasifica a **x** con la etiqueta dominante entre los **k** vecinos



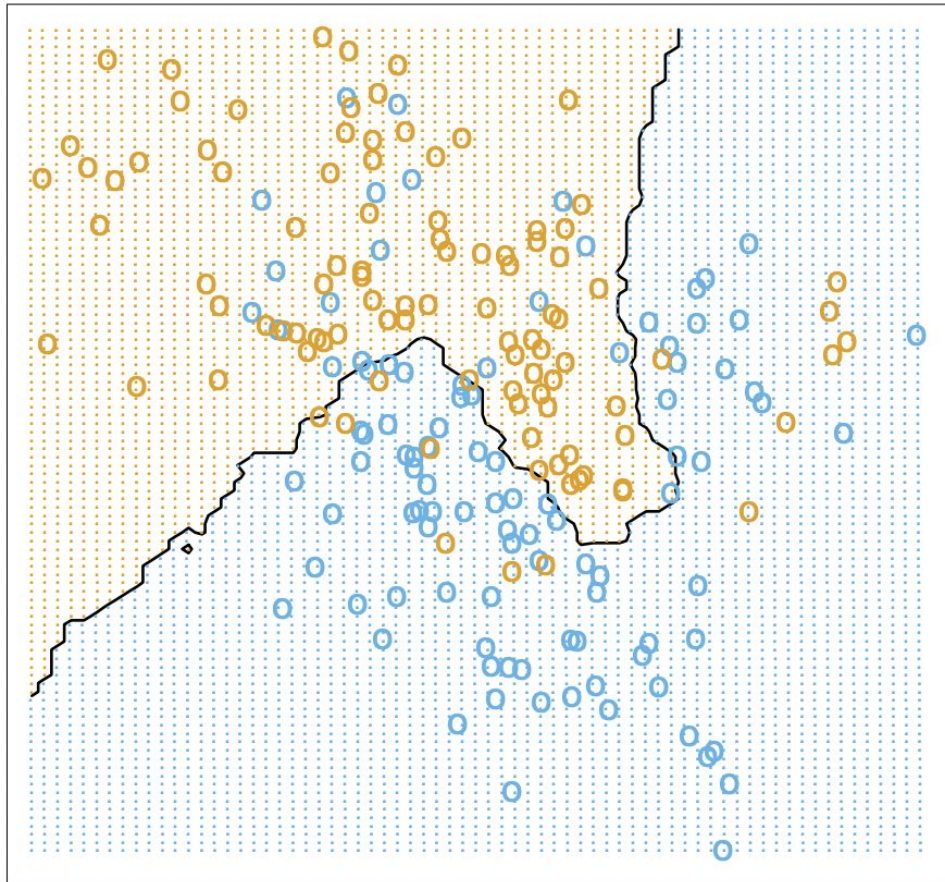
**k** y la **distancia** son hiperparámetros del método.

# Hiperparámetro k (cantidad de vecinos)

1-NN



15-NN



# Hiperparámetro k (cantidad de vecinos)

- k muy chico ( $k \rightarrow 1$ ):
  - sobre ajuste: se ajusta muy bien a los datos de entrenamiento y no es bueno generalizando
  - las fronteras de decisión son irregulares
- k grande:
  - tiende a un mejor comportamiento
  - si k es muy grande se empiezan a considerar puntos que no son vecinos
  - subajuste
  - las fronteras de decisión son más homogéneas

# sklearn.neighbors.KNeighborsClassifier



Install User Guide API Examples Community More ▾

Prev Up Next

scikit-learn 1.4.1

[Other versions](#)

Please [cite us](#) if you use the software.

[sklearn.neighbors.KNeighborsClassifier](#)

[KNeighborsClassifier](#)

[Examples using](#)

[sklearn.neighbors.KNeighborsClassifier](#)

## sklearn.neighbors.KNeighborsClassifier

```
class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None)
```

[\[source\]](#)

Classifier implementing the k-nearest neighbors vote.

Read more in the [User Guide](#).

### Parameters:

**n\_neighbors** : *int*, **default=5**

Number of neighbors to use by default for [kneighbors](#) queries.

**weights** : *{'uniform', 'distance'}*, *callable or None*, **default='uniform'**

Weight function used in prediction. Possible values:

- 'uniform' : uniform weights. All points in each neighborhood are weighted equally.
- 'distance' : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

Refer to the example entitled [Nearest Neighbors Classification](#) showing the impact of the `weights` parameter on the decision boundary.

# Clasificación por árboles de decisión



FACULTAD DE  
INGENIERÍA



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY

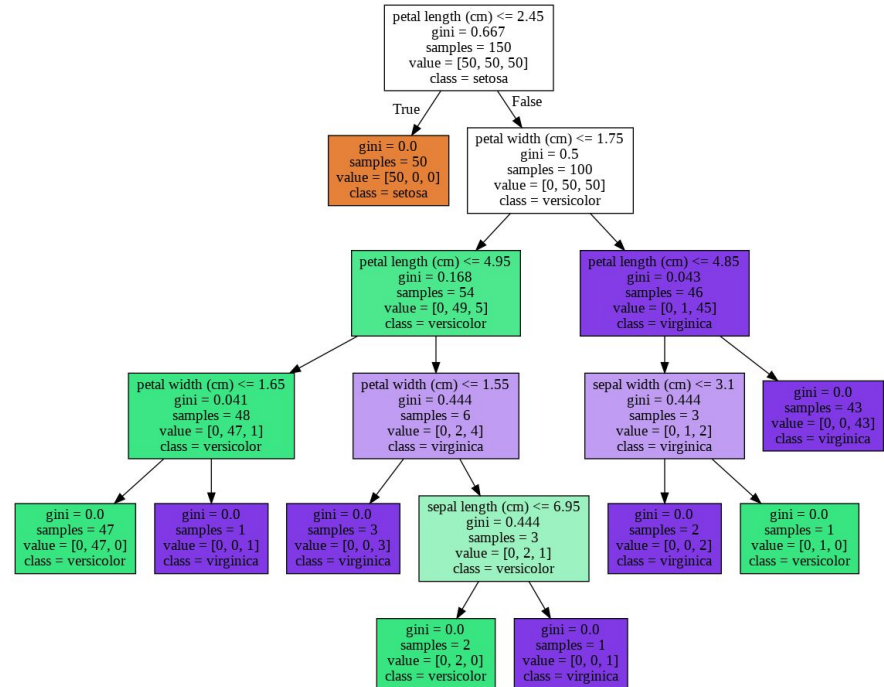
# Árboles de decisión

- Método de **aprendizaje supervisado** utilizado para clasificación y regresión
- Idea principal: predecir el valor de una variable objetivo mediante el **aprendizaje de reglas de decisión sencillas** inferidas a partir de las características de los datos de entrenamiento.



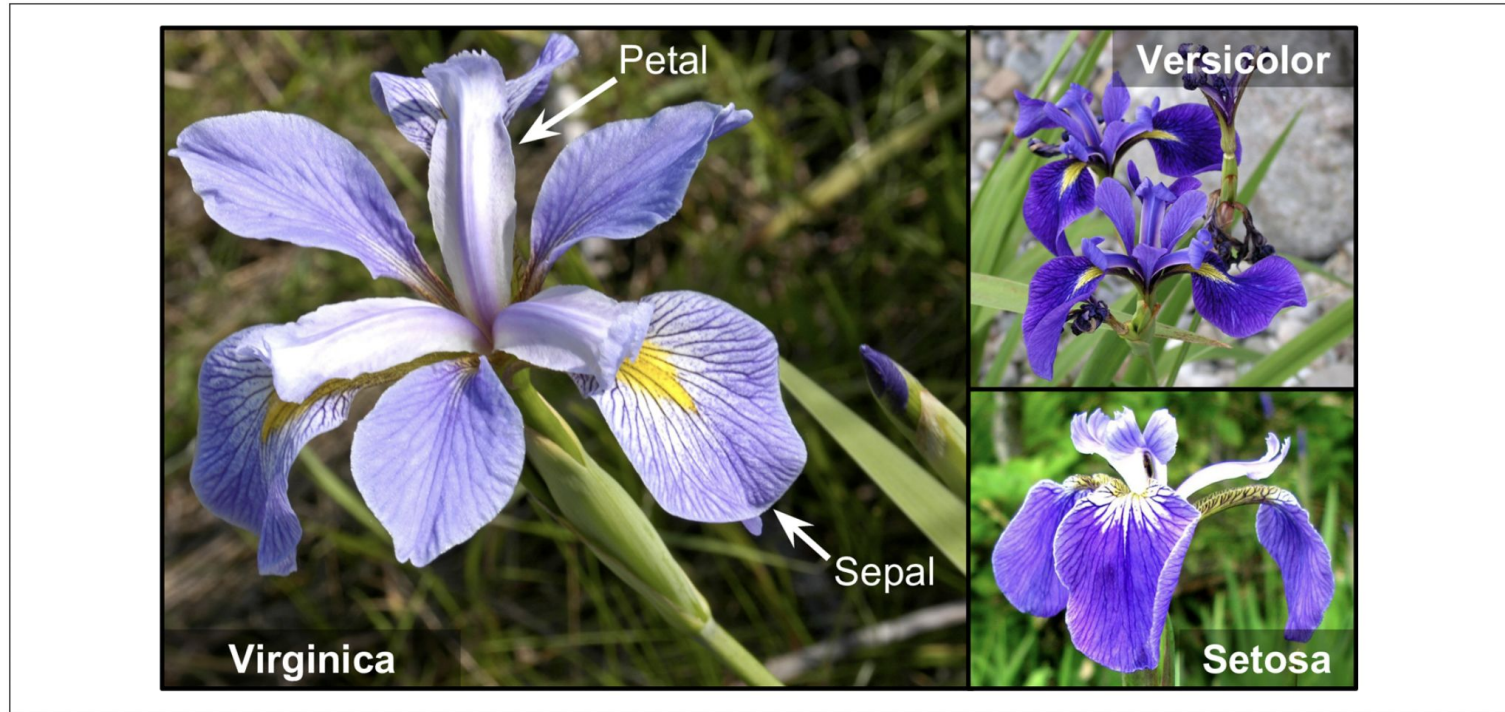
# Árboles de decisión

- Clase/valor se decide por secuencia de preguntas
- Muy intuitivos: fácil de entender e interpretar
- Los árboles se pueden visualizar
- No se precisa tratar los datos previo a la clasificación/regresión
- **Muy efectivos!**



# Dataset Iris

Conjunto de datos de sklearn que contiene la longitud y ancho de los sépalos y pétalos de 150 flores de iris de tres especies diferentes: Iris setosa, Iris versicolor e Iris virginica



# Ejemplo de clasificación para el dataset iris

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris(as_frame=True)
X_iris = iris.data[["petal length (cm)", "petal width (cm)"]].values
y_iris = iris.target

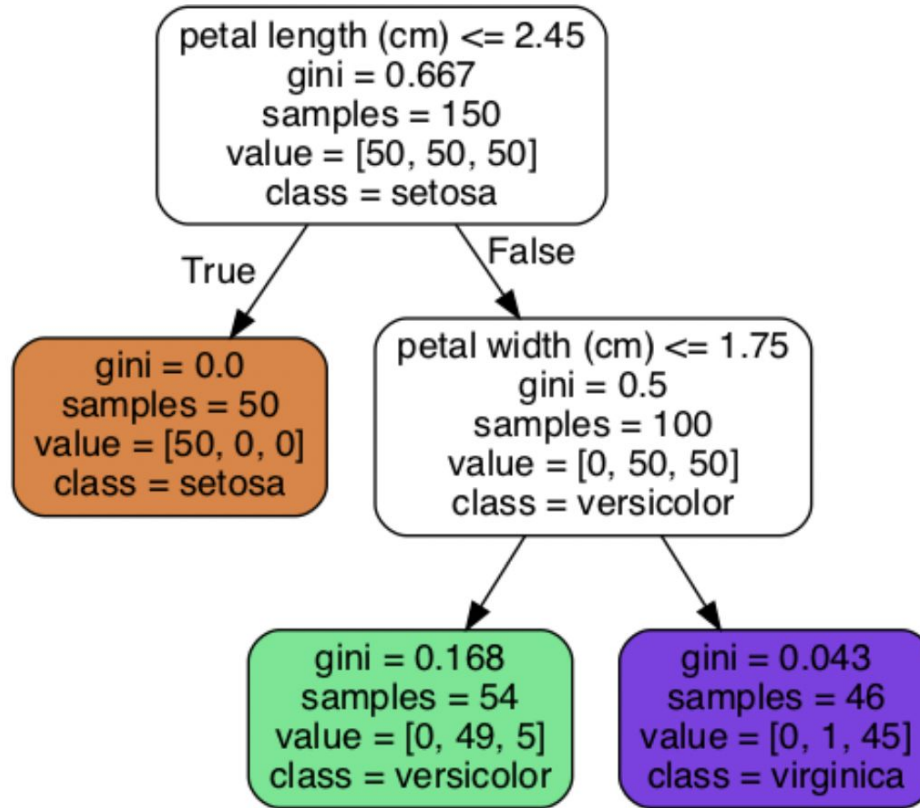
tree_clf = DecisionTreeClassifier(max_depth=2, random_state=42)
tree_clf.fit(X_iris, y_iris)
```

```
from sklearn.tree import export_graphviz

export_graphviz(
    tree_clf,
    out_file=str(IMAGE_PATH / "iris_tree.dot"), # path differs in the book
    feature_names=["petal length (cm)", "petal width (cm)"],
    class_names=iris.target_names,
    rounded=True,
    filled=True
)
```

```
from graphviz import Source

Source.from_file(IMAGE_PATH / "iris_tree.dot") # path differs in the book
```



Árbol de decisión para la clasificación

# Métricas de desempeño



FACULTAD DE  
INGENIERÍA



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY

# Performance measurements

42 17 1 20 4

Total samples:  $N = TP + FP + TN + FN$

Accuracy:  $ACC = \frac{(TP + TN)}{N}$

Sensitivity/Recall:  $TPR = \frac{TP}{(TP + FN)}$

Specificity:  $TNR = \frac{TN}{(TN + FP)}$

Precision:  $PPV = \frac{TP}{(TP + FP)}$

False positive rate:  $FPR = \frac{FP}{(FP + TN)}$

False discovery rate:  $FDR = \frac{FP}{(FP + TP)}$

F-score:  $F = \frac{2TP}{(2TP + FP + FN)}$

¿cuánto se acerca a los valores reales? exactitud

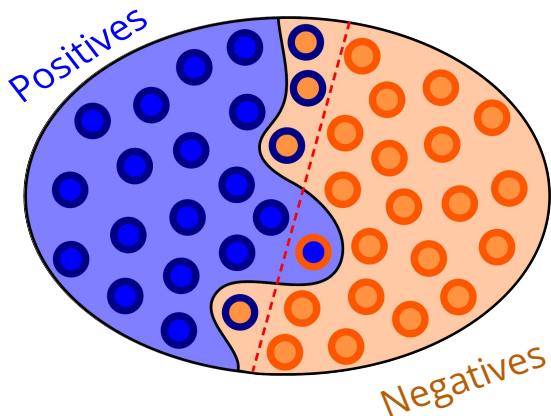
tasa de enfermos correctamente detectados


tasa de sanos correctamente clasificados


de los clasificados positivos ¿cuántos son los realmente positivos?


probabilidad de una falsa alarma.


Media armónica entre TPR y PPV.



 TP: true positives  
success

 TN: true negatives  
correct rejection

 FP: false positives  
type I error (false alarm)

 FN: false negatives  
type II error

system login

disease detection

## Confusion matrix

		Positive (sick)	Negative (healthy)		
		TP	FN		
Real labels	Positive (sick)	TP	FN		
	Negative (healthy)	FP	TN		
		Predicted labels			



FACULTAD DE  
INGENIERÍA



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY