APRENDIZAJE AUTOMÁTICO CON PYTHON

SEGUNDA SESIÓN

TOULOUSE LAUTREC

TOULOUSE LAUTREC

EDUCACIÓN CONTINUA

ALEXANDER VALDEZ PORTOCARRERO

CONTENIDO

Introducción al Machine Learning

SESIÓN - 1

Introducción al Machine Learning

- Fundamentos de Machine Learning
- Tipos de Aprendizaje
- Librerías de Python para Machine Learning
- Algoritmos de Regresión
- Evaluación de Modelos de Regresión

Introducción al Machine Learning

SESIÓN - 2

Clasificación

- Algoritmos de Clasificación
- Balanceo de Datos
- Regresión Logística
- Evaluación de Modelos de Clasificación
- Árboles de Decisión

CONTENIDO

Introducción al Machine Learning

SESIÓN - 3

Clustering

- Algoritmos de Agrupación
- Clustering No Jerárquico: Kmeans, PAM, CLARA
- Clustering Jerárquico: AGNES, DIANA
- Clustering Basado en Densidad: DBSCAN

Introducción al Machine Learning

SESIÓN - 4

Reducción de Dimensionalidad

- Análisis de Componentes Principales
- Análisis Factorial
- Selección de Variables

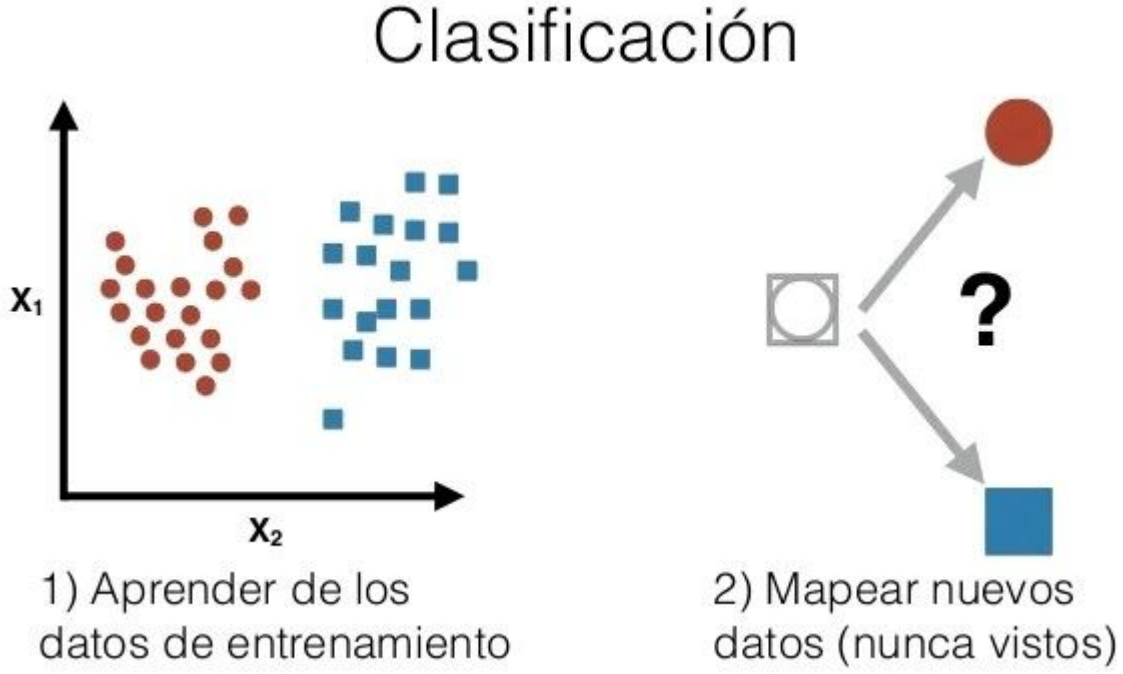
Normas de clase online:

- Habrá un **break de 10 min** después de la parte teórica y antes de la parte práctica en Google Collaboratory.

- La evaluación se realizará **durante la sesión 4 como trabajo final** y obtendrás puntos adicionales en función de las tareas resueltas y enviadas al correo del profesor:

TAREAS PRESENTADAS	PUNTOS ADICIONALES
Ninguna	0
1	+2
2	+4
3	+6

ALGORITMOS DE CLASIFICACIÓN

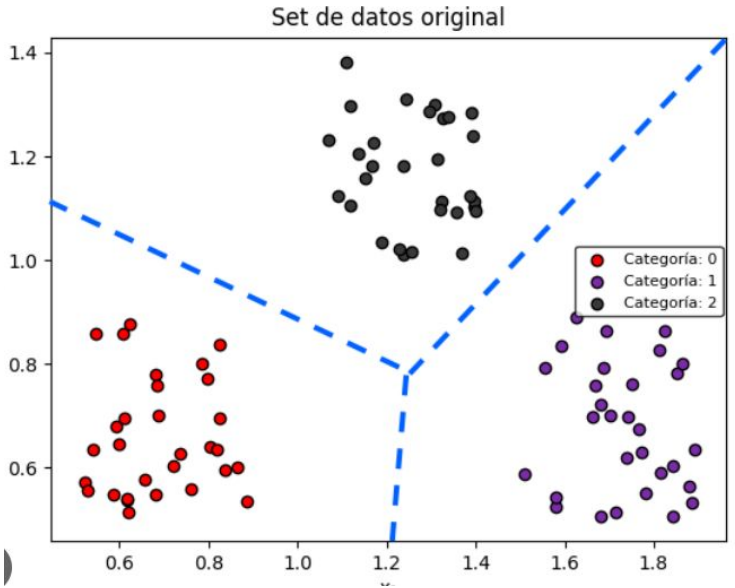


ALGORITMOS DE CLASIFICACIÓN

Los algoritmos de clasificación se utilizan en casos en los que el resultado es un conjunto infinito de resultados.

El ejemplo más utilizado para entender los algoritmos de clasificación es el detector del spam o correo no deseado en el mail. Si buscamos saber si un correo es o no es spam, el algoritmo de clasificación decide a qué tipo pertenece.

A esta metodología también se la conoce por clasificación binaria, pero también podemos hacer una clasificación multiclase. En esta parte nos enfocamos en ver algoritmos de aprendizaje supervisado.



ALGORITMOS DE CLASIFICACIÓN-APRENDIZAJE SUPERVISADO

1. **Regresión Logística**
2. Clasificador Naive Bayes
3. Perceptron
4. Máquinas de Vectores de Soporte (SVM)
5. Árboles de Decisiones
6. **K-vecino más cercano**

REGRESIÓN LOGÍSTICA

La Regresión Logística es un Algoritmo Supervisado y se utiliza para clasificación.

Vamos a clasificar problemas con dos posibles estados “SI/NO”: binario o un número finito de “etiquetas” o “clases”: múltiple. Algunos Ejemplos de Regresión Logística son:

- Clasificar si el correo que llega es Spam o No es Spam
- Dados unos resultados clínicos de un tumor clasificar en “Benigno” o “Maligno”.
- El texto de un artículo a analizar es: Entretenimiento, Deportes, Política ó Ciencia
- A partir de historial bancario conceder un crédito o no

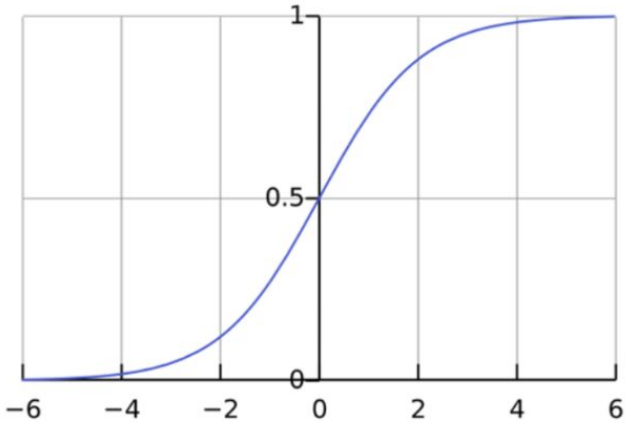
Confiaremos en la implementación del **paquete sklearn en Python** para ponerlo en práctica.

La regresión logística es un método estadístico que trata de modelar la probabilidad de una variable cualitativa binaria (dos posibles valores) en función de una o más variables independientes. La principal aplicación de la regresión logística es la creación de modelos de clasificación binaria.

Se llama regresión logística simple cuando solo hay una variable independiente y regresión logística múltiple cuando hay más de una. Dependiendo del contexto, a la variable modelada se le conoce como variable dependiente o variable respuesta, y a las variables independientes como regresores, predictores o features.

REGRESIÓN LOGÍSTICA

$$f(x) = \frac{1}{1 + e^{-x}}$$



La regresión logística es un modelo estadístico que utiliza la función logística, o función logit, en matemáticas como la ecuación entre *x* e*y*. La función logit mapea *y* como una función sigmoidea de *x*.

Clasificación basada en Naïve Bayes

- Familia de modelos basados en Bayes.
- Veremos Clasificador de Naive Bayes.
- También existen las Redes Bayesianas.

Clasificación Basada en Naïve Bayes

- Modelo que busca modelar la relación **probabilística** entre atributos y clase.
- Modelo generativo, asume una distribución conjunta entre X e Y.
- Supuesto: atributos **independientes** dado la clase (naive assumption).

Clasificador Bayesiano

- Esquema probabilístico para resolver problemas de clasificación.
- Probabilidad condicional:

$$P(C \mid A) = \frac{P(A, C)}{P(A)}$$

$$P(A \mid C) = \frac{P(A, C)}{P(C)}$$

- Teorema de Bayes:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Teorema de Bayes

El teorema de Bayes es una ecuación que **describe la relación de probabilidades condicionales** de cantidades estadísticas. En clasificación bayesiana estamos interesados en encontrar la probabilidad de que ocurra una "clase" dadas unas características observadas (datos). Lo podemos escribir como **P(Clase | Datos)**. El teorema de Bayes nos dice cómo lo podemos expresar en términos de cantidades que podemos calcular directamente:

- **Clase** es una salida en particular, por ejemplo "comprar"
- **Datos** son nuestras características, en nuestro caso los ingresos, gastos, hijos, etc
- **P(Clase|Datos)** se llama posterior (y es el resultado que queremos hallar)
- **P(Datos|Clase)** se llama "verosimilitud" (en inglés likelihood)
- **P(Clase)** se llama anterior (pues es una probabilidad que ya tenemos)
- **P(Datos)** se llama probabilidad marginal

$$P(Clase \mid Datos) = \frac{P(Datos|Clase) * P(Clase)}{P(Datos)}$$

Si estamos tratando de elegir entre dos clases como en nuestro caso, entonces una manera de tomar la decisión **es calcular la tasa de probabilidades a posterior**:

Ejemplo Teorema de Bayes

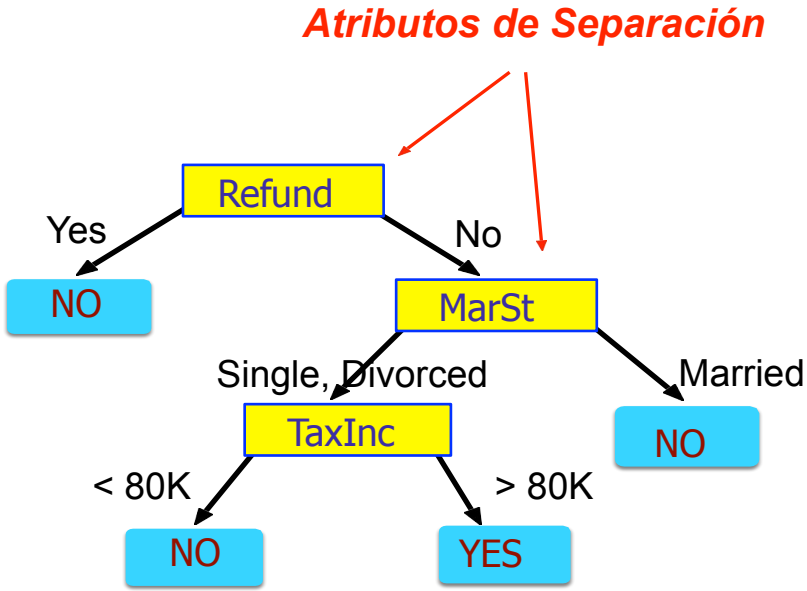
- Dado:
  - Un doctor sabe que la meningitis produce rigidez de cuello el 50% de las veces.
  - La probabilidad previa de que cualquier paciente tenga meningitis es 1/50,000.
  - La probabilidad previa de que cualquier paciente tenga rigidez en el cuello es de 1/20.
- ¿Si un paciente tiene el cuello rígido, cuál es la probabilidad de que tenga meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Árbol de Decisión

	categorical	categorical	continuous	class
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Datos de Entrenamiento



Modelo: Árbol de Decisión

Árbol de Decisión

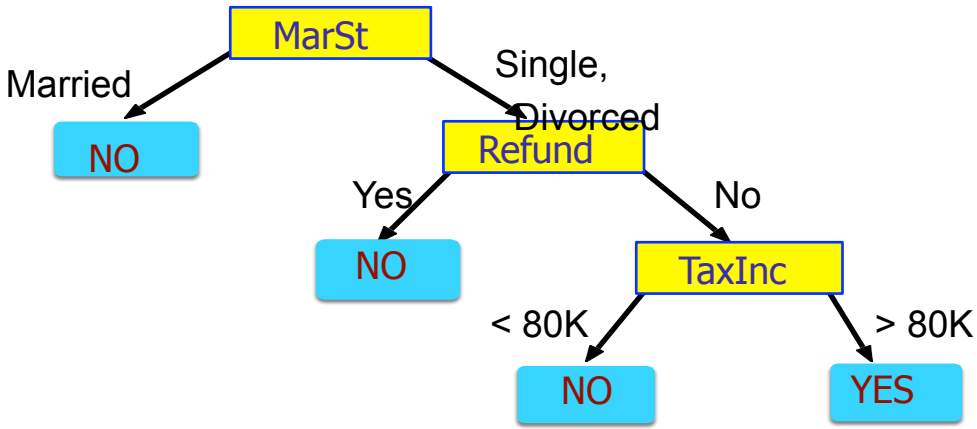
El árbol tiene tres tipos de nodos:

1. Un nodo raíz que no tiene arcos entrantes y tiene arcos salientes.
2. Nodos internos, cada uno de los cuales tiene exactamente un arco entrante y dos o más arcos salientes.
3. Nodos hoja o terminales, cada uno de los cuales tiene exactamente un arco entrante.

- A cada nodo de hoja se le asigna una **etiqueta** de clase.
- Los nodos no terminales, que incluyen la raíz y otros nodos internos, contienen **tests** sobre los atributos para separar los ejemplos que tienen valores diferentes para esos atributos.
- El árbol de decisión **fragmenta** el dataset de manera recursiva hasta asignar los ejemplos a una clase.

Otro Ejemplo

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



¡Puede existir más de un árbol que se ajuste a los datos!

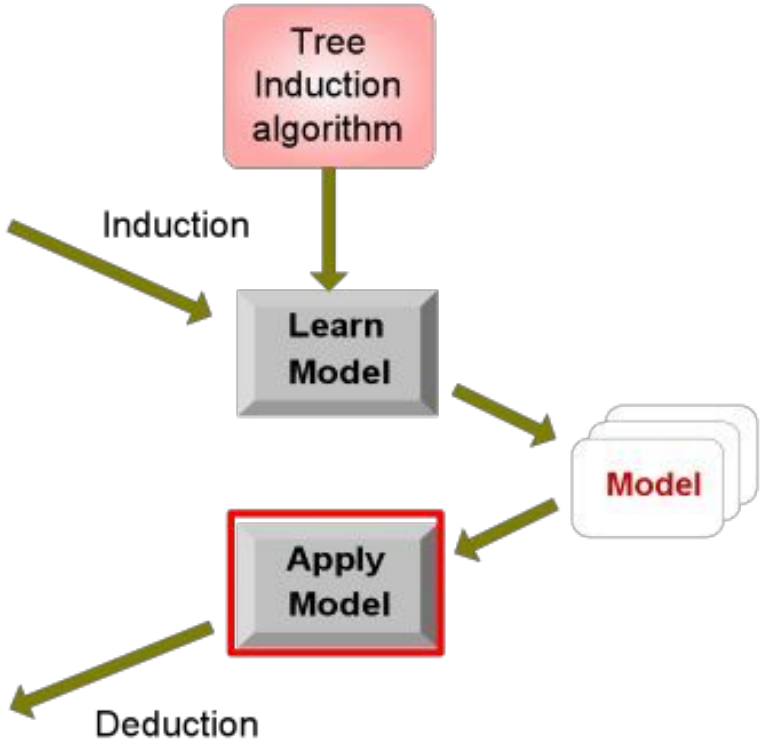
Clasificando con un árbol de decisión

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

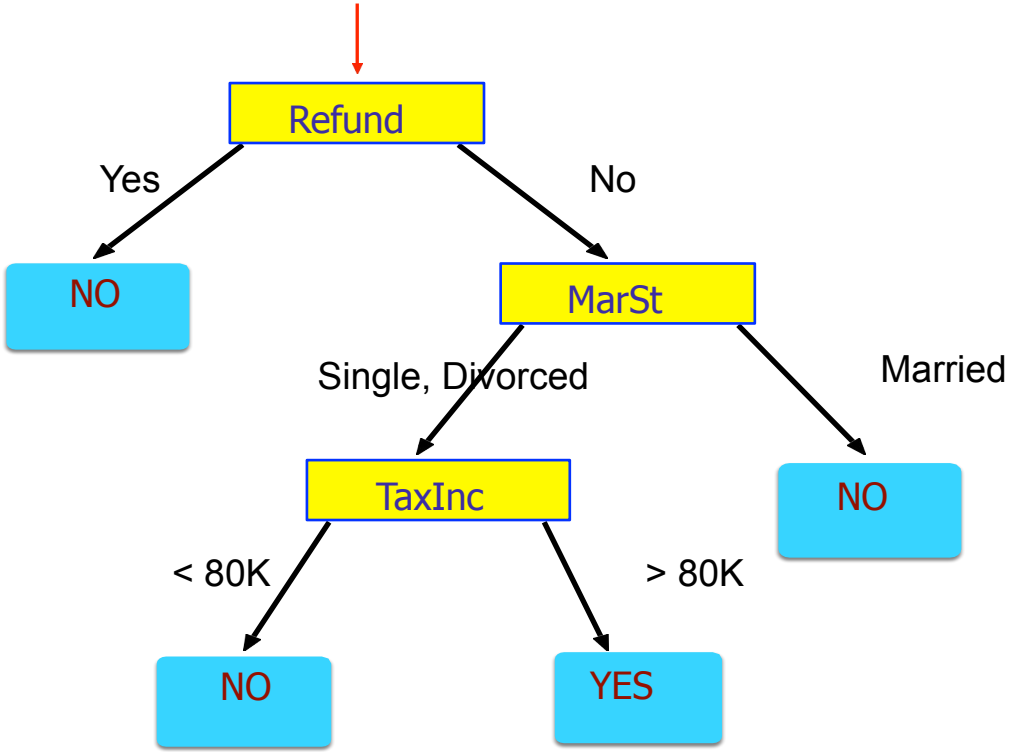


Aplicamos el modelo

Comenzamos en la raíz

Dato de Evaluación

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Clasificando con un árbol de decisión

- Muchos algoritmos
  - CART
  - ID3, C4.5 (J48 en Weka)
  - SLIQ, SPRINT

Construyendo un Árbol de Decisión

- Estrategia: Top down (greedy) - Divide y vencerás recursiva
  - Primero: seleccionar un atributo para el nodo raíz y crear rama para cada valor posible del atributo .
  - Luego: dividir las instancias del dataset en subconjuntos, uno para cada rama que se extiende desde el nodo.
  - Por último: repetir de forma recursiva para cada rama, utilizando sólo las instancias que llegan a ésta.
- Detenerse cuando todas las instancias del nodo sean de la misma clase.

Un árbol de decisión hace cortes perpendiculares a los ejes

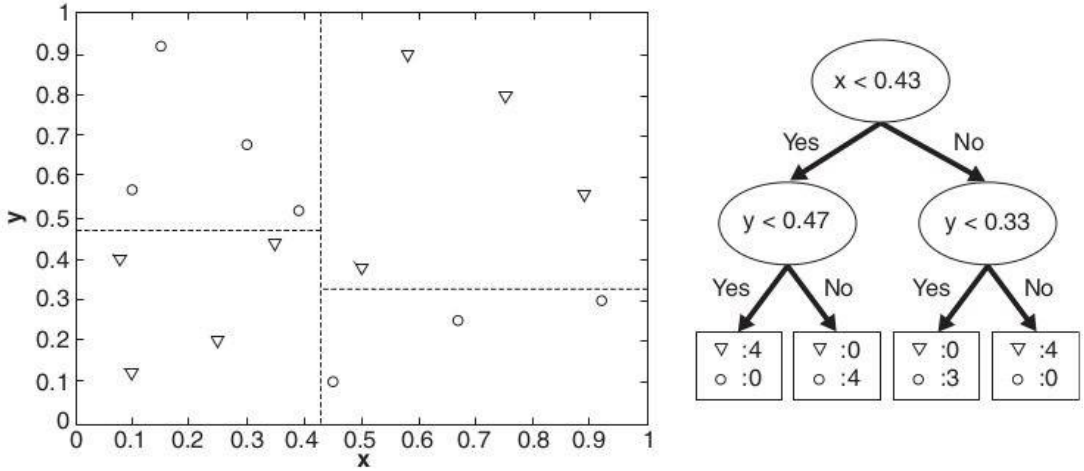


Figure 3.20. Example of a decision tree and its decision boundaries for a two-dimensional data set.

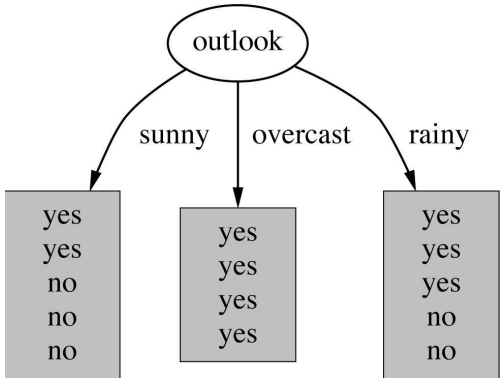
El dataset Weather

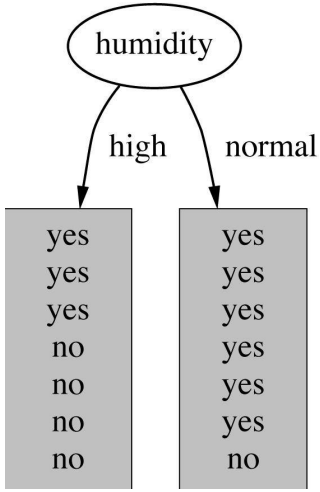
Condiciones para salir a jugar tenis:

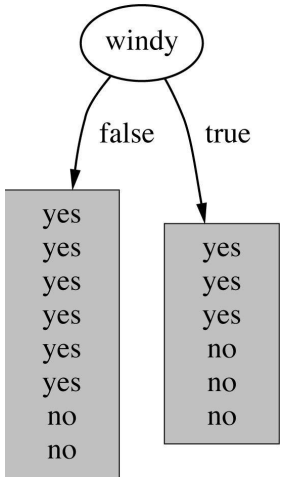
Table 4.6 The weather data with identification codes.

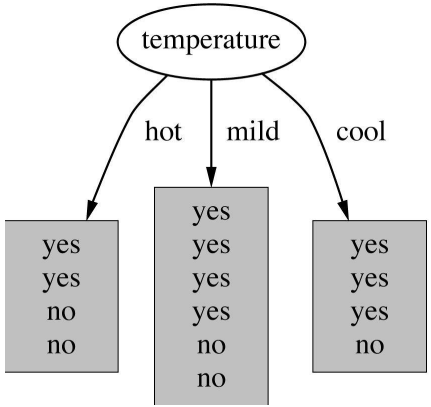
ID code	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
e	rainy	cool	normal	false	yes
f	rainy	cool	normal	true	no
g	overcast	cool	normal	true	yes
h	sunny	mild	high	false	no
i	sunny	cool	normal	false	yes
j	rainy	mild	normal	false	yes
k	sunny	mild	normal	true	yes
l	overcast	mild	high	true	yes
m	overcast	hot	normal	false	yes
n	rainy	mild	high	true	no

¿Cómo escoger atributos?







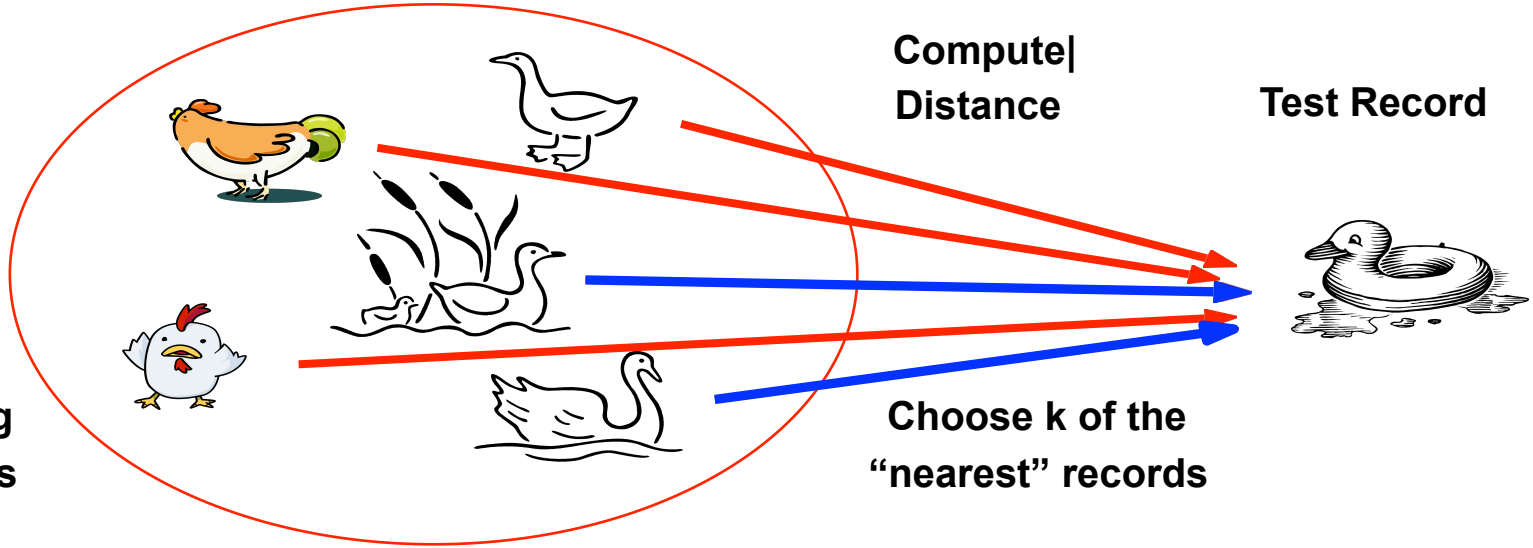


Clasificador KNN

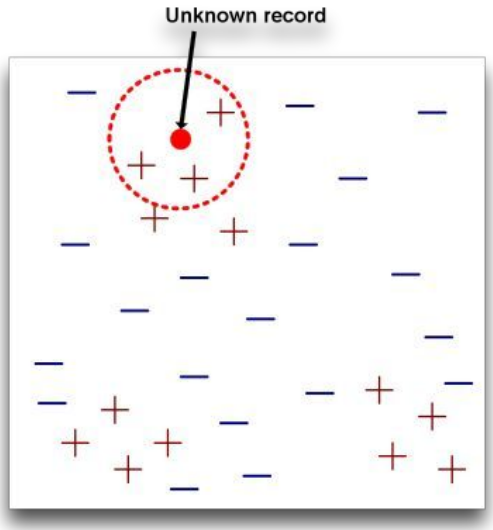
- Nearest Neighbor Classifier (o k-nn)
- Es un clasificador basado en **instancias**
- Conocido como **lazy**
  - Usa los **k** puntos más cercanos (nearest neighbors) para realizar la clasificación

Clasificadores KNN

- Idea:|
  - **If it walks like a duck, quacks like a duck, then it's probably a duck**



Clasificadores KNN



- Necesita 3 cosas
  - –Set de records almacenados.
  - –Métrica de distancia para calcular la distancia entre records.
  - –Valor de *k*, el número de vecinos cercanos a obtener.
- Para clasificar un récord nuevo
  - –Calcular la distancia los los récords almacenados.
  - –Identificar *k* nearest neighbors .
  - – Utilizar la clase de los knn para asignar la clase al record nuevo (e.j. voto de la mayoría).

Métricas de distancia

- Para atributos numéricos usamos la distancia euclidiana:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2},$$

- Una versión más general es la distancia de **Minkowsky** (r=1 => distancia Manhattan, r=2 => distancia euclideana)

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

- Es muy importante que los atributos estén normalizados.

Escalando atributos

- Problemas de escalas
  - Atributos deben ser escalados para prevenir que algún atributo domine la métrica de distancia
  - Ejemplos:
    - La altura de una persona puede variar entre 1.5m a 1.8m
    - El peso puede variar entre 40 kg a 150 kg

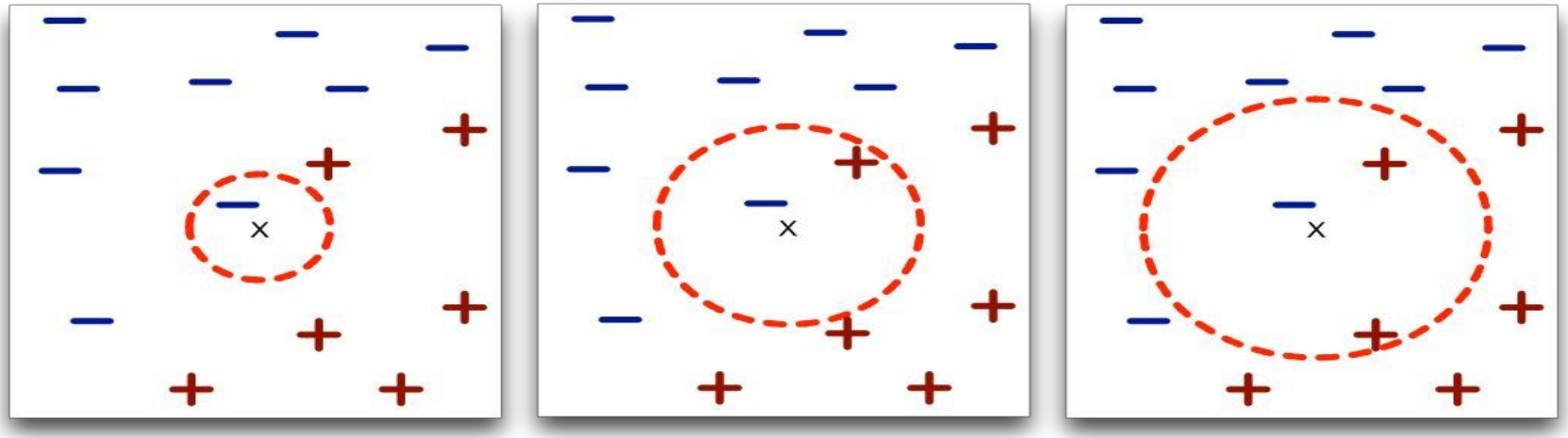
Técnicas para escalar atributos

$$\frac{x-\mu_x}{\sigma_x}$$

- Normalización a media cero y varianza unitaria:

$$\frac{x-min_x}{max_x-min_x}$$

Definición de NN

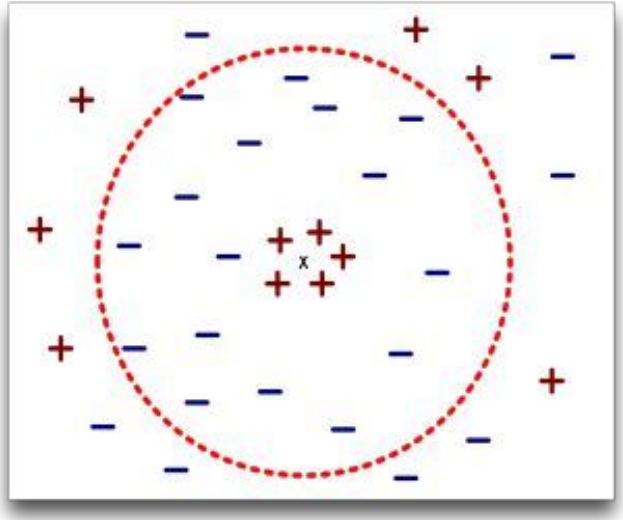


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-NN de un récord x son los puntos que tienen las k menores distancias a x

Eligiendo el valor de K

- k muy pequeño es susceptible a ruido
- k muy grande puede incluir puntos de otra clase



Clasificación kNN

- Los clasificadores k-NN son **lazy learners.**
  - No construyen modelos explícitos, es más flexible ya que no necesita comprometerse con un modelo global a priori.
  - Al contrario de otros **eager learners** como los árboles de decisión o clasificadores basados en reglas.
  - Es independiente del nro. de clases.
  - La clasificación es más costosa (memoria y tiempo).

La Maldición de la Dimensionalidad

- Cuando los datos tienen una alta dimensionalidad KNN está sujeta a la **Maldición de la Dimensionalidad**.

- Fenómeno en que muchos tipos de análisis de datos se vuelven significativamente más difíciles a medida que aumenta la dimensionalidad de los datos.

- Para la clasificación, esto puede significar que no haya suficientes ejemplos para crear un modelo que asigne de forma confiable una clase a todos los ejemplos posibles.

- Para técnicas basadas en distancias (KNN, K-means) las distancias entre objetos se vuelven menos claras cuando hay muchas dimensiones.

APRENDIZAJE AUTOMÁTICO CON PYTHON

Gracias por su atencion.