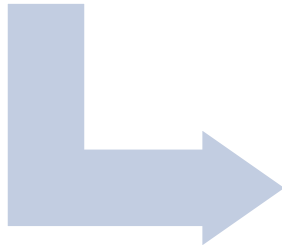


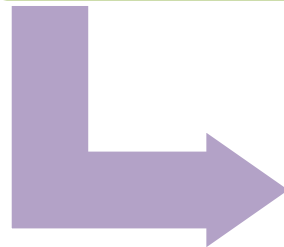
Árboles de Decisión y Conjuntos de Modelos

Aprendizaje
Supervisado



Árboles de
Decisión

- ¿Qué son?
- ¿Cómo se Construyen?



Conjuntos
de Modelos

- Bagging
- Random Forests
- Boosting

APRENDIZAJE SUPERVISADO

- El Aprendizaje Supervisado es un tipo de aprendizaje automático en el que en el conjunto de datos están incluidos los valores objetivo del problema.
- Este conjunto de datos servirá para entrenar el modelo.
- Los árboles de decisión son algoritmos de aprendizaje supervisado que permiten modelar mediante reglas/ árboles el conocimiento que nos lleva a inferir de las entradas las salidas.

Plasma Glucose	BMI	Pregnancies	Diabetes?
112	27,54	3	FALSE
179	31,65	4	TRUE
109	21,39	1	FALSE

- La **clasificación** es aquella subcategoría del aprendizaje supervisado en la que el objetivo es predecir la categoría o categorías a las que pertenecen nuevas instancias basándonos en los ejemplos de entrenamiento. En este caso la etiqueta que acompaña a cada ejemplo, y que es necesario predecir, es discreta, es decir, valores finitos sin necesidad de tener una relación de orden entre ellos, por ejemplo: spam/no spam, bueno/regular/malo, rojo/amarillo, verde; etc.
- El **análisis de regresión** es la parte del aprendizaje supervisado que se ocupa de la predicción de valores numéricos continuos, por ejemplo, cuando a partir de variables como la superficie de una casa, y el número de habitaciones es posible predecir su precio. En este caso, el objetivo del algoritmo es inferir las relaciones entre las variables, que son previamente conocidas y que permiten ofrecer una predicción sobre la salida requerida.

ÁRBOLES DE DECISIÓN

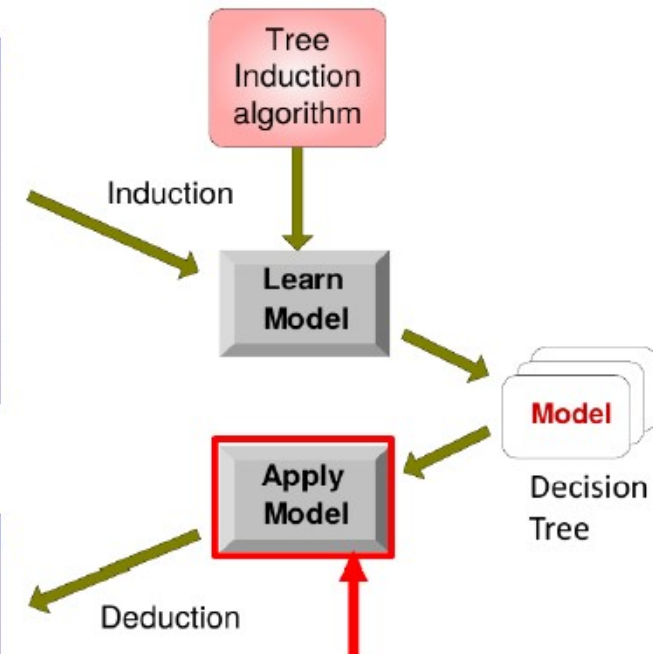
- Los árboles de decisión son una técnica de aprendizaje automático por **inducción** que permiten identificar conceptos (clases de objetos) a partir de las características de un conjunto de ejemplos que los representan.

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

Training Set

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

Test Set



Ejemplo: cliente y préstamos bancario

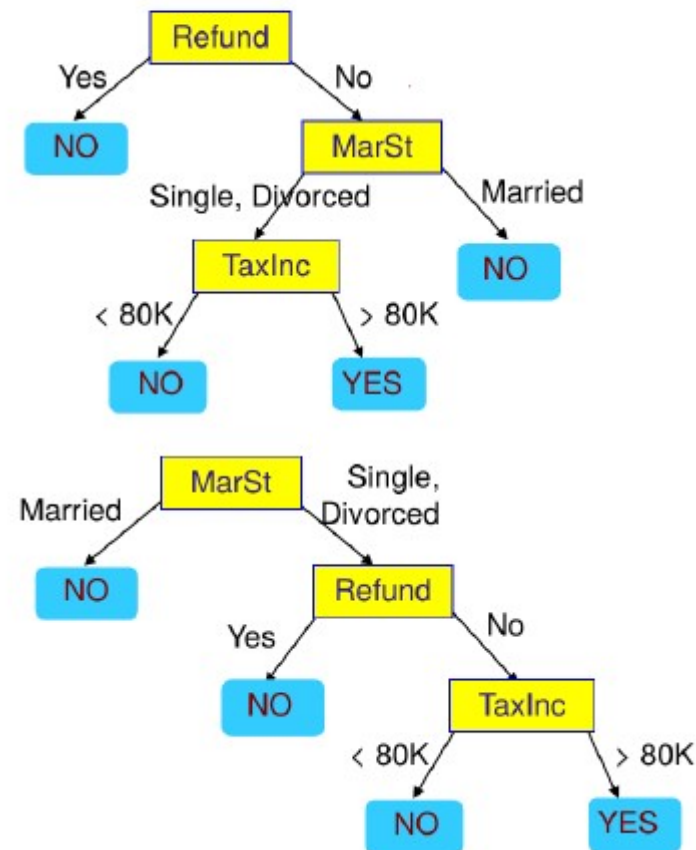
- La información extraída de los mismos queda organizada jerárquicamente en forma de **árbol**, es decir, en forma de **grafo dirigido que consta de nodos y arcos**.
- Los **nodos corresponden a una pregunta** o a un test que se hace a los ejemplos.
- El árbol de decisión se construye a base de ir haciendo preguntas sobre características determinadas a los ejemplos y clasificándolos según la respuesta.

Candidate Concepts/Decision Trees

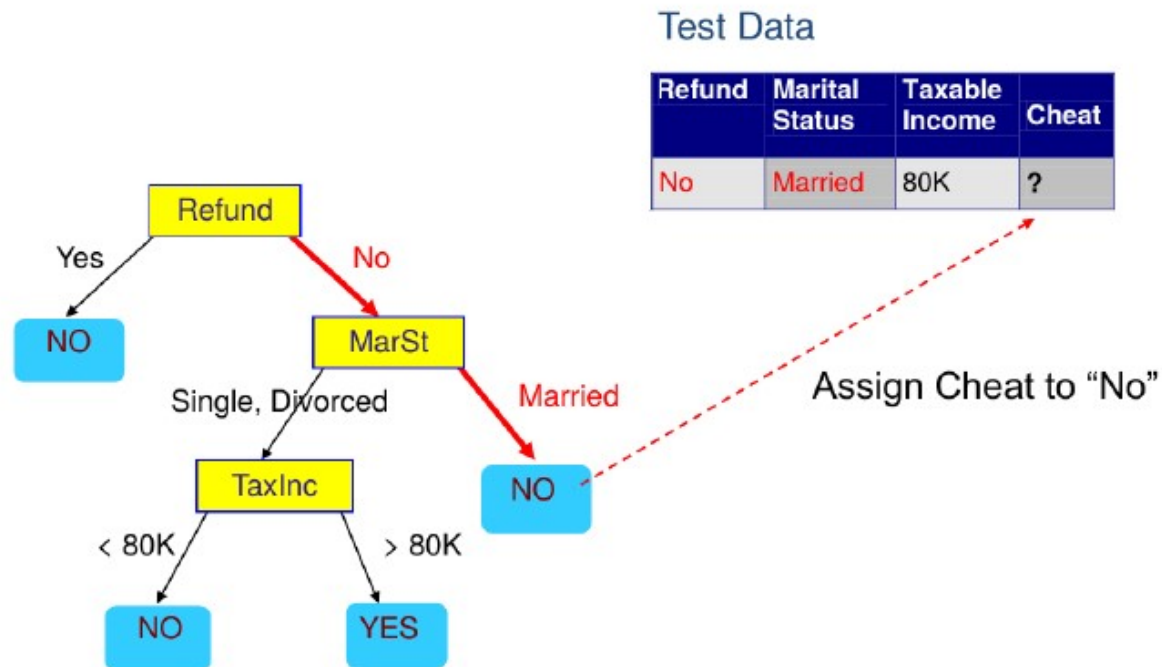
categorical
categorical
continuous
class

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

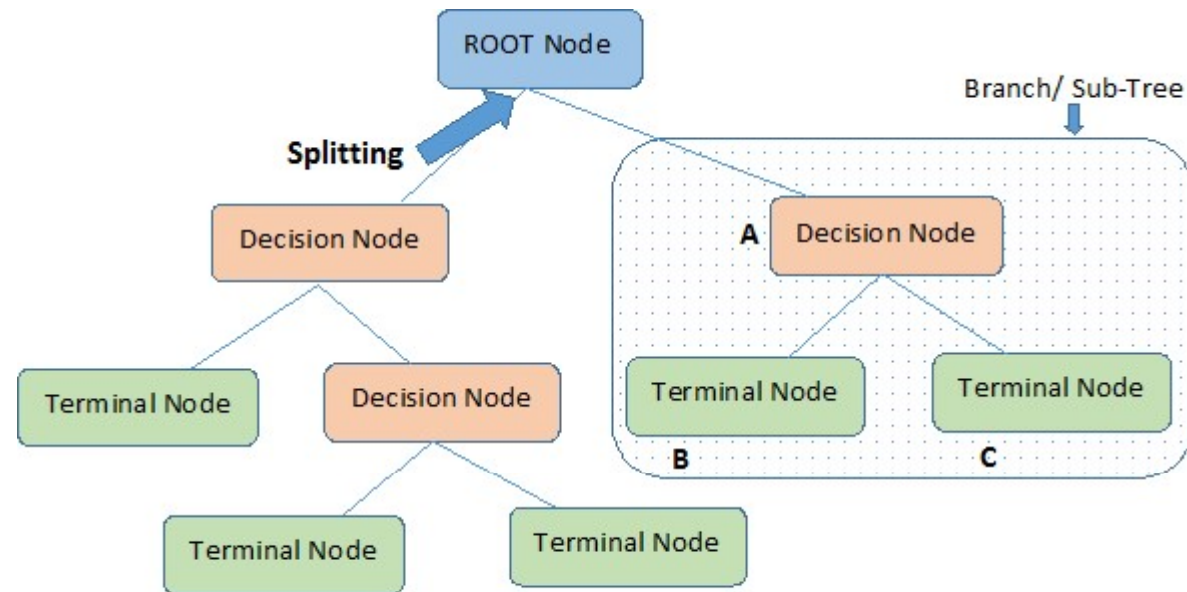
Training Data



- Las diferentes opciones de clasificación (respuesta a las preguntas) son excluyentes entre sí, lo que hace que a partir de casos desconocidos y siguiendo el árbol adecuadamente, se llegue a una única conclusión o decisión a tomar



- Un árbol de decisión tiene un nodo raíz, nodos intermedios y hojas.
- **Nodo raíz:** Es el principal rasgo o descriptor que permite dividir el conjunto original en dos subconjuntos mejores (aporta mayor información)
- **Nodos Intermedios:** Cualquier nodo intermedio puede ser un nodo raíz de un subárbol. Esto conduce a una definición recursiva de árbol de decisión.
 - Cada nodo intermedio y la raíz tienen asociados separadores que formulan una pregunta o realizan un test acerca de la existencia o no de una característica en cada caso ejemplo. Esto permite clasificar los ejemplos y determinar cuáles serían los nodos sucesores.
- **Hojas:** Una hoja en el árbol corresponde a un conjunto de ejemplos que representan una sola clase. La clase de la hoja se asigna por el criterio de a la que pertenezcan la mayoría de los ejemplos en ella. Las hojas del árbol de decisión representan los conceptos extraídos de manera automática

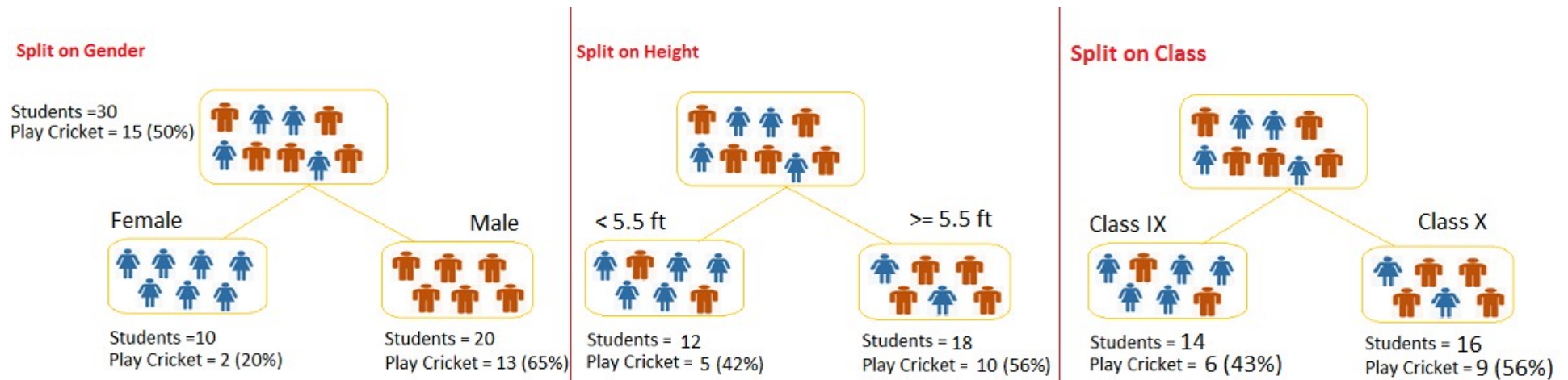


Note:- A is parent node of B and C.

<https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/>

- El algoritmo general para todo método basado en árboles de decisión sería el siguiente:
 - **Se calcula la calidad** (impureza, información, etc.) del nodo (conjunto de ejemplos).
 - Si un conjunto de ejemplos en un nodo no tiene suficiente calidad, se **selecciona el mejor atributo separador**.
 - Una vez que se obtiene, se añade a la base de reglas, y se utiliza para **dividir el conjunto de ejemplos en al menos dos** conjuntos nuevos de mayor calidad.
 - Seguir separando hasta que se cumpla el **criterio de parada**.

- **El rasgo o descriptor** a seleccionar debe de cumplir el objetivo de que su posición en algún punto del árbol **genere un subárbol tan simple como sea posible** y dé una concreta clasificación.
- De esta forma, cuando se construye un árbol de decisión, es necesario tener un medio para **determinar tanto los atributos importantes** requeridos para la clasificación, **como el orden** de uso de esos atributos importantes.
- **Es clave la elección del criterio de selección de separadores** lo cual dará lugar a diferentes algoritmos de construcción de árboles de decisión.



Buen criterio de división

Malos criterios de división

<https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/>

- **Índice de Entropía:** se calcula el índice de entropía para cada una de las variables: Si la variable es categórica, se obtiene sumando el índice de entropía de cada una de sus clases. En cambio, si es numérica, previamente se obtiene uno o varios puntos de corte por métodos iterativos. Se elegirá aquella variable que tenga menor índice de entropía, aquella cuya proporción de casos positivos esté más alejada de 0,5.
- **Chi-cuadrado:** mide el grado de asociación entre dos variables. Se calcula comparando la tabla de contingencia dada por el cruce de la variable objetivo versus cada una de las variables explicativas con una tabla donde no hubiera asociación. Se elegirá la variable con mayor valor del estadístico chi-cuadrado.
- **Índice de Gini:** mide la probabilidad de no sacar dos registros con el mismo valor para la variable objetivo dentro del mismo nodo. Cuando menor es el índice de Gini mayor es la pureza del corte. Por lo tanto, el corte propuesto en primer lugar será aquella variable que tenga menor valor del índice de Gini.

$$\text{Entropy}(t) = - \sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t),$$

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2,$$

$$\text{Classification error}(t) = 1 - \max_i [p(i|t)],$$

p refers to the fraction of records that belong to one of the two classes

Node N_1	Count
Class=0	0
Class=1	6

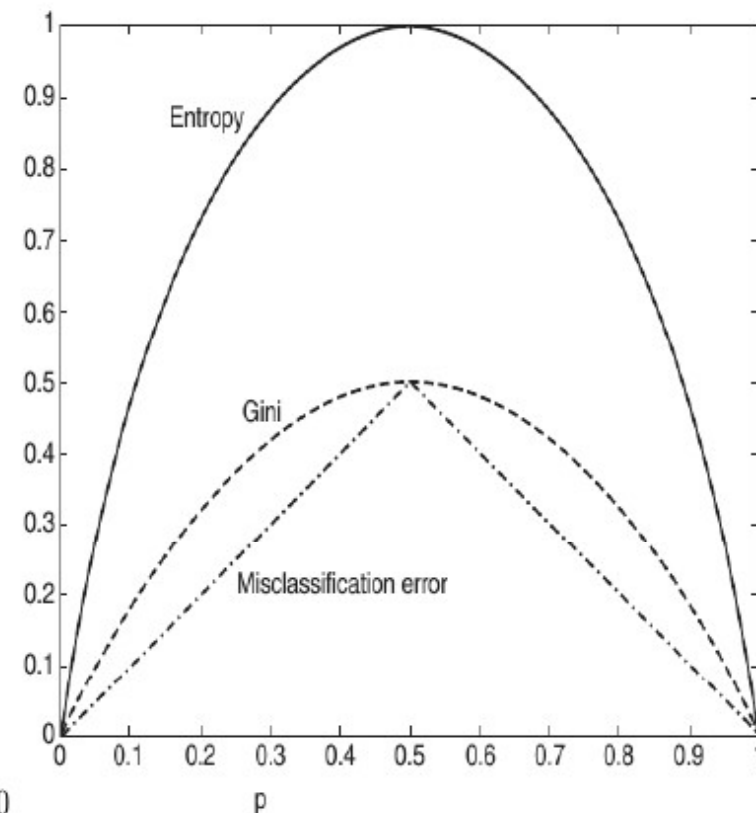
$$\begin{aligned} \text{Gini} &= 1 - (0/6)^2 - (6/6)^2 = 0 \\ \text{Entropy} &= -(0/6) \log_2(0/6) - (6/6) \log_2(6/6) = 0 \\ \text{Error} &= 1 - \max[0/6, 6/6] = 0 \end{aligned}$$

Node N_2	Count
Class=0	1
Class=1	5

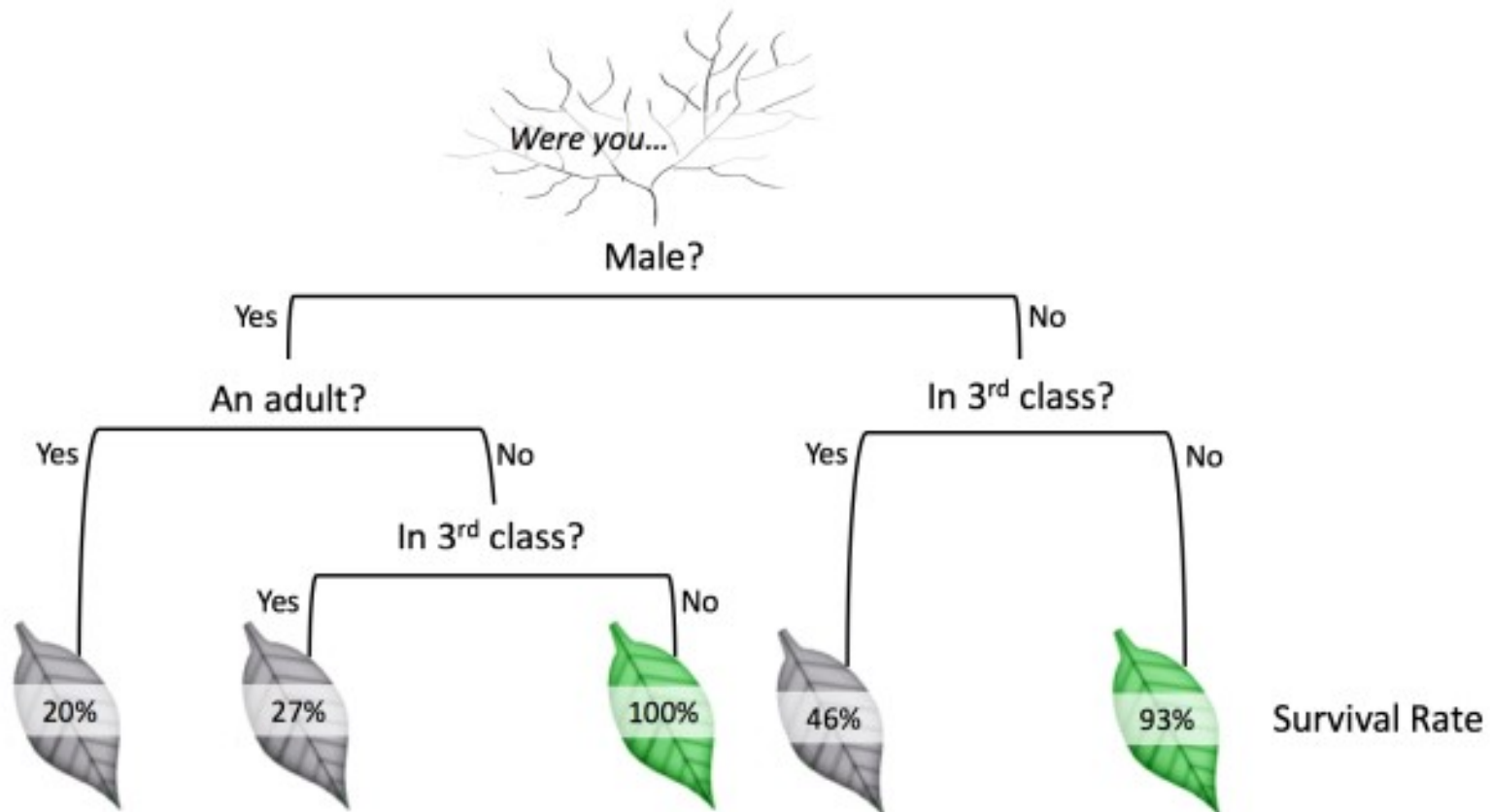
$$\begin{aligned} \text{Gini} &= 1 - (1/6)^2 - (5/6)^2 = 0.278 \\ \text{Entropy} &= -(1/6) \log_2(1/6) - (5/6) \log_2(5/6) = 0.650 \\ \text{Error} &= 1 - \max[1/6, 5/6] = 0.167 \end{aligned}$$

Node N_3	Count
Class=0	3
Class=1	3

$$\begin{aligned} \text{Gini} &= 1 - (3/6)^2 - (3/6)^2 = 0.5 \\ \text{Entropy} &= -(3/6) \log_2(3/6) - (3/6) \log_2(3/6) = 1 \\ \text{Error} &= 1 - \max[3/6, 3/6] = 0.5 \end{aligned}$$

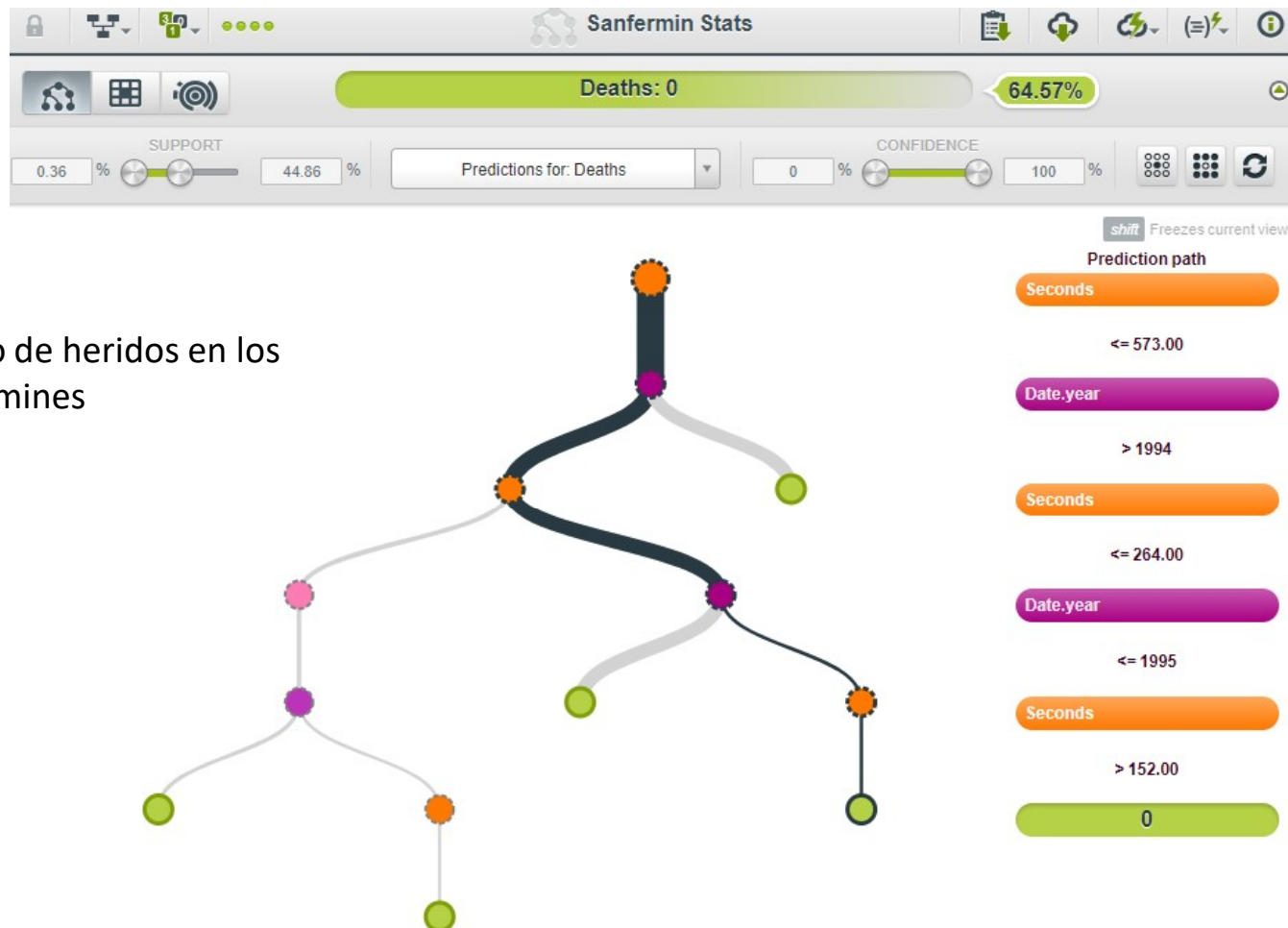


- **Máximo número de ramas por nodo (Maximum Branch):** número de ramas máximo en que puede dividirse un nodo.
- **Número mínimo de observaciones por nodo final (Leaf Size):** número mínimo de observaciones que tiene que tener un nodo final para que se construya la regla.
- **Número mínimo de observaciones para dividir un nodo (Split Size):** número mínimo de observaciones que tiene que tener un nodo para que se pueda cortar por la variable seleccionada.
- **Variables discriminantes (Discriminant Variables):** no encontrar ninguna variable que sea lo suficientemente discriminante en el nodo es motivo de parada.

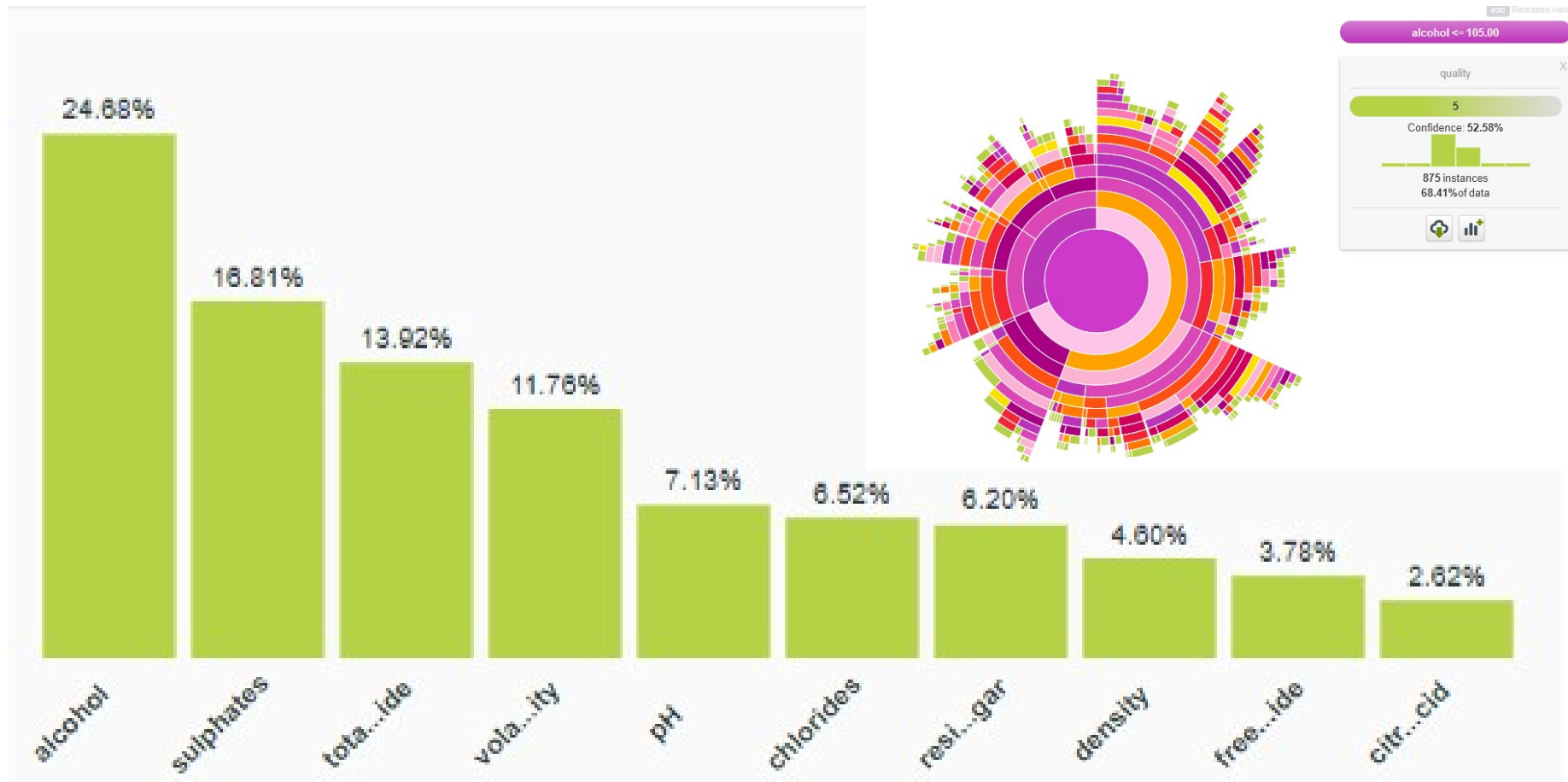


<https://algobeans.com/2016/07/27/decision-trees-tutorial/>

Siempre crean separadores lineales



Número de heridos en los San Fermes



Importancia o relevancia de las variables evaluadas

- **Podar** el árbol consiste en reducirlo, haciéndolo más sencillo dejando sólo los nodos más importantes y a su vez eliminando los redundantes. **IMPORTANTE PARA EVITAR SOBRE-APRENDIZAJE**
- Para ello se sigue habitualmente el siguiente procedimiento:
 - Antes de llamar recursivamente al algoritmo para cada Nodo:
 - Se calcula la tasa de ramificación del conjunto actual.
 - Se calcula la tasa de ramificación de los subconjuntos.
 - Si la tasa de ramificación del Subconjunto 1 es menor que la del conjunto. entonces se aplica el algoritmo sobre él, si no se detiene la expansión.
 - Si la tasa de ramificación del Subconjunto 2 es menor que la del conjunto. entonces se aplica el algoritmo sobre él, si no se detiene la expansión.

	Splitting Criteria	Attribute type	Missing values	Pruning Strategy	Outlier Detection
ID3	Information Gain (Basado en entropía)	Handles only Categorical value	Do not handle missing values.	No pruning is done	Susceptible to outliers
CART	Towing Criteria (Basado en Gini)	Handles both Categorical & Numeric value	Handle missing values.	Cost-Complexity pruning is used	Can handle Outliers
C4.5	Gain Ratio (Basado en entropía)	Handles both Categorical & Numeric value	Handle missing values.	Error Based pruning is used	Susceptible to outliers

<https://www.quora.com/What-are-the-differences-between-ID3-C4-5-and-CART>

VENTAJAS

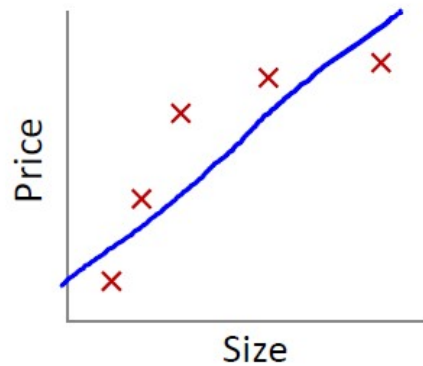
- No son costosos de construir
- Extremadamente rápidos al tratar nuevos ejemplos
- Si el árbol es pequeño, el modelo es muy interpretable
- El rendimiento es similar a otros modelos más complejos.

INCONVENIENTES

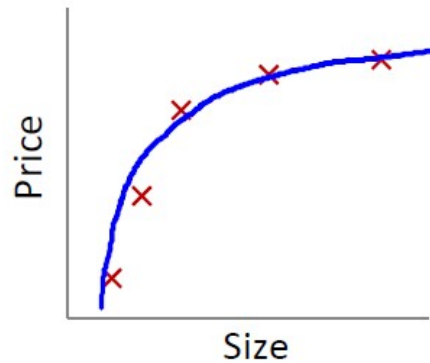
- Los árboles grandes son difíciles de entender y tienden al sobreaprendizaje.
- Algunos resultados dependen del algoritmo elegido.

CONJUNTOS DE MODELOS

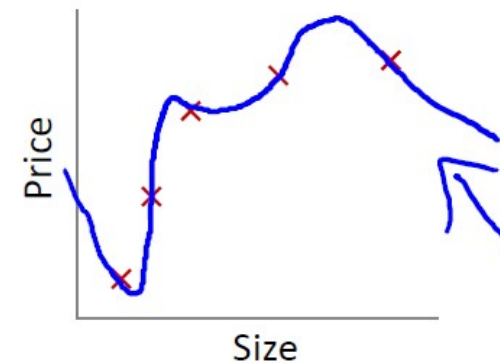
- Uno de los inconvenientes de los modelos individuales es que puede **sobreaprender**, es decir, sobreajustar sus datos, por lo que su rendimiento en su entrenamiento los datos son muy buenos, pero no se generalizan bien a los datos nuevos, lo que hace que los modelos individuales de árboles de decisión sean más peores modelos.
- Para solucionar el problema del sobreaprendizaje aparecen los conjuntos de modelos.



$\rightarrow \theta_0 + \theta_1 x$
 "Underfit" "High bias"



$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2$
 "Just right"



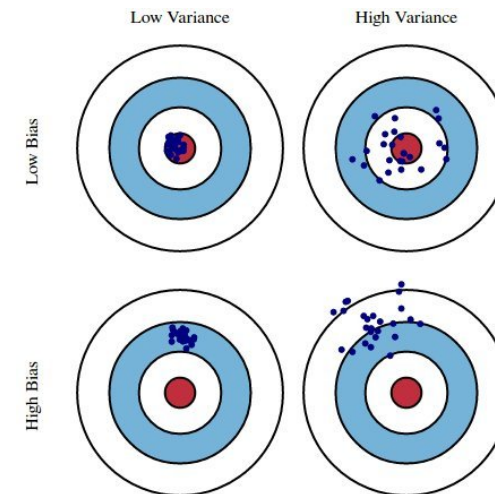
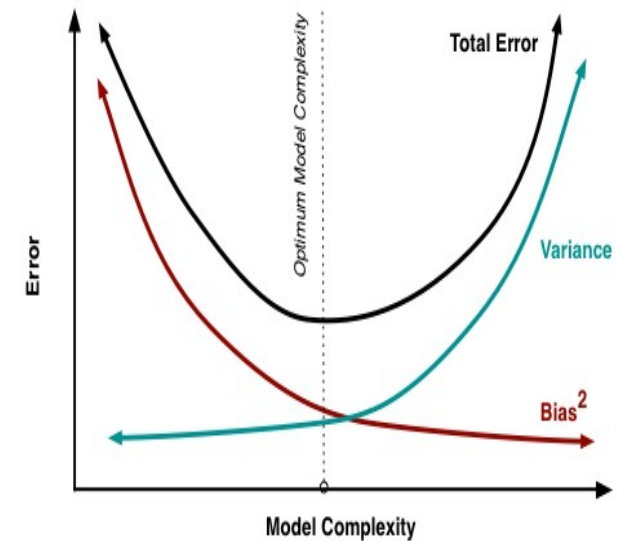
$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$
 "Overfit" "High variance"

- **Bias:**

- Error introducido por aproximar una función compleja con un modelo excesivamente simple.
- Diferencia entre lo predicho y lo real
- Underfitting

- **Variance**

- Capacidad del modelo para adaptarse si se estimara con un conjunto de entrenamiento diferente
- Si un modelo tiene una alta varianza, cualquier mínimo cambio le afectaría.
- Overfitting



- **Un conjunto (ensemble) es una colección de árboles de decisión** múltiples que se combinan para crear un modelo más sólido con un mejor rendimiento predictivo.
- Un conjunto de modelos basados en muestras de los datos puede convertirse en un buen predictor al **promediar los errores de cada modelo individual**.
- Los conjuntos son menos sensibles a los valores atípicos en sus datos de entrenamiento, **evitan el riesgo de sobreajuste y generalizar mejor** cuando se los aplica a nuevos datos.
- Dependiendo de la naturaleza de sus datos y los valores específicos para los parámetros del conjunto, puede aumentar significativamente el rendimiento sobre el uso de un solo modelo además de **permitir la paralelización** para la construcción y explotación del mismo.

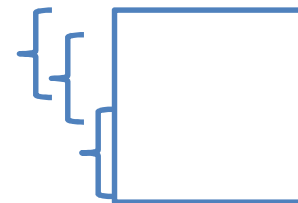
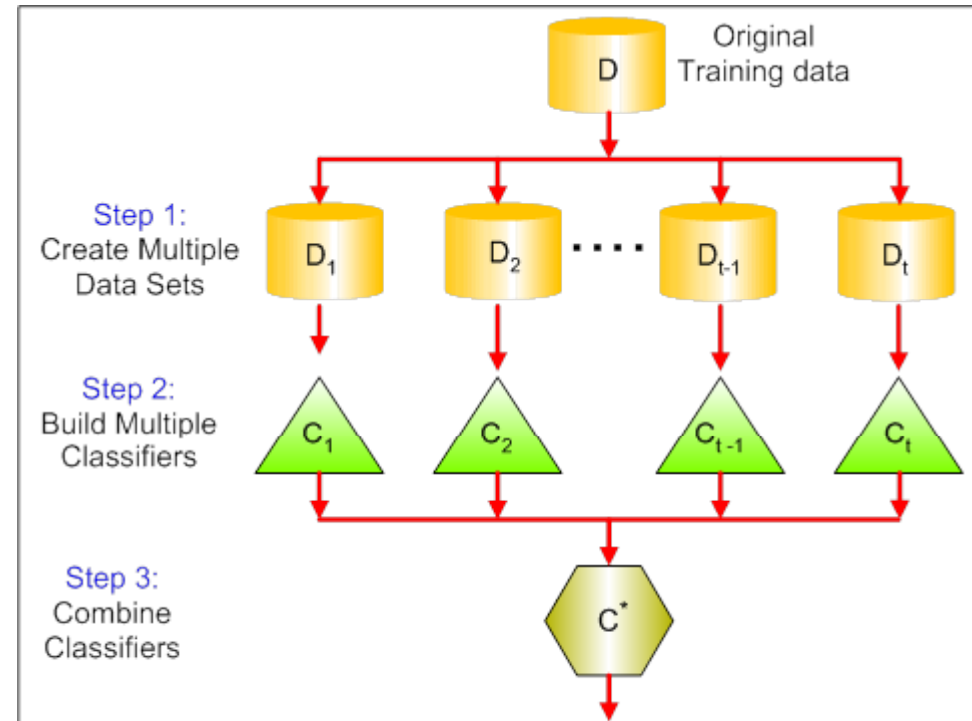
- La idea básica es **remuestrear los datos** y calcular las predicciones sobre el conjunto de datos remuestreados.
- Al promediar varios modelos conjuntamente **se obtiene un mejor ajuste** debido a que se mitigan tanto los modelos con sesgo como los modelos con alta varianza.
- Si el algoritmo predice datos categóricos, entonces el voto de la mayoría dará la clase dominante o con mejor predicción.
- Si se está realizando predicción sobre datos numéricos, entonces se realizará la media sobre las predicciones.

Bagging

Boosting

Random
Forests

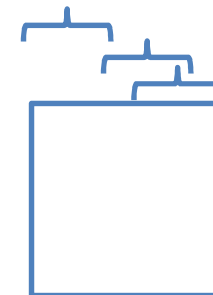
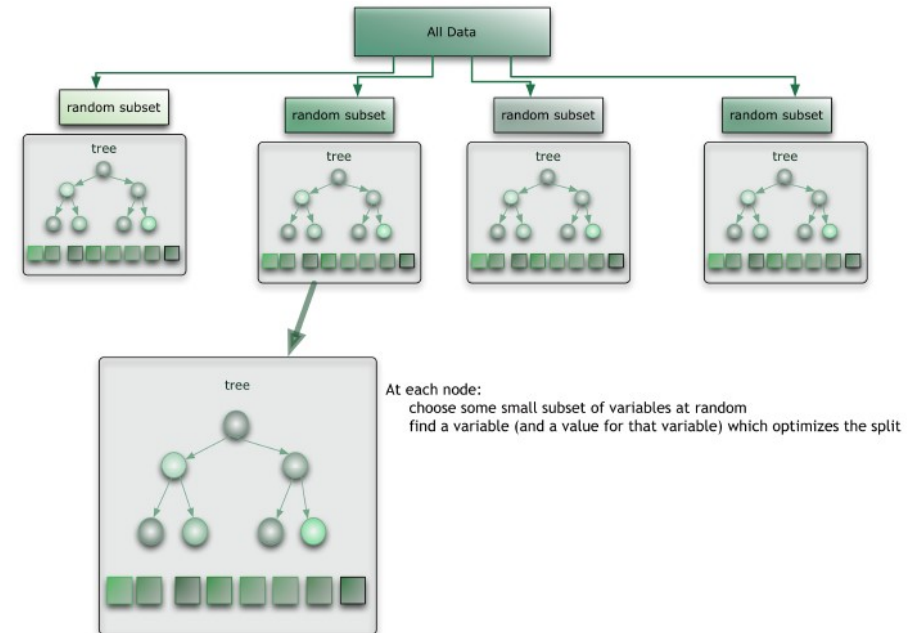
- **Bagging** (Bootstrap Aggregating): este algoritmo crea cada árbol individual a partir del aprendizaje sobre una muestra aleatoria de los elementos del conjunto de datos de entrenamiento.
- Por defecto las muestras son tomadas utilizando un ratio del 100%, reduciéndose si es conjunto de datos es muy grande, con reemplazo..
- El tamaño es el mismo que el conjunto de entrenamiento original pero no su composición.



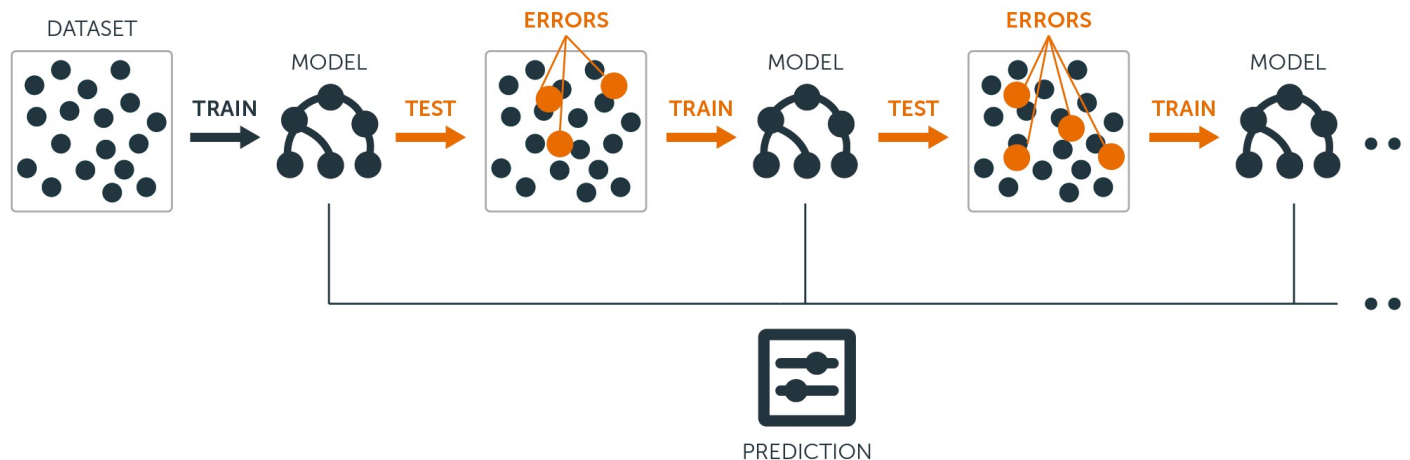
Bagging Example (Opitz, 1999)

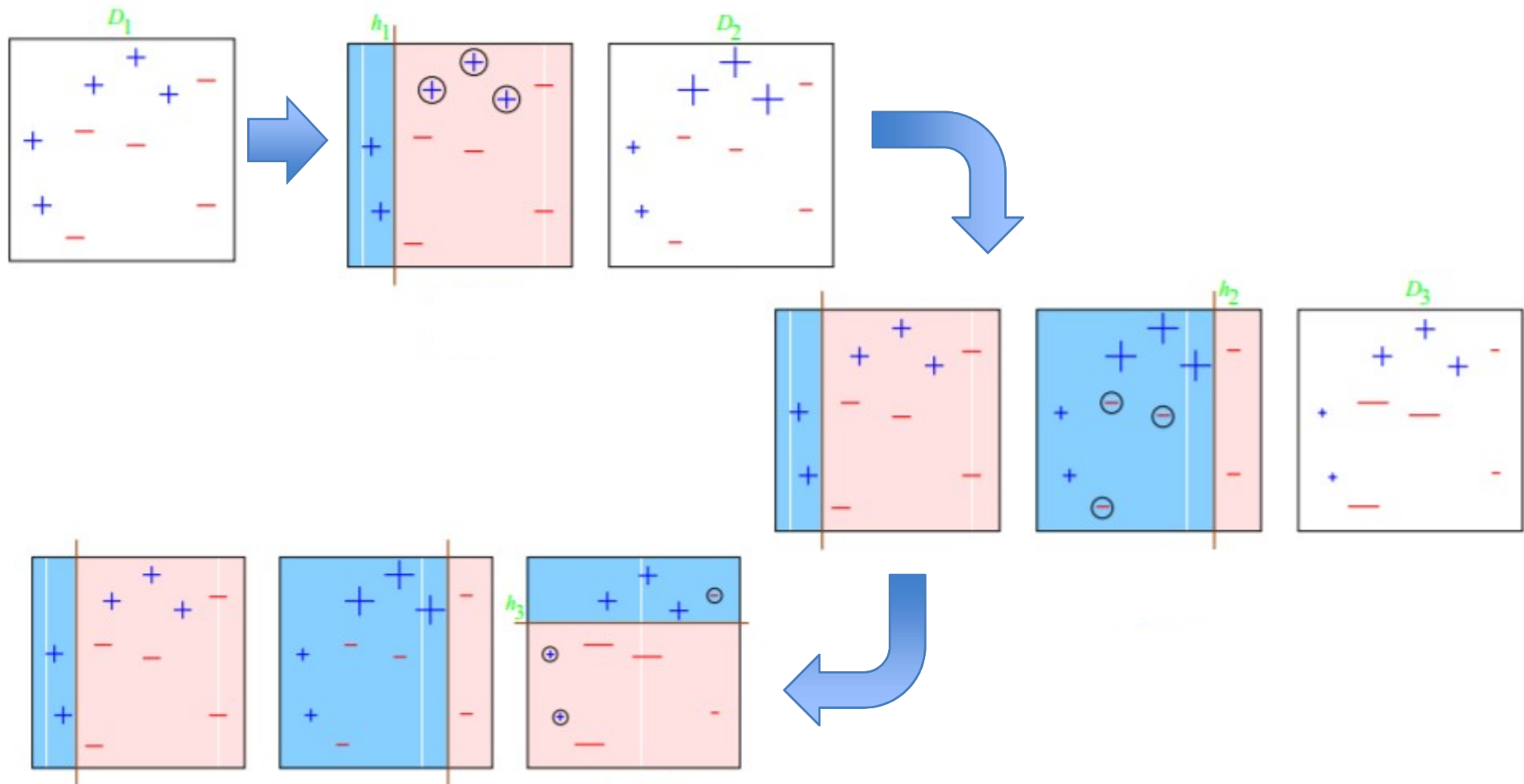
Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	7	8	5	6	4	2	7	1
Training set 3	3	6	2	7	5	6	2	2
Training set 4	4	5	1	4	6	4	3	8

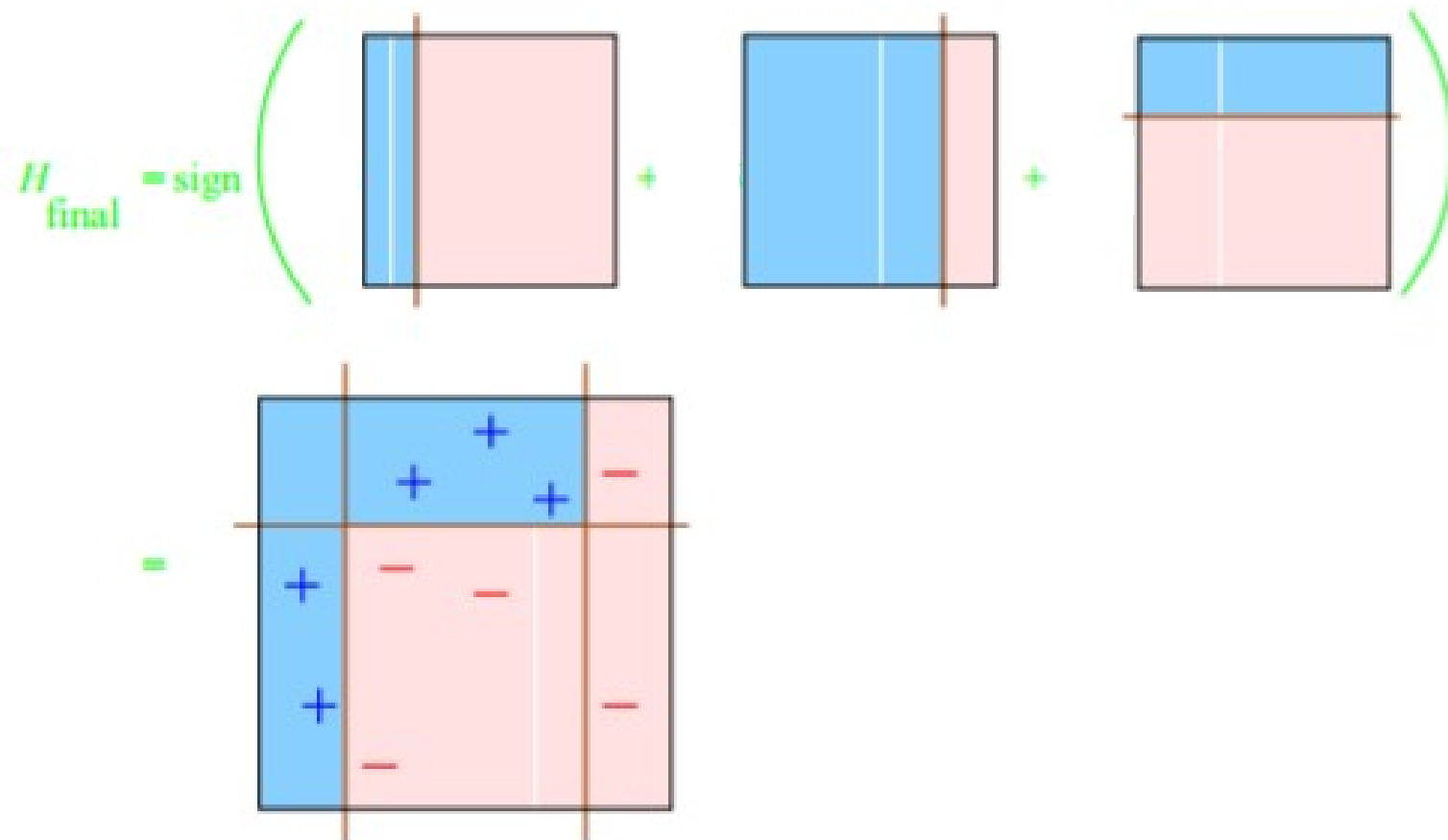
- **Random Forests:** sigue una estrategia similar al Bagging pero añadiendo un elemento adicional de aleatoriedad eligiendo para cada árbol un **subconjunto de las características del conjunto de entrenamiento**
- **Random subspaces**, cada miembro es entrenado con todos los ejemplos, pero con un subconjunto de los atributos.
- Parametrización
 - Número de arboles: SQRT
 - Profundidad del árbol indefinida



- **Boosting** (Adaboost, Gradient Boosted Trees): en este caso el algoritmo construye secuencialmente un conjunto de árboles de decisión que supone modelos débiles (weak learners, un poco mejores que aleatorios).
- En cada iteración del algoritmo, cada modelo individual intenta corregir los errores cometidos en la iteración previa mediante la optimización de una función de pérdida







- Elements: people in Mexico D.F
- Features: Age, Gender, Education, Residence, Industry
- **Class**: Salary Band:
 - Band 1 : Below 40,000
 - Band 2: 40,000 150,000
 - Band 3: More than 150,000
- Random forest:
 - 10k observaciones
 - 1 variable
 - 5 árboles (decisores)

	Salary Band	1	2	3
Age	Below 18	90%	10%	0%
	19-27	85%	14%	1%
	28-40	70%	23%	7%
	40-55	60%	35%	5%
	More than 55	70%	25%	5%

	Salary Band	1	2	3
Education	<=High School	85%	10%	5%
	Diploma	80%	14%	6%
	Bachelors	77%	23%	0%
	Master	62%	35%	3%

	Salary Band	1	2	3
Gender	Male	70%	27%	3%
	Female	75%	24%	1%













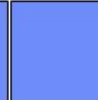


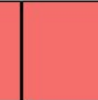







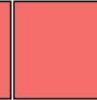
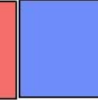


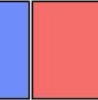

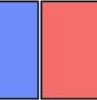
	Salary Band	1	2	3
Residence	Metro	70%	20%	10%
	Non-Metro	65%	20%	15%

	Salary Band	1	2	3
Industry	Finance	65%	30%	5%
	Manufacturing	60%	35%	5%
	Others	75%	20%	5%

<http://www.analyticsvidhya.com/blog/2014/06/introduction-random-forest-simplified/>

- Proceso de Inferencia: Media de Probabilidades: votación
 - EL resultado final será la media de la probabilidad que arroja cada uno de los 5 CARTs.
-
- Example: 1. Age : 35 years , 2, Gender : Male , 3. Highest Educational Qualification : Diploma holder, 4. Industry : Manufacturing, 5. Residence : Metro|

CART	Band	1	2	3
Age	28-40	70%	23%	7%
Gender	Male	70%	27%	3%
Education	Diploma	80%	14%	6%
Industry	Manufacturing	60%	35%	5%
Residence	Metro	70%	20%	10%
Final probability		70%	24%	6%

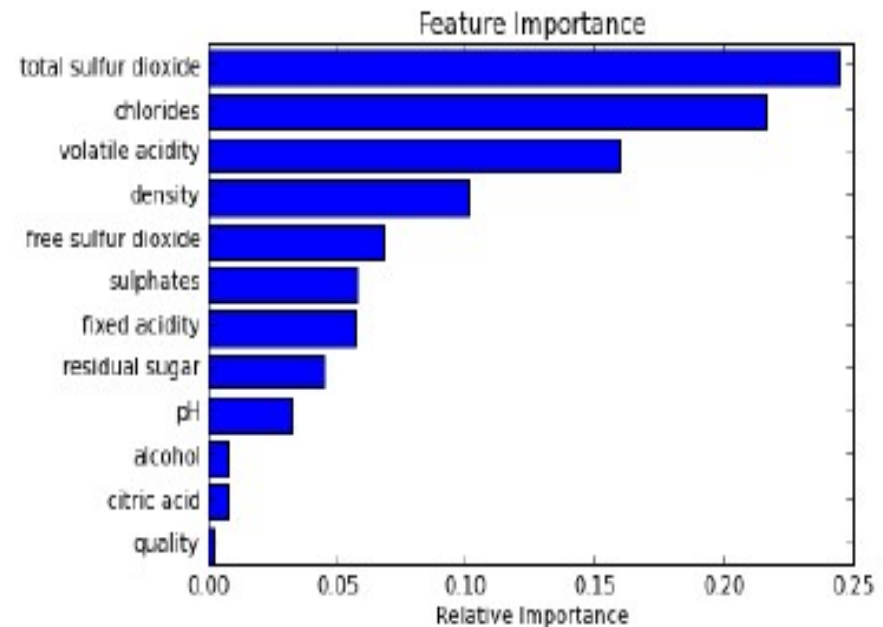
Model 1											7/10 correct
Model 2											7/10 correct
Model 3											6/10 correct
Ensemble Model (Majority Voting)											

For more tutorials: algobears.com

<https://algobears.com/2016/08/25/random-forest-tutorial/>

- Para **medir el error** de random forest se suele utilizar la técnica **denominada out-the-bag error**. Para cada árbol se utiliza el conjunto de objetos no seleccionados por su muestra bootstrap de entrenamiento para ser clasificados con dicho árbol. Promediando sobre el conjunto de árboles de random forest se puede estimar el error del algoritmo.
- Por otro lado, random forest puede ser **paralelizado** eficazmente puesto que cada árbol puede construirse de manera independiente a los otros árboles.
- Un incremento del número de árboles permite una mayor diversidad de los mismos, y por lo tanto reduce el error OOB. Sin embargo, **la mejora se estanca a partir de un determinado número de árboles**

- El mecanismo de construcción de random forest **permite establecer un baremo de la importancia de cada variable en la predicción final.**
- Para ello se calcula el error de la muestra Out of the bag y posteriormente se permutan elementos para calcular el error.
- Así las variables menos importantes deberían alterar menos la diferencia entre el error de la muestra OOB y el error de la muestra OOB permutada, que las variables importantes



VENTAJAS

- No se necesita poda
- Buena precision
- Evitar Overfitting
- Sencilla parametrización
- Se genera importancia de las variables.

INCONVENIENTES

- Variables categóricas con muchos niveles son favorecidas en el algoritmo
(muchos posibles valores ayudan mucho a separar, sin ser necesariamente relevantes).
- Mantenimiento de los rangos de datos en problemas de regression
- Reducida interpretabilidad.

GRACIAS