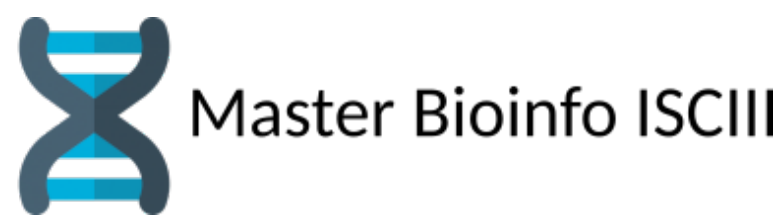


Máster en Bioinformática  
aplicada a la Medicina Personalizada y la Salud  
2022/23

# Supervised Learning Algorithms



Daniel Glez-Peña &  
Hugo López-Fernández &  
Alba Nogueira-Rodríguez  
@SINGgroup  
[www.sing-group.org](http://www.sing-group.org)

# K-Nearest Neighbors

# K-Nearest Neighbors

---

- Para cada muestra a clasificar:
  - 1) Se buscan las K muestras más parecidas (utilizando alguna medida de distancia).
  - 2) Se realiza una predicción:
    - a) En clasificación: se asigna a la nueva muestra la clase mayoritaria entre las K más parecidas.
    - b) En regresión: se asigna a la nueva muestra el valor medio entre las K más parecidas.

# K-Nearest Neighbors

---

- KNN es una de las técnicas más simples: no se ajusta un modelo (fase de entrenamiento).
- Sin embargo, es necesario ajustar los parámetros del algoritmo para obtener un buen resultado:
  - K: número de vecinos a considerar.
  - Función de distancia empleada.
- Importante: es necesario considerar la necesidad de escalar las variables (al igual que en otras técnicas basadas en el cálculo de distancias).

# K-Nearest Neighbors

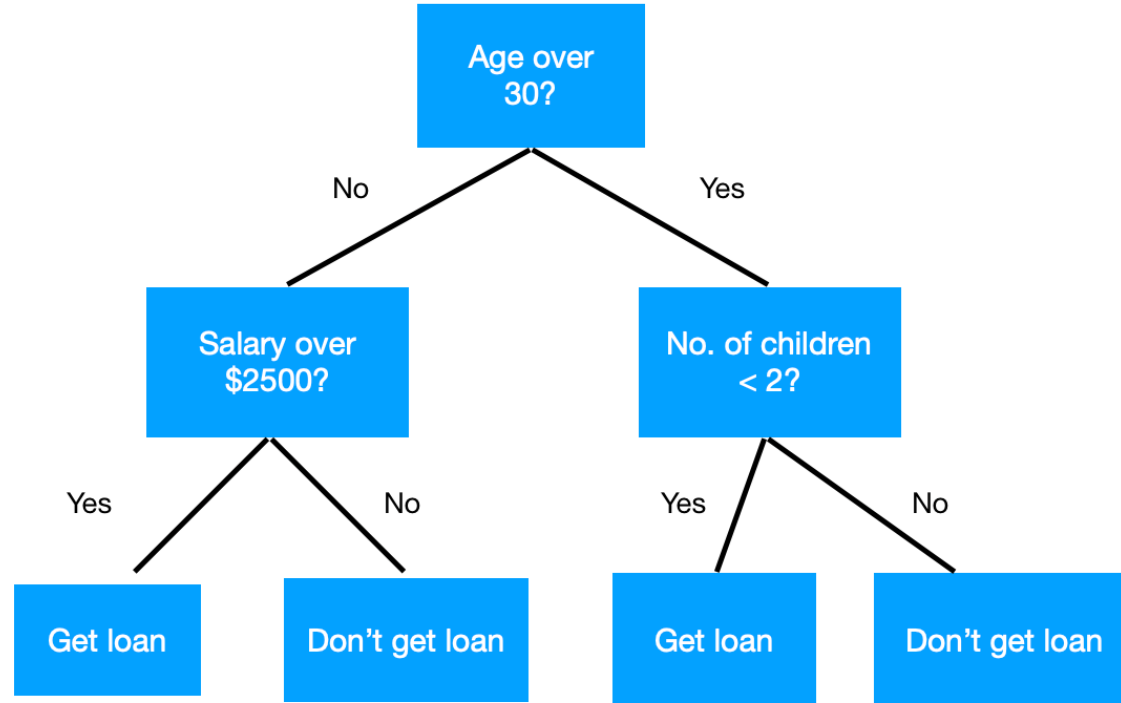
---

- Algunas librerías ofrecen una probabilidad entre 0 y 1 de que una muestra pertenezca a una clase:
  - Esta probabilidad está basada en el porcentaje de los K vecinos cercanos que pertenecen a dicha clase.
- Esto permite establecer otros cortes distintos a 0.5 (clase mayoritaria) para seleccionar la clase a la que pertenece una muestra.
  - Por ejemplo: si tratamos de identificar una clase “rara” (poco prevalente), podemos establecer un corte mucho menor.

# Decision Trees

# Decision Trees

---



Fuente: <https://eloquentarduino.github.io/2020/10/decision-tree-random-forest-and-xgboost-on-arduino/>

# Decision Trees

---

- El árbol de decisión se infiere mediante un proceso de particionado recursivo (*CART: Classification and Regression Tree*):
  - Dado un conjunto de muestras  $A$ , el objetivo del algoritmo es partirlo en  $A_1$  y  $A_2$  utilizando una variable  $X_i$  y un valor específico de dicha variable.



# Algoritmo de particionado recursivo (CART)

---

1. Para cada posible variable  $X_i$ :
  - a) Para cada posible valor  $s_j$  de  $X_i$ :
    - i. Dividir A en dos conjuntos: uno con las muestras con valores  $<$  que  $s_j$  y el resto en otra.
    - ii. Medir la homogeneidad de las clases en cada partición.
  - b) Seleccionar el valor de  $s_j$  que produce la partición más homogénea.
2. Seleccionar la variable  $X_i$  y el valor de partición  $s_j$  que produce la partición más homogénea.
3. Repetir los pasos 1 y 2 para cada nueva partición.

# Medidas de homogeneidad o impuridad

---

- Gini impurity:
  - En un nodo concreto, mide cómo de probable es clasificar mal una muestra de entrenamiento si se etiqueta al azar en base a la distribución de etiquetas en la partición del nodo.
  - Por ejemplo: si la mitad de muestras son del grupo A y la mitad del B, hay una probabilidad del 50% de clasificar mal una muestra si se etiqueta al azar.

# Medidas de homogeneidad o impuridad

---

- Gini impurity:
  - Toma un valor entre 0 (totalmente puro) y 0.5 (totalmente impuro).
  - Vale 0 (totalmente puro) cuando solo hay un posible grupo en un nodo.
  - Este índice representa la probabilidad de clasificar mal una nueva muestra en un nodo determinado del árbol, basado en los datos de entrenamiento.

# Medidas de homogeneidad o impuridad

---

- Information Gain:
  - Basada en “Information Entropy”.
  - <https://victorzhou.com/blog/information-gain/>

# Overfitting

---

- El algoritmo puede crear particiones hasta que en cada hoja solo queden muestras de la misma clase.
- Se habrá adaptado perfectamente a las muestras de entrenamiento pero es posible que las reglas no sean flexibles y el modelo no tenga la capacidad de generalizar ante nuevas muestras.
- Las estrategias para evitar el overfitting en los árboles de decisión pasan por evitar que el árbol crezca demasiado.

# Overfitting

---

- Evitar el overfitting:
  - Establecer un valor máximo para la profundidad del árbol.
  - Establecer un valor mínimo de muestras necesarias para hacer una división.
  - Establecer un valor mínimo en la reducción de la impuridad para poder realizar una partición.

# Decision Trees

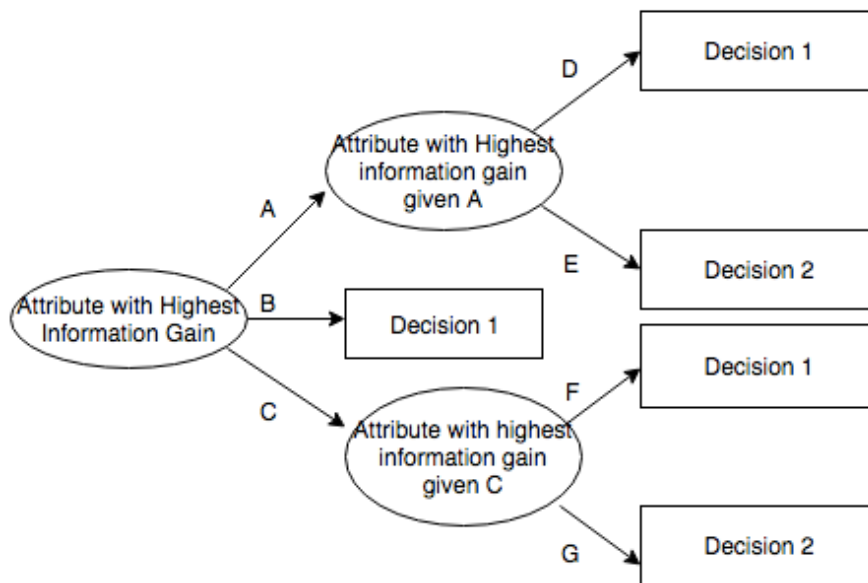
---

- Ventajas:
  - Son modelos “de caja blanca” porque proporcionan un conjunto de reglas de clasificación interpretables por un humano.
  - Se pueden representar gráficamente.
  - Permiten extraer o identificar las variables más relevantes para la clasificación de interés.

# Decision Trees

---

- Otros algoritmos para crear árboles de decisión:
  - ID3 (Iterative Dichotomiser 3)
  - C4.5 (sucesor de ID3)





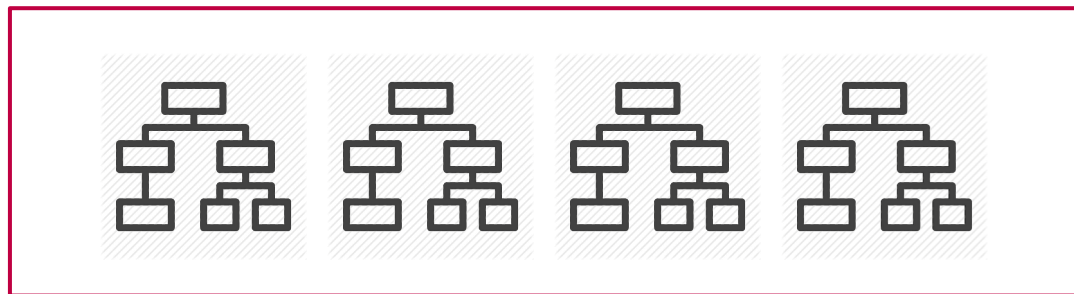
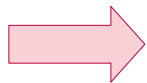
# Decision Trees

---

- ¿Y si en lugar de utilizar un único árbol de decisión utilizamos muchos y hacemos una votación?



Decision Tree



Random Forest

# Random Forest

# Random Forest

---

- Ensemble: hacer una predicción utilizando un conjunto de modelos.
- Bagging: técnica para crear una colección de modelos utilizando muestreos *bootstrapping* de los datos disponibles.
  - Bagging = Bootstrap aggregation.
- Random Forest: aplicación de bagging a árboles de decisión.
  - Random Forest = Bagged decision trees.

# Random Forest

---

1. Tomar una muestra bootstrap (con reemplazamiento) de los datos de entrenamiento.
2. Para hacer la primera partición, tomar una muestra aleatoria (sin reemplazamiento) de  $p < P$  variables predictoras.
3. Para cada variable muestreada, aplicar el algoritmo de particionado recursivo para árboles de decisión.
4. Seleccionar la variable que produzca la partición más homogénea.

# Random Forest

---

5. Repetir los pasos 2 a 4 para las siguientes particiones hasta que el árbol de decisión esté completo.
  - En cada partición se considera un subconjunto aleatorio de variables diferente.
6. Cuando se ha completado el árbol, se vuelve al paso 1 para crear un nuevo árbol de decisión.

# Out-of-bag error

---

- Las muestras no utilizadas para construir un árbol de decisión son un *test set* (*out-of-bag samples*).
- Cálculo del error OOB:
  - Para cada muestra, obtener una predicción utilizando los árboles en los que dicha muestra no ha sido utilizada para crear el árbol.
  - La proporción de muestras OOB que han sido clasificadas incorrectamente es el error OOB.

# Random Forest

---

- Producen predicciones más acertadas que los árboles de decisión.
- Se pierde la interpretabilidad de los árboles de decisión porque no existe un único conjunto de reglas.
- Aún así, es posible medir la importancia de las variables:
  - Reducción en la tasa de acierto del modelo si se permutan los valores de una variable. Para hacer esta estimación se utilizan las muestras OOB.
  - Reducción media en la Gini impurity de cada variable.