Árboles (CARTs), Bagging and Random Forests

Big Data y Machine Learning para Economía Aplicada

Ignacio Sarmiento-Barbieri

Universidad de los Andes

Motivación

- Queremos predecir:

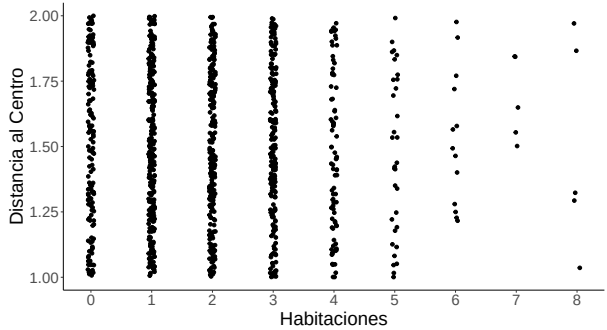
$$Price = f(structural\ attributes, amenities, ...) \tag{1}$$

  - Podemos aplicar linear regression,

$$Price = \beta_0 + \beta_1 Habitaciones + \beta_2 DCBD + u \tag{2}$$

- Aplicar OLS a este problema requiere tomar algunas decisiones.

Motivación


Distancia al Centro
2.00
1.75
1.50
1.25
1.00
0
1
2
3
4
5
6
7
8
Habitaciones

Agenda

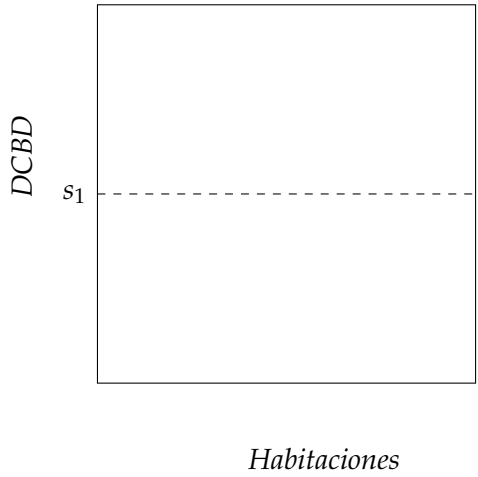
1. Árboles
   - ¿Qué hacen?
   - ¿Cómo lo hacen?
   - Sobreajuste

Agenda

1. Árboles
   - ¿Qué hacen?
   - ¿Cómo lo hacen?
   - Sobreajuste

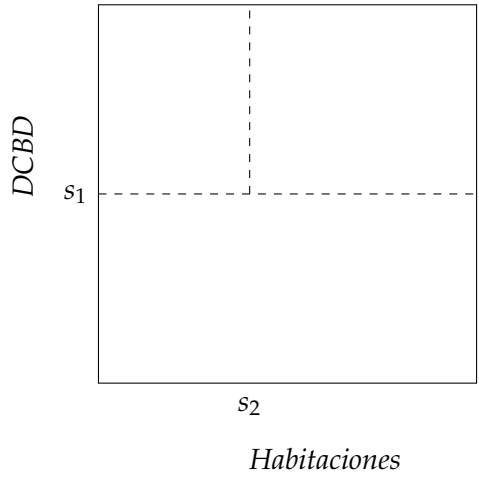
¿Qué hacen?

1. Y es la variable a predecir, los insumos son X_1 y X_2
2. Partimos el espacio (X_1, X_2) en dos regiones, en base a una sola variable .
3. Punto: elegir la variable y el punto de partición de manera óptima.



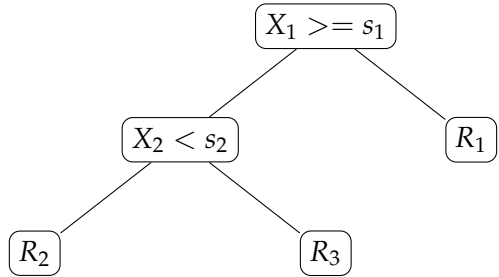
¿Qué hacen?

1. Y es la variable a predecir, los insumos son X_1 y X_2
2. Partimos el espacio (X_1, X_2) en dos regiones, en base a una sola variable .
3. Punto: elegir la variable y el punto de partición de manera óptima.
4. Continuamos partiendo

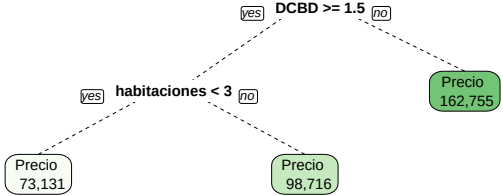


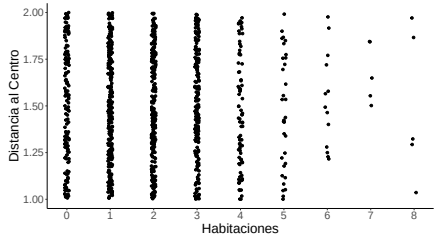
¿Qué hacen?

1. Y es la variable a predecir, los insumos son X_1 y X_2
2. Partimos el espacio (X_1, X_2) en dos regiones, en base a una sola variable .
3. Punto: elegir la variable y el punto de partición de manera optima (mejor ajuste global.
4. Continuamos partiendo



¿Qué hacen?


yes
DCBD >= 1.5
no
yes
habitaciones < 3
no
Precio
162,755
Precio
73,131
Precio
98,716


Distancia al Centro
2.00
1.75
1.50
1.25
1.00
0
1
2
3
4
5
6
7
8
Habitaciones

Agenda

1. Árboles
   - ¿Qué hacen?
   - ¿Cómo lo hacen?
   - Sobreajuste

¿Cómo construimos un árbol de decisión?

- Datos: $y_{n\times 1}$ y $X_{n\times k}$
- Definiciones
  - j es la variable que parte el espacio y s es el punto de partición
  - Defina los siguientes semiplanos

$$R_1(j,s) = \{X|X_j \leq s\} \;\; \& \;\; R_2(j,s) = \{X|X_j > s\} \tag{3}$$

  - El problema: buscar la variable de partición X_j y el punto s de forma tal que

$$\min_{j,s} \left[\min_{y_{R_1}} \sum_{x_i \in R_1(j,s)} (y - y_{R_1})^2 + \min_{y_{R_2}} \sum_{x_i \in R_2(j,s)} (y - y_{R_2})^2 \right] \tag{4}$$

¿Cómo construimos un árbol de decisión?

- ¿Cuál es la solución?

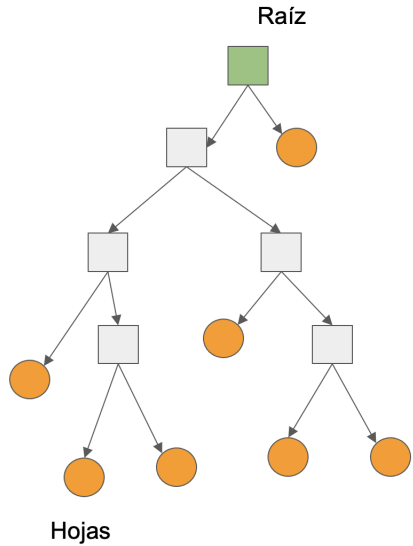
¿Cómo construimos un árbol de decisión?

photo from https://www.dailydot.com/parsec/batman-1966-labels-tumblr-twitter-vine/

Agenda

1. Árboles
   - ¿Qué hacen?
   - ¿Cómo lo hacen?
   - Sobreajuste

Sobreajuste



Sobreajuste. Algunas soluciones

- Fijar la profundidad del árbol.
- Fijar la cantidad de hojas (nodos terminales, regiones).
- Fijar la mínima cantidad de datos que están contenidos dentro de cada hoja.
- Pruning (poda).
  - Dejar crecer un árbol muy grande T_0
  - Luego cortarlo obteniendo sub-árbol (*subtree*)
  - Como cortarlo?

Pruning (poda)

- No es posible calcular el error de predicción usando cross-validation para cada sub-árbol posible
- Solución: *Cost complexity pruning (cortar las ramas mas débiles)*
  - Indexamos los arboles con T.
  - Un sub-árbol $T \in T_0$ es un árbol que se obtuvo colapsando los nodos terminales de otro árbol (cortando ramas).
  - $[T]$ = número de nodos terminales del árbol T

Pruning (poda)

- Cost complexity del árbol T

$$C_\alpha(T) = \sum_{m=1}^{[T]} n_m Q_m(T) + \alpha[T] \tag{5}$$

  - donde $Q_m(T) = \frac{1}{n_m} \sum_{x_i \in R_m} (y_i - \hat{y}_m)^2$ para los árboles de regresión
  - $Q_m(T)$ penaliza la heterogeneidad dentro de la regresión y el número de regiones
  - Objetivo: para un dado α, encontrar el pruning óptimo que minimice $C_\alpha(T)$

Pruning (poda)

- Mecanismo de búsqueda para T_α (pruning óptimo dado α).

  Resultado: para cada α hay un sub-árbol único T_α que minimiza $C\alpha$ (T).

  - Eliminar sucesivamente las ramas que producen un aumento mínimo en $\sum_{m=1}^{[T]} n_m Q_m(T)$

  - Se colapsa hasta el nodo inicial pero va a través de una sucesión de árboles
  - T_α pertenece a esta secuencia. (Breiman et al., 1984)

Pruning (poda)

Algoritmo Completo

1. Utilizamos particiones recursivas binarias para hacer crecer el árbol
2. Para un dado α, aplicamos *cost complexity pruning* al árbol para obtener la secuencia de los subarboles como α.
3. Utilizamos K-fold cross-validation para elegir α.
4. Tenemos entonces una secuencia de subarboles para distintos valores de α
5. Elegimos el α y el subárbol que tienen el menor error de predicción.

Ejemplo

photo from https://www.dailydot.com/parsec/batman-1966-labels-tumblr-twitter-vine/