

**APRENDIZAJE
AUTOMÁTICO**

BRACE YOURSELF

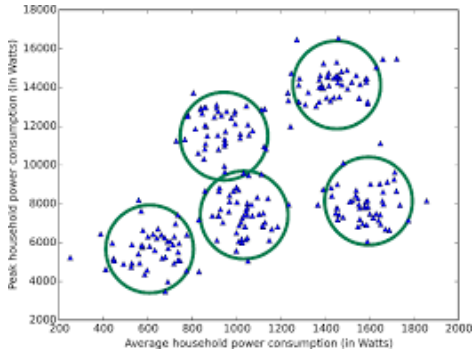
**MACHINE LEARNING IS
COMING**



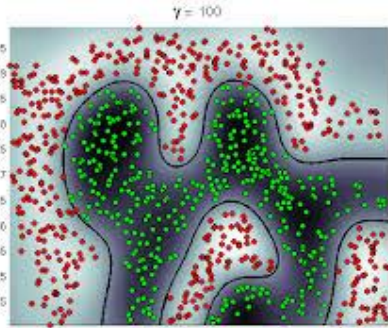
AGENDA



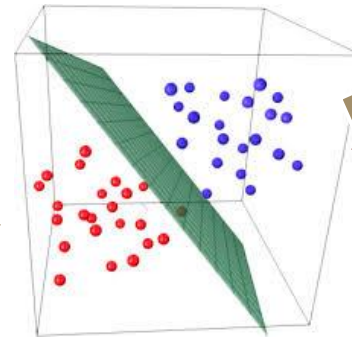
**Aprendizaje
automático**



**Aprendizaje
no supervisado**



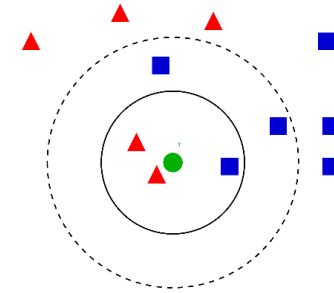
**Aprendizaje
supervisado**



Clasificación

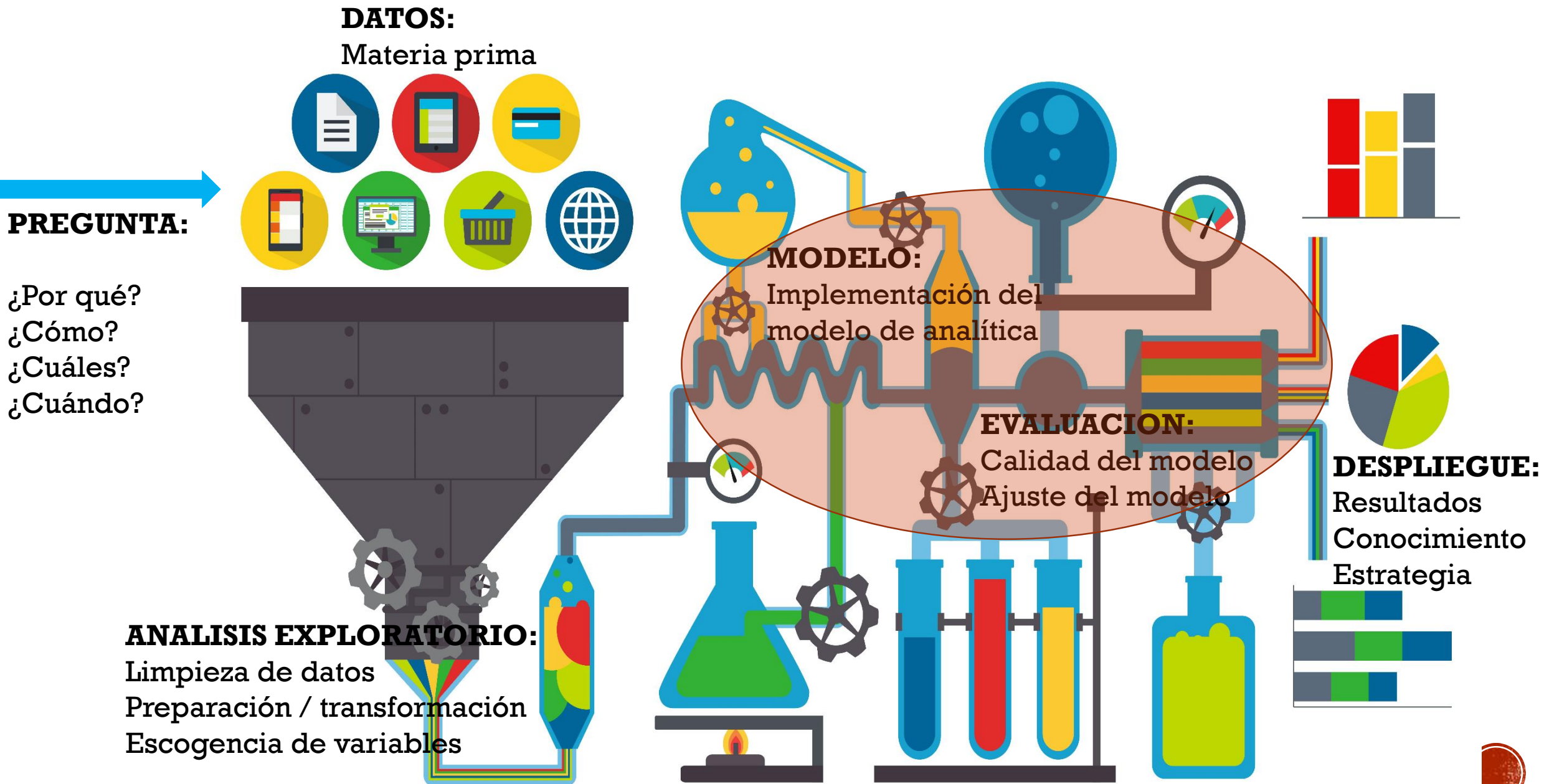


**Métricas de
Evaluación de la
clasificación**



KNN





Machine Learning



what society thinks I
do



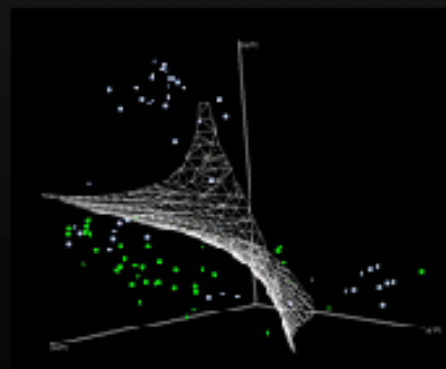
what my friends think
I do



what my parents think
I do

$$\begin{aligned}
 L_T &= \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^n \alpha_i \\
 \alpha_i &\geq 0, \forall i \\
 \mathbf{w} &= \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i, \sum_{i=1}^n \alpha_i y_i = 0 \\
 \nabla g(\theta_t) &= \frac{1}{n} \sum_{i=1}^n \nabla \ell(x_i, y_i; \theta_t) + \nabla r(\theta_t) \\
 \theta_{t+1} &= \theta_t - \eta_t \nabla \ell(x_{t+1}, y_{t+1}; \theta_t) - \eta_t \cdot \nabla r(\theta_t) \\
 \mathbb{E}_{t+1}[\ell(x_{t+1}, y_{t+1}; \theta_t)] &= \frac{1}{n} \sum_{i=1}^n \ell(x_i, y_i; \theta_t)
 \end{aligned}$$

what other programmers
think I do



what I think I do

```
>>> from scipy import svm
```

what I really do



APRENDIZAJE AUTOMÁTICO

- **¿Por qué es necesario?**
 - Tareas complejas extremadamente difíciles de programar
 - Poder computacional disponible para tratar grandes volúmenes de datos

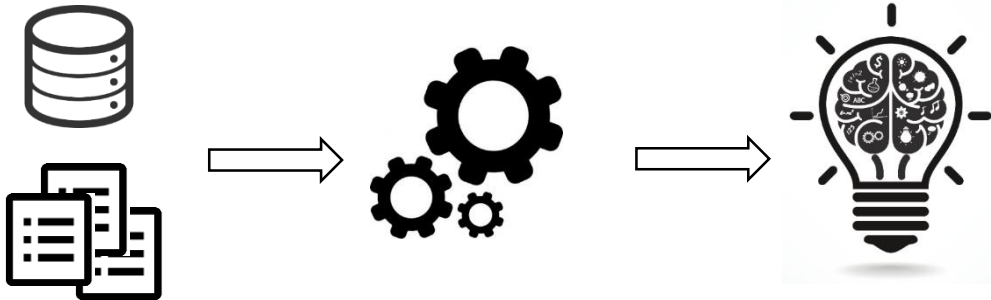
Las máquinas tienen que aprender por sí solas



APRENDIZAJE AUTOMÁTICO

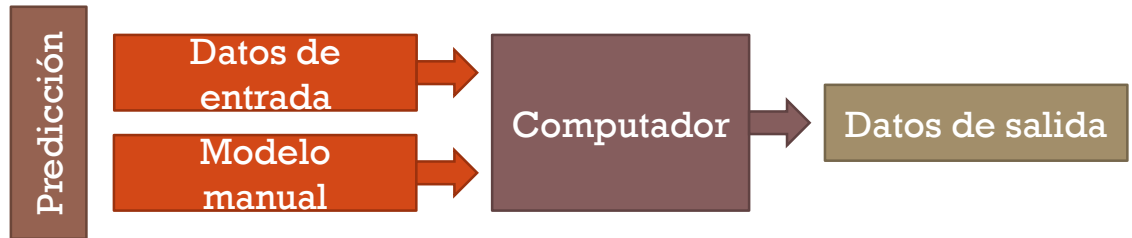
- Definición:

El aprendizaje automático es la ciencia que permite a los computadores aprender, sin ser explícitamente programados¹

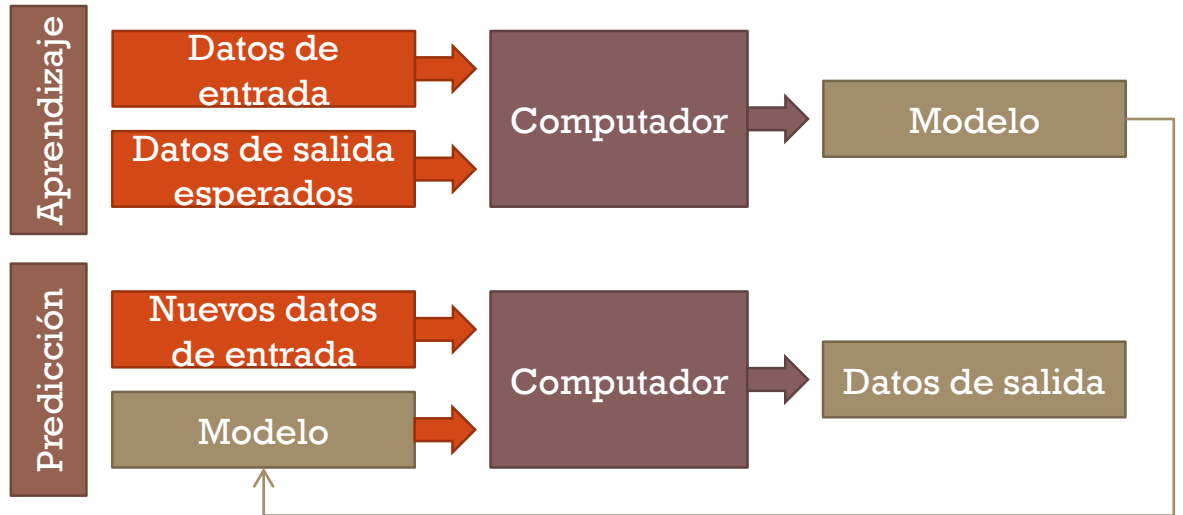


1. Andrew Ng, Stanford University, 2014

Modelo tradicional



Ciencia de datos



APRENDIZAJE AUTOMÁTICO


Aprendizaje supervisado

- Aprender a partir de un “experto”
- Datos de entrenamiento **etiquetados** con una clase o valor:

$(x_1, x_2, \dots, x_n, y)$

Predictores, explicativos,
independientes

Dependiente, objetivo,
salida



- **Meta:** predecir una clase o valor

Aprendizaje no supervisado

- Sin conocimiento de una clase o valor objetivo
- Datos **no** están **etiquetados**

(x_1, x_2, \dots, x_n)

- **Meta:** descubrir factores no observados, estructura, o una representación mas simple de los datos



APRENDIZAJE AUTOMÁTICO

Aprendizaje supervisado

Edad	Ingresos	Tiene carro?
24	1'200.000	NO
23	4'500.000	SI
45	1'250.000	SI
32	1'100.000	NO

Datos etiquetados:
"Respuestas correctas" disponibles

Factores/atributos/variables independientes, predictores, explicativos

Dependiente, objetivo, respuesta, salida

34	3'500.000
----	-----------

¿Cuál es el valor predicho para una instancia dada?

Aprendizaje no supervisado

Edad	Ingresos
24	1'200.000
23	4'500.000
45	1'250.000
32	1'100.000

Factores/atributos/variables

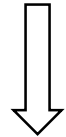
¿Se puede encontrar alguna estructura en los datos?



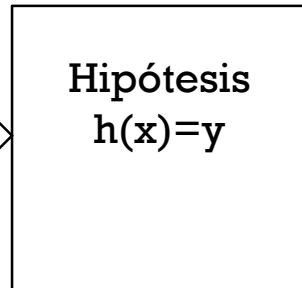
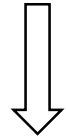
APRENDIZAJE AUTOMÁTICO

Aprendizaje supervisado

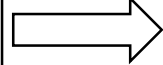
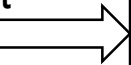
Set de entrenamiento(x_1, x_2, \dots, x_n, y)



Algoritmo de aprendizaje,
estimación de parámetros



Set de text de test
(x_1', x_2', \dots, x_n')



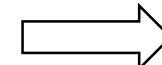
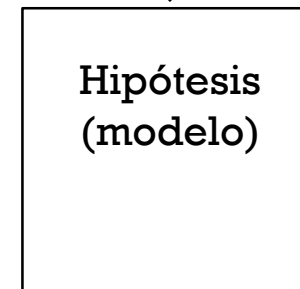
Resultado
(y')

Aprendizaje no supervisado

Set de entrenamiento(x_1, x_2, \dots, x_n)



Algoritmo de aprendizaje,
estimación de parámetros



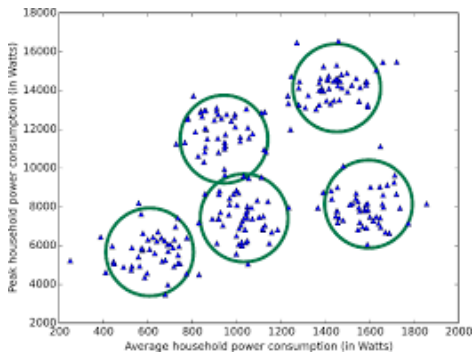
Resultado
(estructura)



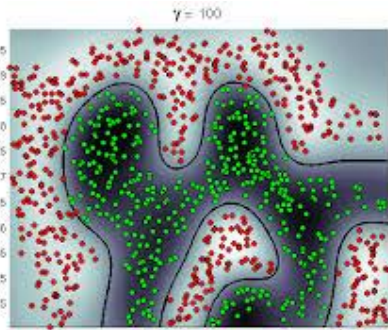
AGENDA



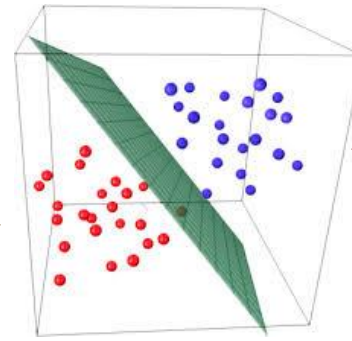
**Aprendizaje
automático**



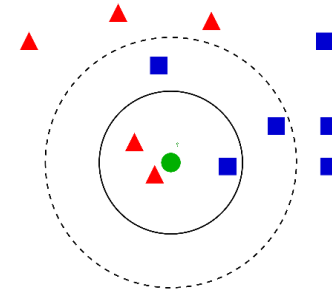
**Aprendizaje
no supervisado**



**Aprendizaje
supervisado**



Clasificación



KNN



CLASIFICACIÓN —

K-NN

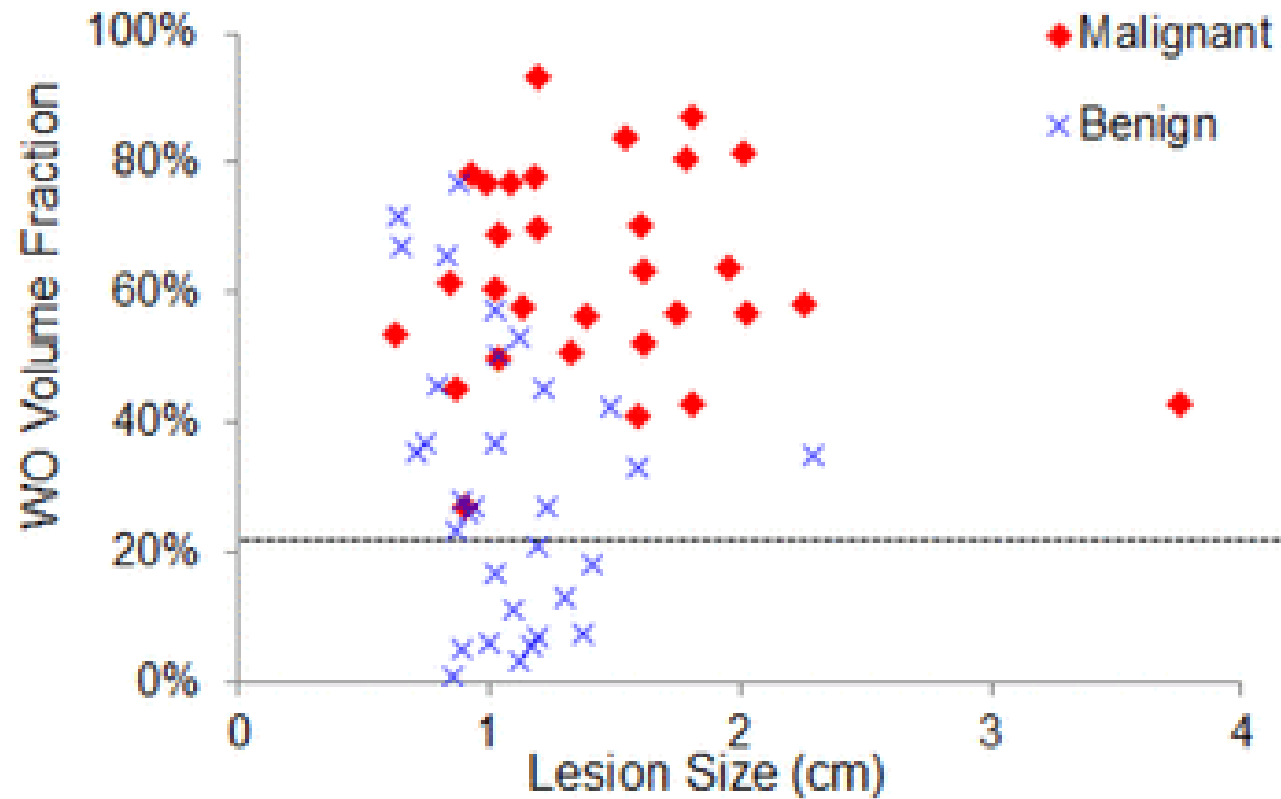
(K-NEAREST NEIGHBORS)



Neighbors

can be a pain in the ass

CLASIFICACIÓN

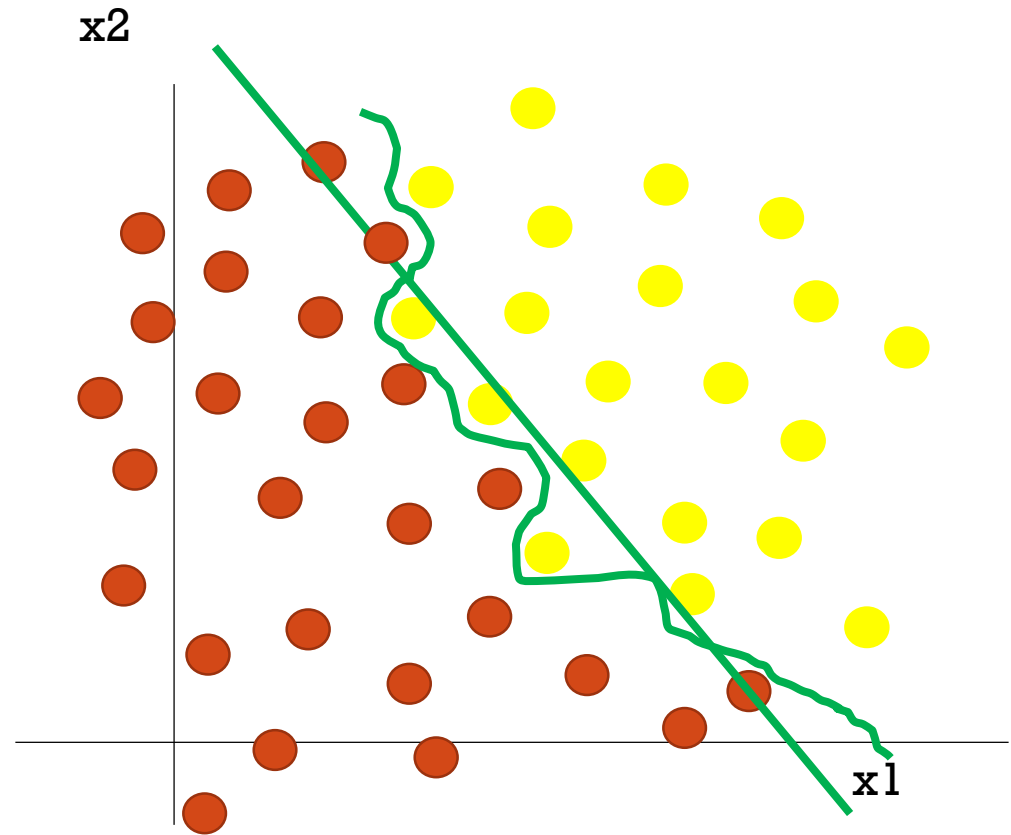


http://www.jacmp.org/index.php/jacmp/article/view/5187/html_374



CLASIFICACIÓN

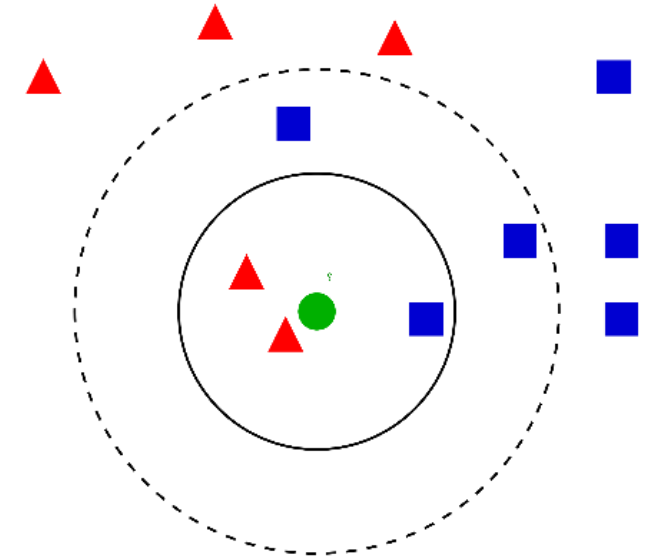
- Encontrar modelos que describan clases para futuras predicciones:
 - KNN
 - Árboles de decisión
 - Regresión logística
 - Redes neuronales
 - ...
- Valores **discretos** de la variable objetivo
- Incluye la estimación de **probabilidades** de clase
- **Baseline**: medida de evaluación dada por un clasificador que escoge siempre la clase mayoritaria



KNN

KNN (K Nearest Neighbors): K Vecinos más Cercanos

- Algoritmo de aprendizaje supervisado para **clasificación** y **regresión**
- **Sencillo**: asignar la clase o valor agregado de las instancias conocidas que se encuentran mas cerca de la instancia a predecir
- Basado en las **instancias** de aprendizaje, no en un modelo subyacente probabilístico/estadístico
- Aprendizaje **perezoso**: en realidad el algoritmo solo se ejecuta en el momento que se requiere predecir una nueva instancia a partir de una predicción local

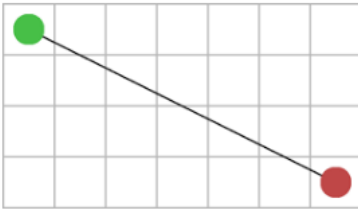


Wikipedia, 2016



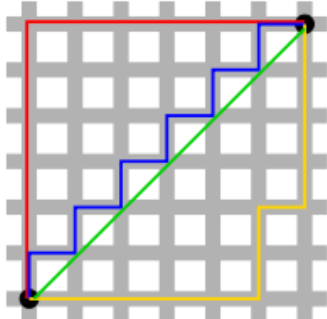
KNN – DISTANCIAS

- Basado en una medida de **similitud** o **distancia** que hay que definir para encontrar los vecinos mas cercanos:
 - **Euclidiana**: tamaño del segmento linear que une las dos instancias comparadas.

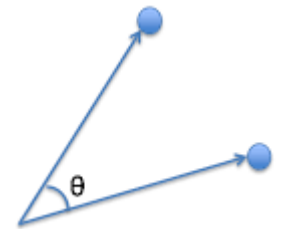


$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2}$$
$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.$$

- **Manhattan**: basada en una organización en bloques rectilíneos



- **Coseno**: coseno del ángulo entre las dos instancias comparadas → Alta dimensionalidad y **big data**

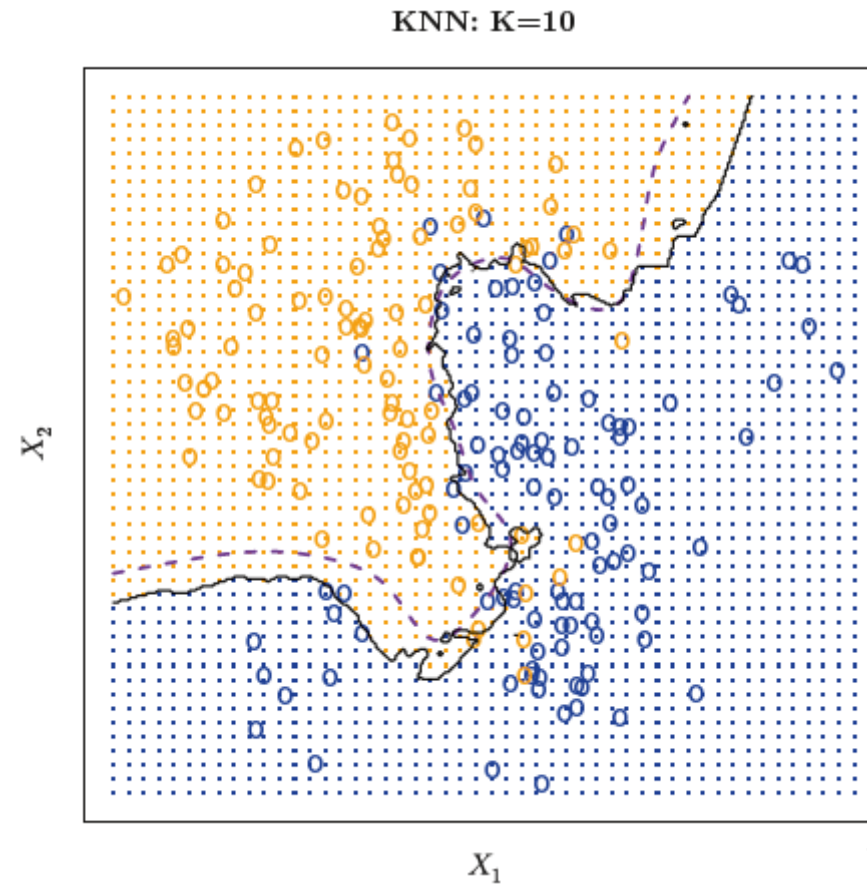
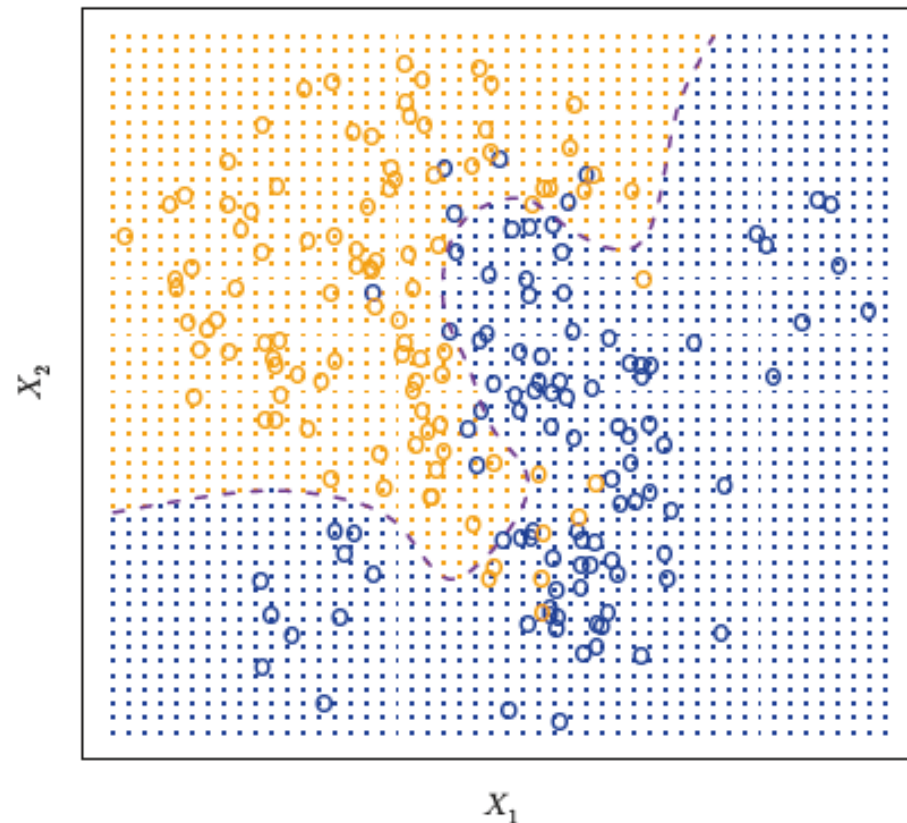


$$\text{sim}(\mathbf{x}, \mathbf{y}) = \cos(\theta_{\mathbf{x}, \mathbf{y}}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_i x_i * y_i}{\sqrt{(\sum_i x_i * x_i) * \sum_i y_i * y_i}}$$



KNN – K

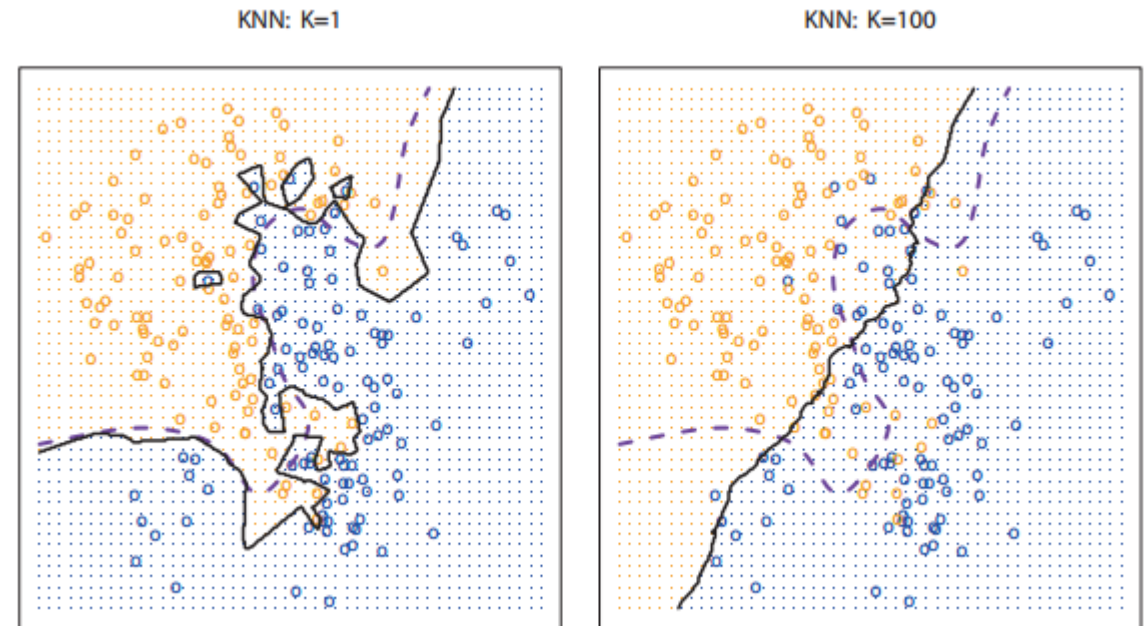
- **Parámetro K:** número de vecinos mas cercanos a considerar para establecer la clase o valor de una nueva instancia



KNN – K

■ Parámetro K

- El resultado puede ser drásticamente diferente para diferentes valores de K
- Un valor de K grande suavizará los límites entre clases/valores (alto sesgo, baja varianza)
- Un valor de K pequeño resultará en límites muy flexibles (bajo sesgo, alta varianza)
- El valor de K óptimo se encuentra empíricamente

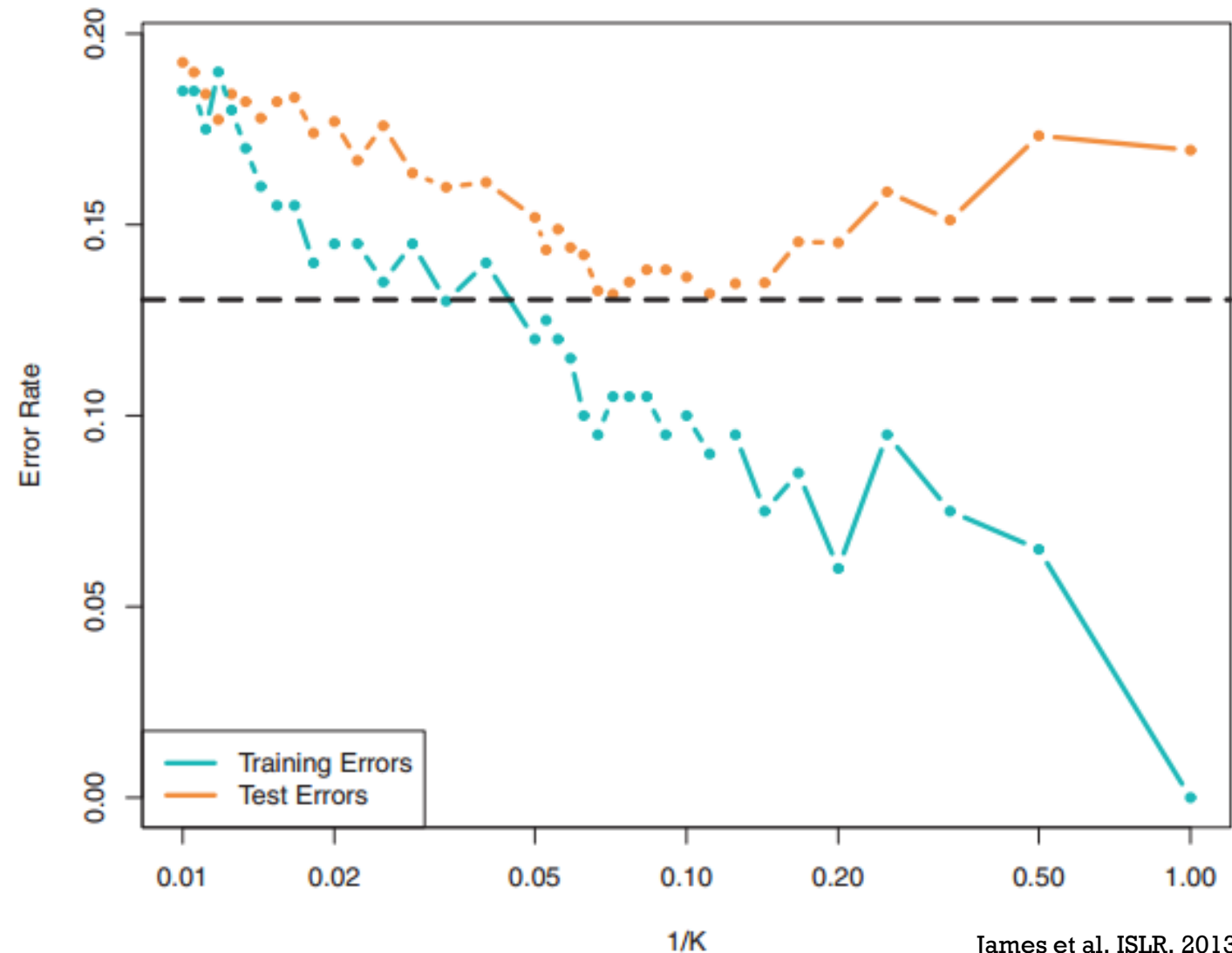


James et al, ISLR, 2013



KNN – K

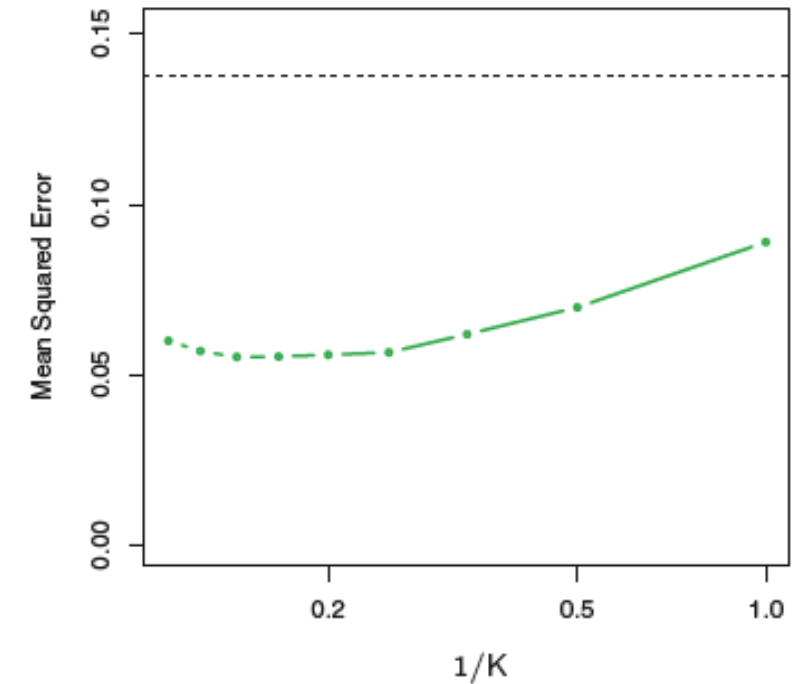
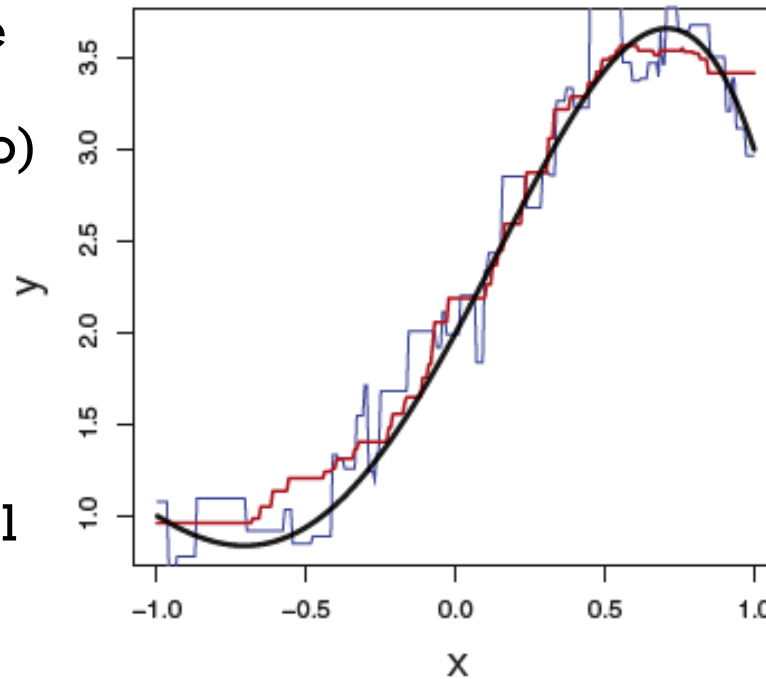
- **Overfitting:** (sobre aprendizaje) a considerar en el momento de escoger el K.
- Modelos mas sencillos previenen el overfitting → K mas grandes
- Igualmente, cuidado con el **underfitting** (sub aprendizaje)



KNN – K

En el caso de la utilización de KNN para la regresión las mismas consideraciones aplican

- En el panel izquierdo: se aplica KNN con un valor de $K=1$ (azul) y $K=9$ (rojo)
- En el panel derecho, se puede ver el valor de RMSE para diferentes valores de K (en verde). También se puede ver, por comparación el nivel de error de la regresión lineal simple (punteada en negro)



James et al, ISLR, 2013



KNN

Consideraciones:

- Perezoso (Lazy learning)
- No paramétrica y no lineal
- **Método local, no generalizable (no hay un modelo construido como tal):**
 - Puede encontrar particularidades muy específicas a ciertas regiones
 - Su uso (sobre todo en regresión) sólo permite estimaciones en los rangos de las variables del set de aprendizaje (extrapolación no tiene mucho sentido)
- Maldición de la **dimensionalidad**
- Muy sensible a la **unidad de medida** de los atributos, y a atributos que no aportan poder predictivo (e.g. el color de los ojos no debería considerarse para predecir la edad de una persona)
- No sabe que hacer con los **missing values**, ni con variables **categoricas**
- **Variaciones:** K-nn ponderado por la distancia, basado en un radio dado.



CNN (CONDENSED NEAREST NEIGHBORS)

- Dificultad de aplicación de KNN cuando se tienen **muchos registros**
- No todos los registros son necesarios para la correcta clasificación
- Aproximación de KNN utilizando un conjunto de datos reducido
- Escogencia de **prototipos** que permitan una clasificación con $K=1$ lo más parecida al resultado utilizando el dataset completo
- Algoritmo: Siendo **\mathbf{X}** el conjunto de datos inicial y **\mathbf{U}** el conjunto reducido:
 - Identificar todos los elementos x de **\mathbf{X}** cuyo vecino más cercano sea de clase diferente
 - Retirar los x identificados (son prototipos) de **\mathbf{X}** y agregarlos a **\mathbf{U}**
 - Repetir hasta que no se agreguen más prototipos a **\mathbf{U}**



TALLER DE CLASIFICACIÓN CON KNN

- DATASET: 150 ejemplos pertenecientes a 3 especies diferentes de la flor Iris
- 4 Atributos: largo y ancho del sépalo, largo y ancho del pétalo
- Reproducir el taller hasta antes de la sección de métricas y matriz de confusión



Iris setosa



Iris versicolor



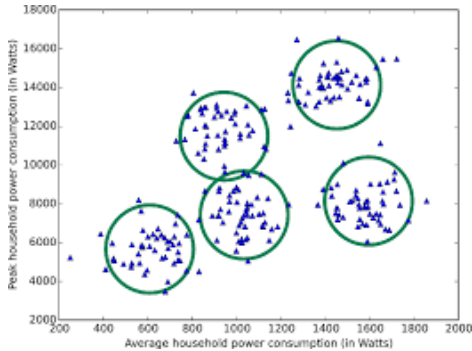
Iris virginica



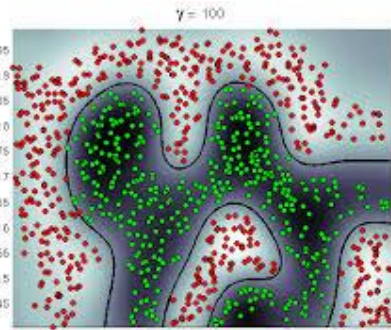
AGENDA



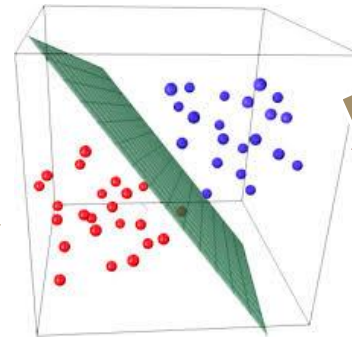
Aprendizaje automático



Aprendizaje no supervisado



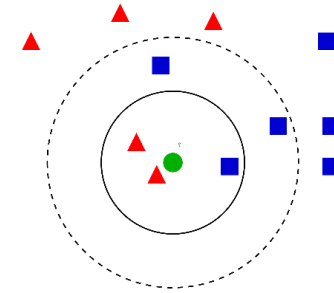
Aprendizaje supervisado



Clasificación



Métricas de Evaluación de la clasificación



KNN



MÉTRICAS DE EVALUACIÓN

- Necesidad de evaluar la calidad de los modelos de aprendizaje automático
- Diferentes criterios a tener en cuenta:
 - Correctitud de la predicción
 - Simplicidad (parsimonia)
 - Interpretabilidad
 - Tiempo de aprendizaje o de predicción
 - Escalabilidad (importante para Big Data)



MÉTRICAS DE CLASIFICACIÓN

- Se usa una **matriz de confusión** para evaluar diferentes métricas de correctitud/error
- Se utilizan dos calificadores para describir cada una de sus casillas:
 - Un calificador de la correctitud de la predicción con respecto a la realidad: Verdadero o Falso
 - Un calificador del tipo de la predicción: Positivo o Falso, con respecto a cada clase de interés (i.e churn)
- Dependiendo del contexto los tipos de error pueden ser mas graves que otros (costos diferentes)

		Predicción	
		Churn ^P	No churn ^N
Realidad	Churn ⁺	VP	FN - Tipo II
	No churn ⁻	FP - Tipo I	VN

- La diagonal (en verde) muestra las instancias correctamente clasificadas. Las demás casillas resume diferentes tipos de error:
 - Tipo I: Falsos positivos
 - Tipo II: Falsos negativos

¿Qué pasa cuando hay mas de dos clases?



MÉTRICAS DE CLASIFICACIÓN

- Interpretarían el caso de la detección de un email spam

TP, TN:

FP: , consecuencia:

FN: , consecuencia:

- Interpretar el caso del diagnóstico de una enfermedad grave?

TP, TN:

FP: , consecuencia:

FN: , consecuencia:

		Predicción	
		Churn ^P	No churn ^N
Realidad	Churn ⁺	VP	FN - Tipo II
	No churn ⁻	FP - Tipo I	VN

- Interpretar el caso de la prospección de clientes de un crédito de consumo (baja aceptación)

TP, TN:

FP: , consecuencia:

FN: , consecuencia:



MÉTRICAS DE CLASIFICACIÓN

- Tasa de correctitud (*accuracy*) = $(VP+VN)/(VP+VN+FP+FN)$
- Error de mala clasificación (contrario de *accuracy*) = $(FP+FN)/(VP+VN+FP+FN)$: probabilidad de error
- Precisión = $VP / (VP+FP)$: valor de predicción positiva, $P(\text{Real+} | \text{Predicho+})$
- *Recall* (o TPR o sensibilidad) = $VP / (VP+FN)$: qué proporción de todos los positivos reales pude identificar como tal, $P(\text{Predicho+} | \text{Real+})$
- Especificidad (o TNR): $VN / (VN+FP)$: qué proporción de todos los negativos reales pude identificar como tal, $P(\text{Predicho-} | \text{Real-})$
- Valor de predicción negativa (FPR) = $FN / (FN+VN)$

		Predicción	
		Churn ^P	No churn ^N
Realidad	Churn ⁺	VP	FN - Tipo I
	No churn ⁻	FP - Tipo I	VN

Imaginemos el problema de detección de spam mail e interpretemos cada métrica

Imaginemos el problema de diagnóstico de cáncer e interpretemos cada métrica



MÉTRICAS DE CLASIFICACIÓN

■ Coeficiente de concordancia **Kappa**

- Para datos nominales u ordinales
- Concordancia entre las predicciones y las clases reales
- Sustrae el efecto de concordancia por suerte (AC) del valor del **accuracy** (concordancia observada - OA)
- Valores van de 0 a 1
- Muy útil sobretodo cuando las clases no están balanceadas
 - Diagnóstico de enfermedades raras
 - Clientes que acepten productos de crédito)

		Predicciones		TOTAL
		+	-	
reales	+	10	4	14
	-	3	2	5
TOTAL		13	6	19

OA = 0,63

AC = 0,59

Kappa = 0,11

Accuracy (OA) = $(10+2)/19=0,63$

(AC) = $(13/19 * 14/19) + (6/19 * 5/19) = 0,59$

Kappa = $(OA-AC)/(1-AC) = 0,11$

		Predicciones		TOTAL
		+	-	
reales	+	0	3	3
	-	0	97	97
TOTAL		0	100	100

OA = 0,97

AC = 0,97

Kappa = 0,00

Accuracy (OA) = $(0+97)/100=0,97$

(AC) = $(0/100 * 3/100) + (100/100 * 97/100) = 0,97$

Kappa = $(OA-AC)/(1-AC) = 0$

		Predicciones		TOTAL
		+	-	
reales	+	1475	988	2463
	-	556	1981	2537
TOTAL		2031	2969	5000

OA = 0,69

AC = 0,50

Kappa = 0,38



TALLER DE CLASIFICACIÓN CON KNN

- DATASET: 150 ejemplos pertenecientes a 3 especies diferentes de la flor Iris
- 4 Atributos: largo y ancho del sépalo, largo y ancho del pétalo
- Ejecutar las tareas de métricas y matriz de confusión para encontrar el K ideal.



Iris setosa



Iris versicolor



Iris virginica



TALLER DE CLASIFICACIÓN CON KNN

1. DATASET: MNIST

- Crear un clasificador K-NN siguiendo las mismas etapas que con el dataset de Iris.
 - Un modelo para cada uno de los 2 dígitos asignados
 - Un modelo para todos los dígitos

2. DATASET: Churn

- Crear un clasificador K-NN siguiendo las mismas etapas que con el dataset de Iris.

