APRENDIZAJE AUTOMÁTICO

Cualquier sistema que se considere "inteligente" debería poseer la habilidad de aprender, es decir, *mejorar automáticamente con la experiencia*. La idea principal detrás del aprendizaje, es que aquello que es percibido por el agente, debería ser usado no sólo para actuar, sino también para mejorar sus habilidades en el futuro. El proceso de aprendizaje de un agente puede resultar de su interacción con el mundo como así también de la observación introspectiva de sus propios procesos internos.

El aprendizaje de máquina o automático (en inglés Machine Learning (ML)), se ha tornado muy importante en muchas aplicaciones prácticas de Inteligencia Artificial, como por ejemplo los Sistemas Expertos, proveyendo una alternativa a las técnicas de adquisición de conocimiento tradicionales.

Los principales aspectos a ser considerados en el proceso de aprendizaje, se encuentran implícitos en la definición de aprendizaje de Herbert Simon: "*cualquier cambio en un sistema que le permite desempe˜narse mejor la próxima vez, sobre la misma tarea u otra tomada de la misma población*".

Esta definición cubre un amplio espectro de actividades que van desde mejorar el rendimiento de un sistema existente (ya sea en eficiencia o en la no reiteración de errores) hasta la adquisición de nuevos conceptos. También esta definición habla de cambios en el agente que aprende, lo que implica que deberá haber alguna manera de representarlos. En este sentido, serán válidos tanto los métodos que modelizan el aprendizaje como la adquisición de conocimiento del dominio representado explícitamente (mediante sentencias en un lenguaje simbólico), como aquellos cuyo conocimiento está implícito (como por ejemplo redes neuronales) y que aprenden modificando su estructura completa (la organización e interacción entre las neuronas).

El Aprendizaje Automático se enfrenta con el desafío de la construcción de programas computacionales que automáticamente mejoren con la experiencia. Estos programas computacionales son sistemas de aprendizaje capaces de adquirir conocimientos de alto nivel y/o estrategias para la resolución de problemas mediante ejemplos, en forma análoga a la mente humana. A partir de los ejemplos provistos por un tutor o instructor y de los conocimientos de base o conocimientos previos, el sistema de aprendizaje crea descripciones generales de conceptos.

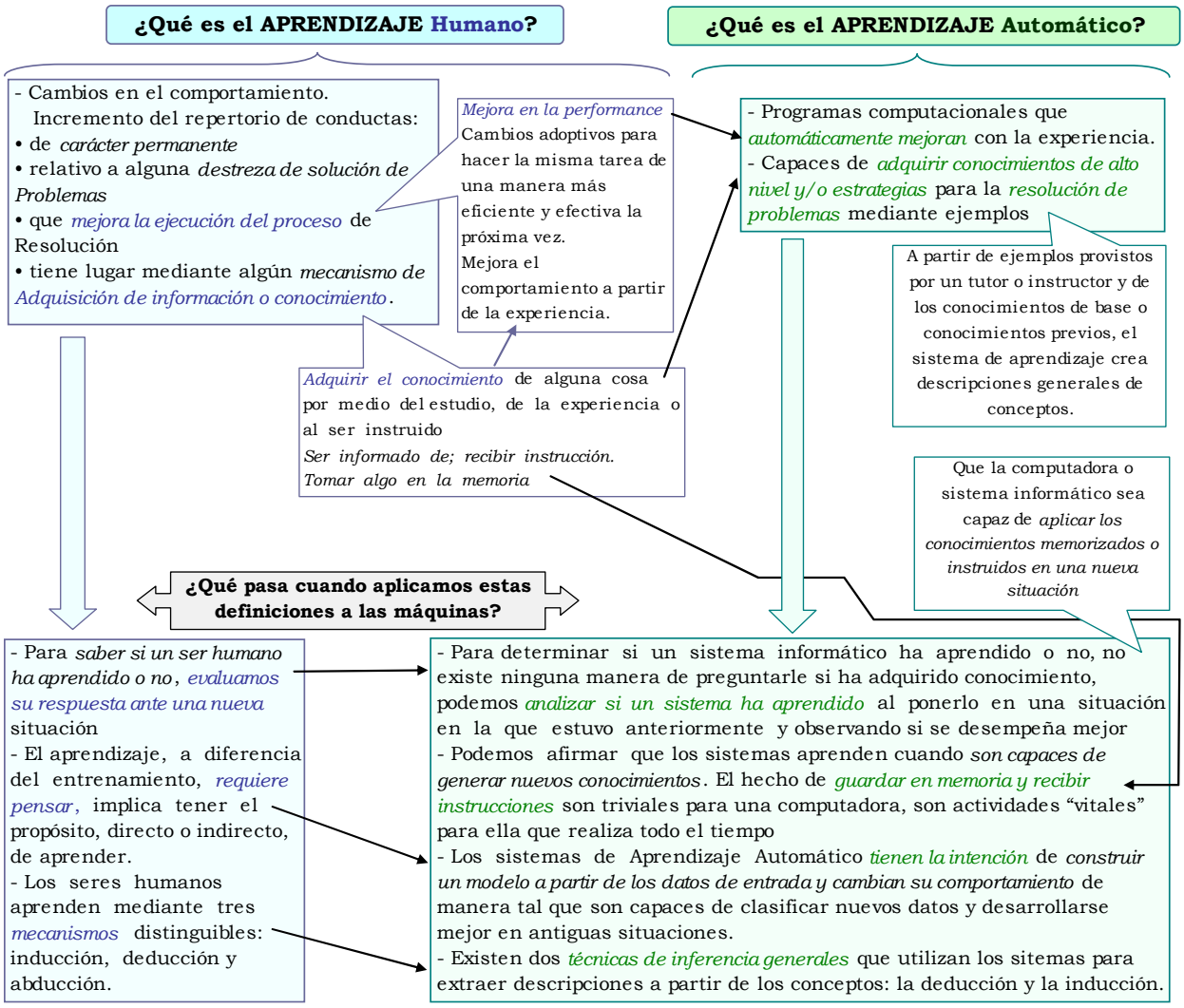
Definición del Aprendizaje Automático.

El Aprendizaje Automático es el campo de la Ingeniería Informática en el que se estudian y desarrollan algoritmos que implementan los distintos modelos de aprendizaje y su aplicación a la resolución de problemas prácticos. Entre los problemas abordados en este campo, está el de inducir conocimientos a partir de datos o ejemplos.

Puede entenderse, de forma general, como el *proceso mediante el cual se consigue que un sistema no humano sintetice el conocimiento preciso que le permite realizar una serie de tareas o labores*, sin que sea necesario indicarle cómo debe hacerlas; bastará con *suministrarle descripciones de las actuación que un experto o grupo de expertos han realizado en el pasado para resolver esa misma tarea*. Cada una de esas actuaciones constituirá para el sistema los *modelos de comportamiento* que trata de clonar.

Un sistema de Aprendizaje Automático está formado por una serie de procesos que le permiten aprender a realizar determinadas tareas. Para ello extraerá el conocimiento necesario mediante la observación de las actuaciones de los expertos en la materia. La solución sintetizada por el sistema puede servir, bien para construir un asistente inteligente y amigable o bien para que forme parte de un sistema físico más complejo en el que el sistema de aprendizaje decidirá las acciones a realizar y los sistemas físicos las ejecutarán.

"Se puede afirmar que un programa computacional es capaz de aprender a partir de la experiencia E con respecto a un grupo de tareas T y según la medida de performance P, si su performance en las tareas T, medida según P, mejora con la experiencia E."



El tema fundamental para construir un sistema de aprendizaje automático es, según Mitchell, *plantear el problema de aprendizaje de manera correcta*. Para ello, debe contar con las tres partes esenciales de la siguiente definición:

"Se puede afirmar que un programa computacional es capaz de aprender a partir de la experiencia *E* con respecto a un grupo de tareas *T* y según la medida de performance *P*, si su performance en las tareas *T*, medida según *P*, mejora con la experiencia *E*."

Como se puede observar, para tener un *problema de aprendizaje bien definido*, debemos identificar estas 3 características: la clase de tareas (T), la medida de perfomance (P) y la fuente de experiencia (E).

Ejemplo:

Tarea (T): *Reconocer y clasificar palabras escritas a mano que se encuentran dentro de imágenes.*
Medida de perfomance (P): *Porcentaje de palabras clasificadas correctamente.*
Experiencia de entrenamiento (E) : *Una base de datos de palabras escritas a mano con sus correspondientes clasificaciones.*

Tarea (T): *Jugar a las damas.*
Medida de perfomance (P): *Porcentaje de juegos ganados en el torneo mundial.*
Experiencia de entrenamiento (E) : *Juegos jugados contra él mismo.*

Con respecto a la *medida de performance*, podemos en general reconocer al menos tres aspectos que pueden ser medidos como una mejora lograda mediante el aprendizaje:

- *Precisión*: ¿el agente realiza una tarea en forma más precisa, es decir, hace las cosas mejor?
- *Velocidad*: ¿el agente realiza la tarea más rápidamente?
- *Descubrimiento*: ¿El agente adquiere nuevas habilidades o comportamientos?

Cuando se *diseña un sistema de aprendizaje*, hay una serie de decisiones que se deben tomar en las distintas etapas:

1. ¿Qué tipo de experiencia de entrenamiento usaré para que el sistema aprenda?
2. ¿Qué es exactamente lo que debe ser aprendido?
3. ¿Cómo se debe representar?
4. ¿Qué algoritmo de aprendizaje debería usar?

Los sistemas de aprendizaje se clasifican en dos categorías generales: ***métodos de caja negra*** **y** ***métodos orientados al conocimiento***. Los primeros desarrollan su *propia representación de conceptos*, que por lo general no es comprensible para las humanos; normalmente, realizan cálculos numéricos de coeficientes, distancias o vectores. Entre estos métodos, se encuentran las redes neuronales y los métodos estadístico-matemáticos. Por otro lado, los métodos orientados al conocimiento tratan de *crear estructuras simbólicas de conocimiento que sean comprensibles para el usuario*. El Aprendizaje Automático pertenece al segundo grupo de métodos.

Ejemplos: Una tarea de aprendizaje

- Pronóstico el grado de aptitud de parcelas de tierras para el cultivo de caña de azúcar

El proceso de evaluación de tierras permite realizar estudios con el fin de valorar si el uso dado a una unidad agrícola es el más adecuado, apoyándose para tales decisiones en factores agro climáticos que inciden en su comportamiento. La importancia que tiene conocer la aptitud física de las tierras es que posibilita realizar un uso correcto de las mismas. De manera que si no es recomendable su uso para este cultivo pudiera emplearse para otras actividades, evitándose así la degradación del suelo en las áreas no aptas y a la vez concentrar los recursos en aquellas parcelas con las condiciones más adecuadas para alcanzar los mayores rendimientos en caña de azúcar.

Tarea: *Determinar la aptitud física de las tierras para el cultivo de la caña de azúcar*.

Experimentos: *Mediciones de factores asociados al suelo, al clima y características agrícolas (pendiente del terreno, pedregosidad, rocosidad, salinidad, acidez del suelo, capacidad de intercambio catiónico, drenaje, compactación, precipitaciones, profundidad efectiva, agrupamiento agroproductivo del suelo y categoría de aptitud de la tierra); realizadas sobre una serie de parcelas cultivadas con caña de azúcar.*

Performance: *Proporción de tierras correctamente clasificadas como: áreas sumamente aptas, moderadamente aptas, marginalmente aptas y áreas no aptas para el cultivo de la caña de azúcar.*

- Detección precoz de insolvencias. La predicción de la insolvencia es uno de los temas centrales del análisis financiero que ha suscitado el interés no sólo del ámbito académico sino también de un amplio abanico de usuarios relacionados con el mundo empresarial.
  T: *predecir crisis*
  E: *los ratios contables*
  P: *porcentaje de predicción de crisis verdaderas.*

- Aprender jugando contra si mismo a las Damas.
  T: *jugar a las damas*,
  E: *juegos jugados contra si mismo.*
  P: *porcentaje de partidas ganadas*

- Identificación de tipos de tumores a partir de datos moleculares (microarrays)
  T: *Clasificar tumores*
  E: *nivel de activación de genes*
  P: *porcentaje de tumores correctamente identificados*

- Problema de control sobre un avión F16
  T: *predecir la acción de control realizada sobre los alerones*
  E: *Registros que describen estados del avión*
  P: *porcentaje de acciones predecías, correctas*

1. Esquema general de un sistema de Aprendizaje Automático

El esquema general de un sistema de Aprendizaje Automático se detalla en la Figura 1. Vemos que el sistema recibe dos grupos de entradas: *los ejemplos* y *los conocimientos previos* y que genera una *descripción de conceptos* como salida. Los ejemplos instancian un cierto concepto, lo describen mediante distintos valores de sus atributos.

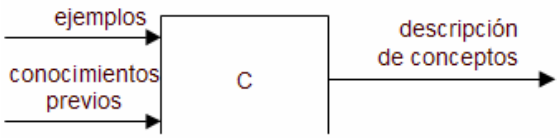


Figura 1 Características generales de un sistema de Machine Learning

Los *conocimientos previos* contienen la información acerca del lenguaje utilizado para describir los ejemplos y los conceptos, es decir, son una suerte de metalenguaje. El sistema utiliza entonces, los conocimientos previos para interpretar los ejemplos y para generar descripciones a partir de ellos.

2. Aprendizaje de conceptos

El Aprendizaje Automático trata de extraer conceptos de los datos que recibe como entrada. Por concepto se entiende una *abstracción para un conjunto de objetos que comparten ciertas propiedades* que los diferencian de otros conceptos .

Los límites entre conceptos no están claramente definidos en todos los casos, y aún en los casos en los cuales los límites están claros, puede no ser fácil clasificar un ejemplo en particular. Por ejemplo, cuál es el límite entre un perro grande y un perro chico. Si decimos que los dobermans son perros grandes, ¿cómo clasificamos a un doberman que por causas naturales es enano? ¿Qué pasa con la "excepción que prueba la regla"?

Existen dos ***técnicas de inferencia generales*** que se utilizan para extraer descripciones a partir de los conceptos: la ***deducción*** y la ***inducción***.

- La deducción es la técnica que *infiere información como una consecuencia lógica de los ejemplos* y *conocimientos de base*.
- La inducción es la técnica que *infiere información generalizada de los ejemplos* y *conocimientos de base*.

En la inducción, podemos trabajar con *jerarquías de generalización*, representadas por árboles o grafos. En una jerarquía de generalización, un concepto puede describirse por los objetos del nivel base o por cualquier objeto en un nivel superior. Analizando la Figura 2, podemos describir a la *Universidad Tecnológica Nacional* los objetos de nivel base, en cuyo caso decimos que es una institución educativa, universitaria y pública; o podemos describirla usando los objetos de nivel superior diciendo que la *Universidad Tecnológica Nacional* es una institución educativa.

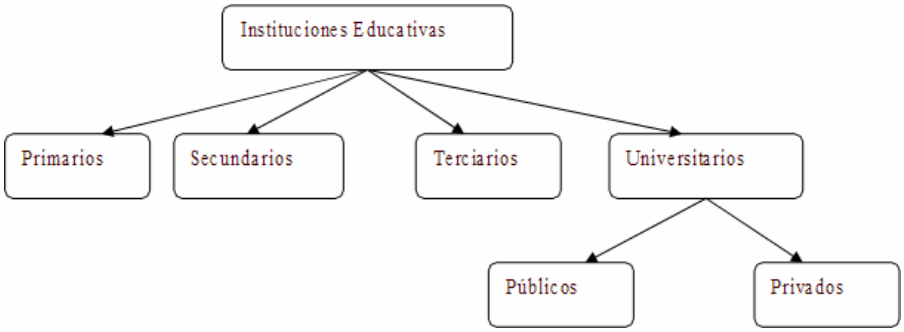


Figura 2 Jerarquía de generalización

En este tipo de jerarquías, podemos identificar tres nociones que relacionan los conceptos: *efecto de nivel básico* (basic-level effect), *tipicalidad* (typicality) y *dependencia contextual* (contextual dependency).

- El *efecto de nivel básico* hace referencia al hecho de que los conceptos de nivel base pueden ser descriptos por características fácilmente identificables por los humanos, lo cual hace que su aprendizaje sea simple para nosotros. Mientras que los conceptos de nivel superior se definen como grupos de conceptos de nivel básico que comparten alguna característica en común.

- La segunda noción, la *tipicalidad*, analiza cuán típico es un concepto. Puede medirse de acuerdo a la cantidad de características comunes que comparte con otros conceptos, y a la cantidad de características heredadas de los superconceptos (conceptos de nivel superior). En el aprendizaje, la tipicalidad es muy importante, por ejemplo, tratar de enseñar el concepto de pájaro con los ejemplos de un pingüino, un ganso y un avestruz, no será muy efectivo. En cambio, utilizar una golondrina, un gorrión y una paloma será exitoso.
- Por último, la *dependencia contextual* es importante porque los conceptos aprendidos dependen del contexto en el que estamos. Al definir estudiantes podemos estar pensando en estudiantes universitarios, estudiantes primarios, o estudiantes del curso de Análisis Matemático; el concepto que estamos tratando de enseñar, dependerá del contexto en el que estamos.

2.1. Representación de conceptos

La primera cuestión que debe solucionar el Aprendizaje Automático al encarar el problema de aprendizaje, es cómo representar los conceptos, es decir, *cómo llevar los ejemplos, conocimientos base y conceptos obtenidos a un lenguaje entendible por una computadora*, que sea también, fácilmente interpretable por el usuario (recordemos que los métodos de Aprendizaje Automático son orientados al conocimiento y no cajas negras).

Algunos métodos que el Aprendizaje Automático utiliza para representar los datos, en orden ascendente en cuanto a complejidad y capacidad expresiva, son: *Lógica de orden cero* (lógica proposicional), *lógica de atributos*, *lógica de predicados de primer orden* y *lógica de segundo orden*, entre otros.

En la lógica de orden cero, los ejemplos y conceptos se describen como conjunciones de constantes booleanas que representan valores de los atributos. El poder descriptivo de este tipo de lógica es bajo, por lo cual, el Aprendizaje Automático lo utiliza únicamente para describir conceptos muy simples. Un ejemplo de una cláusula en lógica de orden cero es: Juego_Tenis <= Día_soleado ^ No_hay_viento ^ Poca_humedad

Para solucionar el problema del bajo poder descriptivo de la lógica de orden cero, puede utilizarse la lógica de atributos. La idea básica detrás de la lógica de atributos es *caracterizar los ejemplos y conceptos como valores de algunos atributos predefinidos*. En lugar de utilizar conjunciones de valores fijos, cada atributo es una variable. El poder descriptivo de la lógica de atributos es mayor que el de la lógica de orden cero, aunque en sentido matemático la expresividad es la misma. Los ejemplos generalmente se presentan en una tabla donde cada fila representa un ejemplo y cada columna, un atributo. La tabla 1 contiene ejemplos positivos y negativos para los días en que es posible jugar al tenis.

Objeto	Pronóstico	Viento	Humedad	Juego_Tenis
Dia 1	Sol	No	Poca	Si
Dia 2	Lluvia	Ráfagas	Poca	No
Dia 3	Nublado	Ventoso	Poca	No
Dia 4	Sol	Ventoso	Alta	Si
Dia 5	Nublado	No	Alta	No

Tabla 1 Ejemplos positivos y negativos del concepto Juego_Tenis <= Pronóstico ^ Viento ^ Humedad

Como lenguaje descriptivo, la lógica de atributos es mucho más práctica que la lógica de orden cero. Por eso, es utilizada en muchos programas de Aprendizaje Automático, como los de la familia TDIDT (Árboles inductivos de arriba hacia abajo - *Top-Down Induction Trees)*.

La lógica de predicados de primer orden utiliza las cláusulas de Horn para representar conceptos. Estas cláusulas simplifican las descripciones complicadas mediante el uso de predicados y variables. Son bastante potentes, incluso permiten la expresión de conceptos recursivos. El lenguaje Prolog se basa en la lógica de predicados de primer orden. Este tipo de lógica se utiliza en algunos programas de Aprendizaje Automático, como el algoritmo FOIL.

Un ejemplo de una cláusula de Horn sería: Abuelo(X,Z) :- Padre(X,Y), Padre(Y,Z)

Por último, la lógica de predicados de segundo orden considera a los nombres de los predicados como variables. La expresión anterior quedaría de la forma: p(X,Z) :- q(X,Y), q(Y,Z) donde p es Abuelo y q es Padre.

Este tipo de lógica es la de mayor poder descriptivo. Sin embargo, dada su complejidad rara vez se utiliza en los sistemas de Aprendizaje Automático.

2.1.1. Presentación de los resultados

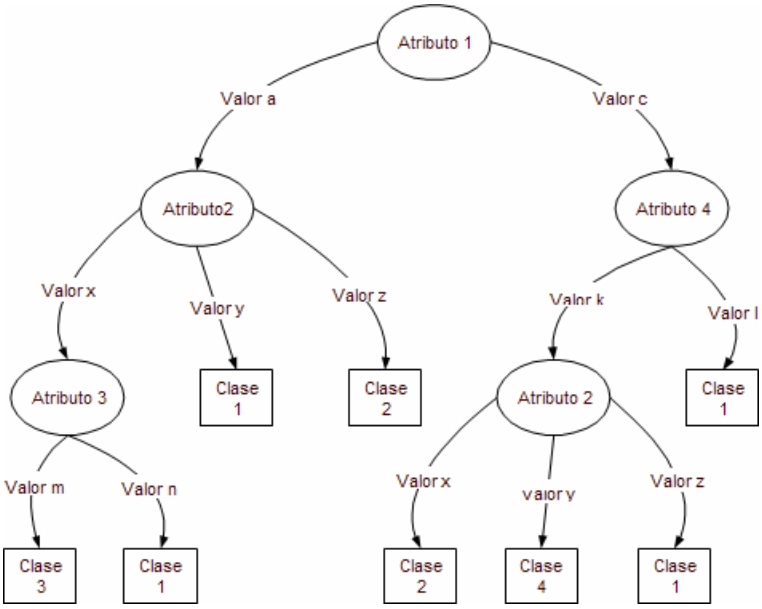
Los resultados de un sistema de Aprendizaje Automático son los conceptos aprendidos. Como ya se explicó, para que los sistemas sean de caja transparente, estos *resultados deben ser comprensibles fácilmente*. Hay varias técnicas que permiten modelar los conceptos aprendidos, a continuación se describen algunas de ellas.

Tablas de Decisión. {atrib1, atrib2, …, atribN, clase}

El problema está en decidir cuál atributo descartar para simplificar la tabla sin afectar el resultado.

Árboles de Decisión. En cada nodo de un árbol de decisión se evalúa un atributo.

Existe una rama por valor del atributo cuando los atributos son discretos, y una rama por rango de valores cuando los atributos son continuos. Al clasificar un nuevo ejemplo, se evalúa el valor del atributo indicado por el nodo actual y se recorre el árbol por la rama de dicho valor.



Cualquier método de aprendizaje que utiliza el método "divide y reinarás" obtiene naturalmente un árbol de decisión.

Reglas de Clasificación Antecedente => Consecuente

Son una alternativa a los árboles de decisión, y todo árbol de decisión puede llevarse a reglas de este tipo.

Ej: Si atributo1="valor a" y atributo2= "valor y", entonces Clase 1

Agregar una nueva regla implica simplemente añadirla a la lista de reglas sin necesidad de hacer cambios de estructura, mientras que agregar una nueva regla en un árbol implicaría rehacer la estructura del mismo.

El orden en que las reglas se interpretan es importante, determina cuáles reglas deben ejecutarse primero, y se avanza en la lista si el ejemplo no cumple con el antecedente de la regla actual.

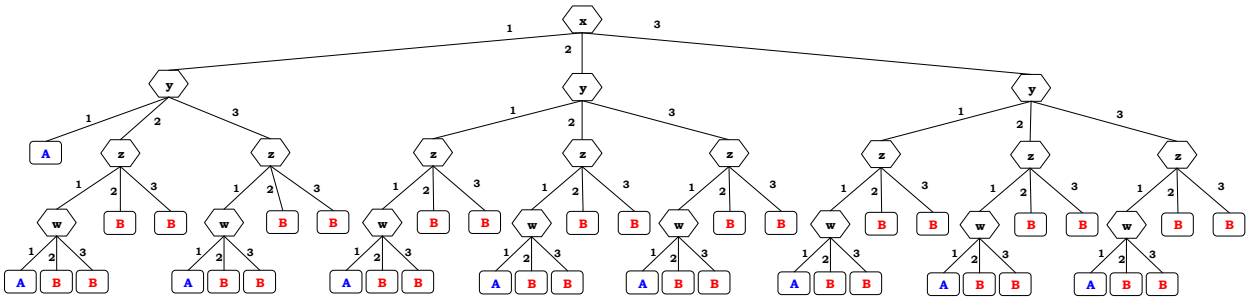
Ejemplo Árbol de decisión para las siguientes reglas:

Si x=1 y y=1 => A

Si z=1 y w=1 => A Sino

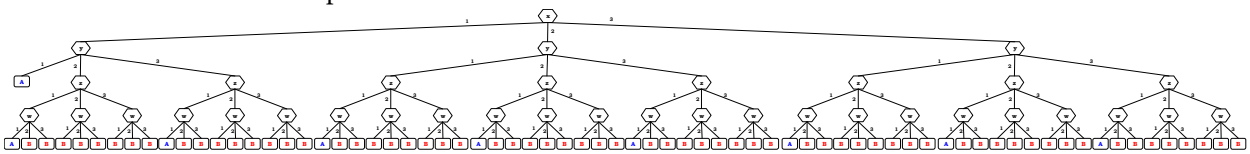
donde los posibles valores de x,y,z,w son {1,2,3}, y las clases posibles son {A,B}

Analizar qué ventajas tiene trabajar con un árbol o con un conjunto de reglas de decisión cuando tenemos una clase por defecto.



Usar Clases por defecto simplifica el problema, sin dejar de contemplar todos las circunstancias. En el caso del árbol permite reducir su amplitud y evitar que sea muy frondoso incorporando reglas irrelevantes. En caso de las reglas permite que solo se evalúen aquellas condiciones relevantes e incorpora el resto de las posibles situaciones en caso de que no se den estas circunstancias.

El árbol sin usar la clase por defecto sería:

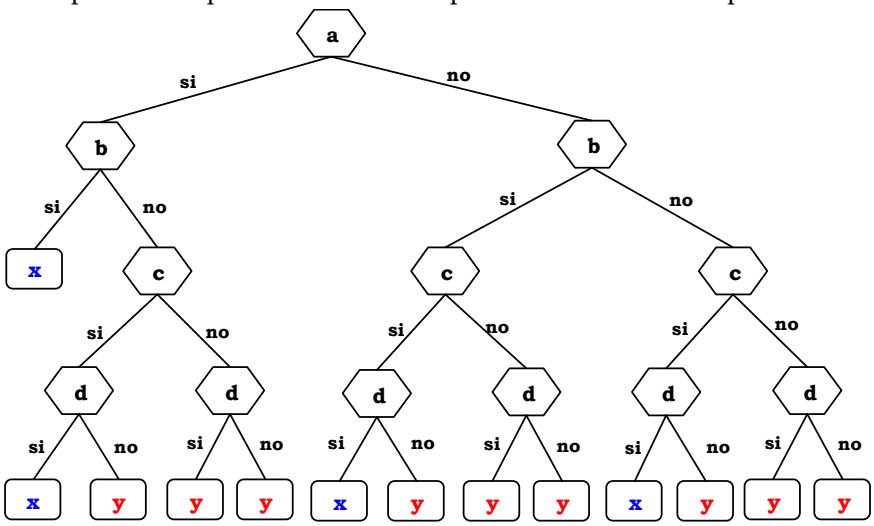


Bastante amplio y difícil de interpretar

Si a y b => x

Si c y d => x donde las clases posibles son {x,y}

El árbol resultante presenta lo que se conoce como el problema del subárbol replicado.



Las reglas pueden expresar disyunciones de manera más fácil que los árboles. Por lo mismo, el extraer reglas directamente de árboles tiende a producir reglas mías complejas de lo necesario. Los árboles tienen lo que se conoce como problema del subárbol replicado (replicated subtree problem), ya que a veces repiten subárboles en varios lados.

Reglas de asociación

Estas reglas no difieren en gran medida de las reglas de clasificación. En lugar de basarse en un atributo en particular (la clase), las reglas de asociación buscan todas las relaciones existentes entre todos los atributos.

La *cobertura* de una regla de asociación es el número de instancias para cual la predicción es correcta. La *precisión* o *confidencia* es el número de instancias que predice correctamente expresado con respecto a todas las instancias a las cuales se aplica.

Reglas de relaciones

Los tipos de relaciones analizados anteriormente son *proposicionales* ya que pertenecen a la lógica proposicional (comparan un valor contra una constante). A veces, esto no alcanza para expresar un concepto y es necesario utilizar la *lógica relacional* que compara los atributos entre sí.

Ej: tenemos un bloque =>

Si ancho>alto entonces acostado

Si alto>ancho entonces parado

Aprendizaje basado en instancias

Se basa en el método más simple de aprendizaje, la memorización. En este caso no se crea ningún modelo, sino que se guardan las instancias en memoria. Entonces, cada vez que se quiere aplicar lo aprendido se realiza una búsqueda entre todas las instancias guardadas. Las instancias en sí representan los conceptos aprendidos, esto implica que el sistema realiza gran parte del trabajo a la hora de aplicar lo aprendido y no antes como en los otros tipos de aprendizaje.

Para determinar qué instancia utilizar en cada caso se utiliza el "método del k-vecino más cercano". Este método determina una medida de distancia, gracias a la cual a cada nueva instancia se le asigna la clase de la instancia más cercana, o la de la mayoría de los K vecinos más cercanos

Clusters

Cuando se aprenden clusters en lugar de un clasificador, el resultado toma la forma de un diagrama que muestra cómo las instancias caen en los clusters. Este diagrama puede ser una tabla tan simple como

{instancia, n°cluster}

Algunos algoritmos de clustering permiten que una instancia pertenezca a más de un cluster, entonces puede representarse como un diagrama de Venn con intersecciones.

Generalmente, al clustering le sigue una etapa de construcción de un árbol o de reglas de decisión que modelizan lo aprendido.

2.2. Ejemplos de sistemas de Aprendizaje Automático

Incremento de la producción en procesos de control químicos:

Este sistema, que se desarrolló en 1984, utiliza métodos estadísticos para predecir la calidad de los cartuchos de Uranio generados a partir del gas. El sistema genera reglas interactuando con el experto y como resultado obtiene una mejor producción.

Decisiones de análisis de créditos:

El sistema fue desarrollado en el Reino Unido para ayudar a los expertos a tomar decisiones acerca de los créditos que deben otorgar. En especial, se buscó reducir las pérdidas debido al análisis erróneo de los casos confusos. El sistema construye árboles de decisión para predecir si los aspirantes fallarán o no a la hora de pagar el crédito. Se trabajó sobre 1014 casos de la base de datos de la compañía y se obtuvo un alto nivel de precisión a la hora de evaluar a los aspirantes, incluso más alto que el nivel obtenido por los expertos humanos. Esto llevó a que American Express del Reino Unido adoptara el sistema.

Diagnóstico de dispositivos médicos:

Se desarrolló un sistema de mantenimiento preventivo de equipos en una gran planta química, buscando la predicción del tipo de falla a punto de ocurrir. El sistema realiza un análisis de Fourier sobre las vibraciones que recibe de los equipos. Las reglas obtenidas son más precisas que las generadas por los expertos de campo.

Clasificación Automática de objetos celestes:

A partir de una base de datos sobre los objetos celestes que ya habían sido clasificados por los expertos, se desarrolló un sistema que genera árboles de decisión para distinguir entre estrellas y galaxias. Se obtuvo una precisión superior en un 94% a la esperada, y hoy en día el sistema se utiliza embebido en la base de datos para catalogar los objetos en estudio.

3. Aprendizaje supervisado y no supervisado

Existen dos tipos de aprendizaje: el ***supervisado*** y el ***no supervisado***.

- En el aprendizaje supervisado o aprendizaje a partir de ejemplos, el *instructor o experto define clases y provee ejemplos de cada una*.

  El sistema debe obtener una descripción para cada clase. Cuando el instructor define una única clase, provee ejemplos positivos (pertenecen a la clase) y negativos (no pertenecen a la clase). En este caso, los ejemplos importantes son los cercanos al límite, porque proveen información útil sobre los límites de la clase. Cuando el instructor define varias clases, el sistema puede optar por realizar *descripciones discriminantes o no*. Un conjunto de descripciones es discriminante si el total de las descripciones cubren todas las clases, pero una sola descripción cubre una clase en particular.

- En el aprendizaje no supervisado o aprendizaje a partir de observaciones y descubrimientos, el sistema debe *agrupar los conceptos1 sin ayuda alguna de un instructor*.

  El sistema recibe los ejemplos, pero no se predefine ninguna clase. Por lo tanto, debe observar los ejemplos y *buscar características en común que permitan formar grupos*. Como resultado, este tipo de aprendizaje genera un *conjunto de descripciones de clases*, que juntas cubren todas las clases y en particular describen a una única clase.

4. Tipos de aprendizaje automático

Existen varios tipos de aprendizaje que pueden clasificarse como supervisados o no supervisados. Acontinuación, se presentan los distintos tipos de aprendizaje automático[García Martínez, 1997].

- Aprendizaje por memorización
- Aprendizaje por instrucción
- Aprendizaje por deducción
- Aprendizaje por analogía
- Aprendizaje por inducción
- Aprendizaje por ejemplos
- Aprendizaje por observación - descubrimiento
- Observación pasiva
- Experimentación activa

En el *aprendizaje por memorización* los sistemas reciben conocimientos del medio ambiente y los guardan sin ningún tipo de procesamiento. Su complejidad se encuentra en el almacenamiento de los conocimientos y no en su adquisición. Lo importante en estos casos es que la información esté disponible cuando se requiera; no hay ningún tipo de inferencia ni procesamiento, por lo tanto, los conocimientos deben ser adquiridos y almacenados en un nivel que los haga directamente utilizables.

En el caso del *aprendizaje por instrucción,* los conocimientos son provistos por un instructor o experto en la materia (aprendizaje supervisado). La información provista es abstracta o de índole general, por lo tanto, el sistema tendrá que inferir los detalles. Es decir, el sistema deberá transformar la información provista en términos abstractos de alto nivel, a reglas que puedan ser utilizadas directamente en la tarea del sistema.

El *aprendizaje por deducción* o *aprendizaje guiado por la especificación* destaca o especifica las relaciones existentes entre conceptos. El sistema transforma las especificaciones recibidas como entrada en un algoritmo que actualiza relaciones.

En el *aprendizaje por analogía*, el sistema, que recibe información relevante a problemas análogos a los que está tratando de resolver, debe descubrir las analogías e inferir reglas aplicables al problema. Se trata de generar nuevos conocimientos utilizando información preexistente

En el *aprendizaje por inducción,* el sistema genera nuevos conocimientos que no están presentes en forma implícita dentro del conocimiento disponible. El aprendizaje por inducción abarca el aprendizaje por ejemplos y el aprendizaje por observación y descubrimiento.

En el *aprendizaje por ejemplos,* el sistema recibe varios ejemplos como entrada y debe generalizarlos en un proceso inductivo para presentarlos como salida. Generalmente, en este tipo de aprendizaje existen dos tipos de ejemplos, los positivos y los negativos. Los ejemplos positivos fuerzan la generalización, mientras que los ejemplos negativos previenen que esta sea excesiva. Se trata de que el conocimiento adquirido cubra todos los ejemplos positivos y ningún ejemplo negativo.

A este tipo de aprendizaje pertenece la familia TDIDT. Debe tenerse en cuenta, que los ejemplos a partir de los cuales aprende el sistema, deben ser representativos de los conceptos que se está tratando de enseñar. Además, la distribución de las clases en el conjunto de ejemplos de entrenamiento, a partir de los que el sistema aprende, debe ser similar a la distribución existente en los datos sobre los cuales se aplicará el sistema.

En el *aprendizaje por observación y descubrimiento*, el sistema forma teorías o criterios de clasificación en jerarquías taxonómicas, a partir de la inducción realizando tareas de descubrimiento. Pertenece al tipo de aprendizaje no supervisado y, como tal, permite que el sistema clasifique la información de entrada para formar conceptos.

En cuanto a la interacción con el entorno, existen dos tipos de sistemas: en aquellos que realizan *observación pasiva*, el sistema clasifica las observaciones de múltiples puntos del medio; y en aquellos que realizan *observación activa*, el sistema interactúa con el entorno, realiza cambios en el mismo, y luego observa los resultados.

MÉTODOS CLÁSICOS DE APRENDIZAJE

Existen dos métodos clásicos de aprendizaje inductivo a partir de ejemplos que debemos conocer: el aprendizaje AQ y el aprendizaje según el método de divide y reinarás.

1. Aprendizaje AQ

El aprendizaje AQ se basa en la idea de cubrir progresivamente los datos de entrenamiento a medida que se generan reglas de decisión. Su esencia está en la búsqueda de un *conjunto de reglas (conjunciones de pares atributo-valor o predicados arbitrarios) que cubran todos los ejemplos positivos y ningún ejemplo negativo*. En lugar de dividir los ejemplos en subconjuntos, el aprendizaje AQ generaliza, paso a paso, las descripciones de los ejemplos positivos seleccionados.

El aprendizaje "divide y reinarás" particiona el conjunto de ejemplos en subconjuntos sobre los cuales se puede trabajar con mayor facilidad. En la lógica proposicional, por ejemplo, se parte el conjunto de acuerdo a los valores de un atributo en particular, entonces, todos los miembros de un subconjunto tendrán un mismo valor para dicho atributo. Dentro de este tipo de aprendizaje, encontramos la familia TDIDT (*Top-Down Induction Trees)*.

2. Algoritmos de Clasificación (Classification Algorithms)

En la *Clasificación de Datos se desarrolla una descripción o modelo para cada una de las clases presentes en la base de datos*. Existen muchos métodos de clasificación como aquellos basados en los árboles de decisión TDIDT como el ID3 y el C4.5, los métodos estadísticos, las redes neuronales, y los conjuntos difusos, entre otros.

A continuación se describen brevemente aquellos métodos de Aprendizaje Automático que han sido aplicados a la Minería de Datos con cierto éxito:

- **Algoritmos estadísticos:** Muchos algoritmos estadísticos han sido utilizados por los analistas para detectar patrones inusuales en los datos y explicar dichos patrones mediante la utilización de modelos estadísticos, como, por ejemplo, los modelos lineales. Estos métodos se han ganado su lugar y seguirán siendo utilizados en los años venideros.
- **Redes Neuronales:** las redes neuronales imitan la capacidad de la mente humana para encontrar patrones. Han sido aplicadas con éxito en aplicaciones que trabajan sobre la clasificación de los datos.
- **Algoritmos genéticos:** técnicas de optimización que utilizan procesos como el entrecruzamiento genético, la mutación y la selección natural en un diseño basado en los conceptos de la evolución natural.
- **Método del vecino más cercano:** es una técnica que clasifica cada registro de un conjunto de datos en base a la combinación de las clases de los k registros más similares. Generalmente se utiliza en bases de datos históricas.
- **Reglas de inducción:** la extracción de reglas si-entonces a partir de datos de importancia estadística.
- **Visualización de los datos:** la interpretación visual de las relaciones entre datos multidimensionales
- **Clasificadores basados en instancias o ejemplos:** Una manera de clasificar un caso es a partir de un caso similar cuya clase es conocida, y predecir que el caso pertenecerá a esa misma clase. Esta filosofía es la base para los sistemas basados en instancias, que clasifican nuevos casos refiriéndose a casos similares recordados. Un clasificador basado en instancias necesita teorías simbólicas. Los problemas centrales de este tipo de sistemas se pueden resumir en tres preguntas: ¿cuáles casos de entrenamiento deben ser recordados?, ¿cómo puede medirse la similitud entre los casos?, y ¿cómo debe relacionarse el nuevo caso a los casos recordados?. Los métodos de aprendizaje basados en reglas de clasificación buscan obtener reglas o árboles de decisión que particionen un grupo de datos en clases predefinidas. Para cualquier dominio real, el espacio de datos es demasiado grande como para realizar una búsqueda exhaustiva en el mismo. En cuanto a los métodos inductivos, la elección del atributo para cada uno de los nodos se basa en la ganancia de entropía generada por cada uno de los atributos. Una vez que se ha recopilado la información acerca de la distribución de todas las clases, la ganancia en la entropía se calcula utilizando la teoría de la información o bien el índice de Gini [Joshi, 1997].
- **Algoritmos de reglas de asociación**

  Una regla de asociación es una regla que implica ciertas relaciones de asociación entre distintos objetos de una base de datos, como puede ser: *"ocurren juntos"* o *"uno implica lo otro"*. Dado un conjunto de transacciones, donde cada transacción es un conjunto de ítems, una regla de asociación es una expresión de la forma XY, donde X e Y son conjuntos de ítems. Un ejemplo de regla de asociación sería: "30% de las transacciones que contienen niños, también contienen pañales; 2% de las transacciones contienen ambas cosas". En este caso el 30% es el nivel de confianza de la regla y 2% es la cantidad de casos que respaldan la regla. La cuestión está en encontrar todas las reglas de asociación que satisfagan los requerimientos de confianza mínima y máxima impuestos por el usuario.

3. Análisis de Secuencias

En este caso se trabaja sobre datos que tienen una cierta secuencia entre sí. Cada dato es una lista ordenada de transacciones (o ítems). Generalmente, existe un tiempo de transacción asociado con cada dato. El problema consiste en encontrar *patrones secuenciales de acuerdo a un límite mínimo* impuesto por el usuario, dicho límite se mide en función al porcentaje de datos que contienen el patrón. Por ejemplo, un patrón secuencial puede estar dado por los usuarios de un video club que alquilan "Arma Mortal", luego "Arma Mortal 2", "Arma Mortal 3" y finalmente "Arma Mortal 4", lo cual no implica que todos lo hagan en ese orden.

APRENDIZAJE AUTOMÁTICO & MINERÍA DE DATOS

La automatización del proceso de aprendizaje se conoce como *Aprendizaje Automático*. La Minería de Datos es un caso especial de *Aprendizaje Automático* donde el escenario observado es una base de datos. Los gráficos que se encuentran a continuación explican este concepto.

En la figura 6, el entorno E representa el mundo real, el entorno sobre el cual se realiza el aprendizaje. E representa un número finito de observaciones u objetos que son codificados en algún formato legible para Aprendizaje Automático. El conjunto de ejemplos codificados constituye el conjunto de entrenamiento para el sistema de aprendizaje automático.

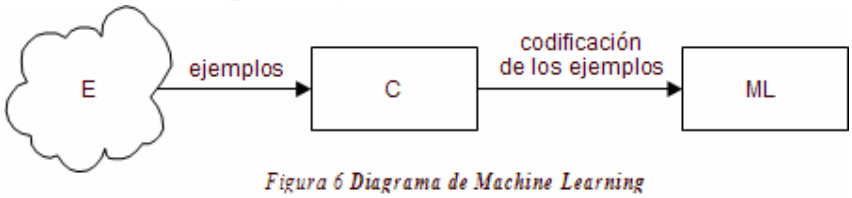


Figura 6 Diagrama de Machine Learning

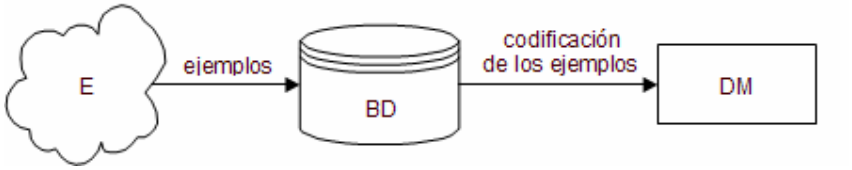


Figura 7 Diagrama de Minería de Datos

Por su lado, en la figura 7, la codificación C es reemplazada por una base de datos, que modela el entorno. Cada estado en la base de datos refleja algún estado de E, y cada transición de estados en la base de datos representa una transición de estados en E. El algoritmo utilizado para realizar la minería de datos construye entonces un modelo a partir de los datos en la base de datos.

Aunque a simple vista, la minería de datos parece muy similar a Aprendizaje Automático, hay importantes diferencias que deben tenerse en cuenta. La base de datos generalmente se construye con fines distintos a la minería de datos, con lo cual la base se diseña según los requerimientos del sistema y no según los requerimientos del algoritmo de aprendizaje.

1. Aplicaciones

A continuación se describen algunos algoritmos de Aprendizaje Automático que han sido utilizados con éxito en la Minería de Datos. Algunos de ellos son generales y pueden ser utilizados en varios dominios de conocimiento, mientras que otros fueron diseñados para un dominio en particular.

- **ID3**

  Este sistema ha sido el que más impacto ha tenido en la Minería de Datos. Desarrollado en los años ochenta por Quinlan, ID3 significa *Induction Decision Trees*, y es un sistema de aprendizaje supervisado que construye árboles de decisión a partir de un conjunto de ejemplos. Estos ejemplos son tuplas, donde el dominio de cada atributo de estas tuplas está limitado a un conjunto de valores. Las primeras versiones del ID3 generaban descripciones para dos clases, como ser positiva y negativa. En las versiones posteriores, se eliminó esta restricción, pero se mantuvo la restricción de clases disjuntas. ID3 genera descripciones que clasifican cada uno de los ejemplos del conjunto de entrenamiento.

  Este sistema tiene una buena performance en un amplio rango de aplicaciones, como dominios médicos, artificiales y el análisis de juegos de ajedrez. El nivel de precisión en la clasificación es alto. Sin embargo, el sistema no hace uso del conocimiento del dominio. Además, los árboles son demasiado frondosos, lo cual conlleva a una difícil interpretación. En esos casos pueden ser transformados en reglas de decisión para hacerlos más comprensibles.

- **C4.5**

El C4.5 es una extensión del ID3 que permite trabajar con valores continuos para los atributos, separando los posibles resultados en dos ramas: una para aquellos Ai<=N y otra para Ai>N. Este algoritmo fue propuesto por Quinlan en 1993. El algoritmo C4.5 genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente. El árbol se construye mediante la estrategia de profundidad-primero *(depth-first)*. El algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en la mayor ganancia de información. Para cada atributo discreto, se considera una prueba con *n* resultados, siendo *n* el número de valores posibles que puede tomar el atributo. Para cada atributo continuo, se realiza una *prueba binaria* sobre cada uno de los valores que toma el atributo en los datos.

- **AQ15**

El AQ15 fue desarrollado por Michalski. Es un sistema de aprendizaje inductivo que genera reglas de decisión, donde el antecedente es una fórmula lógica. Una característica particular de este sistema es la inducción constructiva *(constructive induction)*, es decir, el uso de conocimientos del dominio para generar nuevos atributos que no están presentes en los datos de entrada. Al igual que el ID3, el AQ15 está diseñado para la generación de reglas fuertes, es decir, que para cada clase, se construye una regla que cubre todos los ejemplos positivos y ningún ejemplo negativo. El sistema soluciona el problema de los ejemplos incompletos o inconsistentes mediante un pre o post procesamiento. En el post procesamiento, además, se reducen de forma drástica la cantidad de reglas generadas mediante el truncamiento de reglas, el cual no afecta la precisión de las reglas obtenidas. AQ15 ha sido testeado en dominios médicos, como el diagnóstico en la limfografía, diagnóstico de cáncer de mama y la ubicación del tumor primario. En estos casos, se obtuvieron reglas con el mismo nivel de precisión que el de los expertos humanos. En todos los casos, los datos de entrenamiento son conjuntos chicos, de unos cientos de ejemplos.

- **CN2**

El sistema CN2, desarrollado por Clark y Niblett, es una adaptación del AQ15. La gran desventaja del AQ15 es que elimina los ruidos mediante pre y post procesamiento y no durante la ejecución del algoritmo. El objetivo del CN2 es, entonces, incorporar el manejo de datos ruidosos al algoritmo en sí. Combina entonces las técnicas de poda utilizadas en el ID3, con las técnicas de reglas condicionales utilizadas en el AQ15. El CN2 genera reglas simples y comprensibles en dominios donde los datos pueden tener ruido. Construye reglas probabilísticas, es decir, el antecedente en cada regla cubre ejemplos positivos de una clase, pero también puede cubrir ejemplos de otra clase en menor número. De esta forma no restringe el espacio de búsqueda únicamente a aquellas reglas inferibles a partir de los ejemplos. La performance el ID3, AQ15 y CN2 ha sido comparada en dominios médicos y artificiales. Las estructuras de conocimiento generadas en cada caso son de similar calidad y complejidad.

- **DBLearn**

El sistema DBLearn fue diseñado por Cai, Han y Cercone y utiliza conocimientos del dominio para generar descripciones para subconjuntos predefinidos de una base de datos relacional. Las características especiales de este sistema son su estrategia de búsqueda de abajo hacia arriba *(bottom up)*; el uso de conocimientos del dominio como jerarquías de valores de atributos y el uso del álgebra relacional. El conjunto de entrenamiento es una tabla de datos relacional con n-tuplas. El sistema DBLearn es relativamente simple, ya que utiliza solo dos operaciones de generalización para construir los descriptores. La generalización está orientada a los atributos, lo cual limita el conjunto de descriptores que pueden ser construidos. La performance del sistema es buena, y la complejidad en el tiempo está en el orden de los O(N logN), siendo N la cantidad inicial de tuplas.

- **Meta-Dendral**

El sistema Meta-Dendral es un sistema especial para la generación de reglas de conocimiento en la estereoscopia. Esta ciencia estudia la estructura tridimensional de la molécula. El Meta-Dendral es interesante porque utiliza un sistema de representación de conocimientos totalmente diferente a los anteriores. Al buscar generar reglas que puedan predecir dónde se romperá la estructura de una molécula, toma las estructuras moleculares como entrada. El sistema ha sido exitoso para encontrar reglas de fragmentación desconocidas hasta el momento. Sin embargo, la estrategia de búsqueda es ineficiente, ya que genera muchas reglas de decisión que luego son eliminadas en la etapa de optimización. Es muy difícil encontrar heurísticas que guíen la búsqueda y no existen técnicas explícitas que ayuden a eliminar ruidos o a destacar casos especiales.

- **RADIX/RX**

El sistema RX se utiliza para el descubrimiento de relaciones en bases de datos clínicas. La diferencia importante con otros sistemas es que incorpora la noción de *tiempo*: un dato es un conjunto de ejemplos que guardan información de un paciente en diferentes momentos, y los conocimientos generados son de naturaleza causal. El sistema divide su proceso de descubrimiento en dos etapas: primero genera hipótesis y, luego, utiliza técnicas avanzadas de estadística para validarlas. El sistema RX fue utilizado en una base de reumatología y sirvió para probar hipótesis acerca de la cantidad de droga prodnisone que aumenta el colesterol en la sangre. Sin embargo, la principal desventaja de este sistema es que no utiliza información del dominio para guiar la búsqueda. Una versión mejorada del RX, el RADIX, sí lo hace.

- **BACON**

  El sistema BACON utiliza algoritmos de análisis de datos para descubrir relaciones matemáticas entre datos numéricos. Ha redescubierto leyes como la ley de Ohm para circuitos eléctricos y la ley de desplazamiento de Arquímides. Los datos de entrenamiento son numéricos y, normalmente, son generadas en algún experimento previo. Cada tupla esta constituida por los valores de las mediciones durante el experimento. El sistema BACON tiene varias desventajas: no considera el ruido en los datos, ni la inconsistencia o los datos incompletos. Además, considera que todas las variables son relevantes, y explora todas las soluciones posibles utilizando un grafo, lo cual empeora considerablemente su performance.

- **SLIQ**

  El algoritmo SLIQ (*Supervised Learning In Quest*) fue desarrollado por el equipo Quest de IBM. Este algoritmo utiliza los árboles de decisión para clasificar grandes cantidades de datos. El uso de técnicas de pre-ordenamiento en la etapa de crecimiento del árbol, evita los costos de ordenamiento en cada uno de los nodos. SLIQ mantiene una lista ordenada independiente de cada uno de los valores de los atributos continuos y una lista separada de cada una de las clases. Un registro en la lista ordenada de atributos consiste en el valor del atributo y un índice a la clase correspondiente en la lista de clases. SLIQ construye el árbol de forma ancho-primero *(breadth-first)*. Para cada uno de los atributos busca en la lista correspondiente y calcula los valores de entropía para cada uno de los nodos de la frontera simultáneamente. A partir de la información obtenida se particionan los nodos de la frontera, y se expanden para obtener una nueva frontera. Aunque SLIQ trabaja con datos que pueden estar en disco mientras se ejecuta el algoritmo, necesita que cierta información resida en memoria permanentemente durante la totalidad de la ejecución del mismo. Dicha información crece proporcionalmente a la cantidad de registros de entrada, lo cual limita en gran medida la cantidad de registros de entrenamiento. Para solucionar este problema el equipo de desarrollo del Quest, ha desarrollado otro algoritmo de clasificación basado en árboles de decisión: el SPRINT *(**S**calable **P**a**R**allelizable **IN**duction of decision **T**rees)*. El SPRINT elimina todas las restricciones de memoria presentes en el SLIQ.

2. Inferencia de Reglas Simples

A continuación se detalla una manera sencilla de generar reglas de clasificación a partir de un grupo de instancias. El algoritmo "**1-Rule**" o **1R** genera un *árbol de decisión de un nivel*, o, visto de otra manera, una regla de decisión que evalúa un solo atributo. Recordemos que las reglas de decisión son de la forma: Si atributo1= valorX y atrib2 = valorY y . . . y atributoN = valorZ entonces Clase = C

Entonces, una regla simple que evalúa un único atributo será de la forma:

Si atributoi=valorA entonces Clase=C

El algoritmo del 1R se detalla a continuación:

Para cada atributo
  Para cada valor del atributo crear una regla de la siguiente manera:
    Contar las ocurrencias de cada una de las clases
    Encontrar la clase más frecuente
    Crear la regla que asigne esa clase al correspondiente atributo-valor
    Calcular la proporción de error de las reglas
    Elegir las reglas con la menor proporción de error

La proporción de error de las reglas se determina fácilmente *contando los errores en los datos de entrenamiento, es decir, las instancias que no pertenecen a la clase mayoritaria*. Cada atributo genera un conjunto de reglas diferente, con una regla por cada valor del atributo. Se evalúa la proporción de error para cada conjunto de reglas y se elige el conjunto con menor error.

Aunque parezca sorprendente, varias pruebas realizadas con este algoritmo han demostrado que genera buenos modelos para los conjuntos de datos simples. El **1R** ha tenido resultados casi tan precisos como otras técnicas de aprendizaje automático que generan árboles de decisión de más niveles o conjuntos de reglas más extensos.

Es importante encontrar la técnica que mejor se adapta a cada caso, teniendo en cuenta los datos de entrada, el tipo de modelo que se quiere obtener y los recursos con que se cuenta. El 1R es una buena opción cuando la *rapidez y la simplicidad prevalecen* sobre la *exactitud del resutlado*.

3. Construcción de reglas de decisión

Hay varios métodos para construirse reglas de decisión directamente a partir de los datos, analizaremos el método conocido como "*separa y reinarás*".

El enfoque básico de dicho método consiste en *tomar cada una de las clases y buscar la manera de cubrir todas las instancias que pertenecen a ella, excluyendo las instancias que no pertenecen*. La idea es tomar una clase y construir una regla que cubra todas las instancias que pertenecen a ella. La regla se va construyendo condición por condición.

Cada condición se elige teniendo en cuenta:

- Sea ***t*** la cantidad de instancias cubiertas por la regla.
- Sea ***p*** la cantidad de instancias cubiertas por la regla de clase *i*.
- Entonces, ***t-p*** será la cantidad de instancias cubiertas por la regla pertenecientes a una clase distinta de ***i.***
- La condición que se agrega a la regla es aquella que maximice **p/t**

La idea básica del método "separa y reinarás" fue implementada en el algoritmo PRISM que se detalla a continuación.

Algoritmo PRISM

```
Para cada clase C
    Inicializar E con todas las instancias del conjunto
    Mientras E contenga instancias de clase C
        Crear una regla R con el antecedente vacio que predice clase C
        Repetir hasta que R sea perfecta (o no haya más atributos)
            Para cada atributo de A no mencionado en R y cada valor v
                Considerar agregar la condición A=v al antecedente de R
                Seleccionar A y v para maximizar la precisión p/t (resolver los empates
                eligiendo la condición con mayor p)
            Fin para
            Agregar A=v a R
        Fin repetir
        Eliminar de E las instancias cubiertas por R
    Fin mientras
Fin para
```

Ejemplo

Dados los siguientes datos:

Edad	Prescripción anteojos	de Astigmatismo	Proporción de producción de lágrimas	Lentes recomendados
joven	miope	no	reducida	ninguno
joven	miope	no	normal	suaves
joven	miope	si	reducida	ninguno
joven	miope	si	normal	duros
joven	hipermétrope	no	reducida	ninguno
joven	hipermétrope	no	normal	suaves
joven	hipermétrope	si	reducida	ninguno
joven	hipermétrope	si	normal	duros

Edad	Prescripción anteojos	de Astigmatismo	Proporción de producción de lágrimas	Lentes recomendados
adulta	miope	no	reducida	ninguno
adulta	miope	no	normal	suaves
adulta	miope	si	reducida	ninguno
adulta	miope	si	normal	duros
adulta	hipermétrope	no	reducida	ninguno
adulta	hipermétrope	no	normal	suaves
adulta	hipermétrope	si	reducida	ninguno
adulta	hipermétrope	si	normal	ninguno
mayor	miope	no	reducida	ninguno
mayor	miope	no	normal	ninguno
mayor	miope	si	reducida	ninguno
mayor	miope	si	normal	duros
mayor	hipermétrope	no	reducida	ninguno
mayor	hipermétrope	no	normal	suaves
mayor	hipermétrope	si	reducida	ninguno
mayor	hipermétrope	si	normal	ninguno

Construimos las reglas de decisión según el algoritmo PRISM

Tomamos la clase *duros*
Si ? entonces lentes recomendados = duros
Edad = joven 2/8
Edad = adulta 1/8
Edad = mayor 1/8
Anteojos = miope 3/12
Anteojos = hipremétrope 1/12
Astigmatismo = no 0/12
Astigmatismo = si 4/12
Lágrimas = reducida 0/12
Lágrimas = normal 4/12
Elegimos *Astigmatismo* = si que es la condición de mayor p/t
Si Astigmatismo = si y ? entonces lentes recomendados = duros
Edad = joven 2/4
Edad = adulta 1/4
Edad = mayor 1/4
Anteojos = miope 3/6
Anteojos = hipremétrope 1/6
Lágrimas = reducida 0/6
Lágrimas = normal 4/6
Si Astigmatismo = si y Lágrimas = normal y ? entonces lentes recomendados = duros
Edad = joven 2/2
Edad = adulta 1/2
Edad = mayor 1/2
Anteojos = miope 3/3
Anteojos = hipremétrope 1/3
Si Astigmatismo = si y Lágrimas = normal y Anteojos = miope entonces lentes recomendados = duros
Esta es una de las reglas finales, pero cubre sólo tres casos para los cuales la recomendación es lentes duros. Con lo cual, si aplicamos nuevamente el algoritmo para la clase duros, sin tener en cuenta los casos cubiertos por la regla anterior, obtendremos:
Si Edad = joven y Astigmatismo = si y Lágrimas = normal entonces lentes recomendados = duros
Una vez que cubrimos todos los casos de clase dura, se repite el procedimiento para las otras dos clases.

LA FAMILIA TDIDT

La familia de los *Top Down Induction Trees* (TDIDT) pertenece a los métodos inductivos del Aprendizaje Automático que aprenden a partir de ejemplos preclasificados. En Minería de Datos, se utiliza para modelar las clasificaciones en los datos mediante árboles de decisión

1. Construcción de los árboles de decisión

Los árboles TDIDT, a los cuales pertenecen los generados por el ID3 y el C4.5, se construyen a partir del método de Hunt. El esqueleto de este método para construir un árbol de decisión a partir de un conjunto *T* de datos de entrenamiento es muy simple. Sean las clases *{C1, C2,. . ., Ck}*. Existen tres posibilidades:

1. *T* contiene uno o más casos, todos pertenecientes a un única clase *Cj*: El árbol de decisión para *T* es una hoja identificando la clase *Cj* .
2. *T* no contiene ningún caso: El árbol de decisión es una hoja, pero la clase asociada debe ser determinada por información que no pertenece a *T*. Por ejemplo, una hoja puede escogerse de acuerdo a conocimientos de base del dominio, como ser la clase mayoritaria.
3. *T* contiene casos pertenecientes a varias clases: En este caso, la idea es refinar *T* en subconjuntos de casos que tiendan, o parezcan tender hacia una colección de casos pertenecientes a una única clase. Se elige una prueba basada en un único atributo, que tiene uno o más resultados, mutuamente excluyentes *{O1, O2,. . ., On}*. *T* se particiona en los subconjuntos *T1, T2,. . ., Tn* donde *Ti* contiene todos los casos de *T* que tienen el resultado *Oi* para la prueba elegida. El árbol de decisión para *T* consiste en un nodo de decisión identificando la prueba, con una rama para cada resultado posible. El mecanismo de construcción del árbol se aplica recursivamente a cada subconjunto de datos de entrenamientos, para que la *i*-ésima rama lleve al árbol de decisión construido por el subconjunto *Ti* de datos de entrenamiento.

A continuación se presenta el algoritmo del método ID3 para la construcción de árboles de decisión en función de un conjunto de datos previamente clasificados.

Función ID3

(R: conjunto de atributos no clasificadores, C: atributo clasificador, S: conjunto de entrenamiento) devuelve un árbol de decisión;

```
Comienzo
    Si S está vacio,
        devolver un único nodo con Valor Falla;
    Si todos los registros de S tienen el mismo valor para el atributo clasificador,
        Devolver un único nodo con dicho valor;
    Si R está vacio, entonces
        devolver un único nodo con el valor más frecuente del atributo clasificador en los
        registros de S [Nota: habrá errores, es decir, registros que no estarán bien clasificados en
        este caso];
    Si R no está vacio, entonces
        D   atributo con mayor Ganancia(D,S) entre los atributos de R;
        Sean {dj| j=1,2, .., m} los valores del atributo D;
        Sean {Sj| j=1,2, .., m} los subconjuntos de S correspondientes a los valores de dj
        respectivamente;
        Devolver un árbol con la raiz nombrada como D y con los arcos nombrados d1, d2, .., dm
        que van respectivamente a los árboles
        ID3(R-{D}, C, S1), ID3(R-{D}, C, S2), .., ID3(R-{D}, C, Sm);
Fin
```

1. Cálculo de la Ganancia de Información

En los casos, en los que el conjunto *T* contiene ejemplos pertenecientes a distintas clases, se realiza una prueba sobre los distintos atributos y se realiza una partición según el "mejor" atributo. Para encontrar el "mejor" atributo, se utiliza la teoría de la información, que sostiene que la información se maximiza cuando la entropía se minimiza. La entropía determina la azarosidad o desestructuración de un conjunto.

Supongamos que tenemos ejemplos positivos y negativos. En este contexto la entropía de un subconjunto Si, H(Si), puede calcularse como:

$$H(S_i) = -\left[(p_i^+)\cdot Log_2(p_i^+)\right] - \left[(p_i^-)\cdot Log_2(p_i^-)\right]$$

Donde +

es la probabilidad de que un ejemplo tomado al azar de Si sea positivo. Esta probabilidad puede calcularse como:

$$p^+ = \frac{n^+}{n^+ + n^-}$$ *Probabilidad de que un ejemplo de S pertenezca a la clase +*

Si el atributo *at* divide el conjunto *S* en los subconjuntos *Si, i* = 1,2, ... , n, entonces, la entropía total del sistema de subconjuntos será:

$$H(S, A) = \sum_{i=1}^{n}\left[P(S_i) * H(S_i)\right]$$

Donde

H (*Si*) es la entropía del subconjunto *S i* y

P(*Si*) es la probabilidad de que un ejemplo pertenezca

a *S i* . Puede calcularse, utilizando los tamaños relativos de los subconjuntos, como:

$$P(S_i) = \frac{|S_i|}{|S|}$$

La ganancia en información puede calcularse como la disminución en entropía. Es decir:

$$I(S, A) = H(S) - H(S, A)$$

Donde

H (*S*) es el valor de la entropía a priori, antes de realizar la subdivisión, y

H (*S* , *at*) es el valor de

la entropía del sistema de subconjuntos generados por la partición según *at*.

El uso de la entropía para evaluar el mejor atributo no es el único método existente o utilizado en *Aprendizaje Automático*. Sin embargo, es el utilizado por Quinlan al desarrollar el ID3 y su sucesor el C4.5.

2. Entropía

El estudio de la entropía, como la define la teoría de la Información, merece un capítulo aparte. Supongamos que tenemos una moneda que está arreglada para que siempre salga cara. Entonces, P(cara)=1. Si tiramos la moneda, podemos predecir que el resultado será cara. ¿Cuánta información hemos aprendido? (o, ¿cuán inciertos fuimos al predecir?). No hemos aprendido nada.

¿Qué pasaría si la moneda no estuviese tocada? ¿Cuánto aprendemos si sale cara? 1bit (no estábamos seguros acerca de cuál de los dos resultados igualmente probables saldría y ahora lo sabemos). Entonces, una distribución "uniforme" tiene poca incertidumbre y, por ende, aprendemos menos porque aprendemos a partir de resultados altamente probables.

Supongamos ahora, que tenemos una moneda en la cual la probabilidad de que salga cara es muy alta (0.99). Si tiramos la moneda y sale ceca, nos sorprendemos, aprendemos más si nos dicen que un evento muy poco probable ha ocurrido. En general, la cantidad de aprendizaje es inversamente proporcional a la probabilidad del evento.

Ahora supongamos que tenemos un dado normal. Si nos dicen el resultado de una tirada, ¿cuánto hemos aprendido? 2.6 bits (cada posibilidad ocurre 1/6 de las veces, log2(1/(1/6))

En general la información de un evento es I(e)= log2(1/P(e)) = -log2P(e)

Ahora, supongamos que S es una variable aleatoria con *n* valores posibles. Entonces, como explicamos anteriormente, la entropía se define como

$$H(S_i) = -\sum_{i-1}^{n} p_i \cdot \log_2 p_i$$

donde pi es la probabilidad de que un ejemplo tomado al azar pertenezca a la clase *i* y se calcula en base a la frecuencia de los datos de dicha clase en los datos de entrenamiento. Vemos que la entropía es simplemente la cantidad de información esperada de observar un evento que ocurre según una distribución de probabilidades. La entropía mide la cantidad de incertidumbre que tenemos dada una distribución de probabilidades.

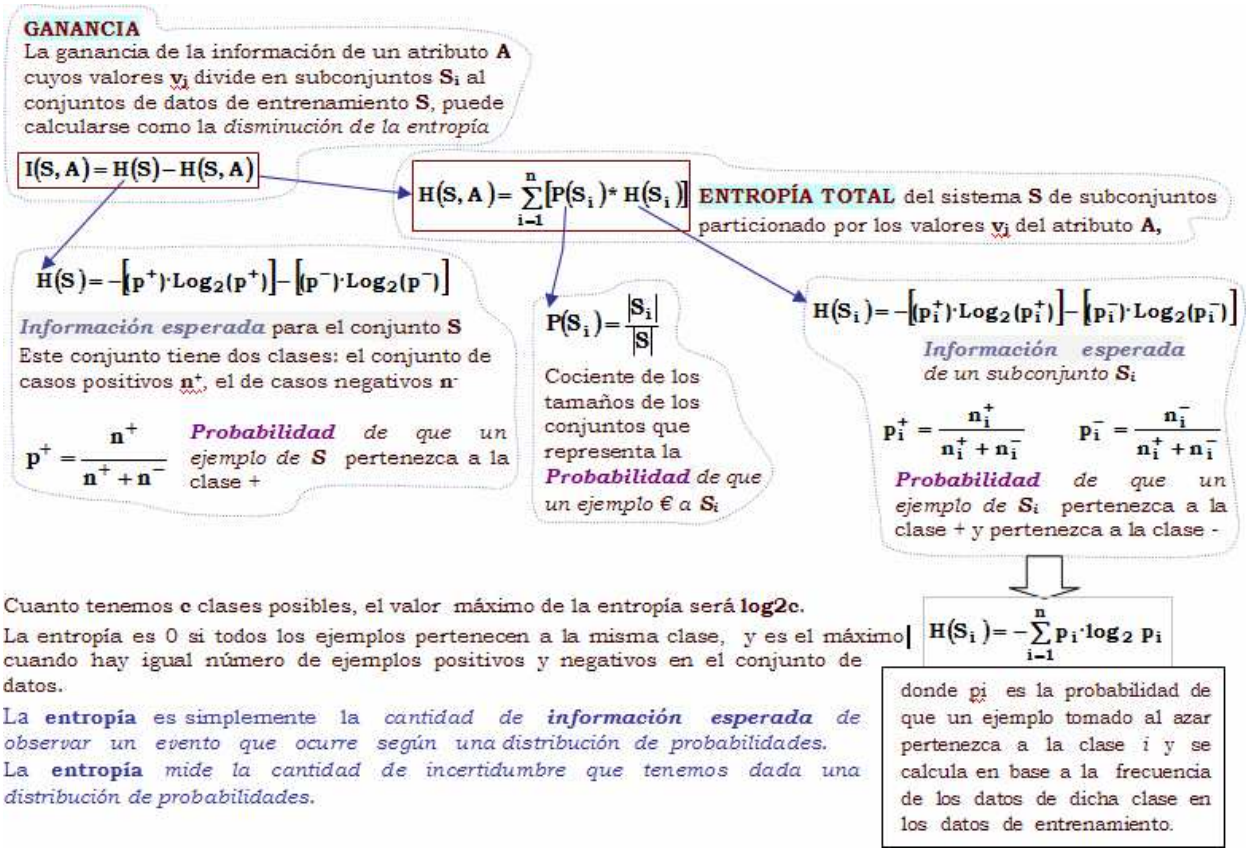
Si tomamos un conjunto con elementos positivos y negativos, la entropía variará entre 0 y 1

Notemos que la entropía es 0 si todos los ejemplos pertenecen a la misma clase, y es 1 cuando hay igual número de ejemplos positivos y negativos en el conjunto de datos.

Cuanto tenemos *c* clases posibles, el valor máximo de la entropía será *log2c*.

Una interpretación de la entropía desde el punto de vista de la Teoría de la Información es que especifica el número mínimo de bits de información necesarios para codificar un ejemplo arbitrario *S* (un ejemplo sacado al azar del conjunto de datos). Por ejemplo, si p+ es 1, el receptor sabe que el ejemplo es positivo, con lo cual no es necesario enviar ningún bit. En cambio, si p+ es 0.5, se necesita 1 bit para indicar si el ejemplo es positivo o negativo. Si p+ es 0.8, se pueden utilizar varios mensajes de menos de 1 bit para indicar la clase de varios ejemplos.

Resumen para el **Cálculo**: se analizan los ejemplos que pertenecen a distintas clases, se hace una prueba sobre los atributos y una partición según el *mejor atributo* → en función de la Teoría de Información (la información se maximiza cuando la entropía se minimiza)



Cuanto tenemos **c** clases posibles, el valor máximo de la entropía será **log2c**.

La entropía es 0 si todos los ejemplos pertenecen a la misma clase, y es el máximo cuando hay igual número de ejemplos positivos y negativos en el conjunto de datos.

La **entropía** es simplemente la *cantidad de* ***información esperada*** *de observar un evento que ocurre según una distribución de probabilidades.*

La **entropía** *mide la cantidad de incertidumbre que tenemos dada una distribución de probabilidades.*

$$H(S_i) = -\sum_{i-1}^{n} p_i \cdot \log_2 p_i$$

donde pi es la probabilidad de que un ejemplo tomado al azar pertenezca a la clase *i* y se calcula en base a la frecuencia de los datos de dicha clase en los datos de entrenamiento.

Genéricamente

Para calcular la **GANANCIA DE INFORMACIÓN** en cada atributo:

- Primero calculamos la **INFORMACIÓN ESPERADA para clasificar cada ejemplo dado:**

$$I(s_1,\dots,s_n) = -\sum_{i=1}^{m}(s_i/s)\log_2(s_i/s)$$

Un atributo **A** con valores $\{a_1,a_2,\dots,a_v\}$ puede ser usado para particionar **S** en subconjuntos $\{S_1, S_2, \dots, S_v\}$ donde cada S_j contiene los ejemplos de **S** que tienen el valor a_j de **A**. Dado S_j que contiene S_{ij} ejemplos de la clase C_i.

- Luego se calcula la **ENTROPÍA de cada atributo**. Calculamos el valor esperado de información para cada distribución.

$$E(A) = \sum_{j=1}^{v}\left[\frac{s_{1j}+\dots+s_{mj}}{s}\right] I(s_1,\dots,s_n)$$

La información esperada necesaria para clasificar un determinado ejemplo si los ejemplos están particionados según la especialidad. La información buscada en base a la partición de **A** es conocida como Entropía de **A**, este es el promedio sopesado

- Ahora calculamos la **GANANCIA de información** de dicha partición es:

$$Ganancia(A) = I(s_i,\dots,s_m) - E(A)$$

Se calcula la ganancia para cada atributo definiendo los ejemplos en **S**, *El atributo con la ganancia más alta es el que se considera como el más discriminante dentro del conjunto.*

Procesando la información ganada para cada atributo se obtiene un ranking de los mismos, el cual puede ser usado para seleccionar los mejores atributos.

EJEMPLO

Si tenemos **S** el conjunto de 250 alumnos, cuyos datos servirán como ejemplos de entrenamiento.

De este conjunto **S,** hay 120 (**n+**) que corresponden a la clase C_1 **"graduados"** y 130 (**n-**) que corresponden a la clase C_2 **"no graduados".**

El atributo **A**, representa al "**Sexo"**, que dividen el conjunto **S** en 2 subconjuntos: S_1 para el grupo de "mujeres" y S_2 para el grupo de "hombres". A su vez estos subconjuntos se dividen en las dos clases: graduados y no graduados.

Siendo S_1C_1 las mujeres graduadas ($n+_1 = 65$) y S_1C_2 las mujeres no graduadas ($n-_1 = 68$)

Siendo S_2C_1 los hombres graduados ($n+_2 = 55$) y S_2C_2 los hombres no graduados ($n-_2 = 62$)

La ganancia es $I(S,A) = H(S) - H(S,A)$, por lo tanto necesito averiguar la información esperada de S y la entropía de A $H(S) = -[(p^+)\cdot Log_2(p^+)] - [(p^-)\cdot Log_2(p^-)]$

1. Calculo la ***Información esperada S***.

Siendo $p^+ = \frac{n^+}{n^+ + n^-} = \frac{120 \quad graduados}{250}$

$p^- = \frac{n^-}{n^+ + n^-} = \frac{130 \quad no\_graduados}{250}$

$$H(S) = -\left[\frac{120}{250}\cdot Log_2\frac{120}{250}\right] - \left[\frac{130}{250}\cdot Log_2\frac{130}{250}\right] = -(-0.508) - (-0.491) = 0.9988$$

2. Calculo la ***Entropía de A*** (sexo) respecto al conjunto S. $H(S,A) = \sum_{i=1}^{n}[P(S_i) * H(S_i)]$

Calculo la ***probabilidad*** para los subconjunto S_1 y S_2

$P(S_i) = \frac{|S_i|}{|S|}$ $\quad P(S_{1femenino}) = \frac{|S_1|}{|S|} = \frac{133}{250}$ $\quad P(S_{2masculino}) = \frac{|S_2|}{|S|} = \frac{117}{250}$

Calculo la ***información esperada*** para los subconjunto S_1 y S_2 $H(S_i) = -[(p_i^+)\cdot Log_2(p_i^+)] - [(p_i^-)\cdot Log_2(p_i^-)]$

$p_1^+ = \frac{n_1^+}{n_1^+ + n_1^-} = \frac{65 \quad graduados}{133}$ $\quad p_1^- = \frac{n_1^-}{n_1^+ + n_1^-} = \frac{68 \quad no\_grad}{133}$ $\quad H(S_{1femenino}) = -\left[\frac{65}{133}\cdot Log_2\frac{65}{133}\right] - \left[\frac{68}{133}\cdot Log_2\frac{68}{133}\right] = 0.9996$

$p_2^+ = \frac{n_2^+}{n_2^+ + n_2^-} = \frac{55 \quad graduados}{117}$ $\quad p_2^- = \frac{n_2^-}{n_2^+ + n_2^-} = \frac{62 \quad no\_grad}{117}$ $\quad H(S_{2masculino}) = -\left[\frac{55}{117}\cdot Log_2\frac{55}{117}\right] - \left[\frac{62}{117}\cdot Log_2\frac{62}{117}\right] = 0.9974$

$$H(S,A) = \left[\frac{133}{250} * 0.9996\right] + \left[\frac{117}{250} * 0.9974\right] = 0.9986$$

3. Calculo la ***Ganancia de A*** (sexo) respecto al conjunto S. $I(S,A) = H(S) - H(S,A)$

$$I(S,A) = 0.9988 - 0.9986 = 0.0002$$

GENERICAMENTE

1. Cálculo de la ***información esperada para clasificar cada ejemplo*** dado: $I(s_1,\dots,s_n) = -\sum_{i=1}^{m}(s_i/s)\log_2(s_i/s)$

$$I(S_1,S_2) = I(120,130) = -\left(\left(\frac{120}{250} * LOG_2\left(\frac{120}{250}\right)\right) + \left(\frac{130}{250} * LOG_2\left(\frac{130}{250}\right)\right)\right) = 0.9988$$

2. Luego se calcula ***la entropía de cada atributo***.

- Necesitaremos observar la distribución de estudiantes graduados y no graduados según el valor de **SEXO**. Calculamos el ***valor esperado de información para cada distribución***:

  Para el **Sexo F**: $S_1C_1 = 65$ $S_1C_2 = 68$

$$I(S_{11},S_{21}) = I(65,68) = -\left(\left(\frac{65}{133} * LOG_2\left(\frac{65}{133}\right)\right) + \left(\frac{68}{133} * LOG_2\left(\frac{68}{133}\right)\right)\right) = 0.9996$$

  Para el **sexo M**: $S_2C_1 = 55$ $S_2C_2 = 62$

$$I(S_{12},S_{22}) = I(55,62) = -\left(\left(\frac{55}{117} * LOG_2\left(\frac{55}{117}\right)\right) + \left(\frac{62}{117} * LOG_2\left(\frac{62}{117}\right)\right)\right) = 0.9974$$

- Calculamos la ***información esperada*** *necesaria para clasificar un determinado ejemplo si los ejemplos están particionados* ***según el Sexo***:

$E(A) = \sum_{j=1}^{v}\left[\frac{s_{1j}+\dots+s_{mj}}{s}\right] I(s_1,\dots,s_n)$

$$E(sexo) = -\left(\left(\frac{133}{250} * 0.9996\right) + \left(\frac{117}{250} * 0.9974\right)\right) = 0.9986$$

3. Ahora calculamos la ganancia de información de dicha partición es: $Ganancia(A) = I(s_i,\dots,s_m) - E(A)$

$$Ganancia(sexo) = I(S_1,S_2) - E(sexo) = 0.9988 - 0.9986 = 0.0002$$

3. Proporción de ganancia

Volviendo al tema de la elección del "mejor" atributo para realizar la división de los datos, encontramos que *la ganancia favorece a aquellos atributos que tienen muchos valores frente* a los que tienen pocos valores. Tomemos, por ejemplo, unos registros diarios, cada uno con la fecha. Si particionamos el conjunto de datos según el campo fecha, dicha partición será perfecta, sin embargo, el árbol resultante no servirá para clasificar casos futuros. Como el campo fecha tiene tantos valores, divide a los datos de entrenamiento en conjuntos pequeños, con lo cual tendrá una alta ganancia de información en relación a los datos de entrenamiento.

Para evitar esto puede utilizarse otra medida para dividir a los datos. Una de estas medida alternativas es la *ganancia de información*. Esta medida penaliza a los atributos como fecha al incorporar el término de información de la división, que es sensible a que tan amplia y uniformemente cada atributo divide a los datos:

$$I\_división(X) = -\sum_{i=1}^{n}\frac{|T_i|}{|T|} \bullet Log_2\left(\frac{|T_i|}{|T|}\right)$$

La información de la división no es otra cosa que la entropía del conjunto con respecto al atributo i. Se define, entonces, a la proporción de ganancia como:

$$proporcion\_ganancia(X) = \frac{Ganancia\_I(T,X)}{I\_división(X)}$$

Cabe destacar que la información de la división penalizará a aquellos atributos con muchos valores uniformemente distribuidos. Si tenemos n datos separados perfectamente por un atributo, la información de la división para ese caso será log2n. En cambio, un atributo que divide a los ejemplos en dos mitades, tendrá una información de la división de 1.

¿Qué pasa cuando la información de la división es cercana a cero? Para resolver este inconveniente pueden aplicarse varias heurísticas. Por ejemplo, puede utilizarse la ganancia como medida y utilizar la proporción de ganancia sólo para los atributos que estén sobre el promedio.

Encontramos que la **ganancia** favorece a aquellos atributos que tienen muchos valores frente a los que tienen pocos valores.

Tomemos, por ejemplo, unos registros diarios, cada uno con la fecha. Si particionamos el conjunto de datos según el campo fecha, dicha partición será perfecta, sin embargo, el árbol resultante no servirá para clasificar casos futuros. Como el campo fecha tiene tantos valores, divide a los datos de entrenamiento en conjuntos pequeños, con lo cual tendrá una alta ganancia de información en relación a los datos de entrenamiento

⇩

Para evitar esto puede utilizarse otra medida para dividir a los datos.
LA PROPORCIÓN DE GANANCIA

Penaliza a los atributos como fecha al incorporar el término de información de la división, que es sensible a que tan amplia y uniformemente cada atributo divide a los datos

$$\mathbf{I\_división(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \bullet Log_2\left(\frac{|T_i|}{|T|}\right)} \Longrightarrow \mathbf{proporcion\_ganancia(X) = \frac{Ganancia\_I(T,X)}{I\_división(X)}}$$

La información de la división no es otra cosa que la entropía del conjunto con respecto al atributo *i*
Penalizará a aquellos atributos con muchos valores uniformemente distribuidos.
Si tenemos n datos separados perfectamente por un atributo, la información de la división para ese caso será log2n. Puede utilizarse la ganancia como medida y utilizar la proporción de ganancia sólo para los atributos que estén sobre el promedio.

En el ejemplo anterior:

La ganancia para el atributo Sexo es $\mathbf{I(S,A) = 0.9988 - 0.9986 = 0.0002}$

De los 250 casos de entrenamiento: Hay 117 hombre, y 133 mujeres

$$\mathbf{I\_división(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \bullet Log_2\left(\frac{|T_i|}{|T|}\right) = -\left[\left(\frac{117}{250}\log_2\frac{117}{250}\right)+\left(\frac{133}{250}\log_2\frac{133}{250}\right)\right] = -[(-0.51)+(-0.48)] = 0.9970}$$

$$\mathbf{proporcion\_ganancia(X) = \frac{Ganancia\_I(T,X)}{I\_división(X)} = \frac{0.0002}{0.9970} = 0.0002006}$$

EJEMPLO PASO POR PASO.

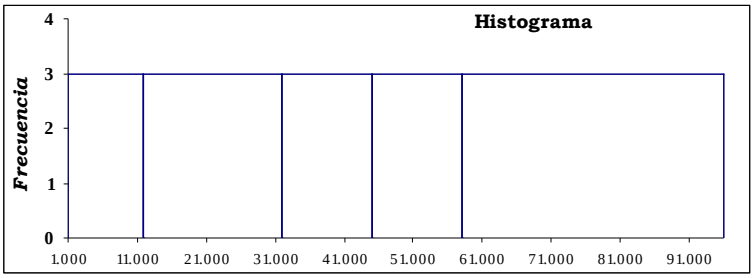
Dado los siguientes datos

Ciudad de nacimiento	[1]Edad	Sexo	Target para campaña publicitaria (clase)
Rosario	12	F	S
Santa Rosa	34	F	N
Santa Rosa	55	F	N
Santa Rosa	78	F	N
Rosario	96	F	S
Santa Rosa	45	F	N
Rosario	40	F	S
Santa Rosa	23	M	S
Santa Rosa	25	M	S
Rosario	53	M	N
Santa Rosa	58	M	N
Santa Rosa	1	M	S
Rosario	10	M	S
Rosario	32	M	S
Rosario	67	M	N

[1] En más adelante (datos numéricos) se aplicó el algoritmo, tratando la variable Edad como un **atributo numérico**. Se ordenan los ejemplos en función de la edad, se identifican ejemplos adyacentes que tengan valor de la clase diferente y se considera como candidatos los puntos medios de la división. A cada partición se calcula la proporción de ganancia. Cada posible partición entra en competencia con el resto de los atributos, seleccionando el de mayor ganancia.

DISCRETIZACIÓN DEL ATRIBUTO EDAD: Edad es un atributo numérico que discretizo en 5 rangos con el método de Frecuencias iguales:

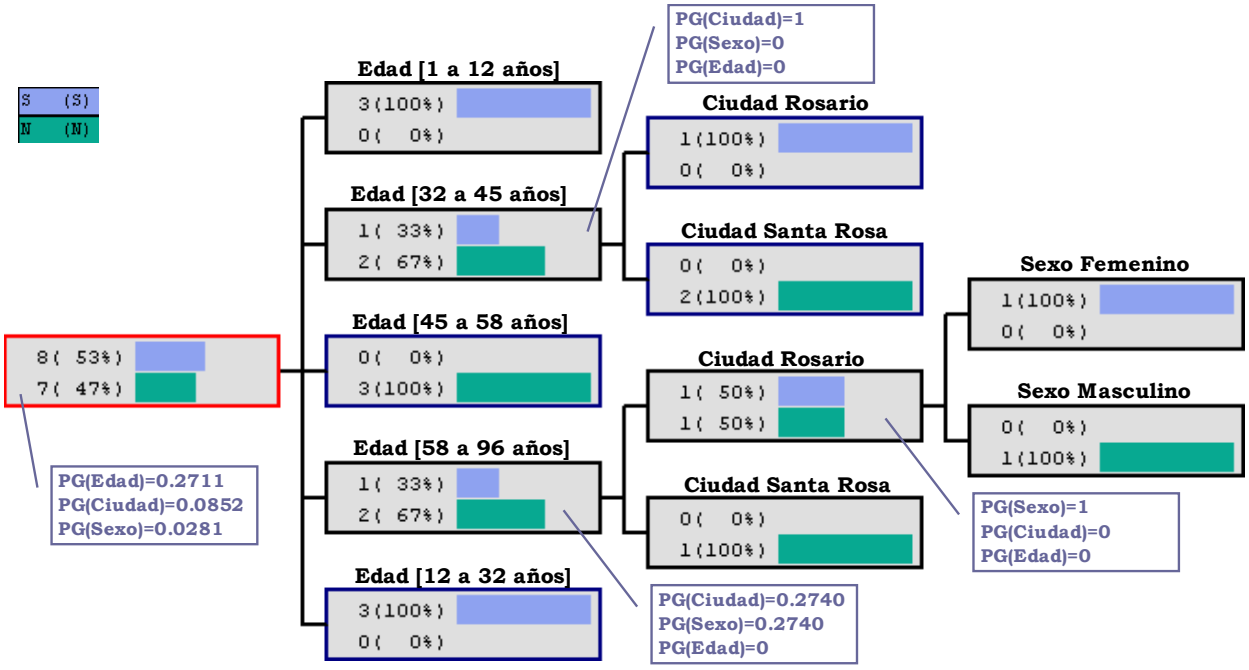
Discretización e histograma:					
Clase	**Límite inferior**	**Límite superior**	**Centro**	**Frecuencia**	**Prb. Densidad**
1	1	12	6.5	3	0.018
2	12	32	22	3	0.010
3	32	45	38.5	3	0.015
4	45	58	51.5	3	0.015
5	58	96	77	3	0.005



La Tabla quedaría:

id	Ciudad de nacimiento	Edad	Rangos Edad	Sexo	Target para campaña publicitaria (clase)
1	Rosario	12	[1 a 12 años]	F	S
2	Santa Rosa	34	[32 a 45 años]	F	N
3	Santa Rosa	55	[45 a 58 años]	F	N
4	Santa Rosa	78	[58 a 96 años]	F	N
5	Rosario	96	[58 a 96 años]	F	S
6	Santa Rosa	45	[32 a 45 años]	F	N
7	Rosario	40	[32 a 45 años]	F	S
8	Santa Rosa	23	[12 a 32 años]	M	S
9	Santa Rosa	25	[12 a 32 años]	M	S
10	Rosario	53	[45 a 58 años]	M	N
11	Santa Rosa	58	[45 a 58 años]	M	N
12	Santa Rosa	1	[1 a 12 años]	M	S
13	Rosario	10	[1 a 12 años]	M	S
14	Rosario	32	[12 a 32 años]	M	S
15	Rosario	67	[58 a 96 años]	M	N

a. ÁRBOL DE DECISIÓN UTILIZANDO EL MÉTODO ID3



A partir de todos los datos disponibles, el ID3 analiza todas las divisiones posibles según los distintos atributos y calcula la ganancia y/o la proporción de ganancia.

1º *Evaluamos cada atributo y seleccionamos el mejor. (Determino el nodo raíz)*

Ciudad de nacimiento	Rangos Edad	Sexo	Target para campaña publicitaria (clase)
Santa Rosa	[32 a 45 años]	F	N
Santa Rosa	[32 a 45 años]	F	N
Santa Rosa	[45 a 58 años]	F	N
Rosario	[45 a 58 años]	M	N
Santa Rosa	[45 a 58 años]	M	N
Santa Rosa	[58 a 96 años]	F	N
Rosario	[58 a 96 años]	M	N
Rosario	[1 a 12 años]	F	S
Santa Rosa	[1 a 12 años]	M	S
Rosario	[1 a 12 años]	M	S
Santa Rosa	[12 a 32 años]	M	S
Santa Rosa	[12 a 32 años]	M	S
Rosario	[12 a 32 años]	M	S
Rosario	[32 a 45 años]	F	S
Rosario	[58 a 96 años]	F	S

S es el conjunto de todos los ejemplos donde de 15 individuos a 7 pertenecen a la clase "N" y "8" a la clase "S".

Analizamos la ganancia y proporción de ganancia de Ciudad, Edad y Sexo para determinar cual de los 3 atributos particionera los datos.

1. Cálculo de la ***ENTROPÍA TOTAL del conjunto respeto a la clase***: $I(s_1,\dots,s_n) = -\sum_{i=1}^{m}(s_i/s)\log_2(s_i/s)$

S = 15 $S_{SI} = 8$ $S_{NO} = 7$

Información esperada para clasificar cada ejemplo

$$I(S_S, S_N) = -\left[\left(\frac{8}{15}\log_2 \bullet \frac{8}{15}\right) + \left(\frac{7}{15}\log_2 \bullet \frac{7}{15}\right)\right] = -[(-0.4837) + (-0.5131)] = \boxed{0.9968}$$

2. Luego calculo ***la ENTROPÍA DE CADA ATRIBUTO.***

$$I(s_1,\dots,s_n) = -\sum_{i=1}^{m}(s_i/s)\log_2(s_i/s)$$

❖ Calculamos el **valor esperado de información para cada distribución**:

a. Comencemos analizando el atributo ***Ciudad de nacimiento***

El atributo ***Ciudad de Nacimiento*** tiene la siguiente distribución de datos:

Para **Rosario**: $S_{Rosario} = 7$ $S_{Rosario_{SI}} = 5$ $S_{Rosario_{NO}} = 2$

$$I(S_{Rosario_{SI}}, S_{Rosario_{NO}}) = -\left[\left(\frac{5}{7}\log_2 \bullet \frac{5}{7}\right) + \left(\frac{2}{7}\log_2 \bullet \frac{2}{7}\right)\right] = -[(-0.3467) + (-0.5164)] = \boxed{0.8631}$$

Para **Santa Rosa**: $S_{SantaRosa} = 8$ $S_{SantaRosa_{SI}} = 3$ $S_{SantaRosa_{NO}} = 5$

$$I(S_{SRosa_{SI}}, S_{SRosa_{NO}}) = -\left[\left(\frac{3}{8}\log_2 \bullet \frac{3}{8}\right) + \left(\frac{5}{8}\log_2 \bullet \frac{5}{8}\right)\right] = -[(-0.5306) + (-0.4238)] = \boxed{0.9544}$$

b. Comencemos analizando el atributo ***Edad***

El atributo ***Edad*** tiene la siguiente distribución de datos:

Para **[1 a 12 años]**: $S_{1a12} = 3$ $S_{1a12_{SI}} = 3$ $S_{1a12_{NO}} = 0$

$$I(S_{1a12_{SI}}, S_{1a12_{NO}}) = -\left[\left(\frac{3}{3}\log_2 \bullet \frac{3}{3}\right) + \left(\frac{0}{3}\log_2 \bullet \frac{0}{3}\right)\right] = \boxed{0}$$

Para **[12 a 32 años]**: $S_{12a32} = 3$ $S_{12a32_{SI}} = 3$ $S_{12a32_{NO}} = 0$

$$I(S_{12a32_{SI}}, S_{12a32_{NO}}) = -\left[\left(\frac{3}{3}\log_2 \bullet \frac{3}{3}\right) + \left(\frac{0}{3}\log_2 \bullet \frac{0}{3}\right)\right] = \boxed{0}$$

Para **[32 a 45 años]**: $S_{32a45} = 3$ $S_{32a45_{SI}} = 1$ $S_{32a45_{NO}} = 2$

$$I(S_{32a45_{SI}}, S_{32a45_{NO}}) = -\left[\left(\frac{1}{3}\log_2 \bullet \frac{1}{3}\right) + \left(\frac{2}{3}\log_2 \bullet \frac{2}{3}\right)\right] = \boxed{0.9183}$$

Para **[45 a 58 años]**: $S_{45a58} = 3$ $S_{45a58_{SI}} = 0$ $S_{45a58_{NO}} = 3$

$$I(S_{45a58_{SI}}, S_{45a58_{NO}}) = -\left[\left(\frac{0}{3}\log_2 \bullet \frac{0}{3}\right) + \left(\frac{3}{3}\log_2 \bullet \frac{3}{3}\right)\right] = \boxed{0}$$

Para **[58 a 96 años]**: $S_{58a96} = 3$ $S_{58a96_{SI}} = 1$ $S_{58a96_{NO}} = 2$

$$I(S_{58a96_{SI}}, S_{58a96_{NO}}) = -\left[\left(\frac{1}{3}\log_2 \bullet \frac{1}{3}\right) + \left(\frac{2}{3}\log_2 \bullet \frac{2}{3}\right)\right] = \boxed{0.9183}$$

c. Comencemos analizando el atributo ***Sexo***

El atributo ***Sexo*** tiene la siguiente distribución de datos:

Para **F "femenino"**: $S_F = 7$ $S_{F_{SI}} = 3$ $S_{F_{NO}} = 4$

$$I(S_{F_{SI}}, S_{F_{NO}}) = -\left[\left(\frac{3}{7}\log_2 \bullet \frac{3}{7}\right) + \left(\frac{4}{7}\log_2 \bullet \frac{4}{7}\right)\right] = \boxed{0.9852}$$

Para **M "masculino"**: $S_M = 8$ $S_{M_{SI}} = 5$ $S_{M_{NO}} = 3$

$$I(S_{M_{SI}}, S_{M_{NO}}) = -\left[\left(\frac{5}{8}\log_2 \bullet \frac{5}{8}\right) + \left(\frac{3}{8}\log_2 \bullet \frac{3}{8}\right)\right] = \boxed{0.9544}$$

$$E(A) = \sum_{j=1}^{v}\left[\frac{s_{1j}+\dots+s_{mj}}{s}\right] I(s_1,\dots,s_n)$$

❖ Calculamos la **información esperada** *necesaria para clasificar un determinado ejemplo si los ejemplos están particionados* según:

a. Comencemos analizando el atributo ***Ciudad de nacimiento***

$$E(Ciudad) = \left[\left(\frac{S_{Rosario}}{S} \bullet I(S_{Rosario})\right) + \left(\frac{S_{SRosa}}{S} \bullet I(S_{SRosa})\right)\right] =$$

$$= \left[\left(\frac{7}{15} \bullet 0.8631\right) + \left(\frac{8}{15} \bullet 0.9544\right)\right] = \boxed{0.9118}$$

b. Comencemos analizando el atributo ***Edad***

$$E(Edad) = \left[\left(\frac{S_{1a12}}{S} \bullet I(S_{1a12})\right) + \left(\frac{S_{12a32}}{S} \bullet I(S_{12a32})\right) + \left(\frac{S_{32a45}}{S} \bullet I(S_{32a45})\right) + \left(\frac{S_{45a58}}{S} \bullet I(S_{45a58})\right) + \left(\frac{S_{58a96}}{S} \bullet I(S_{58a96})\right)\right] =$$

$$= \left[\left(\frac{3}{15} \bullet 0\right) + \left(\frac{3}{15} \bullet 0\right) + \left(\frac{3}{15} \bullet 0.9183\right) + \left(\frac{3}{15} \bullet 0\right) + \left(\frac{3}{15} \bullet 0.9183\right)\right] = \boxed{0.3673}$$

c. Comencemos analizando el atributo ***Sexo***

$$E(\text{Sexo}) = \left[\left(\frac{S_F}{S} \bullet I(S_F)\right) + \left(\frac{S_M}{S} \bullet I(S_M)\right)\right] =$$

$$= \left[\left(\frac{7}{15} \bullet 0.9852\right) + \left(\frac{8}{15} \bullet 0.9544\right)\right] = \boxed{0.9688}$$

3. Ahora calculamos la ***GANANCIA DE INFORMACIÓN*** de dicha partición es:

$Ganancia(A) = I(s_i,\ldots,s_m) - E(A)$

a. Comencemos analizando el atributo ***Ciudad de nacimiento***

$$G(\text{Ciudad}) = I(S_{SI}, S_{NO}) - E(\text{Ciudad}) = 0.9968 - 0.9118 = \boxed{0.0850}$$

b. Comencemos analizando el atributo ***Edad***

$$G(\text{Edad}) = I(S_{SI}, S_{NO}) - E(\text{Edad}) = 0.9968 - 0.3673 = \boxed{0.6295}$$

c. Comencemos analizando el atributo ***Sexo***

$$G(\text{Sexo}) = I(S_{SI}, S_{NO}) - E(\text{Sexo}) = 0.9968 - 0.9688 = \boxed{0.0280}$$

4. Ahora calculamos la ***PROPORCIÓN DE GANANCIA DE INFORMACIÓN*** de dicha partición es:

❖ Calculamos la **información esperada de la división:** $I\_división(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \bullet Log_2\left(\frac{|T_i|}{|T|}\right)$

a. Comencemos analizando el atributo ***Ciudad de nacimiento***

$$I_{división}(\text{Ciudad}) = -\left[\left(\frac{S_{Rosario}}{S} \bullet \log_2 \frac{S_{Rosario}}{S}\right) + \left(\frac{S_{SRosa}}{S} \bullet \log_2 \frac{S_{SRosa}}{S}\right)\right] =$$

$$= -\left[\left(\frac{7}{15} \bullet \log_2 \frac{7}{15}\right) + \left(\frac{8}{15} \bullet \log_2 \frac{8}{15}\right)\right] = \boxed{0.9968}$$

b. Comencemos analizando el atributo ***Edad***

$$I_{división}(\text{Edad}) = -\left[\left(\frac{S_{1a12}}{S} \bullet \log_2 \frac{S_{1a12}}{S}\right) + \ldots + \left(\frac{S_{58a96}}{S} \bullet \log_2 \frac{S_{58a96}}{S}\right)\right] =$$

$$= -\left[\left(\frac{3}{15} \bullet \log_2 \frac{3}{15}\right) + \left(\frac{3}{15} \bullet \log_2 \frac{3}{15}\right) + \left(\frac{3}{15} \bullet \log_2 \frac{3}{15}\right) + \left(\frac{3}{15} \bullet \log_2 \frac{3}{15}\right) + \left(\frac{3}{15} \bullet \log_2 \frac{3}{15}\right)\right] = \boxed{2.3219}$$

c. Comencemos analizando el atributo ***Sexo***

$$I_{división}(\text{Sexo}) = \left[\left(\frac{S_F}{S} \bullet \log_2 \frac{S_F}{S}\right) + \left(\frac{S_M}{S} \bullet \log_2 \frac{S_M}{S}\right)\right] =$$

$$= \left[\left(\frac{7}{15} \bullet \log_2 \frac{7}{15}\right) + \left(\frac{8}{15} \bullet \log_2 \frac{8}{15}\right)\right] = \boxed{0.9968}$$

❖ Calculamos la **Proporción de ganacia:** $proporcion\_ganancia(X) = \frac{Ganancia\_I(T,X)}{I\_división(X)}$

a. Comencemos analizando el atributo ***Ciudad de nacimiento***

$$PG(\text{Ciudad}) = \frac{0.0850}{0.9968} = \boxed{0.0852}$$

b. Comencemos analizando el atributo ***Edad***

$$PG(\text{Edad}) = \frac{0.6295}{2.3219} = \boxed{0.2711}$$

c. Comencemos analizando el atributo ***Sexo***

$$PG(\text{Sexo}) = \frac{0.0280}{0.9968} = \boxed{0.0281}$$

Una vez que hemos calculado las ganancias y proporciones de ganancia para todos los atributos disponibles, elegimos el atributo según el cual dividiremos a este conjunto de datos. Tanto en el caso de la ganancia como en el de la proporción de ganancia, el mejor atributo para la división es aquel que la maximiza.

En este estudio, la división según el atributo Edad es la que mayor ganancia y proporción de ganancia ofrece. Esto significa que el nodo raíz del árbol será un nodo que evalúa el atributo Edad.

Nodo 0. EDAD.

2º Ahora analizamos cada conjunto de individuos de cada rango de edad.

Grupo: Edad de [1 a 12 años] → Clase Target "S"

Ciudad de nacimiento	Rangos Edad	Sexo	Target para campaña publicitaria (clase)
Rosario	[1 a 12 años]	F	S
Santa Rosa	[1 a 12 años]	M	S
Rosario	[1 a 12 años]	M	S

S es el conjunto de todos los ejemplos de edad entre 1 y 12 años donde de 3 individuos todos pertenecen a la clase "S".

Si recordamos que si todos los registros de S tienen el mismo valor para el atributo clasificador, creamos un único nodo con dicho valor; entonces dado que todos los ejemplos de S son de una única clase para esta rama del árbol creamos un nodo terminal (hoja) con la clase "S"

Grupo: Edad de [32 a 45 años] → Clase Target "N"

Ciudad de nacimiento	Rangos Edad	Sexo	Target para campaña publicitaria (clase)
Santa Rosa	[32 a 45 años]	F	N
Santa Rosa	[32 a 45 años]	F	N
Rosario	[32 a 45 años]	F	S

S es el conjunto de todos los ejemplos de edad entre 32 y 45 años de sexo Femenino, donde de 3 individuos 1 pertenecen a la clase "S" y 2 pertenece a la clase "N".

$\mathbf{S = 3 \quad S_{SI} = 1 \qquad S_{NO} = 2}$

$$I(S_S, S_N) = -\left[\left(\frac{1}{3}\log_2 \bullet \frac{1}{3}\right) + \left(\frac{2}{3}\log_2 \bullet \frac{2}{3}\right)\right] = 0.9183$$

El atributo ***Ciudad de Nacimiento*** tiene la siguiente distribución de datos:

Para **Rosario**: $\mathbf{S_{Rosario} = 1 \qquad S_{Rosario_{SI}} = 1}$

$$I(S_{Rosario_{SI}}, S_{Rosario_{NO}}) = -\left[\left(\frac{1}{1}\log_2 \bullet \frac{1}{1}\right) + \left(\frac{0}{1}\log_2 \bullet \frac{0}{1}\right)\right] = 0$$

Para **Santa Rosa**: $\mathbf{S_{SantaRosa} = 2 \qquad S_{SantaRosa_{NO}} = 2}$

$$I(S_{SRosa_{SI}}, S_{SRosa_{NO}}) = -\left[\left(\frac{0}{2}\log_2 \bullet \frac{0}{2}\right) + \left(\frac{2}{2}\log_2 \bullet \frac{2}{2}\right)\right] = 0$$

$$E(Ciudad) = \left[\left(\frac{S_{Rosario}}{S} \bullet I(S_{Rosario})\right) + \left(\frac{S_{SRosa}}{S} \bullet I(S_{SRosa})\right)\right] =$$

$$= \left[\left(\frac{1}{3} \bullet 0\right) + \left(\frac{2}{3} \bullet 0\right)\right] = 0$$

$$G(Ciudad) = I(S_{SI}, S_{NO}) - E(Ciudad) = 0.9183 - 0 = 0.9183$$

$$I_{división}(Ciudad) = -\left[\left(\frac{S_{Rosario}}{S} \bullet \log_2 \frac{S_{Rosario}}{S}\right) + \left(\frac{S_{SRosa}}{S} \bullet \log_2 \frac{S_{SRosa}}{S}\right)\right] =$$

$$= -\left[\left(\frac{1}{3} \bullet \log_2 \frac{1}{3}\right) + \left(\frac{2}{3} \bullet \log_2 \frac{2}{3}\right)\right] = 0.9183$$

$$PG(Ciudad) = \frac{0.9183}{0.9183} = 1$$

El atributo ***Sexo*** tiene la siguiente distribución de datos:

Para **F "femenino"**: $\mathbf{S_F = 3 \qquad S_{F_{SI}} = 1 \qquad S_{F_{NO}} = 2}$

$$I(S_{F_{SI}}, S_{F_{NO}}) = -\left[\left(\frac{1}{3}\log_2 \bullet \frac{1}{3}\right) + \left(\frac{2}{3}\log_2 \bullet \frac{2}{3}\right)\right] = 0.9183$$

Para **M "masculino"**: $\mathbf{S_M = 0}$

$$I(S_{M_{SI}}, S_{M_{NO}}) = 0$$

$$E(Sexo) = \left[\left(\frac{S_F}{S} \bullet I(S_F)\right) + \left(\frac{S_M}{S} \bullet I(S_M)\right)\right] =$$

$$= \left[\left(\frac{3}{3} \bullet 0.9183\right) + \left(\frac{0}{3} \bullet 0\right)\right] = 0.9183$$

$$I_{división}(Sexo) = \left[\left(\frac{S_F}{S} \bullet \log_2 \frac{S_F}{S}\right) + \left(\frac{S_M}{S} \bullet \log_2 \frac{S_M}{S}\right)\right] =$$

$$= -\left[\left(\frac{3}{3} \bullet \log_2 \frac{3}{3}\right) + \left(\frac{0}{3} \bullet \log_2 \frac{0}{3}\right)\right] = 0$$

$$G(Sexo) = I(S_{SI}, S_{NO}) - E(Sexo) = 0.9183 - 0.9183 = 0 \qquad PG(Sexo) = 0$$

Una vez que hemos calculado las ganancias y proporciones de ganancia para Ciudad y Sexo, elegimos el atributo según el cual dividiremos a este conjunto de datos. Tanto en el caso de la ganancia como en el de la proporción de ganancia, el mejor atributo para la división es aquel que la maximiza.

En este caso, la división según el atributo Ciudad es la que mayor ganancia y proporción de ganancia ofrece.

Esto significa que se genera un nodo que evalúa el atributo Ciudad.

Este nodo divide los datos en dos grupos: Rosarinos que son todos de la clase "S" y de Santa Rosa que son todos de la clase "N". Se genera los nodos terminales correspondientes a cada grupo.

Grupo: Edad de [32 a 45 años] y Ciudad "Rosario" → Clase Target "S"

Ciudad de nacimiento	Rangos Edad	Sexo	Target
Rosario	[32 a 45 años]	F	S

Si recordamos que si todos los registros de S tienen el mismo valor para el atributo clasificador, creamos un único nodo con dicho valor; entonces dado que todos los ejemplos de S son de una única clase para esta rama del árbol creamos un nodo terminal (hoja) con la clase "S"

Grupo: Edad de [32 a 45 años] y Ciudad "Santa Rosa" → Clase Target "N"

Ciudad de nacimiento	Rangos Edad	Sexo	Target
Santa Rosa	[32 a 45 años]	F	N
Santa Rosa	[32 a 45 años]	F	N

Si recordamos que si todos los registros de S tienen el mismo valor para el atributo clasificador, creamos un único nodo con dicho valor; entonces dado que todos los ejemplos de S son de una única clase para esta rama del árbol creamos un nodo terminal (hoja) con la clase "N"

Grupo: Edad de [45 a 58 años] → Clase Target "N"

Ciudad de nacimiento	Rangos Edad	Sexo	Target para campaña publicitaria (clase)
Santa Rosa	[45 a 58 años]	F	N
Rosario	[45 a 58 años]	M	N
Santa Rosa	[45 a 58 años]	M	N

S es el conjunto de todos los ejemplos de edad entre 45 y 58 años, donde de 3 individuos todos pertenecen a la clase "N".

Si recordamos que si todos los registros de S tienen el mismo valor para el atributo clasificador, creamos un único nodo con dicho valor; entonces dado que todos los ejemplos de S son de una única clase para esta rama del árbol creamos un nodo terminal (hoja) con la clase "N"

Grupo: Edad de [58 a 96 años]

Ciudad de nacimiento	Rangos Edad	Sexo	Target para campaña publicitaria (clase)
Santa Rosa	[58 a 96 años]	F	N
Rosario	[58 a 96 años]	M	N
Rosario	[58 a 96 años]	F	S

S es el conjunto de todos los ejemplos de edad entre 58 y 96 años, donde de 3 individuos 1 pertenecen a la clase "S" y 2 pertenece a la clase "N".

Analizamos la ganancia y proporción de ganancia de Ciudad, Edad y Sexo para determinar cual de los 3 atributos particionera los datos.

$$S = 3 \quad S_{SI} = 1 \qquad S_{NO} = 2$$

$$I(S_S, S_N) = -\left[\left(\frac{1}{3}\log_2 \bullet \frac{1}{3}\right) + \left(\frac{2}{3}\log_2 \bullet \frac{2}{3}\right)\right] = 0.9183$$

El atributo ***Ciudad de Nacimiento*** tiene la siguiente distribución de datos:

Para **Rosario**: $S_{Rosario} = 2 \qquad S_{Rosario_{SI}} = 1 \qquad S_{Rosario_{NO}} = 1$

$$I(S_{Rosario_{SI}}, S_{Rosario_{NO}}) = -\left[\left(\frac{1}{2}\log_2 \bullet \frac{1}{2}\right) + \left(\frac{1}{2}\log_2 \bullet \frac{1}{2}\right)\right] = 1$$

Para **Santa Rosa**: $S_{SantaRosa} = 1 \qquad S_{SantaRosa_{NO}} = 1$

$$I(S_{SRosa_{SI}}, S_{SRosa_{NO}}) = -\left[\left(\frac{0}{1}\log_2 \bullet \frac{0}{1}\right) + \left(\frac{1}{1}\log_2 \bullet \frac{1}{1}\right)\right] = 0$$

$$E(Ciudad) = \left[\left(\frac{S_{Rosario}}{S} \bullet I(S_{Rosario})\right) + \left(\frac{S_{SRosa}}{S} \bullet I(S_{SRosa})\right)\right] =$$

$$= \left[\left(\frac{2}{3} \bullet 1\right) + \left(\frac{1}{3} \bullet 0\right)\right] = 0.6667$$

$$G(Ciudad) = I(S_{SI}, S_{NO}) - E(Ciudad) = 0.9183 - 0.6667 = 0.2516$$

$$I_{división}(Ciudad) = -\left[\left(\frac{S_{Rosario}}{S} \bullet \log_2 \frac{S_{Rosario}}{S}\right) + \left(\frac{S_{SRosa}}{S} \bullet \log_2 \frac{S_{SRosa}}{S}\right)\right] =$$

$$= -\left[\left(\frac{2}{3} \bullet \log_2 \frac{2}{3}\right) + \left(\frac{1}{3} \bullet \log_2 \frac{1}{3}\right)\right] = 0.9183$$

$$PG(Ciudad) = \frac{0.2516}{0.9183} = 0.2740$$

El atributo ***Sexo*** tiene la siguiente distribución de datos:

Para **F "femenino"**: $S_F = 2 \qquad S_{F_{SI}} = 1 \qquad S_{F_{NO}} = 1$

$$I(S_{F_{SI}}, S_{F_{NO}}) = -\left[\left(\frac{1}{2}\log_2 \bullet \frac{1}{2}\right) + \left(\frac{1}{2}\log_2 \bullet \frac{1}{2}\right)\right] = 1$$

Para **M "masculino"**: $S_M = 1 \qquad S_{M_{NO}} = 1$

$$I(S_{M_{SI}}, S_{M_{NO}}) = -\left[\left(\frac{0}{1}\log_2 \bullet \frac{0}{1}\right) + \left(\frac{1}{1}\log_2 \bullet \frac{1}{1}\right)\right] = 0$$

$$E(\text{Sexo}) = \left[\left(\frac{S_F}{S} \bullet I(S_F)\right) + \left(\frac{S_M}{S} \bullet I(S_M)\right)\right] =$$

$$= \left[\left(\frac{2}{3} \bullet 1\right) + \left(\frac{1}{3} \bullet 0\right)\right] = 0.6667$$

$$G(\text{Sexo}) = I(S_{SI}, S_{NO}) - E(\text{Sexo}) = 0.9183 - 0.6667 = 0.2516$$

$$I_{\text{división}}(\text{Sexo}) = \left[\left(\frac{S_F}{S} \bullet \log_2 \frac{S_F}{S}\right) + \left(\frac{S_M}{S} \bullet \log_2 \frac{S_M}{S}\right)\right] =$$

$$= -\left[\left(\frac{2}{3} \bullet \log_2 \frac{2}{3}\right) + \left(\frac{1}{3} \bullet \log_2 \frac{1}{3}\right)\right] = 0.9183$$

$$PG(\text{Sexo}) = \frac{0.2516}{0.9183} = 0.2740$$

Ambos atributos tienen la misma proporción de ganancia, elijo el primero analizado, Ciudad, para dividir los datos.

De esta manera genero dos nuevos nodos para Rosario y Santa Rosa.

Grupo: Edad de [58 a 96 años] y Ciudad "Santa Rosa" → Clase Target "N"

Ciudad de nacimiento	Rangos Edad	Sexo	Target
Santa Rosa	[58 a 96 años]	F	N

Dado que todos los ejemplos de S son de una única clase para esta rama del árbol creamos un nodo terminal (hoja) con la clase "S".

Grupo: Edad de [58 a 96 años] y Ciudad "Rosario"

Ciudad de nacimiento	Rangos Edad	Sexo	Target
Rosario	[58 a 96 años]	M	N
Rosario	[58 a 96 años]	F	S

S es el conjunto de todos los ejemplos de edad entre 58 y 96 años donde de 2 individuos 1 pertenecen a la clase "S" y el otro a la clase "N".

Por el sexo puedo dividir este grupo en dos nodos terminales:

Grupo: Edad de [58 a 96 años] y Ciudad "Rosario" y Sexo "Femenino" → Clase Target "S"

Ciudad de nacimiento	Rangos Edad	Sexo	Target
Rosario	[58 a 96 años]	F	S

Dado que todos los ejemplos de S son de una única clase para esta rama del árbol creamos un nodo terminal (hoja) con la clase "S".

Grupo: Edad de [58 a 96 años] y Ciudad "Rosario" y Sexo "Masculino" → Clase Target "N"

Ciudad de nacimiento	Rangos Edad	Sexo	Target
Rosario	[58 a 96 años]	M	N

Dado que todos los ejemplos de S son de una única clase para esta rama del árbol creamos un nodo terminal (hoja) con la clase "N".

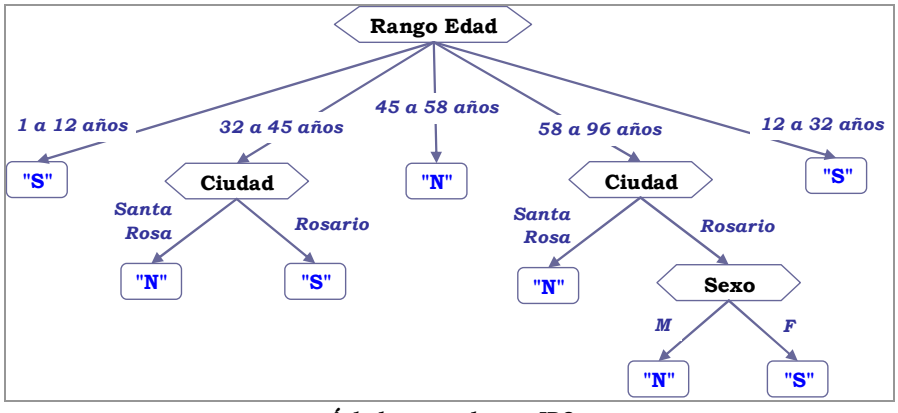
Grupo: Edad de [12 a 32 años] → Clase Target "S"

Ciudad de nacimiento	Rangos Edad	Sexo	Target para campaña publicitaria (clase)
Santa Rosa	[12 a 32 años]	M	S
Santa Rosa	[12 a 32 años]	M	S
Rosario	[12 a 32 años]	M	S

S es el conjunto de todos los ejemplos de edad entre 12 y 32 años donde de 3 individuos todos pertenecen a la clase "S".

Si recordamos que si todos los registros de S tienen el mismo valor para el atributo clasificador, creamos un único nodo con dicho valor; entonces dado que todos los ejemplos de S son de una única clase para esta rama del árbol creamos un nodo terminal (hoja) con la clase "S".

Generación de Reglas a partir del árbol:



Árbol generado por ID3

Reglas	S	C
Si Edad [1 a 12 años] Entonces *Target (clase)*=S	3/15	3/3
Si Edad [45 a 58 años] Entonces *Target (clase)*=N	3/15	3/3
Si Edad [12 a 32 años] Entonces *Target (clase)*=S	3/15	3/3
Si Ciudad "Santa Rosa" Y Edad [32 a 45 años] Entonces *Target(clase)*=N	2/15	2/2
Si Ciudad "Rosario" Y Edad [32 a 45 años] Entonces *Target (clase)*=S	1/15	1/1
Si Ciudad "Santa Rosa" Y Edad [58 a 96 años] Entonces *Target (clase)*=N	1/15	1/1
Si Sexo "F" Y Ciudad "Rosario" Y Edad [58 a 96 años] Entonces *Target (clase)*=S	1/15	1/1
Si Sexo "M" Y Ciudad "Rosario" Y Edad [58 a 96 años] Entonces *Target (clase)*=N	1/15	1/1

b. CONJUNTO DE REGLAS DE DECISIÓN UTILIZANDO EL ALGORITMO PRISM

Target	Ciudad de nacimiento	Sexo	Rangos Edad
S	Rosario	F	[1 a 12 años]
N	Santa Rosa	F	[32 a 45 años]
N	Santa Rosa	F	[45 a 58 años]
N	Santa Rosa	F	[58 a 96 años]
S	Rosario	F	[58 a 96 años]
N	Santa Rosa	F	[32 a 45 años]
S	Rosario	F	[32 a 45 años]
S	Santa Rosa	M	[12 a 32 años]
S	Santa Rosa	M	[12 a 32 años]
N	Rosario	M	[45 a 58 años]
N	Santa Rosa	M	[45 a 58 años]
S	Santa Rosa	M	[1 a 12 años]
S	Rosario	M	[1 a 12 años]
S	Rosario	M	[12 a 32 años]
N	Rosario	M	[58 a 96 años]

Para cada clase C
  Sea E = ejemplos de entrenamiento
  Mientras E tenga ejemplos de clase C
    Crea una regla R con LHS vacío y clase C
    Until R es perfecta, do
      Para cada atributo A no incluido en R y cada valor v,
        Considera añadir la condición $A = v$ al LHS de R
        Selecciona el par $A = v$ que maximice p/t
        (en caso de empates, selecciona la que tenga p mayor)
      Añade $A = v$ a R
    Elimina de E los ejemplos cubiertos por R

La condición para generar la REGLA es MAXIMIZAR p/t

instancias que cumplen el valor **v** de **A**

instancias que cumplen la clase i

Reglas para la clase Target "S"

Target	Ciudad de nacimiento	Sexo	Rangos Edad	Edad
S	Rosario	F	[58 a 96 años]	96
S	Rosario	F	[32 a 45 años]	40
S	Santa Rosa	M	[12 a 32 años]	23
S	Santa Rosa	M	[12 a 32 años]	25
S	Rosario	M	[12 a 32 años]	32
S	Rosario	F	[1 a 12 años]	12
S	Santa Rosa	M	[1 a 12 años]	1
S	Rosario	M	[1 a 12 años]	10
N	Santa Rosa	F	[58 a 96 años]	78
N	Rosario	M	[58 a 96 años]	67
N	Santa Rosa	F	[45 a 58 años]	55
N	Rosario	M	[45 a 58 años]	53
N	Santa Rosa	M	[45 a 58 años]	58
N	Santa Rosa	F	[32 a 45 años]	34
N	Santa Rosa	F	[32 a 45 años]	45

Si ? entonces Target = ”S”	**p/t**
Atributo Ciudad	
Si Ciudad = "Rosario" entonces Target = ”S”	5/7
Si Ciudad = "Santa Rosa" entonces Target = ”S”	3/8
Atributo Edad	
Si Edad = [1 a 12 años] entonces Target = ”S”	**3/3**
Si Edad = [12 a 32 años] entonces Target = ”S”	**3/3**
Si Edad = [32 a 45 años] entonces Target = ”S”	1/3
Si Edad = [45 a 58 años] entonces Target = ”S”	0/3
Si Edad = [58 a 96 años] entonces Target = ”S”	1/3
Atributo Sexo	
Si Sexo = "F" entonces Target = ”S”	3/7
Si Sexo = "M" entonces Target = ”S”	5/8

Si Edad = [1 a 12 años] Y ? entonces Target = ”S”	**p/t**
Atributo Ciudad	
Si Edad = [1 a 12 años] Y Ciudad = "Rosario" entonces Target = ”S”	**2/2**
Si Edad = [1 a 12 años] Y Ciudad = "Santa Rosa" entonces Target = ”S”	1/1
Atributo Sexo	
Si Edad = [1 a 12 años] Y Sexo = "F" entonces Target = ”S”	1/1
Si Edad = [1 a 12 años] Y Sexo = "M" entonces Target = ”S”	**2/2**

Elimino los ejemplos de la primer regla **Si Edad = [1 a 12 años] entonces Target = ”S”**

Target	Ciudad de nacimiento	Sexo	Rangos Edad	Edad
S	Rosario	F	[58 a 96 años]	96
S	Rosario	F	[32 a 45 años]	40
S	Santa Rosa	M	[12 a 32 años]	23
S	Santa Rosa	M	[12 a 32 años]	25
S	Rosario	M	[12 a 32 años]	32
N	Santa Rosa	F	[58 a 96 años]	78
N	Rosario	M	[58 a 96 años]	67
N	Santa Rosa	F	[45 a 58 años]	55
N	Rosario	M	[45 a 58 años]	53
N	Santa Rosa	M	[45 a 58 años]	58
N	Santa Rosa	F	[32 a 45 años]	34
N	Santa Rosa	F	[32 a 45 años]	45

Si ? entonces Target = ”S”	**p/t**
Atributo Ciudad	
Si Ciudad = "Rosario" entonces Target = ”S”	3/5
Si Ciudad = "Santa Rosa" entonces Target = ”S”	2/7
Atributo Edad	
Si Edad = [12 a 32 años] entonces Target = ”S”	**3/3**
Si Edad = [32 a 45 años] entonces Target = ”S”	1/3
Si Edad = [45 a 58 años] entonces Target = ”S”	0/3
Si Edad = [58 a 96 años] entonces Target = ”S”	1/3
Atributo Sexo	
Si Sexo = "F" entonces Target = ”S”	2/6
Si Sexo = "M" entonces Target = ”S”	3/6

Si Edad = [12 a 32 años] Y ? entonces Target = ”S”	**p/t**
Atributo Ciudad	
Si Edad = [12 a 32 años] Y Ciudad = "Rosario" entonces Target = ”S”	1/1
Si Edad = [12 a 32 años] Y Ciudad = "Santa Rosa" entonces Target = ”S”	2/2
Atributo Sexo	
Si Edad = [12 a 32 años] Y Sexo = "F" entonces Target = ”S”	0
Si Edad = [12 a 32 años] Y Sexo = "M" entonces Target = ”S”	**3/3**

Si Edad = [12 a 32 años] Y Sexo = "M" Y ? entonces Target = ”S”	**p/t**
Atributo Ciudad	
Si Edad = [12 a 32 años] Y Sexo = "M" Y Ciudad = "Rosario" entonces Target = ”S”	1/1
Si Edad = [12 a 32 años] Y Sexo = "M" Y Ciudad = "Santa Rosa" entonces Target = ”S”	**2/2**

Elimino los ejemplos de la regla **Si Edad = [12 a 32 años] entonces Target = ”S”**

Target	Ciudad de nacimiento	Sexo	Rangos Edad	Edad
S	Rosario	F	[58 a 96 años]	96
S	Rosario	F	[32 a 45 años]	40
N	Santa Rosa	F	[58 a 96 años]	78
N	Rosario	M	[58 a 96 años]	67
N	Santa Rosa	F	[45 a 58 años]	55
N	Rosario	M	[45 a 58 años]	53
N	Santa Rosa	M	[45 a 58 años]	58
N	Santa Rosa	F	[32 a 45 años]	34
N	Santa Rosa	F	[32 a 45 años]	45

Si ? entonces Target = ”S”	**p/t**
Atributo Ciudad	
Si Ciudad = "Rosario" entonces Target = ”S”	**2/4**
Si Ciudad = "Santa Rosa" entonces Target = ”S”	0/5
Atributo Edad	
Si Edad = [32 a 45 años] entonces Target = ”S”	1/3
Si Edad = [45 a 58 años] entonces Target = ”S”	0/3
Si Edad = [58 a 96 años] entonces Target = ”S”	1/3
Atributo Sexo	
Si Sexo = "F" entonces Target = ”S”	2/6
Si Sexo = "M" entonces Target = ”S”	0/3

Si Ciudad = "Rosario" Y ? entonces Target = ”S”	**p/t**
Atributo Edad	
Si Ciudad = "Rosario" Y Edad = [32 a 45 años] entonces Target = ”S”	1/1
Si Ciudad = "Rosario" Y Edad = [45 a 58 años] entonces Target = ”S”	0/1
Si Ciudad = "Rosario" Y Edad = [58 a 96 años] entonces Target = ”S”	1/2
Atributo Sexo	
Si Ciudad = "Rosario" Y Sexo = "F" entonces Target = ”S”	**2/2**
Si Ciudad = "Rosario" Y Sexo = "M" entonces Target = ”S”	0/2

Reglas para la clase Target “N”

Target	Ciudad de nacimiento	Sexo	Rangos Edad	Edad
S	Rosario	F	[58 a 96 años]	96
S	Rosario	F	[32 a 45 años]	40
S	Santa Rosa	M	[12 a 32 años]	23
S	Santa Rosa	M	[12 a 32 años]	25
S	Rosario	M	[12 a 32 años]	32
S	Rosario	F	[1 a 12 años]	12
S	Santa Rosa	M	[1 a 12 años]	1
S	Rosario	M	[1 a 12 años]	10
N	Santa Rosa	F	[58 a 96 años]	78
N	Rosario	M	[58 a 96 años]	67
N	Santa Rosa	F	[45 a 58 años]	55
N	Rosario	M	[45 a 58 años]	53
N	Santa Rosa	M	[45 a 58 años]	58
N	Santa Rosa	F	[32 a 45 años]	34
N	Santa Rosa	F	[32 a 45 años]	45

Si ? entonces Target = ”N”	**p/t**
Atributo Ciudad	
Si Ciudad = "Rosario" entonces Target = ”N”	2/7
Si Ciudad = "Santa Rosa" entonces Target = ”N”	5/8
Atributo Edad	
Si Edad = [1 a 12 años] entonces Target = ”N”	0/3
Si Edad = [12 a 32 años] entonces Target = ”N”	0/3
Si Edad = [32 a 45 años] entonces Target = ”N”	2/3
Si Edad = [45 a 58 años] entonces Target = ”N”	**3/3**
Si Edad = [58 a 96 años] entonces Target = ”N”	2/3
Atributo Sexo	
Si Sexo = "F" entonces Target = ”N”	4/7
Si Sexo = "M" entonces Target = ”N”	3/8

Si Edad = [45 a 58 años] Y ? entonces Target = ”N”	**p/t**
Atributo Ciudad	
Si Edad = [45 a 58 años] Y Ciudad = "Rosario" entonces Target = ”N”	1/1
Si Edad = [45 a 58 años] Y Ciudad = "Santa Rosa" entonces Target = ”N”	**2/2**
Atributo Sexo	
Si Edad = [45 a 58 años] Y Sexo = "F" entonces Target = ”N”	1/1
Si Edad = [45 a 58 años] Y Sexo = "M" entonces Target = ”N”	**2/2**

Elimino los ejemplos de la regla **Si Edad = [45 a 58 años] entonces Target = ”N”**

Target	Ciudad de nacimiento	Sexo	Rangos Edad
S	Rosario	F	[58 a 96 años]
S	Rosario	F	[32 a 45 años]
S	Santa Rosa	M	[12 a 32 años]
S	Santa Rosa	M	[12 a 32 años]
S	Rosario	M	[12 a 32 años]
S	Rosario	F	[1 a 12 años]
S	Santa Rosa	M	[1 a 12 años]
S	Rosario	M	[1 a 12 años]
N	Santa Rosa	F	[58 a 96 años]
N	Rosario	M	[58 a 96 años]
N	Santa Rosa	F	[32 a 45 años]
N	Santa Rosa	F	[32 a 45 años]

Si ? entonces Target = ”N”	**p/t**
Atributo Ciudad	
Si Ciudad = "Rosario" entonces Target = ”N”	1/6
Si Ciudad = "Santa Rosa" entonces Target = ”N”	3/6
Atributo Edad	
Si Edad = [1 a 12 años] entonces Target = ”N”	0/3
Si Edad = [12 a 32 años] entonces Target = ”N”	0/3
Si Edad = [32 a 45 años] entonces Target = ”N”	**2/3**
Si Edad = [58 a 96 años] entonces Target = ”N”	**2/3**
Atributo Sexo	
Si Sexo = "F" entonces Target = ”N”	3/6
Si Sexo = "M" entonces Target = ”N”	1/6

Si Edad = [32 a 45 años] Y ? entonces Target = ”N”	**p/t**
Atributo Ciudad	
Si Edad = [32 a 45 años] Y Ciudad = "Rosario" entonces Target = ”N”	0/1
Si Edad = [32 a 45 años] Y Ciudad = "Santa Rosa" entonces Target = ”N”	**2/2**
Atributo Sexo	
Si Edad = [32 a 45 años] Y Sexo = "F" entonces Target = ”N”	2/3
Si Edad = [32 a 45 años] Y Sexo = "M" entonces Target = ”N”	0

Si Edad = [32 a 45 años] Y Ciudad = "Santa Rosa" Y ? entonces Target = ”N”	**p/t**
Atributo Sexo	
Si Edad = [32 a 45 años] Y Ciudad = "Santa Rosa" Y Sexo = "F" entonces Target = ”N”	**2/2**
Si Edad = [32 a 45 años] Y Ciudad = "Santa Rosa" Y Sexo = "M" entonces Target = ”N”	0

Elimino los ejemplos de la regla **Si Edad = [32 a 45 años] entonces Target = ”N”**

Target	Ciudad de nacimiento	Sexo	Rangos Edad	Edad
S	Rosario	F	[58 a 96 años]	96
S	Rosario	F	[32 a 45 años]	40
S	Santa Rosa	M	[12 a 32 años]	23
S	Santa Rosa	M	[12 a 32 años]	25
S	Rosario	M	[12 a 32 años]	32
S	Rosario	F	[1 a 12 años]	12
S	Santa Rosa	M	[1 a 12 años]	1
S	Rosario	M	[1 a 12 años]	10
N	Santa Rosa	F	[58 a 96 años]	78
N	Rosario	M	[58 a 96 años]	67

Si ? entonces Target = ”N”	**p/t**
Atributo Ciudad	
Si Ciudad = "Rosario" entonces Target = ”N”	1/5
Si Ciudad = "Santa Rosa" entonces Target = ”N”	1/4
Atributo Edad	
Si Edad = [1 a 12 años] entonces Target = ”N”	0/3
Si Edad = [12 a 32 años] entonces Target = ”N”	0/3
Si Edad = [32 a 45 años] entonces Target = ”N”	0/1
Si Edad = [58 a 96 años] entonces Target = ”N”	**2/3**
Atributo Sexo	
Si Sexo = "F" entonces Target = ”N”	1/3
Si Sexo = "M" entonces Target = ”N”	1/6

Si Edad = [58 a 96 años] Y ? entonces Target = ”N”	**p/t**
Atributo Ciudad	
Si Edad = [58 a 96 años] Y Ciudad = "Rosario" entonces Target = ”N”	1/2
Si Edad = [58 a 96 años] Y Ciudad = "Santa Rosa" entonces Target = ”N”	**1/1**
Atributo Sexo	
Si Edad = [58 a 96 años] Y Sexo = "F" entonces Target = ”N”	1/2
Si Edad = [58 a 96 años] Y Sexo = "M" entonces Target = ”N”	**1/1**

Si Edad = [58 a 96 años] Y Ciudad = "Santa Rosa" Y ? entonces Target = ”N”	**p/t**
Atributo Sexo	
Si Edad = [58 a 96 años] Y Ciudad = "Santa Rosa" Y Sexo = "F" entonces Target = ”N”	**1/1**
Si Edad = [58 a 96 años] Y Ciudad = "Santa Rosa" Y Sexo = "M" entonces Target = ”N”	0

Reglas generadas:

Reglas	p/t (Confianza)	S
Si Edad = [1 a 12 años] entonces Target = "S"	3/3	3/15
Si Edad = [12 a 32 años] entonces Target = "S"	3/3	3/15
Si Edad = [12 a 32 años] Y Sexo = "M" entonces Target = "S"	3/3	3/15
Si Edad = [45 a 58 años] entonces Target = "N"	3/3	3/15
Si Edad = [1 a 12 años] Y Ciudad = "Rosario" entonces Target = "S"	2/2	2/15
Si Edad = [1 a 12 años] Y Sexo = "M" entonces Target = "S"	2/2	2/15
Si Edad = [12 a 32 años] Y Sexo = "M" Y Ciudad = "Santa Rosa" entonces Target = "S"	2/2	2/15
Si Ciudad = "Rosario" Y Sexo = "F" entonces Target = "S"	2/2	2/15
Si Edad = [45 a 58 años] Y Ciudad = "Santa Rosa" entonces Target = "N"	2/2	2/15
Si Edad = [45 a 58 años] Y Sexo = "M" entonces Target = "N"	2/2	2/15
Si Edad = [32 a 45 años] Y Ciudad = "Santa Rosa" entonces Target = "N"	2/2	2/15
Si Edad = [32 a 45 años] Y Ciudad = "Santa Rosa" Y Sexo = "F" entonces Target = "N"	2/2	2/15
Si Edad = [58 a 96 años] Y Ciudad = "Santa Rosa" entonces Target = "N"	1/1	1/15
Si Edad = [58 a 96 años] Y Sexo = "M" entonces Target = "N"	1/1	1/15
Si Edad = [58 a 96 años] Y Ciudad = "Santa Rosa" Y Sexo = "F" entonces Target = "N"	1/1	1/15
Si Edad = [32 a 45 años] entonces Target = "N"	2/3	2/15
Si Edad = [58 a 96 años] entonces Target = "N"	2/3	2/15
Si Ciudad = "Rosario" entonces Target = "S"	2/4	2/15

c. CONJUNTO DE REGLAS DE DECISIÓN UTILIZANDO EL ALGORITMO 1R

Ciudad de nacimiento	Rangos Edad	Sexo	Target (clase)
Santa Rosa	[32 a 45 años]	F	N
Santa Rosa	[32 a 45 años]	F	N
Rosario	[45 a 58 años]	M	N
Santa Rosa	[45 a 58 años]	F	N
Santa Rosa	[45 a 58 años]	M	N
Rosario	[58 a 96 años]	M	N
Santa Rosa	[58 a 96 años]	F	N
Santa Rosa	[1 a 12 años]	M	S
Rosario	[1 a 12 años]	M	S
Rosario	[1 a 12 años]	F	S
Santa Rosa	[12 a 32 años]	M	S
Santa Rosa	[12 a 32 años]	M	S
Rosario	[12 a 32 años]	M	S
Rosario	[32 a 45 años]	F	S
Rosario	[58 a 96 años]	F	S

Para cada atributo
  Para cada valor de cada atributo, crea una regla:
    cuenta cuántas veces ocurre la clase
    encuentra la clase más frecuente
    asigna esa clase a la regla
  Calcula el error de todas las reglas
Selecciona las reglas con el error más bajo

Cantidad de instancias que no pertenecen a la clase mayoritaria

Relación de Cantidad de ejemplos con ese valor pero que son de la clase contraria y la cantidad total que tiene ese valor

Cantidad de ejemplos con ese valor de esa clase,

Regla cuyo consecuente es el de más ejemplos para ese antecedente

Cantidad de ejemplos con ese valor pero que son de la clase contraria

Ciudad	S		
Santa Rosa	3	5	8
Rosario	5	2	7
	8	7	15

Reglas	Error		S	
SI Ciudad = "Santa Rosa" ENTONCES Target = "N"	3	38%	33%	62%
SI Ciudad = "Rosario" ENTONCES Target = "S"	**2**	29%	13%	71%
		promedio de error	3	

Edad	S	N	
[1 a 12 años]	3	0	3
[12 a 32 años]	3	0	3
[32 a 45 años]	1	2	3
[45 a 58 años]	0	3	3
[58 a 96 años]	1	2	3
	8	7	15

Reglas	Error		S	C
SI Edad = "[1 a 12 años]" ENTONCES Target = "S"	**0**	0%	20%	100%
SI Edad = "[12 a 32 años]" ENTONCES Target = "S"	**0**	0%	20%	100%
SI Edad = "[32 a 45 años]" ENTONCES Target = "N"	1	33%	13%	67%
SI Edad = "[45 a 58 años]" ENTONCES Target = "N"	**0**	0%	20%	100%
SI Edad = "[58 a 96 años]" ENTONCES Target = "N"	1	33%	13%	67%
		promedio de error	0	

Sexo	S	N	
F	3	4	7
M	5	3	8
	8	7	15

Reglas	Error		S	C
SI Sexo = "F" ENTONCES Target = "N"	3	43%	27%	57%
SI Sexo = "M" ENTONCES Target = "S"	**3**	38%	33%	63%
		promedio de error	3	

Reglas generadas:

Reglas	Error		S	C
SI Edad = "[1 a 12 años]" ENTONCES Target = "S"	**0**	0%	20%	100%
SI Edad = "[12 a 32 años]" ENTONCES Target = "S"	**0**	0%	20%	100%
SI Edad = "[32 a 45 años]" ENTONCES Target = "N"	1	33%	13%	67%
SI Edad = "[45 a 58 años]" ENTONCES Target = "N"	**0**	0%	20%	100%
SI Edad = "[58 a 96 años]" ENTONCES Target = "N"	1	33%	13%	67%

COMPARACIÓN DE LOS RESULTADOS

árbol de desición

ID3	error	S	C
Si Edad [1 a 12 años] Entonces Target (clase)=S	0%	20%	100%
Si Edad [45 a 58 años] Entonces Target (clase)=N	0%	20%	100%
Si Edad [12 a 32 años] Entonces Target (clase)=S	0%	20%	100%
Si Ciudad "Santa Rosa" Y Edad [32 a 45 años] Entonces Target(clase)=N	0%	13%	100%
Si Ciudad "Rosario" Y Edad [32 a 45 años] Entonces Target (clase)=S	0%	7%	100%
Si Ciudad "Santa Rosa" Y Edad [58 a 96 años] Entonces Target (clase)=N	0%	7%	100%
Si Sexo "F" Y Ciudad "Rosario" Y Edad [58 a 96 años] Entonces Target (clase)=S	0%	7%	100%
Si Sexo "M" Y Ciudad "Rosario" Y Edad [58 a 96 años] Entonces Target (clase)=N	0%	7%	100%

Algoritmos de Generación de Reglas

PRISM	error	S	C
Si Edad = [1 a 12 años] entonces Target = "S"	0%	20%	100%
Si Edad = [12 a 32 años] entonces Target = "S"	0%	20%	100%
Si Edad = [12 a 32 años] Y Sexo = "M" entonces Target = "S"	0%	20%	100%
Si Edad = [45 a 58 años] entonces Target = "N"	0%	20%	100%
Si Edad = [1 a 12 años] Y Ciudad = "Rosario" entonces Target = "S"	0%	13%	100%
Si Edad = [1 a 12 años] Y Sexo = "M" entonces Target = "S"	0%	13%	100%
Si Edad = [12 a 32 años] Y Sexo = "M" Y Ciudad = "Santa Rosa" entonces Target = "S"	0%	13%	100%
Si Ciudad = "Rosario" Y Sexo = "F" entonces Target = "S"	0%	13%	100%
Si Edad = [45 a 58 años] Y Ciudad = "Santa Rosa" entonces Target = "N"	0%	13%	100%
Si Edad = [45 a 58 años] Y Sexo = "M" entonces Target = "N"	0%	13%	100%
Si Edad = [32 a 45 años] Y Ciudad = "Santa Rosa" entonces Target = "N"	0%	13%	100%
Si Edad = [32 a 45 años] Y Ciudad = "Santa Rosa" Y Sexo = "F" entonces Target = "N"	0%	13%	100%
Si Edad = [58 a 96 años] Y Ciudad = "Santa Rosa" entonces Target = "N"	0%	7%	100%
Si Edad = [58 a 96 años] Y Sexo = "M" entonces Target = "N"	0%	7%	100%
Si Edad = [58 a 96 años] Y Ciudad = "Santa Rosa" Y Sexo = "F" entonces Target = "N"	0%	7%	100%
Si Edad = [32 a 45 años] entonces Target = "N"	33%	13%	67%
Si Edad = [58 a 96 años] entonces Target = "N"	33%	13%	67%
Si Ciudad = "Rosario" entonces Target = "S"	50%	13%	50%

IR	error	S	C
SI Edad = "[1 a 12 años]" ENTONCES Target = "S"	0%	20%	100%
SI Edad = "[12 a 32 años]" ENTONCES Target = "S"	0%	20%	100%
SI Edad = "[32 a 45 años]" ENTONCES Target = "N"	33%	13%	67%
SI Edad = "[45 a 58 años]" ENTONCES Target = "N"	0%	20%	100%
SI Edad = "[58 a 96 años]" ENTONCES Target = "N"	33%	13%	67%

Los algoritmos para la construcción de reglas, están enfocados en *crear una regla con la máxima exactitud*, en cambio, los algoritmos para la construcción de árboles se *concentran en examinar la separación entre clases*. En cada caso se trata de encontrar un atributo para dividir, pero el criterio de selección es diferente.

La *generación de reglas* se parece mucho a *los árboles de decisión*, excepto que las reglas generadas no dividen la base de datos en subconjuntos mutuamente excluidos. Ningún registro preparativo de la base de datos será clasificado según más de una regla en el algoritmo de árbol de decisión, pero determinado registro de aprendizaje puede emparejar cualquier número de reglas en el sistema de generación de reglas, incluida ni una sola regla.

Los árboles de decisión generan el más eficaz y menor posible conjunto de reglas que crea un modelo predictivo óptimo. Si dos predictores se sobrepusiesen, se elegiría el mejor de ellos. No obstante, en el sistema de generación de reglas los dos podrían estar bien representados y uno ser menos preciso o tener un poco menos de cobertura, lo que podría ser capturado como dato según la regla. Dado que un árbol de decisión se centra en un problema de predicción particular, no es tan eficaz como un sistema de generación de reglas en encontrar todas las posibles reglas "interesantes".

La generación de reglas por IR es más simple que PRISM, la idea es obtener reglas que prueban un solo par atributo-valor. Se prueban todos los pares atributo-valor y se selecciona el que ocasione el menor número de errores. Las Listas de decisión (PRISM) son conjuntos de reglas que son evaluadas en orden. Esto facilita la evaluación, aunque disminuye su modularidad. El tener reglas que pueden ser evaluadas independientemente de su orden (IR), permite que existan mas de una predicción para un solo ejemplo y dificulta el producir un solo resultado.

ATRIBUTOS CON DATOS NUMÉRICOS

Los árboles de decisión pueden generarse tanto a partir de atributos discretos como de atributos numéricos. Cuando se trabaja con atributos discretos, la partición del conjunto según el valor de un atributo es simple. Por ejemplo, agrupamos todos los animales que tengan pico, siendo *tiene_pico* un atributo y sus posibles valores *si* y *no*. En el caso de los atributos numéricos esta división no es tan fácil. Por ejemplo, si queremos partir los días de un mes en función a la cantidad de lluvia caída, es casi imposible que encontremos dos días con exactamente la misma cantidad de precipitaciones caídas.

Para los **Atributos continuos** se hace una división simple.

1. Primero se ordenan los atributos con respecto a la clase
2. Se sugieren puntos de partición en cada lugar donde cambia la clase.
3. Si existen dos clases diferentes con el mismo valor, se mueve el punto de partición a un punto intermedio con el siguiente valor hacia arriba o abajo dependiendo de donde está la clase mayoritaria. Se exige que cada partición tenga un número mínimo de ejemplos de la clase mayoritaria. Cuando hay clases adyacentes con la misma clase mayoritaria, estas se juntan.
4. A cada partición se le cálculo la proporción de ganancia, por lo que, cada posible partición entra en competencia con el resto de atributos y el de mayor proporción de ganancia es seleccionado. En el caso de Valores numéricos, pueden aparecer varias veces en una rama.

Ciudad de nacimiento	Edad	Sexo	Target para campaña publicitaria (clase)	Valores Medios
Santa Rosa	1	M	S	
Rosario	10	M	S	
Rosario	12	F	S	
Santa Rosa	23	M	S	
Santa Rosa	25	M	S	
Rosario	32	M	S	**33**
Santa Rosa	34	F	N	37
Rosario	40	F	S	42.5
Santa Rosa	45	F	N	
Rosario	53	M	N	
Santa Rosa	55	F	N	
Santa Rosa	58	M	N	
Rosario	67	M	N	
Santa Rosa	78	F	N	**87**
Rosario	96	F	S	

a. ÁRBOL DE DECISIÓN UTILIZANDO EL MÉTODO ID3

A partir de todos los datos disponibles, el ID3 analiza todas las divisiones posibles según los distintos atributos y calcula la ganancia y/o la proporción de ganancia.

1° Evaluamos cada atributo y seleccionamos el mejor. (Determino el nodo raíz)

S es el conjunto de todos los ejemplos donde de 15 individuos a 7 pertenecen a la clase "N" y "8" a la clase "S".

Analizamos la ganancia y proporción de ganancia de Ciudad, Edad (partición 1 y 2) y Sexo para determinar cual de los 3 atributos particionera los datos.

$S = 15 \qquad S_{SI} = 8 \qquad S_{NO} = 7$

$$I(S_S, S_N) = -\left[\left(\frac{8}{15}\log_2 \bullet \frac{8}{15}\right) + \left(\frac{7}{15}\log_2 \bullet \frac{7}{15}\right)\right] = -[(-0.4837) + (-0.5131)] = \boxed{0.9968}$$

a. Comencemos analizando el atributo ***Ciudad de nacimiento,*** tiene la siguiente distribución de datos:

Para **Rosario**: $S_{Rosario} = 7 \qquad S_{Rosario_{SI}} = 5 \qquad S_{Rosario_{NO}} = 2$

Para **Santa Rosa**: $S_{SantaRosa} = 8 \qquad S_{SantaRosa_{SI}} = 3 \qquad S_{SantaRosa_{NO}} = 5$

$$I(S_{Rosario_{SI}}, S_{Rosario_{NO}}) = -\left[\left(\frac{5}{7}\log_2 \bullet \frac{5}{7}\right) + \left(\frac{2}{7}\log_2 \bullet \frac{2}{7}\right)\right] = 0.8631$$

$$I(S_{SRosa_{SI}}, S_{SRosa_{NO}}) = -\left[\left(\frac{3}{8}\log_2 \bullet \frac{3}{8}\right) + \left(\frac{5}{8}\log_2 \bullet \frac{5}{8}\right)\right] = 0.9544$$

$$E(Ciudad) = \left[\left(\frac{7}{15} \bullet 0.8631\right) + \left(\frac{8}{15} \bullet 0.9544\right)\right] = 0.9118$$

$$I_{división}(Ciudad) = -\left[\left(\frac{7}{15} \bullet \log_2 \frac{7}{15}\right) + \left(\frac{8}{15} \bullet \log_2 \frac{8}{15}\right)\right] = 0.9968$$

$$G(Ciudad) = I(S_{SI}, S_{NO}) - E(Ciudad) = 0.9968 - 0.9118 = 0.0850$$

$$PG(Ciudad) = \frac{0.0850}{0.9968} = 0.0852$$

c. Comencemos analizando el atributo ***Sexo***, tiene la siguiente distribución de datos:

Para **F "femenino"**:	$S_F = 7$	$S_{F_{SI}} = 3$	$S_{F_{NO}} = 4$
Para **M "masculino"**:	$S_M = 8$	$S_{M_{SI}} = 5$	$S_{M_{NO}} = 3$

$$I(S_{F_{SI}}, S_{F_{NO}}) = -\left[\left(\frac{3}{7}\log_2 \bullet \frac{3}{7}\right) + \left(\frac{4}{7}\log_2 \bullet \frac{4}{7}\right)\right] = 0.9852$$

$$I(S_{M_{SI}}, S_{M_{NO}}) = -\left[\left(\frac{5}{8}\log_2 \bullet \frac{5}{8}\right) + \left(\frac{3}{8}\log_2 \bullet \frac{3}{8}\right)\right] = 0.9544$$

$$E(\text{Sexo}) = \left[\left(\frac{7}{15} \bullet 0.9852\right) + \left(\frac{8}{15} \bullet 0.9544\right)\right] = 0.9688$$

$$G(\text{Sexo}) = I(S_{SI}, S_{NO}) - E(\text{Sexo}) = 0.9968 - 0.9688 = 0.0280$$

$$I_{división}(\text{Sexo}) = \left[\left(\frac{7}{15} \bullet \log_2 \frac{7}{15}\right) + \left(\frac{8}{15} \bullet \log_2 \frac{8}{15}\right)\right] = 0.9968$$

$$PG(\text{Sexo}) = \frac{0.0280}{0.9968} = 0.0281$$

b. Comencemos analizando el atributo ***Edad***

El atributo ***Edad_1*** tiene la siguiente distribución de datos:

Para **[=<33 años]**:	$S_{=<33} = 6$	$S_{=<33_{SI}} = 6$	$S_{=<33_{NO}} = 0$
Para **[>33 años]**:	$S_{>33} = 9$	$S_{>33_{SI}} = 2$	$S_{>33_{NO}} = 7$

$$I(S_{=<33_{SI}}, S_{=<33_{NO}}) = -\left[\left(\frac{6}{6}\log_2 \bullet \frac{6}{6}\right) + 0\right] = 0$$

$$I(S_{>33_{SI}}, S_{>33_{NO}}) = -\left[\left(\frac{2}{9}\log_2 \bullet \frac{2}{9}\right) + \left(\frac{7}{9}\log_2 \bullet \frac{7}{9}\right)\right] = 0.7642$$

$$E(\text{Sexo}) = \left[\left(\frac{6}{15} \bullet 0\right) + \left(\frac{9}{15} \bullet 0.7642\right)\right] = 0.4585$$

$$G(\text{Sexo}) = I(S_{SI}, S_{NO}) - E(\text{Sexo}) = 0.9968 - 0.4585 = 0.5383$$

$$I_{división}(\text{Sexo}) = \left[\left(\frac{6}{15} \bullet \log_2 \frac{6}{15}\right) + \left(\frac{9}{15} \bullet \log_2 \frac{9}{15}\right)\right] = 0.9710$$

$$PG(\text{Sexo}) = \frac{0.5383}{0.9710} = 0.5544$$

El atributo ***Edad_2*** tiene la siguiente distribución de datos:

Para **[=<87 años]**:	$S_{=<87} = 14$	$S_{=<87_{SI}} = 7$	$S_{=<87_{NO}} = 7$
Para **[>87 años]**:	$S_{>87} = 1$	$S_{>87_{SI}} = 1$	$S_{>87_{NO}} = 0$

$$I(S_{=<87_{SI}}, S_{=<87_{NO}}) = -\left[\left(\frac{7}{14}\log_2 \bullet \frac{7}{14}\right) + \left(\frac{7}{14}\log_2 \bullet \frac{7}{14}\right)\right] = 1$$

$$I(S_{>87_{SI}}, S_{>87_{NO}}) = -\left[\left(\frac{1}{1}\log_2 \bullet \frac{1}{1}\right) + 0\right] = 0$$

$$E(\text{Sexo}) = \left[\left(\frac{14}{15} \bullet 1\right) + \left(\frac{1}{15} \bullet 0\right)\right] = 0.9333$$

$$G(\text{Sexo}) = I(S_{SI}, S_{NO}) - E(\text{Sexo}) = 0.9968 - .9333 = 0.0635$$

$$I_{división}(\text{Sexo}) = \left[\left(\frac{14}{15} \bullet \log_2 \frac{14}{15}\right) + \left(\frac{1}{15} \bullet \log_2 \frac{1}{15}\right)\right] = 0.9968$$

$$PG(\text{Sexo}) = \frac{0.0635}{0.9968} = 0.4322$$

Una vez que hemos calculado las ganancias y proporciones de ganancia para todos los atributos disponibles, elegimos el atributo según el cual dividiremos a este conjunto de datos. Tanto en el caso de la ganancia como en el de la proporción de ganancia, el mejor atributo para la división es aquel que la maximiza.

En este caso, la división según el atributo Edad_1 (particiones =<33 y >33) es la que mayor ganancia y proporción de ganancia ofrece. Esto significa que el nodo raíz del árbol será un nodo que evalúa el atributo Edad.

2º *Evaluamos la primera partición de edad =<33 años*

S es el conjunto de todos los ejemplos donde de 6 individuos a 6 pertenecen a la clase "S" y 0 a la clase "N". en este caso creamos un nodo terminal (hoja) con la clase "S".

Ciudad de nacimiento	Edad	Sexo	Target
Santa Rosa	1	M	S
Rosario	10	M	S
Rosario	12	F	S
Santa Rosa	23	M	S
Santa Rosa	25	M	S
Rosario	32	M	S

3º *Evaluamos la segunda partición de edad <33 años*

S es el conjunto de todos los ejemplos donde de 9 individuos a 2 pertenecen a la clase "S" y 7 a la clase "N".

Ciudad de nacimiento	Edad	Sexo	Target
Santa Rosa	34	F	N
Rosario	40	F	S
Santa Rosa	45	F	N
Rosario	53	M	N
Santa Rosa	55	F	N
Santa Rosa	58	M	N
Rosario	67	M	N
Santa Rosa	78	F	N
Rosario	96	F	S

Analizamos la ganancia y proporción de ganancia de Ciudad, Edad (partición 2) y Sexo para determinar cual de los 3 atributos particionara los datos.

$S = 9$ $S_{SI} = 2$ $S_{NO} = 7$

$$I(S_S, S_N) = -\left[\left(\frac{2}{9}\log_2 \bullet \frac{2}{9}\right) + \left(\frac{7}{9}\log_2 \bullet \frac{7}{9}\right)\right] = 0.7642$$

a. Comencemos analizando el atributo ***Ciudad de nacimiento,*** tiene la siguiente distribución de datos:

Para **Rosario**: $S_{Rosario} = 4$ $S_{Rosario_{SI}} = 2$ $S_{Rosario_{NO}} = 2$

Para **Santa Rosa**: $S_{SantaRosa} = 5$ $S_{SantaRosa_{SI}} = 0$ $S_{SantaRosa_{NO}} = 5$

$$I(S_{Rosario_{SI}}, S_{Rosario_{NO}}) = -\left[\left(\frac{2}{4}\log_2 \bullet \frac{2}{4}\right) + \left(\frac{2}{4}\log_2 \bullet \frac{2}{4}\right)\right] = 1$$

$$I(S_{SRosa_{SI}}, S_{SRosa_{NO}}) = -\left[0 + \left(\frac{5}{5}\log_2 \bullet \frac{5}{5}\right)\right] = 0$$

$$E(Ciudad) = \left[\left(\frac{4}{9} \bullet 1\right) + \left(\frac{5}{9} \bullet 0\right)\right] = 0.4444$$

$$G(Ciudad) = I(S_{SI}, S_{NO}) - E(Ciudad) = 0.7642 - 0.4444 = 0.3198$$

$$I_{división}(Ciudad) = -\left[\left(\frac{4}{9} \bullet \log_2 \frac{4}{9}\right) + \left(\frac{5}{9} \bullet \log_2 \frac{5}{9}\right)\right] = 0.9911$$

$$PG(Ciudad) = \frac{0.3198}{0.9911} = 0.3226$$

c. Comencemos analizando el atributo ***Sexo***, tiene la siguiente distribución de datos:

Para **F "femenino"**:	$S_F = 6$	$S_{F_{SI}} = 2$	$S_{F_{NO}} = 4$
Para **M "masculino"**:	$S_M = 3$	$S_{M_{SI}} = 0$	$S_{M_{NO}} = 3$

$$I(S_{F_{SI}}, S_{F_{NO}}) = -\left[\left(\frac{2}{6}\log_2 \bullet \frac{2}{6}\right) + \left(\frac{4}{6}\log_2 \bullet \frac{4}{6}\right)\right] = 0.9183$$

$$I(S_{M_{SI}}, S_{M_{NO}}) = -\left[0 + \left(\frac{3}{3}\log_2 \bullet \frac{3}{3}\right)\right] = 0$$

$$E(\text{Sexo}) = \left[\left(\frac{6}{9} \bullet 0.9183\right) + \left(\frac{3}{9} \bullet 0\right)\right] = 0.6122$$

$$G(\text{Sexo}) = I(S_{SI}, S_{NO}) - E(\text{Sexo}) = 0.7642 - 0.6122 = 0.1520$$

$$I_{\text{división}}(\text{Sexo}) = \left[\left(\frac{6}{9} \bullet \log_2 \frac{6}{9}\right) + \left(\frac{3}{9} \bullet \log_2 \frac{3}{9}\right)\right] = 0.9183$$

$$PG(\text{Sexo}) = \frac{0.1520}{0.9183} = 0.1070$$

b. Comencemos analizando el atributo ***Edad***

El atributo ***Edad_2*** tiene la siguiente distribución de datos:

Para **[=<87 años]**:	$S_{=<87} = 8$	$S_{=<87_{SI}} = 1$	$S_{=<87_{NO}} = 7$
Para **[>87 años]**:	$S_{>87} = 1$	$S_{>87_{SI}} = 1$	$S_{>87_{NO}} = 0$

$$I(S_{=<87_{SI}}, S_{=<87_{NO}}) = -\left[\left(\frac{1}{8}\log_2 \bullet \frac{1}{8}\right) + \left(\frac{7}{8}\log_2 \bullet \frac{7}{8}\right)\right] = 0.5436$$

$$I(S_{>87_{SI}}, S_{>87_{NO}}) = -\left[\left(\frac{1}{1}\log_2 \bullet \frac{1}{1}\right) + 0\right] = 0$$

$$E(\text{Sexo}) = \left[\left(\frac{8}{9} \bullet .5436\right) + \left(\frac{1}{9} \bullet 0\right)\right] = 0.4832$$

$$G(\text{Sexo}) = I(S_{SI}, S_{NO}) - E(\text{Sexo}) = 0.7642 - 0.4832 = 0.2810$$

$$I_{\text{división}}(\text{Sexo}) = \left[\left(\frac{8}{9} \bullet \log_2 \frac{8}{9}\right) + \left(\frac{1}{9} \bullet \log_2 \frac{1}{9}\right)\right] = 0.5033$$

$$PG(\text{Sexo}) = \frac{0.2810}{0.5033} = 0.5524$$

Elegimos el atributo según el cual dividiremos a este conjunto de datos. En este caso, la división según el atributo Edad_2 (particiones =<87 y >87) es la que mayor ganancia y proporción de ganancia ofrece.

4º *Evaluamos la primera partición de edad >87 años*

S es el conjunto de todos los ejemplos donde 1 individuos pertenecen a la clase "S". Creamos un nodo terminal (hoja) con la clase "S".

Ciudad de nacimiento	Edad	Sexo	Target
Rosario	96	F	S

5º *Evaluamos la segunda partición de edad =<87 años*

S es el conjunto de todos los ejemplos donde de 8 individuos a 1 pertenece la clase "S" y 7 a la clase "N".

Ciudad de nacimiento	Edad	Sexo	Target
Santa Rosa	34	F	N
Rosario	40	F	S
Santa Rosa	45	F	N
Rosario	53	M	N
Santa Rosa	55	F	N
Santa Rosa	58	M	N
Rosario	67	M	N
Santa Rosa	78	F	N

Analizamos la ganancia y proporción de ganancia de Ciudad y Sexo para determinar cual de los 2 atributos particionara los datos.

$S = 8 \qquad S_{SI} = 1 \; S_{NO} = 7$

$$I(S_S, S_N) = -\left[\left(\frac{1}{8}\log_2 \bullet \frac{1}{8}\right) + \left(\frac{7}{8}\log_2 \bullet \frac{7}{8}\right)\right] = 0.5436$$

a. Comencemos analizando el atributo ***Ciudad de nacimiento,*** tiene la siguiente distribución de datos:

Para **Rosario**:	$S_{Rosario} = 3$	$S_{Rosario_{SI}} = 1$	$S_{Rosario_{NO}} = 2$
Para **Santa Rosa**:	$S_{SantaRosa} = 5$	$S_{SantaRosa_{SI}} = 0$	$S_{SantaRosa_{NO}} = 5$

$$I(S_{Rosario_{SI}}, S_{Rosario_{NO}}) = -\left[\left(\frac{1}{3}\log_2 \bullet \frac{1}{3}\right) + \left(\frac{2}{3}\log_2 \bullet \frac{2}{3}\right)\right] = 0.9183$$

$$I(S_{SRosa_{SI}}, S_{SRosa_{NO}}) = -\left[0 + \left(\frac{5}{5}\log_2 \bullet \frac{5}{5}\right)\right] = 0$$

$$E(Ciudad) = \left[\left(\frac{3}{8} \bullet 0.9183\right) + \left(\frac{5}{8} \bullet 0\right)\right] = 0.3444$$

$$G(Ciudad) = I(S_{SI}, S_{NO}) - E(Ciudad) = 0.5436 - 0.3444 = 0.1992$$

$$I_{división}(Ciudad) = -\left[\left(\frac{3}{8} \bullet \log_2 \frac{3}{8}\right) + \left(\frac{5}{8} \bullet \log_2 \frac{5}{8}\right)\right] = 0.9544$$

$$PG(Ciudad) = \frac{0.3444}{0.9544} = 0.2087$$

Elegimos el atributo según el cual dividiremos a este conjunto de datos. En este caso, la división según el atributo Ciudad, es la que mayor ganancia y proporción de ganancia ofrece.

6° Evaluamos los ejemplos de Santa Rosa

S es el conjunto de todos los ejemplos donde 5 individuos pertenecen a la clase "N". Creamos un nodo terminal (hoja) con la clase "N".

Ciudad de nacimiento	Edad	Sexo	Target
Santa Rosa	34	F	N
Santa Rosa	45	F	N
Santa Rosa	55	F	N
Santa Rosa	58	M	N
Santa Rosa	78	F	N

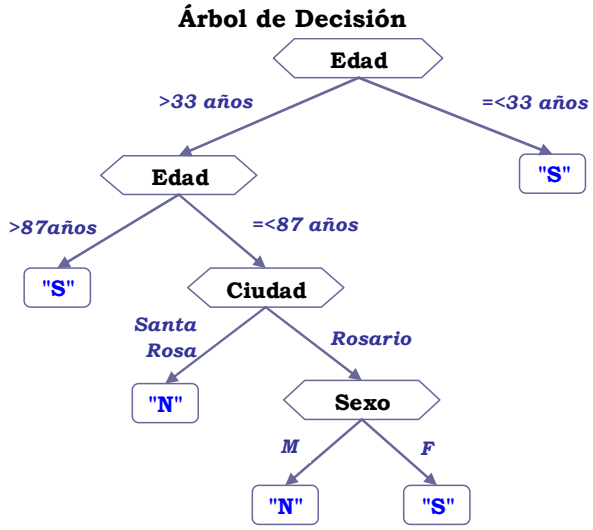
7° Evaluamos los ejemplos de Rosario

S es el conjunto de todos los ejemplos donde de 3 individuos a 1 pertenece la clase "S" y 2 a la clase "N". en este caso el atributo Sexo tiene una proporción de ganancia de 1, donde 1 ejemplo es F y pertenece a la clase "S" y los otros de dos ejemplos son M y pertenecen a la clase "N".

Ciudad de nacimiento	Edad	Sexo	Target
Rosario	40	F	S
Rosario	53	M	N
Rosario	67	M	N

Ciudad de nacimiento	Edad	Sexo	Target
Rosario	40	F	S

Ciudad de nacimiento	Edad	Sexo	Target
Rosario	53	M	N
Rosario	67	M	N



Reglas	S	C
Si Edad =<33 Entonces Target (clase)=S	40%	100%
Si Edad >87 Entonces Target (clase)=S	6.6%	100%
Si Edad >33 Y =<87 Y Ciudad = "Santa Rosa" Entonces Target (clase)=N	33%	100%
Si Edad >33 Y =<87 Y Ciudad = "Rosario" Y Sexo = "F" Entonces Target (clase)=S	6.6%	100%
Si Edad >33 Y =<87 Y Ciudad = "Rosario" Y Sexo = "M" Entonces Target (clase)=N	13.3%	100%

c. Comencemos analizando el atributo ***Sexo***, tiene la siguiente distribución de datos:

Para **F "femenino"**:	$S_F = 5$	$S_{F_{SI}} = 1$	$S_{F_{NO}} = 5$
Para **M "masculino"**:	$S_M = 3$	$S_{M_{SI}} = 0$	$S_{M_{NO}} = 3$

$$I(S_{F_{SI}}, S_{F_{NO}}) = -\left[\left(\frac{1}{5}\log_2 \bullet \frac{1}{5}\right) + \left(\frac{5}{5}\log_2 \bullet \frac{5}{5}\right)\right] = 0.7219$$

$$I(S_{M_{SI}}, S_{M_{NO}}) = -\left[0 + \left(\frac{3}{3}\log_2 \bullet \frac{3}{3}\right)\right] = 0$$

$$E(\text{Sexo}) = \left[\left(\frac{5}{8} \bullet 0.7219\right) + \left(\frac{3}{8} \bullet 0\right)\right] = 0.4512$$

$$G(\text{Sexo}) = I(S_{SI}, S_{NO}) - E(\text{Sexo}) = 0.5436 - 0.4512 = 0.0924$$

$$I_{\text{división}}(\text{Sexo}) = \left[\left(\frac{5}{8} \bullet \log_2 \frac{5}{8}\right) + \left(\frac{3}{8} \bullet \log_2 \frac{3}{8}\right)\right] = 0.9544$$

$$PG(\text{Sexo}) = \frac{0.0924}{0.9544} = 0.0968$$

Poda de los árboles generados

Varias *razones* para la poda de los árboles generados por los métodos de TDIDT : la sobregeneralización, la evaluación de atributos poco importantes o significativos, y el gran tamaño del árbol obtenido

Existen dos *enfoques* para podar los árboles: la pre-poda *(preprunning)* y la post-poda *(postprunning)*.

En el primer caso se detiene el crecimiento del árbol cuando la ganancia de información producida al dividir un conjunto no supera un umbral determinado. tiene la atracción de que no se pierde tiempo en construir una estructura que luego será simplificada en el árbol final. El método típico en estos casos es buscar la mejor manera de partir el subconjunto y evaluar la partición desde el punto de vista estadístico mediante la teoría de la ganancia de información, reducción de errores, etc. Si esta evaluación es menor que un límite predeterminado, la división se descarta y el árbol para el subconjunto es simplemente la hoja más apropiada. este tipo de método tiene la contra de que no es fácil detener un particionamiento en el momento adecuado, un límite muy alto puede terminar con la partición antes de que los beneficios de particiones subsiguientes parezcan evidentes, mientras que un límite demasiado bajo resulta en una simplificación demasiado leve

En la post-poda se podan algunas ramas una vez que se ha terminado de construir el árbol. Es utilizado por el ID3 y el C4.5. Una vez construido el árbol se procede a su simplificación según los criterios propios de cada uno de los algoritmos

COMPARACIÓN DE LOS MÉTODOS DE EVALUACIÓN DE RESULTADOS DE APRENDIZAJE AUTOMÁTICO.

Para problemas de clasificación es natural medir la performance del clasificador con una proporción de error. El clasificador predice la clase de cada instancia: si la predicción es correcta, estamos ante un éxito, si no lo es, estamos ante un error. La proporción de error entonces es la *cantidad de errores sobre la cantidad total de instancias clasificadas*. (Error de aprendizaje o sustitución) Lo que interesa es calcular la proporción de error sobre *nuevos datos*, no utilizados durante la construcción del modelo (conjunto de prueba). (Error de predicción o clasificación).

Holdout

método de retención para estimar la proporción de error

Puede aplicarse cuando existe una cantidad limitada de datos de entrenamiento y prueba.

Reserva una cierta cantidad de datos al azar para prueba y utiliza el resto para el entrenamiento. En general, se reserva un tercio para prueba y se utilizan dos tercios como datos de entrenamiento.

- Enfoque tradicional (más simplista)
- Se divide el conjunto D en D_l y D_t, tal que $\frac{|D_l|}{|D|} = 0.75(approx.)$
- Aprendizaje sobre D_l, estimación sobre D_t

Una manera de evitar la tendencia introducida por los datos retenidos, es repetir el proceso completo (entrenamiento y prueba) varias veces con distintas divisiones de los datos. En cada iteración, una misma proporción de los datos se retiene al azar para las pruebas y el resto se utiliza para el entrenamiento. Las proporciones de error obtenidas en las múltiples iteraciones se promedian para obtener una proporción de error general.

Al dividir al azar los datos preclasificados entre los conjuntos de entrenamiento y prueba, debemos garantizar que cada clase esté correctamente representada tanto en los datos de prueba como en los de entrenamiento.

Este procedimiento se conoce como *estratificación (stratification)*, y podemos hablar de una *retención estratificada*.

Validación cruzada de k pliegues

En la validación cruzada, se determina con anterioridad una cierta cantidad de *pliegos* o particiones de los datos

Supongamos que los datos se dividen al azar en tres particiones de aproximadamente la misma cantidad, y cada una a su turno se utiliza para prueba mientras que las otras dos se utilizan para entrenamiento.

Por lo tanto, utilizamos un tercio para prueba y dos tercios para entrenamiento, y repetimos el procedimiento tres veces. Las tres proporciones de error obtenidas se promedian para llegar a una proporción de error general.

Este procedimiento conocido como *validación cruzada de tres pliegues (threefold cross-validation)*, puede trabajar con datos estratificados, en cuyo caso sería *validación cruzada de tres pliegues estratificada*

- Se divide el conjunto D en k partes $\{D_1, D_2, \ldots, D_k\}$ iguales, y disjuntas
- Se realizan k procesos de aprendizaje, usando en el proceso i el cojunto D_i como test, y el resto para el aprendizaje
- Estimación del error según

$$acc_{cv} = \frac{1}{n} \sum_{j=1}^{n} \sum_{<v_i, y_i> \in D_j} \delta(\mathcal{I}(D - D_j, v_i), y_i)$$

en donde

- $|D| = n$,
- $\mathcal{I}(A, v)$ la salida para el valor v del modelo inducido por $\mathcal{I}$ en el conjunto A y
- $\delta(x, y)$ es la diferencia entre las predicciones x e y

Podemos generalizar el método para llegar a una *validación cruzada de k pliegues*, estratificada o no. El caso más utilizado para predecir la proporción de error de una técnica de aprendizaje es utilizar una validación cruzada de diez pliegues.

Ventajas: El error final se cálcula como la media aritmética de los k errores, de esta manera recoge la media de los experimentos con k subconjuntos de prueba independientes. Otra ventaja es que permite estimar la variabilidad del método de aprendizaje con respecto a la evidencia.

Desventaja: los subconjuntos de entrenamiento no son independientes, por ejemplo en una validación cruzada con k=10, cada subconjunto de entrenamiento comparte el 80% de los datos; este solapamiento entre subconjuntos de entrenamiento podría afectar la calidad de la estimación

Dejar-uno-afuera
(Leave-one-out)

Esta técnica es simplemente una validación cruzada de k pliegues donde k es el número de instancias del conjunto de datos. Por turnos, cada una de las instancias se deja afuera y se entrena el clasificador con el resto de las instancias. Se lo evalúa según el resultado de la clasificación de la instancia que había quedado afuera. Los resultados de las k evaluaciones luego se promedian para determinar la proporción de error.

Ventajas: Primero, se utiliza la mayor cantidad de ejemplos posibles para el entrenamiento, lo cual se presume incrementa la posibilidad de que el clasificador sea correcto. Segundo, el procedimiento es determinístico: no se parten los datos al azar. Además, no tiene sentido repetir el procedimiento diez ni cien veces, ya que siempre se obtendrá el mismo resultado.

Desventajas: Debe tenerse en cuenta que dado el alto costo computacional de aplicar este método, no es factible utilizarlo para grandes conjunto de datos. Sin embargo, este método es el mejor para pequeños conjuntos de datos porque, en cierta medida, evalúa todas las posibilidades.

Bootstraping

Se utiliza en casos en los que todavía se tiene menos ejemplos, especialmente indicada en estos casos

- Sea O de tamaño n.
- Una muestra bootstrap se hace tomando n muestras, del conjunto con remplazamiento.
- La probabilidad de que una instancia cualquiera no se haya escogido es de $(1-1/n)^n \approx e^{-1} \approx 0{,}368$

Este método está basado en el procedimiento estadístico de obtener muestras con sustitución.

Tenemos N ejemplos, a partir de este conjunto realizamos un muestreo aleatorio con reposición de N ejemplos. Esta muestra será el conjunto de entrenamiento, al ser con reemplazamiento puede tener ejemplos repetidos (que se mantienen). Significa que no contendrá algunos ejemplos de conjunto original. Precisamente los ejemplos no elegidos por la la muestra se reserva para el conjunto de prueba. Esto nos da un conjunto de de entrenamiento de N ejemplos y un conjunto de prueba de aproximadamente 0.368 x N ejemplos. Con estos ejemplos se entrena y evalúa un modelo.

Todo el proceso de *bootstrap* se repite un número prefijado de k veces (por ejemplo 10 veces), y después se promedian los errores

Ventajas: las k repeticiones del proceso son independientes y esto es más robusto estadísticamente

- Con esas n muestras se compone el conjunto de entrenamiento, D_l.
- El resto van al conjunto de test. D_t.
- Con n razonablemente grande, el conjunto de test contendrá un 36,8 % de las instancias
- El conjunto de aprendizaje contendrá un 63,2 %.
- El estimador bootstrap se obtiene mediante la expresión $$acc_{boot} = \frac{1}{b}\sum_{i=1}^{b}(0{,}632e_{l_i} + 0{,}368acc_D)$$
  - Siendo b el número de clasificadores obtenidos.
  - e_{l_i} el error de evaluación para el clasificador i
  - acc_s es el estimador de rendimiento en el conjunto D.

La probabilidad de que una instancia particular sea elegida para el conjunto de entrenamiento es de $1/n$, y, por lo tanto, hay un $1\text{-}1/n$ de probabilidad de que no sea elegida. Si multiplicamos esto según la n oportunidades de ser elegida, obtenemos la siguiente probabilidad de que no sea escogida

$$\left(1-\frac{1}{n}\right)^n = e^{-1} = 0.368$$

Entonces, un conjunto de datos lo suficientemente grande contendrá un 36.8% de instancias de prueba y un 63.2% de entrenamiento. Esta es la razón por la cual este método se conoce como el *0.632 bootstrap*

El error estimado sobre el conjunto de prueba será pesimista porque el clasificador tiene en cuenta sólo el 63% de los datos del conjunto original, lo cual es poco frente al 90% de la validación cruzada de diez pliegues. Para compensar el error del conjunto de entrenamiento se combina con el error en el conjunto de prueba de la siguiente manera: $e = 0.632 \times e_{prueba} + 0.368 \times e_{entrenamiento}$

Estimación del costo

Hasta ahora no hemos considerado el costo de tomar malas decisiones y malas clasificaciones. La optimización de las proporciones de clasificación sin considerar el costo de los errores, generalmente lleva a resultados extraños

Si los costos son conocidos, pueden incluirse en el análisis de los métodos. Restringiremos nuestro análisis a los casos que tienen clases *sí* y *no* únicamente. Los cuatro resultados posibles de una predicción pueden listarse en una matriz de confusión

		Clase predicha	
		SI	NO
Clase verdadera	SI	*Verdadero positivo*	*Falso positivo*
	NO	*Falso positivo*	*Verdadero negativo*

No obstante, la mayoría de los algoritmos de aprendizaje automático no tienen en cuenta el costo al aprender. Existen, sin embargo, dos maneras de transformarlo fácilmente. La primera idea para transformar un clasificador para que tome en cuenta el costo, es variar la cantidad de ejemplos positivos y negativos en los datos de entrenamiento de acuerdo a la importancia de cada uno de los errores. Otra idea es ponderar las instancias. Por ejemplo, al generar un árbol de decisión, una instancia puede dividirse en partes con un esquema de ponderación que indique la proporción con que debe tomarse cada rama

Ventajas: el aprendizaje sensible al costo puede considerarse como una generalización más realista del aprendizaje predictivo. La calidad del modelo se mide en términos de minimización de costos en vez de minimización de errores.

Los verdaderos positivos y verdaderos negativos son los casos sin error.

Los falsos positivos corresponden a aquellas instancias negativas que fueron clasificadas como positivas, mientras que los falsos negativos son aquellas instancias clasificadas como negativas cuando en realidad son positivas.

Estos dos casos de errores generalmente tienen distintos costos, como los casos clasificados correctamente tienen distintos beneficios. El hecho de pensar en el costo genera mejores decisiones