# One Rule (1R)

* Árbol de decisión de un nivel.
* Simple y barato computacionalmente hablando.
* Generación de reglas que evalúan un único atributo (ramificaciones de un nivel).
* Predicción de la clase con datos de entrenamiento más frecuentes.
* Tasa de error: cantidad de datos que no se pueden predecir.
* Valores ausentes: se agregan como otro valor del atributo.
* Valores numéricos: se discretizan.

# Naive Bayes

* Todos los atributos son igualmente importantes e independientes.
* Se multiplican las probabilidades directamente para hallar las probabilidades de la clase deseada. Luego se normaliza los resultados para obtener el porcentaje real.
* Regla de Bayes: P(H|E) = P(E|H) \* P(H) / P(E).
* Probabilidad combinada: P(Yes|E) = P(E1|yes) \* P(E2|yes) … \* P(En|yes) \* P(yes) / P(E)
* Si hay valores faltantes se compensan (se le agrega +1 a todo).
* Si no se contase con un valor, no se agrega a la productoria (EJ: Hay 5 variables, pero no se conoce una).
* Valores numéricos:

# Overfitting y Underfitting

* Overfitting: poco sesgo y alta varianza. Modelo complicado.
* Underfitting: alto sesgo y poca varianza. Modelo simple.

# Matriz de confusión

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Clase predicha | |
|  |  | YES | NO |
| Clase real | YES | a | b |
| NO | c | d |

AC = (a+d)/(a+b+c+d)

# Métricas

* Precisión: fracción de instancias recuperadas que son relevantes. (TP/(TP+FP))
* Recall: fracción de instancias relevantes que han sido recuperadas. (TP/(TP+FN))
* True Positives: instancias que pertenecen a la clase que se clasifican correctamente en dicha clase.
* True Negatives: instancias que no pertenecen a la clase que se clasifican correctamente en dicha clase.
* False Positives: instancias que no pertenecen a la clase pero se clasifican como dicha clase.
* False negatives: instancias pertenecientes a la clase pero que no se clasifican como dicha clase.
* F-Measure:
* B = 1: Precisión igual de importante que Recall
* B > 1: Más importante el Recall.
* B < 1: Más importante la precisión

# Árboles de decisión

* Aprendizaje de árboles de decisión a partir de tuplas de entrenamiento etiquetadas con determinada clase.
* Estructura donde:
  + Cada nodo interno (no hoja) denota una prueba.
  + Cada rama representa un resultado.
  + Cada nodo hoja tiene la etiqueta clase.
* Métodos de selección de atributos:
  + Ganancia de información: minimiza la información necesaria para clasificar tuplas. Garantiza que se encuentre un árbol simple.
  + Relación de ganancia: sesgada a conjuntos con muchos resultados. Normaliza la ganancia de información con un valor de información dividida.
  + Índice Gini: usa el algoritmo de CART. Mide la impureza. Considera una división binaria para cada atributo.

# Evaluación de modelos algorítmicos

* Método de retención:
  + Se dividen los datos aleatoriamente en 2 conjuntos: entrenamiento y prueba.
  + La precisión se halla con el de prueba.
* Sub-muestreo aleatorio
  + Variación de modelo de retención pero que se repite k veces.
  + Precisión general es el promedio de las precisiones obtenidas en cada iteración.
* Cross-Validation:
  + Los datos iniciales se dividen en k subconjuntos (folds) mutuamente exclusivos.
  + Entrenamiento y pruebas se realizan k veces.
  + En la iteración i, se reserva una partición i como conjunto de prueba y las restantes entrenan al modelo.
  + Cada muestra se usa el mismo numero de veces (K-1) y una vez de prueba.
  + La precisión es el numero de clasificaciones correctas entre el numero total de tuplas iniciales.
  + Recomendación: k=10 debido al sesgo y varianza baja.