

Aprendizaje automático

Clasificación, Árboles de decisión y KNN

María Auxiliadora Mora, ITCR, julio 2023



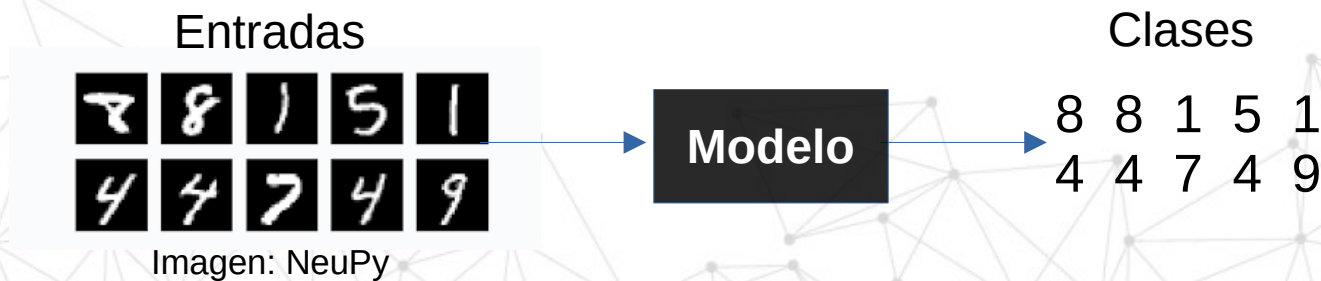
Contenidos

- Introducción
- Aprendizaje inductivo
 - Árboles de decisión
- Algoritmos basados en instancias
 - K vecinos más cercanos (KNN)
- Métricas



Clasificación

- La clasificación es una tarea que requiere el uso de algoritmos de aprendizaje automático para generar modelos que aprenden a **asignar una etiqueta de clase a ejemplos del dominio del problema.**
- Ejemplos:
 - Dado un correo electrónico, clasificar el texto en spam o no.
 - Dado un carácter escrito a mano, clasificarlo como uno de los caracteres conocidos.
 - Dado el comportamiento reciente del usuario, clasificarlo como con interés o no.



Clasificación

Tipos de clasificación:

- Binaria
- Multi-clase
- Multi-etiqueta



Clasificación binaria

- Divide el conjunto de datos en **dos clases**.
- **Ejemplos:**
 - Correo spam (sí/no)
 - Diagnóstico de cáncer de piel (positivo/negativo)
- Los **algoritmos populares** que se pueden usar para la clasificación binaria incluyen:
 - Logistic Regression
 - k-Nearest Neighbors
 - Decision Trees
 - Support Vector Machine
 - Naive Bayes
 - Redes neuronales



Clasificación multi-clase

- Divide el conjunto de datos en **más de dos clases**.
- **Ejemplos:**
 - Clasificación de dígitos escritos a mano.
 - Clasificación de especies en imágenes.
 - Análisis de sentimientos en textos.
- Los **algoritmos populares** (muchos sirven en ambos casos):
 - k-Nearest Neighbors
 - Decision Trees
 - Naive Bayes
 - Random Forest
 - Gradient Boosting
 - Redes neuronales



Clasificación multi-clase

- Los algoritmos que están diseñados para la **clasificación binaria se pueden adaptar para su uso en problemas de clasificación multi-clase.**
- **Estrategias:**
 - **One-vs-Rest:** estrategia de ajuste de múltiples modelos de clasificación binaria para cada clase frente a todas las demás clases.
 - **One-vs-One:** Un modelo para cada par de clases.

Ejemplos de algoritmos:

- Logistic Regression
- Support Vector Machine



Clasificación multi-etiqueta

- Tareas de clasificación que tienen **dos o más etiquetas de clase asignadas a una muestra.**
- **Ejemplo:**
 - Imagen que incluye más de una especie.
 - Más de un vehículo presente en una imagen.

Algunos **algoritmos tienen versiones de etiquetas múltiples** por ejemplo:

- Multi-label Decision Trees
- Multi-label Random Forests
- Multi-label Gradient Boosting

Otro enfoque es utilizar un algoritmo de clasificación independiente para predecir las etiquetas de cada clase.



Ejemplo de clasificación binaria

Situaciones en las que se decide **esperar o no por una mesa**.

Example	Input Attributes										Goal
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
x₁	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>French</i>	<i>0-10</i>	<i>y₁ = Yes</i>
x₂	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Thai</i>	<i>30-60</i>	<i>y₂ = No</i>
x₃	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Some</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Burger</i>	<i>0-10</i>	<i>y₃ = Yes</i>
x₄	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Thai</i>	<i>10-30</i>	<i>y₄ = Yes</i>
x₅	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>Full</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>French</i>	<i>>60</i>	<i>y₅ = No</i>
x₆	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$</i>	<i>Yes</i>	<i>Yes</i>	<i>Italian</i>	<i>0-10</i>	<i>y₆ = Yes</i>
x₇	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>None</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Burger</i>	<i>0-10</i>	<i>y₇ = No</i>
x₈	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$</i>	<i>Yes</i>	<i>Yes</i>	<i>Thai</i>	<i>0-10</i>	<i>y₈ = Yes</i>
x₉	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>No</i>	<i>Full</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Burger</i>	<i>>60</i>	<i>y₉ = No</i>
x₁₀	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>Italian</i>	<i>10-30</i>	<i>y₁₀ = No</i>
x₁₁	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>None</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Thai</i>	<i>0-10</i>	<i>y₁₁ = No</i>
x₁₂	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Burger</i>	<i>30-60</i>	<i>y₁₂ = Yes</i>

Columnas: Alt = alternative; Bar = bar; Fri = Friday; Hun = hungry; Pat = people; Res = reservation; Est = wait estimated.

La clasificación de los ejemplos es positiva (T) o negativa (F)

(Russell & Norvig, 2009)



Arboles de decisión

Es un tipo de **aprendizaje inductivo**, que se basa en el descubrimiento de patrones a partir de ejemplos.

Un **árbol de decisión representa una función**:

- que toma como entrada un vector de valores de atributo y
- devuelve una "decisión": un único valor de salida.
- Los valores de entrada y salida pueden ser discretos o continuos.

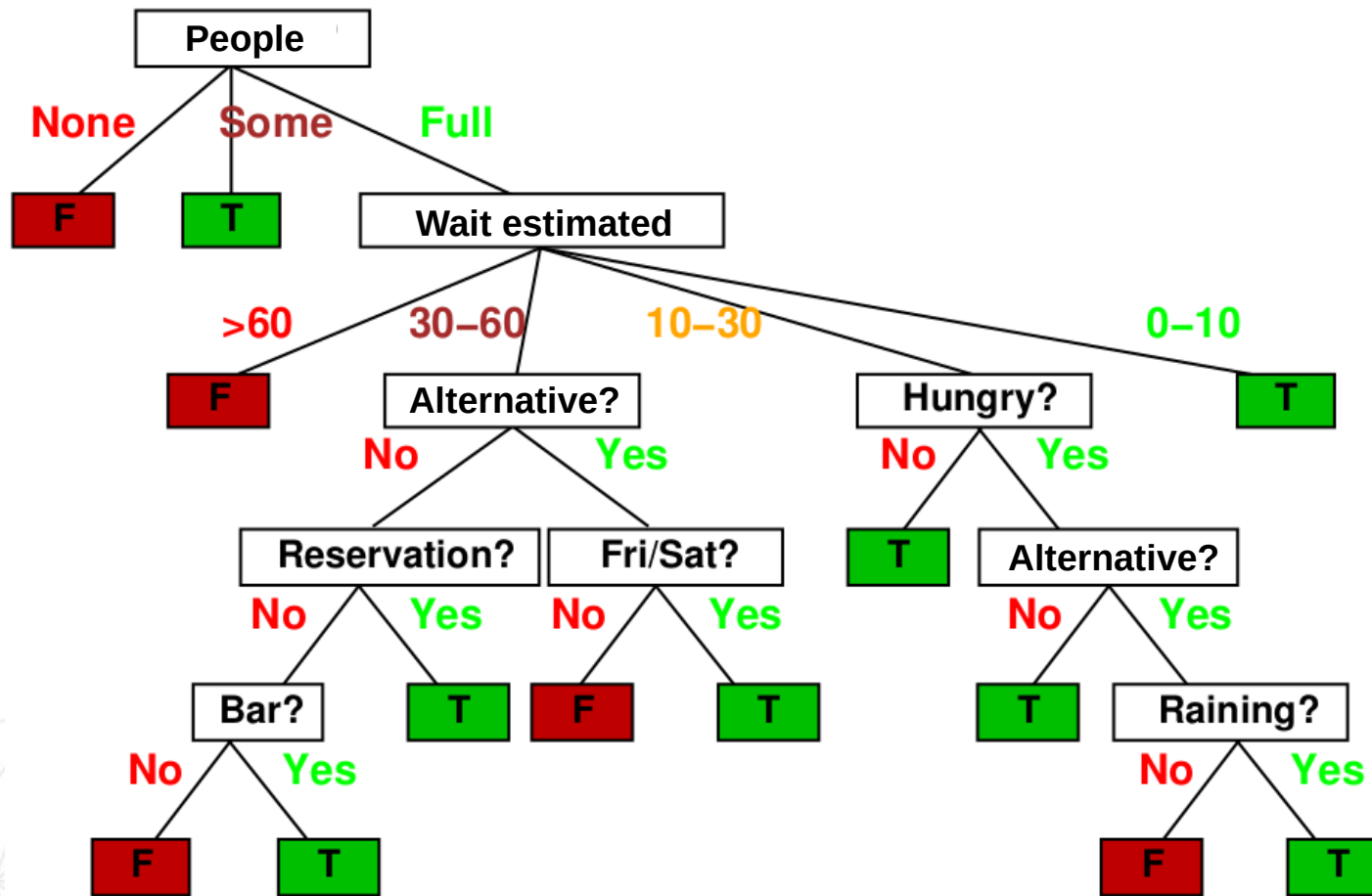
La **inducción en árboles de decisión** es una de las formas más simples y exitosas de aprendizaje automático.



Árboles de decisión

Una posible **representación** para la hipótesis.

Ej. árbol para decidir si esperar o no por una mesa en un restaurante (**WillWait**):



Expresividad en árboles de decisión

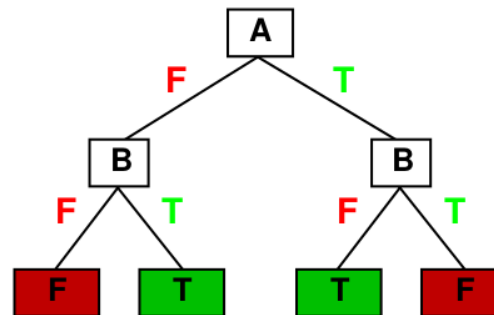
- Un **árbol de decisión binario** es lógicamente equivalente a la afirmación:

$$\text{Goal} \Leftrightarrow (\text{Path1} \vee \text{Path2} \vee \dots)$$

donde cada ruta es una conjunción de pruebas de valor de atributo requeridas para seguir esa ruta.

Ej. xor

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



- Cualquier **sentencia en lógica proposicional** puede ser expresada como un árbol de decisión. Ej.

$$\text{Path} = (\text{People} = \text{Full} \wedge \text{WaitEstimate} = 0-10)$$



Construir árboles de decisión a partir de ejemplos

¿Cuántos árboles de decisión distintos se pueden formar con n atributos booleanos?

- Una tabla de verdad con n atributos tiene 2^n filas $\Rightarrow 2^{n-1}$ árboles distintos.

Con heurísticas \Rightarrow se puede encontrar una buena solución aproximada: un árbol poco profundo y consistente.

Estrategia (divide y conquistarás): siempre se prueba primero el atributo:

- Atributo que hace la mayor diferencia en la clasificación de un ejemplo.
- **Objetivo:** llegar a la clasificación correcta de un ejemplo con un pequeño número de pruebas, lo que significa que todos los caminos en el árbol serán cortos (árbol menos profundo)
- Esta prueba divide el problema en sub-problemas más pequeños que luego se pueden resolver de forma recursiva.

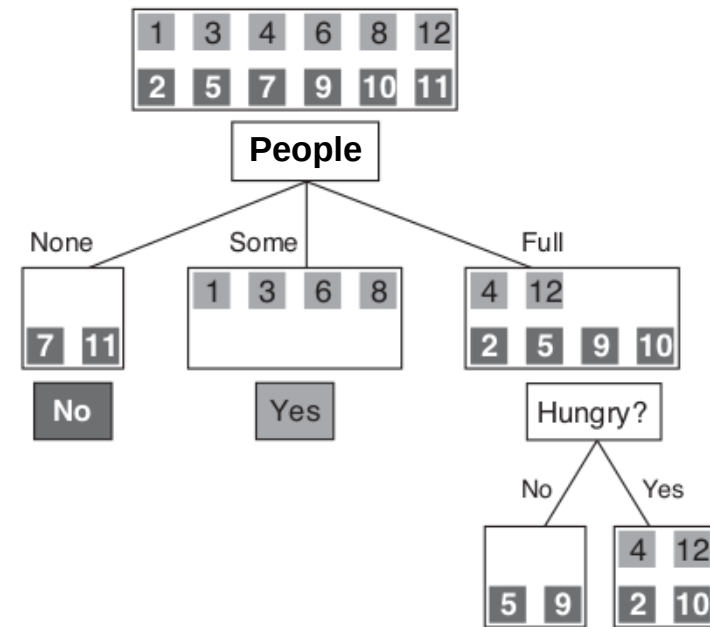
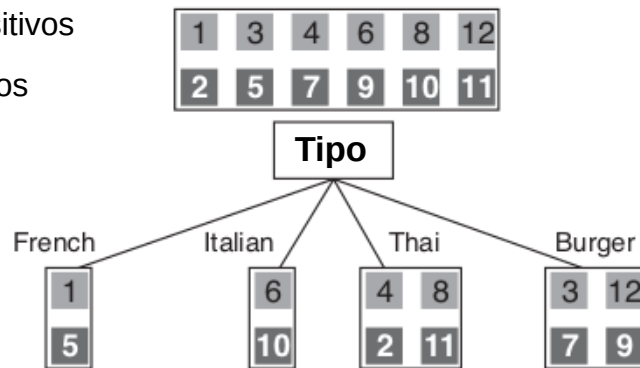


¿Cómo escoger el mejor atributo?

Idea: un buen atributo divide los ejemplos en subconjuntos que son (idealmente) **todos positivos o todos negativos**.

Ejemplos positivos

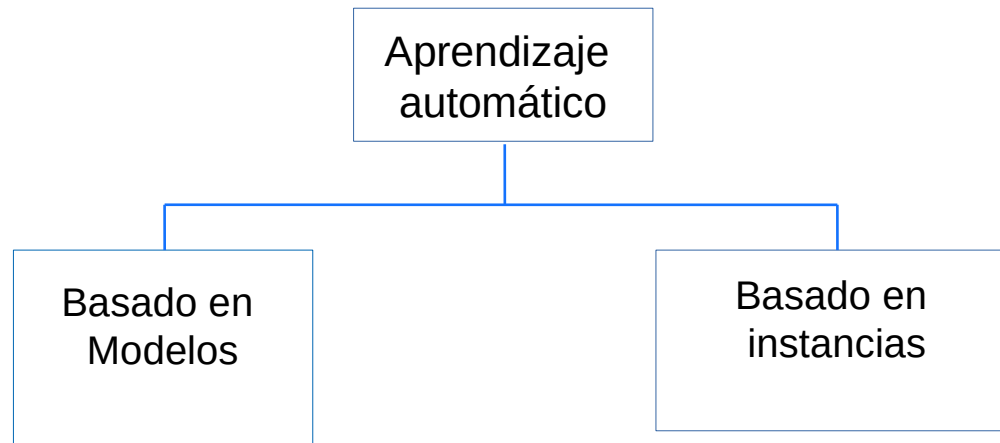
Negativos



People: Es una mejor opción porque da información sobre la clasificación.

Formas de aprendizaje

Otra forma de clasificación del aprendizaje



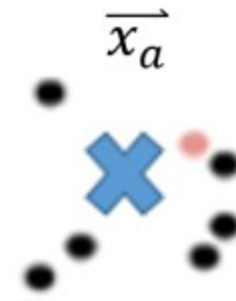
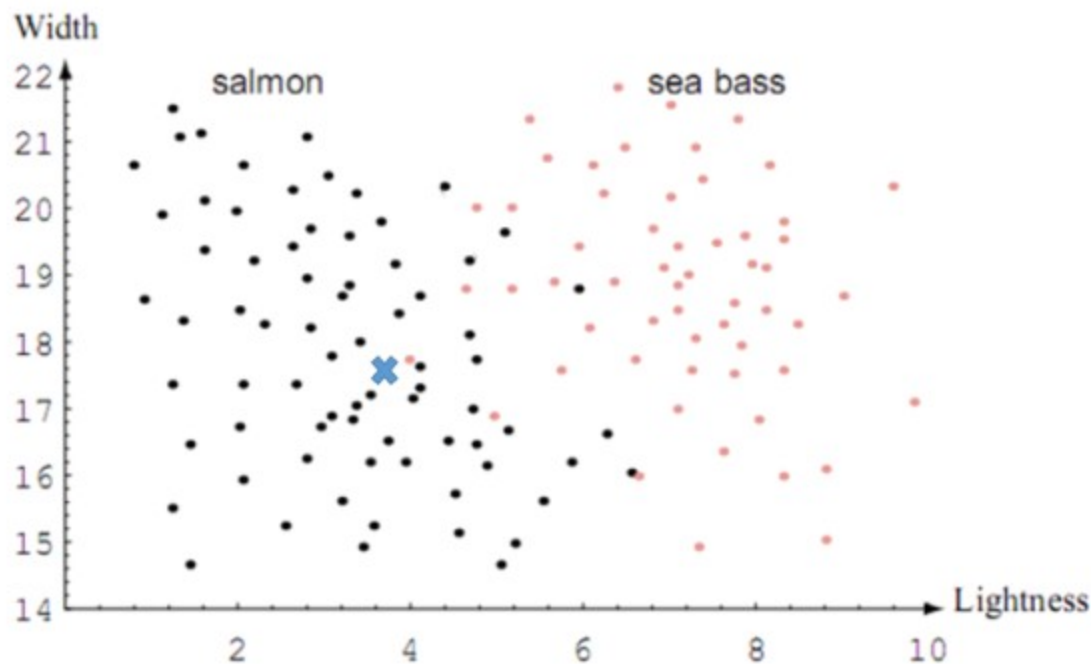
Aprendizaje basado en instancias

- Algoritmos que **almacenan los ejemplos** de entrenamiento en lugar de aprender una función hipótesis explícita.
 - El **generalizar** a partir de los ejemplos **se pospone** hasta que se realiza una nueva clasificación.
 - Se buscan **relaciones entre los ejemplos** almacenados y las **nuevas instancias x_q** para asignar estas a una clase.
- El aprendizaje basado en instancias incluye métodos como:
 - **K Vecinos más cercanos,**
 - Regresión ponderada localmente y
 - Razonamiento basados en casos.



K vecinos más cercanos

- **K vecinos más cercanos:** Para un punto nuevo \vec{x}_a , se calculan los $K = 6$ vecinos más cercanos, usando la distancia Euclidiana,
- En este caso $t_a = 1$ (C_1 corresponde a la clase salmón)



Aprendizaje basado en instancias

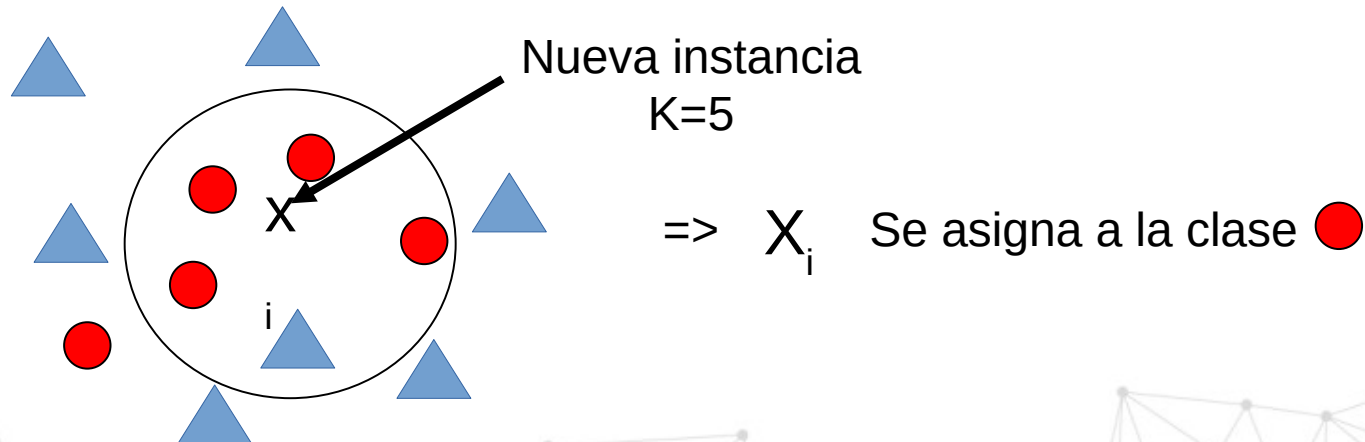
- Algoritmos **perezosos**.
- Una **ventaja** clave:
 - en lugar de **estimar la función objetivo una vez** para todo el espacio de instancias, estos métodos **pueden estimarla localmente** y de manera diferente para cada nueva instancia que se clasifique.
- **Desventaja** pueden ser más **lentos clasificando** nuevas instancias



K vecinos más cercanos (KNN)

Base del algoritmo:

- ¿Cómo medir **distancia** entre instancias?
- ¿Cómo definir la cantidad de muestras más cercanas (**k**) a considerar?



K vecinos más cercanos (K-Nearest Neighbors)

Idea básica: si la nueva instancia a clasificar es más parecida a algunos ejemplos => puede que sean miembros de la misma clase.

Instancias y ejemplos están representados en un espacio n-dimensional R^n .

1) Datos iniciales

- Se requiere un conjunto D de ejemplos $\{x_1, x_2 \dots x_i\}$ y una función objetivo $f(x_i)$.
- Se define una función de similitud o distancia.
- Ejemplos similares (x_i) en D comparten etiqueta (clase), es decir $f(x_i)$ es similar.
- Definir la cantidad de ejemplo cercanos a evaluar (k).



K vecinos más cercanos (K-Nearest Neighbors)

2) Algoritmo:

Nuevas instancias x_q se clasifican midiendo la distancia $d(x_q, x)$ de x_q a todos los ejemplos $x \in D$, con D el conjunto de ejemplos.

Se seleccionan los k ejemplos más cercanos $D_{x_q} \subseteq D$.

- Clasificación:
 - $f(x_q)$ = la clase más frecuente entre las k instancias más cercanas (la clase con mayor cantidad de votos).
- Si la función f es continua se calcula la media de todos los $f(x_i)$ más cercanos.



K vecinos más cercanos (KNN)

- Es una **técnica para clasificar objetos** de acuerdo a los ejemplos más cercanos en el espacio característico.
 - **Espacio** característico está **dado por los ejemplos**.
- Las instancias más cercanas son definidas en términos de una **función de similitud** (ej. distancia euclídea).
- De acuerdo con IEEE, KNN está entre los 10 algoritmos más utilizados de **minería de datos**¹.
- Todo el cálculo se realiza a la hora de clasificar una nueva instancia (algoritmo **perezoso**).

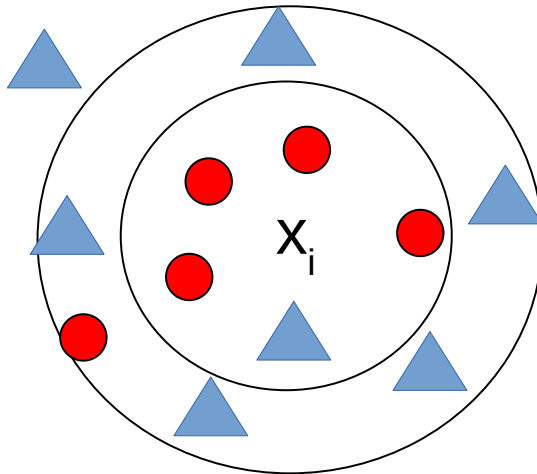
K vecinos más cercanos (KNN)

- ¿Cómo definir la cantidad de muestras más cercanas (k) a considerar?
- Problema:

Clasificar nueva instancia X_i

Si $K = 5 \Rightarrow$ círculo

Si $K = 10 \Rightarrow$ triángulo



K vecinos más cercanos (KNN)

¿Cómo definir la cantidad de muestras más cercanas (k) a considerar?

- Cada conjunto de datos tiene sus propios requisitos.
- En el caso de k pequeño, el ruido tendrá una mayor influencia en el resultado.
- K pequeño tienen un ajuste más flexible, un sesgo bajo pero una varianza alta, y un k grande tendrá un límite de decisión más suave, lo que significa una varianza menor pero un sesgo más alto.
- Generalmente, se selecciona un número impar si el número de clases es par.
- Se puede seleccionar el k evaluando el desempeño con diferentes valores.



K vecinos más cercanos (KNN)

Escala

- Los **atributos deben escalarse** para evitar que las medidas de distancia sean dominadas por uno de los atributos.
- Ejemplos
 - La altura de una persona puede variar de 1.2 – 2.5mt.
 - El peso de una persona puede variar de 45 kilos a 200 kilos
 - El costo de un producto puede variar de 1,000 a 50,000 colones.



Ejemplo con variables no numéricas

¿Dar o no propina en un restaurante?

Ejemplos:

	Food (3)	Chat (2)	Fast (2)	Price (3)	Bar (2)	BigTip
1	great	yes	yes	normal	no	yes
2	great	no	yes	normal	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	normal	yes	yes

Clasificación: Similitud en términos de cantidad de atributos compartidos: Para **k=2**

- $X1 = (\text{great}, \text{no}, \text{no}, \text{normal}, \text{no}) \Rightarrow \text{Yes}$
- - Dos más similares $y = 2$ (1 atributo no coincide, 4 coincidencias) = **Yes**
 $y = 1$ (2 atributos no coinciden, 3 coincidencias) = **Yes**
- $X2 = (\text{mediocre}, \text{yes}, \text{no}, \text{normal}, \text{no}) \Rightarrow \text{Yes/No}$
 - Dos más similares $y = 3$ (1 atributo no coincide, 4 coincidencias) = **No**
 $y = 1$ (2 atributos no coinciden, 3 coincidencias) = **Yes**

Enfoque general

- **Recolección de los datos:** con cualquier método.
- **Preprocesamiento:**
 - Limpieza general.
 - Dependiendo de la medida de distancia, las variables deben ser codificadas como datos numéricos.
- **Entrenamiento:** no aplica.
- **Pruebas:** calcular el error con los datos de prueba.
- **Uso:** algoritmo supervisado para **clasificación o regresión.**

K vecinos más cercanos: Ventajas y desventajas

Ventajas:

- Fácil de implementar y comprender.
- Requiere poco afinamiento y es muy intuitivo.
- Generalmente se comporta bien.

Desventajas

- Necesita una medida de distancia / similitud.
- Para grandes conjuntos de datos puede ser ineficiente clasificando.
 - Debe de medir la distancia de la nueva instancia a todos los elementos del conjunto de ejemplos.
- La precisión de la predicción puede degradarse rápidamente cuando aumenta el número de atributos (no más de 20 atributos).
- Es muy sensible a atributos irrelevantes y a datos atípicos.



Métricas para problemas de clasificación



Clasificación: medidas de desempeño

Matriz de confusión: Precisión, exhaustividad y F1 de cada clase

	Predicted class POSITIVE (spam 📧)	Predicted class NEGATIVE (ad 📧)	Predicted class NEGATIVE (normal 📧)
Actual class POSITIVE (spam 📧)	TRUE POSITIVES 📧 📧 27	FALSE NEGATIVES 📧 📧 286 📧 📧 40	
Actual class NEGATIVE (ad 📧)	📧 📧 1 FALSE POSITIVES	📧 📧 37 TRUE NEGATIVES	📧 📧 9
Actual class NEGATIVE (normal 📧)	📧 📧 5	📧 📧 16	📧 📧 500

Precisión ¿Qué proporción de identificaciones positivas fue correcta?

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

Exhaustividad (recall) ¿Qué proporción de positivos reales se identificó correctamente?

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

Exactitud es la fracción de predicciones que el modelo realizó correctamente.

$$\text{Exactitud} = \frac{VP + VN}{VP + VN + FP + FN}$$

F1 equilibrio entre la precisión y el recall.=
 $2 * ((\text{precision} * \text{recall}) / (\text{precision} + \text{recall}))$

Referencias

- Russell, S.J., Norvig P (2009). Artificial Intelligence : A Modern Approach. Prentice Hall Series.
- Winston, P. (1992). Artificial Intelligence (3era Edición ed.). Massachusetts: Addison-Wesley.
- Brownlee, J. (2020). 4 Types of Classification Tasks in Machine Learning. Recuperado de <https://machinelearningmastery.com/types-of-classification-in-machine-learning/>

