

C5//

Arbol de decision: - Metodo para inferencia inductiva

- Predice valor objetivo en funcion de las Reglas.

- Es una Disyunción de Conjunciones sobre valores de Atributos

Nodo: Representa test sobre un Atributo de la instancia

Ej: Corresponde a un valor para ese Atributo

Util

↳ Instancias Representados por pares Atributo-valor

↳ Función Objetivo tiene valor de Salida Discreto

↳ Se pueden Repetir hipótesis disyuntivas.

↳ posible conjunto de entrenamiento Ruidoso

↳ " valores de Atributos Faltantes

- 1) Eleccion de mejor Atributo para nodo Actual "A"
 - 2) Asignar "A" como nodo Actual.
 - 3) Para cada valor que toma "A" crear un hijo.
 - 4) Clasificar (Repartir) instancias en los nuevos nodos segun valor de "A".
 - 5) Si las instancias estan bien clasificadas: Terminar
- SINO: Iterar Sobre Nuevos nodos

Information Gain (Entropia)

- Reducción esperada de entropia por partir ejemplos basados en ese atributo

Mide
Impureza de
S

$$\text{Entropia}(S) = \sum_{\text{clases}} -p_c \log_2(p_c)$$

$$\text{Gain}(S, A) = \text{Entropia}(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} \text{Entropia}(S_v)$$

Partición por Atributo
"A". Reducción de
Entropia

Gain Ratio: Corrige preferencias de
Information Gain sobre
Atributos c/ muchos valores

Gini: Se mide impureza de la muestra (conjunto de entrenamiento)

Impureza
$$Gini(S) = 1 - \sum_{c \in \text{clases}} \left(\frac{|S_c|}{|S|} \right)^2$$

S_c : conjunto de instancias que pertenecen a la clase c

GiniGain
$$GiniGain(S, A) = Gini(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} Gini(S_v)$$

Posibles valores del atributo A

$$S_v = \{s \in S \mid A(s) = v\}$$

se elige atributo con MAYOR GiniGain

INFORMATION GAIN

$$\text{Gain}(S, A) = \text{Entropia}(S) - \sum_{\text{reabresca}} \frac{|S_v|}{|S|} \text{Entropia}(S_v)$$

$$\text{Entropia}(S) = \sum_{C \in \text{clases}} -p_c \log_2 p_c \quad , \quad p_c: \text{proporci3n de instancias de } S \text{ que pertenecen a la clase } C \text{ (ya a correr o no?)}$$

$$\text{Entropia}([9+, 5-]) = -\left(\frac{9}{14}\right) \log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) \log_2 \left(\frac{5}{14}\right) = 0.940$$

Particiones por HUMEDAD: Alta y Normal

$$|S_v| = 7 \text{ Altas y } 7 \text{ Normales}$$

$$|S| = 14$$

$$\text{Entropia}(S_{v=\text{Humedad Alta}}) = -\underbrace{\left(\frac{3}{7}\right) \log_2 \left(\frac{3}{7}\right)}_{-1.222} - \underbrace{\left(\frac{4}{7}\right) \log_2 \left(\frac{4}{7}\right)}_{-0.807} = 0.984$$

+0.523 + 0.467

Cuántas de los
Altas son Si
y cuántas son
No.

$$\text{Entropia}(S_{v=\text{Humedad Normal}}) = -\underbrace{\left(\frac{6}{7}\right) \log_2 \left(\frac{6}{7}\right)}_{-0.222} - \underbrace{\left(\frac{1}{7}\right) \log_2 \left(\frac{1}{7}\right)}_{-2.907} = 0.591$$

+0.190 +0.401

$$\text{Gain}(S, \text{Humedad}) = 0.940 - \left[\frac{7 \times 0.984}{14} + \frac{7 \times 0.591}{14} \right] = \boxed{0.152}$$

$$\frac{|S_v|}{|S|} \rightarrow \begin{matrix} \# \text{ Humedad} \\ \text{Alta} \\ \# \text{ total} \end{matrix}$$

Particiones for Viento \rightarrow

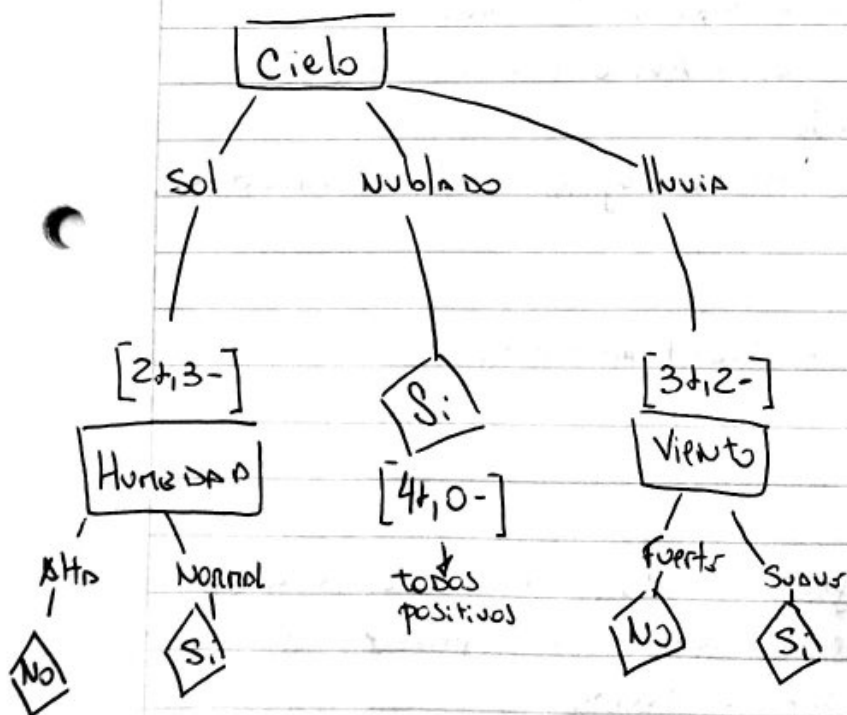
- debil (8/14) \rightarrow Si (6/8) \rightarrow NO (2/8)
- fueras (6/14) \rightarrow Si (3/6) \rightarrow NO (3/6)

$$\text{Entropia}(S_{V=\text{Viento Fuertes}}) = -\underbrace{(3/6) \log_2(3/6)}_{-1} - \underbrace{(3/6) \log_2(3/6)}_{-1} = 1$$

$$\text{Entropia}(S_{V=\text{Viento Debil}}) = -\underbrace{(6/8) \log_2(6/8)}_{-0.415} - \underbrace{(2/8) \log_2(2/8)}_{-0.5} = 0.811$$

$$\text{Gain}(S, \text{Viento}) = 0.940 - \left[\frac{6}{14} \times 1 + \frac{8}{14} \times 0.811 \right] = \boxed{0.048}$$

$$\text{Gain}(S, \text{Cielo}) = \boxed{0.246} \checkmark$$



5 casos

$$\text{Gain}(S_{\text{Sol}}, \text{Humedad}) = \frac{\text{H Alta } 3/5}{\text{H Baja } 2/5}$$

$$\text{Gain}(S_{\text{Sol}}, \text{Humedad}) = \boxed{0.970}$$

$$\text{Gain}(S_{\text{Sol}}, \text{Temp}) = 0.570$$

$$\text{Gain}(S_{\text{Sol}}, \text{Wind}) = 0.19$$

$$\text{Entropia}(S_{\text{Sol}}) = -\underbrace{\frac{3}{5} \log_2(\frac{3}{5})}_{-0.136} - \underbrace{\frac{2}{5} \log_2(\frac{2}{5})}_{-1.321} = 0.97$$

$$= 0.441 + 0.529$$

- Espacio de Hipótesis Completo
- Una sola hipótesis de salida
- No hay backtracking
- Robusto a errores de datos de entrenamiento
- Bias Inductivo
 - ↳ Se prefiere el árbol más corto
 - ↳ con atributos con Information Gain Alto cerca de la Raíz

Navaja de Occam: Se prefiere hipótesis más corta para satisfacer los datos.

Overfitting

- Árbol muy profundo
- La h es mejor sobre entrenamiento pero h' generaliza mejor
 - ↳ Detener crecimiento árbol antes que clasifique perfecto a los datos
 - ↳ Podarlo
 - ↳ Conjunto de Entrenamiento a Validación
 - ↳ Usar todos los datos + aplicar test estadístico para ver si expando o no
 - ↳ MDL: uso medida de complejidad. Se detiene crecimiento

Reduce error Pruning

- ↳ Producir tamaño menor del subárbol más exacto

Rule post Pruning

- Convertir árbol en conjunto de reglas (camino de raíz a nodo hoja)
- Podar cada regla indep de las demás, removiendo precond que mejoran accuracy
- Ordenar las podadas según su accuracy.

C61

Data Entrenamiento \rightarrow Hacer modelo que aproxime f .

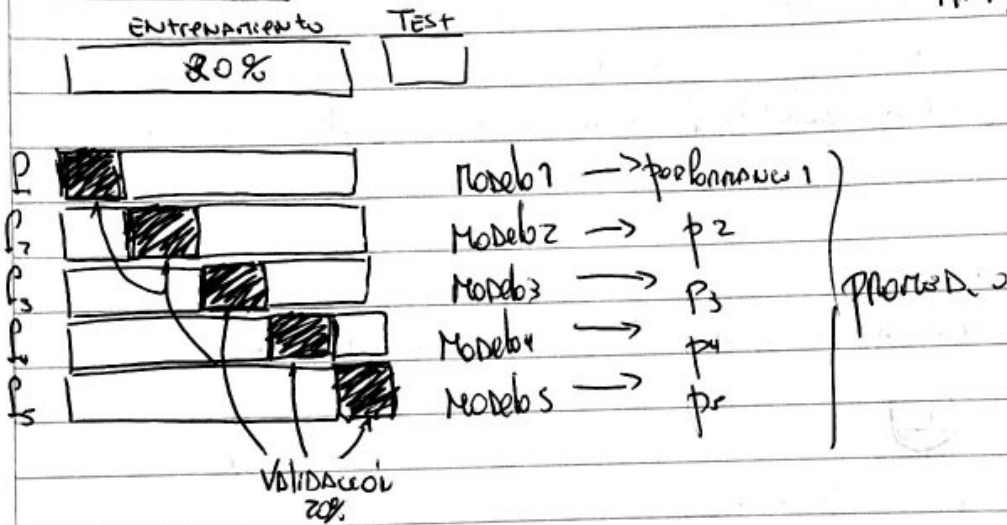
Accuracy Sobre \rightarrow Mala idea; el modelo puede memorizar los datos de entrenamiento y tener eficacia a Hn.

- Medida por forma sobre entrenamiento tiende a sobreestimar los resultados.

Se evalúa el modelo contra Test (Held-out)

Cross-validation (K-Fold con $k=5$)

HAY que Mezclar los Datos.



Comparación entre

\neq Atributos

\neq Algoritmos

\neq Hiperparámetros

- \rightarrow Criterio de Selección de Atributos Nois.
- \rightarrow Criterio de Parámetros
- \rightarrow Estrategias de Búsqueda

Exploramos espacio de búsqueda usando CV K-Fold para medir el desempeño en cada iteración y Al final fuimos con 1 combinación

Random Search

explora a1

AZOR

Grid Search

explora todas

las combinaciones

Matriz De Confusion

| | | Valores Reales | |
|---------------|----------|----------------|----------|
| | | Positivo | Negativo |
| Clasificación | Positivo | TP | FP |
| | Negativo | FN | TN |

TP: true positivos

FP: false positivos

TN: true negativos

FN: false negativos

Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$

No dicen nada sobre los tipos de acierto x de error.

Autentica por voz \rightarrow FP: Autentica a un impostor (mas grave)
 \rightarrow FN: No autentica a un usuario valido

Precision: $\frac{TP}{TP+FP}$

- CUAN utiles son los Resultados de busqueda
- De las clasificadas como positivas, cuantas lo son.
- Test de embarazo.

Recall/Cubrimiento: $\frac{TP}{TP+FN}$

= TPR

- De las instancias positivas, cuantas fueron clasificadas como positivas.
- Cuan completos son los Resultados.
- Dónde clasifican mal las positivas es lo peor: Clasificar + o - de una enfermedad.
- % de usuarios validos autenticados.

Specificity = TNR = $\frac{TN}{TN+FP}$

- % de pacientes sanos correctamente diagnosticados

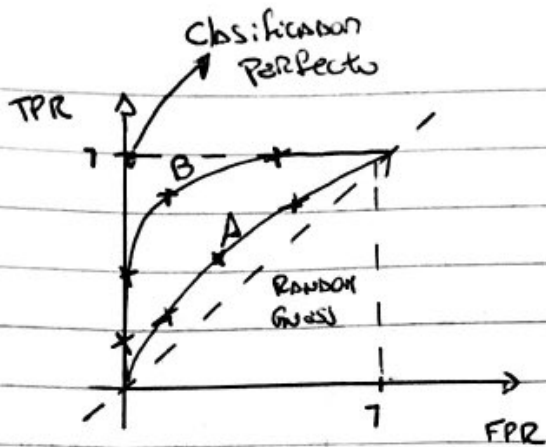
FPR = $\frac{FP}{FP+TN}$

- % de impostores que aceptamos erroneamente.

ROC: TPR vs FPR

(% de usuarios validos autenticados

vs % de impostores que aceptamos erroneamente)



- mas X sobre TPR es porque priorizo de las positivas cuando fueran positivas,
- mas X fuera de TPR, es porque Acepto mas FPR (% de impostores que pasan como validos)

Dados dos Roc, me fue do con la que mas AUC tenga

$$AUC \in (0,1)$$

Varío el umbral de detección entre 0 y 100%. Para cada valor calculo TPR y FPR y obtengo punto de curva.

Factores para la elección

De modelos

- ↳ Tasa de Error
- ↳ Velocidad de entrenamiento y vel test
- ↳ Interpretabilidad
- ↳ Fácil Desarrollo.

Media Armónica:

$$F\text{-Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Cs/

Clasificación: Develva clas mas
Normal Probable

$$P(h|D) = \frac{P(D|h) P(h)}{P(D)}$$

Naive Bayes

- Cualquier problema

De clasificación: Clasificar texto

SPAM

CARPETA PARA MAIL

ANALISIS SENTIMENTAL

TEMA DE UN ARTICULO

etc

- Clasificación probabilística

→ predecir distribución de probabilidades
Sobre un conjunto de clases.

- F se proporcionan probas de ocurrencia de diferentes resultados posibles de un experimento

Texto: "El DRAGON sigue VIVO"

| diccionario | BAG-OF-WORDS |
|-------------|--------------|
| Al | 0 |
| DRAGON | 1 |
| MUERE | 0 |
| VIVO | 1 |
| NO | 0 |

$$X_i = \begin{cases} 1^*, & \text{si palabra esta en diccionario} \\ 0, & \text{si no} \end{cases}$$

palabras son la # de palabras en vez de si esta o no

- Se asume independencia de las probabilidades de los atributos
- Muy usado para clasificar textos
- Muy rapido
- Pocos requerimientos de almacenamiento
- busca la h mas probable dado el conjunto de datos D.

$$C_{\text{map}} = \underset{c \in C}{\text{argmax}} P(d|c) P(c) = \underset{c \in C}{\text{argmax}} P(x_1, x_2, \dots, x_n | c) P(c)$$