

Decision Tree Classifier and Ensembles

Dr. Mauricio Toledo-Acosta

Diplomado Ciencia de Datos con Python

Table of Contents

1 Árboles de Decisión

2 Random Forest

- Bagging

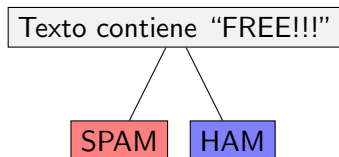
Árboles de Decisión

Árboles de Decisión

Un **árbol de decisión** es un modelo de aprendizaje automático utilizado tanto para tareas de **clasificación** como de **regresión**. Su estructura se asemeja a un árbol, con nodos que representan decisiones basadas en las características de los datos, y ramas que representan los resultados de esas decisiones.

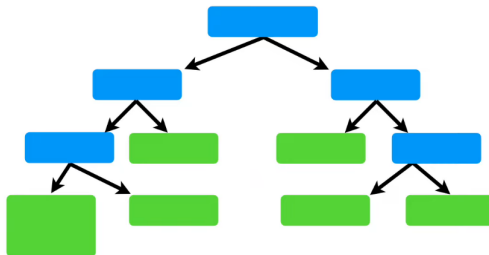
Un árbol de decisión plantea una pregunta en un nodo y se toma una decisión basada en la respuesta a la pregunta.

Un árbol de decisión



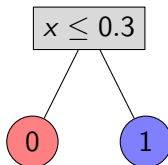
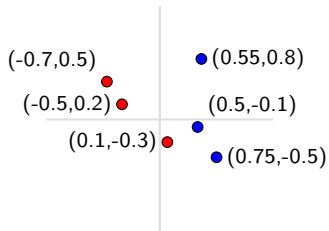
Si la decisión tomada es asignar una clase, es un árbol de decisión clasificador, si la decisión es sobre un valor continuo, es un árbol de decisión regresor.

Terminología

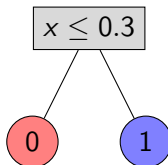
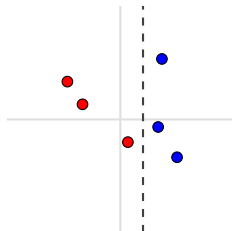


- Nodos (nodes)
- Hojas (leaves)

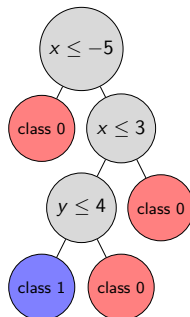
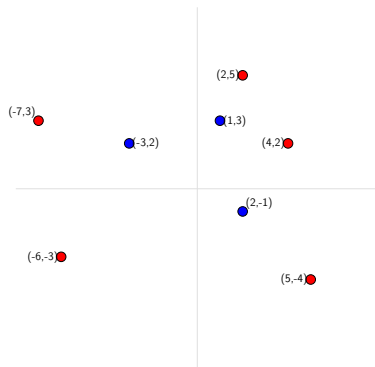
Un Ejemplo de clasificación



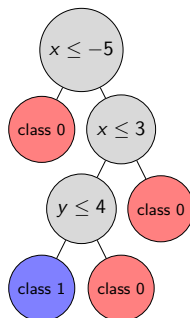
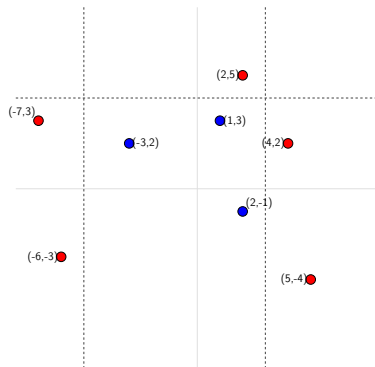
Un Ejemplo de clasificación



Otro Ejemplo de clasificación



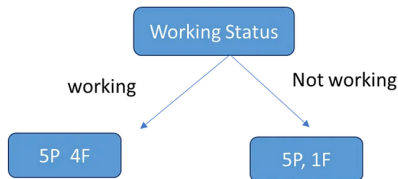
Otro Ejemplo de clasificación



¿Cómo se construye el árbol?

- 1 Analizamos cómo cada feature se correlaciona con las etiquetas.
 - Si es una variable categórica, tomamos posible valor.
 - Si es una variable continua, tomamos los puntos medios entre valores ordenados consecutivos.

Resp srl no	Target variable	Predictor variable	Predictor variable	Predictor variable
	Exam Result	Other online courses	Student backgroun d	Working Status
1	Pass	Y	Maths	NW
2	Fail	N	Maths	W
3	Fail	y	Maths	W
4	Pass	Y	CS	NW
5	Fail	N	Other	W
6	Fail	Y	Other	W
7	Pass	Y	Maths	NW
8	Pass	Y	CS	NW
9	Pass	n	Maths	W
10	Pass	n	CS	W
11	Pass	y	CS	W
12	Pass	n	Maths	NW
13	Fail	y	Other	W
14	Fail	n	Other	NW
15	Fail	n	Maths	W



Ejemplo, Video

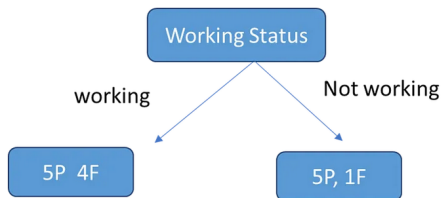
¿Cómo se construye el árbol?

2 Para cuantificar cómo se relaciona la feature con las etiquetas usamos:

- Information Gain
- Gini impurity

$$I_G(A) = 1 - \sum_{i=1}^J p_i^2$$

para un conjunto A de elementos pertenecientes a J clases $\{1, \dots, J\}$ con probabilidades p_1, \dots, p_J .

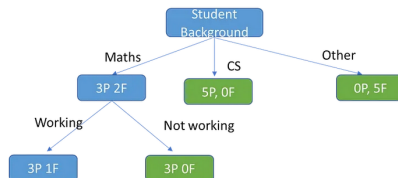


Ejemplo, Video

¿Cómo se construye el árbol?

- 8 Continuamos con el resto de features de la misma forma.

Resp srl no	Target variable	Predictor variable	Predictor variable	Predictor variable
	Exam Result	Other online courses	Student background	Working Status
1	Pass	Y	Maths	NW
2	Fail	N	Maths	W
3	Fail	y	Maths	W
4	Pass	Y	CS	NW
5	Fail	N	Other	W
6	Fail	Y	Other	W
7	Pass	Y	Maths	NW
8	Pass	Y	CS	NW
9	Pass	n	Maths	W
10	Pass	n	CS	W
11	Pass	y	CS	W
12	Pass	n	Maths	NW
13	Fail	y	Other	W
14	Fail	n	Other	NW
15	Fail	n	Maths	W



Ejemplo, Video

Parámetros Importantes

Los parámetros más importantes para un árbol de decisión son:

- Criterio (`criterion`) para evaluar las divisiones: gini, entropy, etc.

Parámetros Importantes

Los parámetros más importantes para un árbol de decisión son:

- Criterio (`criterion`) para evaluar las divisiones: gini, entropy, etc.
- Profundidad máxima (`max_depth`) es la máxima profundidad del árbol. De otra forma, los nodos se van expandiendo hasta que son puros o que quedan menos que el siguiente parámetro.

Parámetros Importantes

Los parámetros más importantes para un árbol de decisión son:

- Criterio (`criterion`) para evaluar las divisiones: gini, entropy, etc.
- Profundidad máxima (`max_depth`) es la máxima profundidad del árbol. De otra forma, los nodos se van expandiendo hasta que son puros o que quedan menos que el siguiente parámetro.
- Mínimo número de instancias para dividir (`min_samples_split`).

Ventajas y Desventajas

Ventajas:

- Los DTs requieren menos esfuerzo en el preprocesamiento.
- DTs no requieren normalización o re-escalamiento de los datos.
- Los valores faltantes no afectan el algoritmo de manera significativa.
- La mezcla de variables binarias (categóricas) y numéricas no es problema.
- Los DTs son intuitivos y fáciles de explicar.

Desventajas:

- Un pequeño cambio en los datos puede causar un gran cambio en la estructura del árbol causando inestabilidad.
- Los DTs a menudo involucran un tiempo grande para entrenar el modelo, al menos relativamente.

Table of Contents

1 Árboles de Decisión

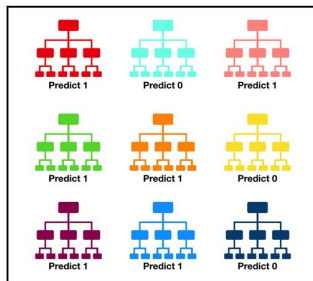
2 Random Forest

- Bagging

Random Forest

Random Forest

Un bosque aleatorio (Random Forest) es un método de clasificación que funciona mediante la construcción de varios árboles de decisión en el entrenamiento. La salida de un random forest es la clase seleccionada por la mayoría de árboles.



Bagging: Bootstrap Aggregating

Bagging

Bootstrap aggregating, o **bagging**, es un **meta-algoritmo de ensambles de Machine Learning** diseñado para mejorar la estabilidad y precisión de algoritmos de Machine Learning de clasificación y regresión, además de reducir el over-fitting. Usualmente se aplica a árboles de decisión.

El **Bagging** consiste de dos pasos:

- **Boostrapping**: Generar varios conjuntos de datos muestreando el conjunto de datos original, con reemplazo. En cada conjunto de datos entrenar un árbol de decisión.

Bagging: Bootstrap Aggregating

Bagging

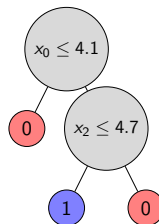
Bootstrap aggregating, o **bagging**, es un **meta-algoritmo de ensambles de Machine Learning** diseñado para mejorar la estabilidad y precisión de algoritmos de Machine Learning de clasificación y regresión, además de reducir el over-fitting. Usualmente se aplica a árboles de decisión.

El **Bagging** consiste de dos pasos:

- **Boostrapping**: Generar varios conjuntos de datos muestreando el conjunto de datos original, con reemplazo. En cada conjunto de datos entrenar un árbol de decisión.
- **Aggregating**: Para clasificar una instancia, el ensamble junta las predicciones hechas por cada árbol y toma la clase predicha por la mayoría.

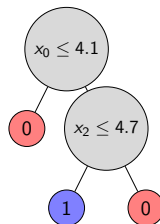
Bagging

	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	6.5	4.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1



Bagging

	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	6.5	4.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1



Bootstrapping: Se entrenan varios árboles usando muestro aleatorio de instancias, con reemplazo.

	x_0	x_1	y
2			
0			
2			
4			
5			
5			

	x_0	x_1	y
2			
1			
3			
1			
4			
4			

	x_0	x_1	y
4			
1			
3			
0			
0			
2			

	x_0	x_1	y
3			
3			
2			
5			
1			
2			

Bagging

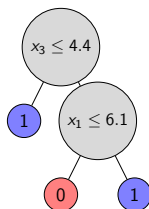
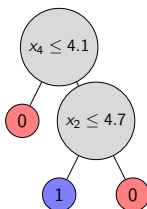
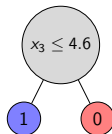
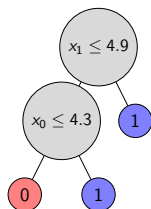
	x_0	x_1	y
2			
0			
2			
4			
5			
5			

	x_0	x_1	y
2			
1			
3			
1			
4			
4			

	x_0	x_1	y
4			
1			
3			
0			
0			
2			

	x_0	x_1	y
3			
3			
2			
5			
1			
2			

Aggregating: La predicción del ensamble es la predicción mayoritaria.



Ventajas y Desventajas

Ventajas:

- Es muy intuitivo, los hiper-parámetros son fáciles de entender.

Desventajas:

Ventajas y Desventajas

Ventajas:

- Es muy intuitivo, los hiper-parámetros son fáciles de entender.
- Es posible conocer la importancia de las features, de acuerdo a los criterios de impureza.

Desventajas:

Ventajas y Desventajas

Ventajas:

- Es muy intuitivo, los hiper-parámetros son fáciles de entender.
- Es posible conocer la importancia de las features, de acuerdo a los criterios de impureza.
- Tienden a evitar el over-fitting.

Desventajas:

Ventajas y Desventajas

Ventajas:

- Es muy intuitivo, los hiper-parámetros son fáciles de entender.
- Es posible conocer la importancia de las features, de acuerdo a los criterios de impureza.
- Tienden a evitar el over-fitting.

Desventajas:

- La gran cantidad de árboles y la profundidad de estos pueden hacer al método lento para usar en tiempo real.

Ventajas y Desventajas

Ventajas:

- Es muy intuitivo, los hiper-parámetros son fáciles de entender.
- Es posible conocer la importancia de las features, de acuerdo a los criterios de impureza.
- Tienden a evitar el over-fitting.

Desventajas:

- La gran cantidad de árboles y la profundidad de estos pueden hacer al método lento para usar en tiempo real.
- Se pierde un poco la intuición respecto a un solo árbol de decisión.