Aprendizaje Estadístico Supervisado

Natalia da Silva

2024

Plan de hoy

Árboles de regresión y clasificación. (Capítulo 8)

- Introducción: idea global
- Atributos relevantes
- Construcción y poda (CART)
- Propiedades y limitaciones
- Variantes de los árboles.

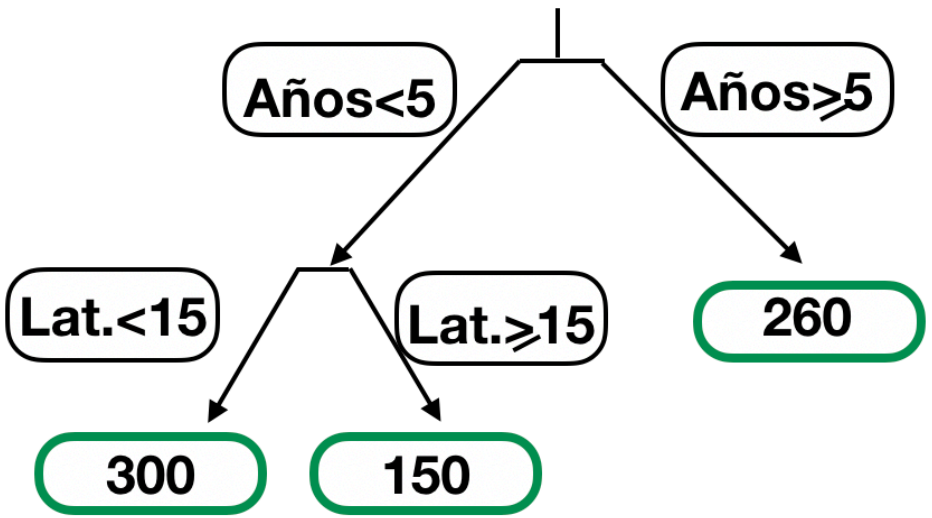
Introducción

Introducción

- Fifty Years of Classification and Regression Trees (Loh 2014)
- Primer algoritmo de árbol de regresión (Automatic Interaction Detection, AID) (Morgan and Sonquist 1963).
- Veremos el algoritmo de CART (Breiman et al. 1984).

Introducción

Queremos predecir el precio de una vivienda en función de su locación geográfica y sus años de construcción.

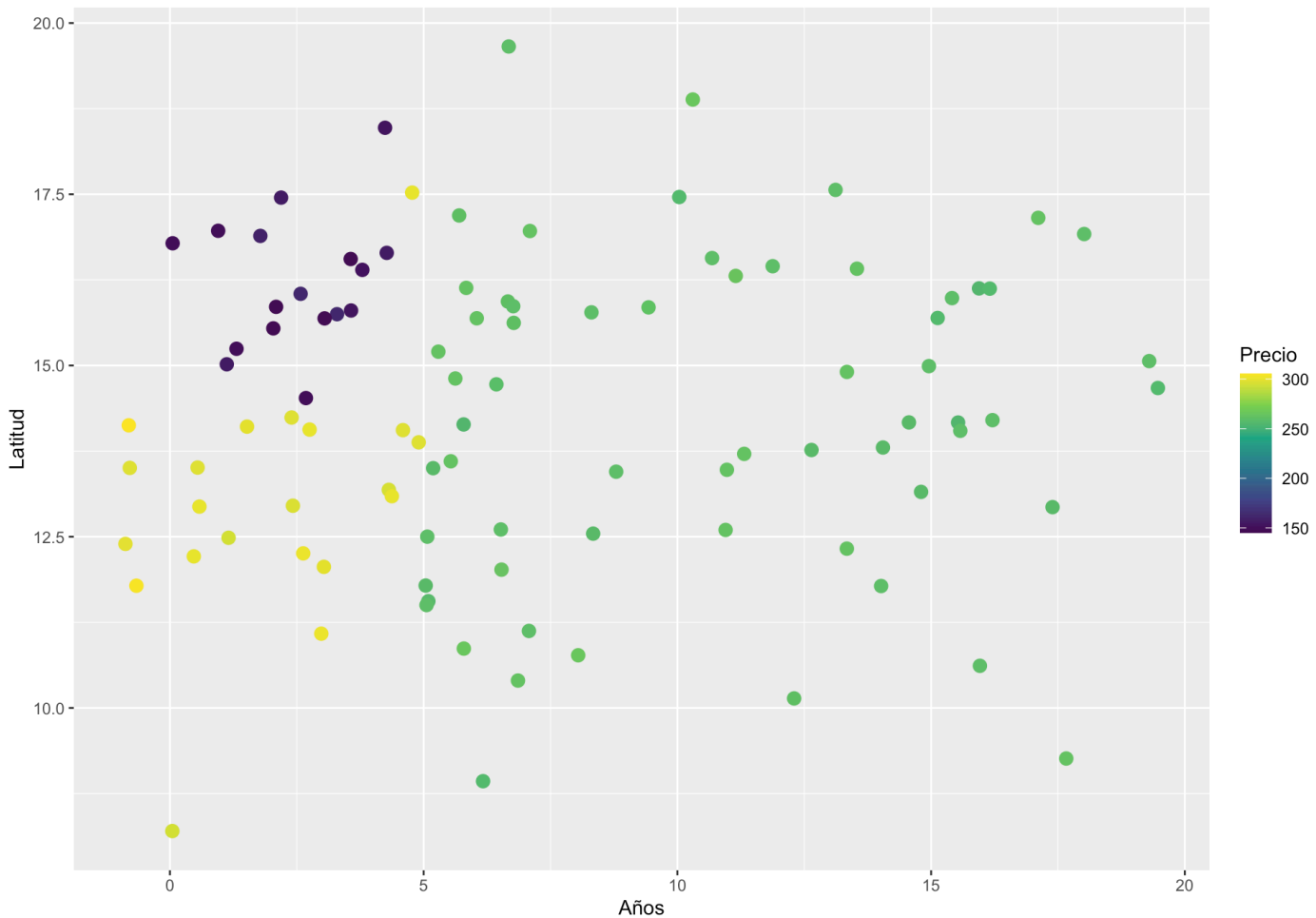


- ¿Resulta intuitivo el *modelo*?
- ¿Cómo se predice el precio de una vivienda nueva?

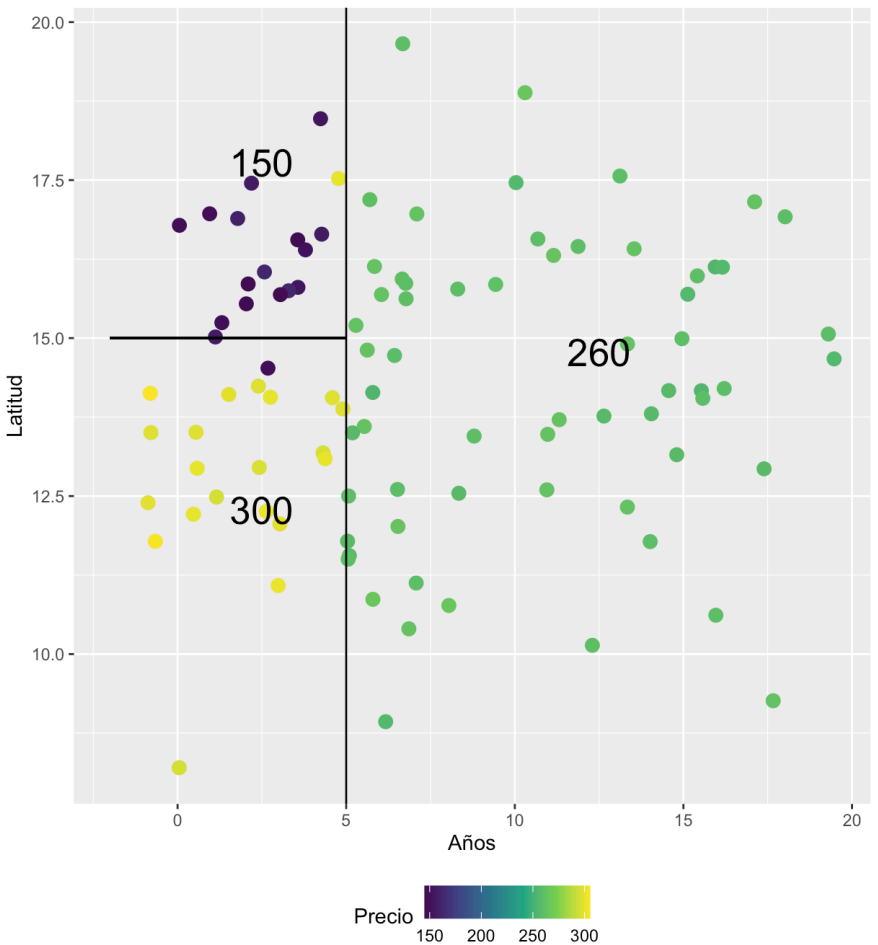
Árbol, compuesto de nodos, ramas y hojas. Nodos terminales son hojas.

Regiones en espacio de predictores

Los árboles de decisión forman de particiones anidadas que dividen el espacio de los predictores ($\mathbf{X}$s) donde en cada partición se usa un modelo simple para predecir la respuesta.



Regiones en espacio de predictores



Se forman 3 regiones:

$$R_1 = \{\text{Año} \geq 5\}$$

$$R_2 = \{\text{Año} < 5\} \cap \{\text{Latitud} < 15\}$$

$$R_3 = \{\text{Año} < 5\} \cap \{\text{Latitud} \geq 15\}$$

En cada región, estima un precio:

$$\hat{y_i} = 260 \text{ si } i \in R_1$$

$$\hat{y_i} = 300 \text{ si } i \in R_2$$

$$\hat{y_i} = 150 \text{ si } i \in R_3$$

Modelo de árbol

$$Y = f(X) + \epsilon$$

Se aproxima $f(X)$ con funciones constantes en regiones del espacio de predictores,

$$f(X) = \sum_{m=1}^{M} c_m I\{(X) \in R_m\}$$

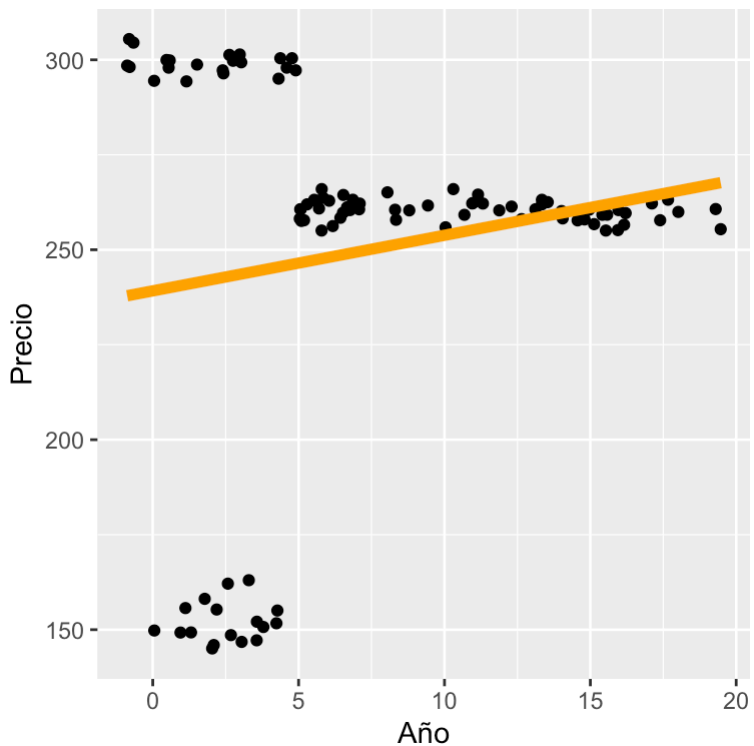
- ¿Cómo se construyen R_m?
- ¿Cómo se calcula c_m? para problemas de regresión y clasificación?
- ¿Cuánto vale M?

LM vs Tree, ejemplo vivienda

Los *árboles* pueden ajustar efectos no lineales automáticamente.

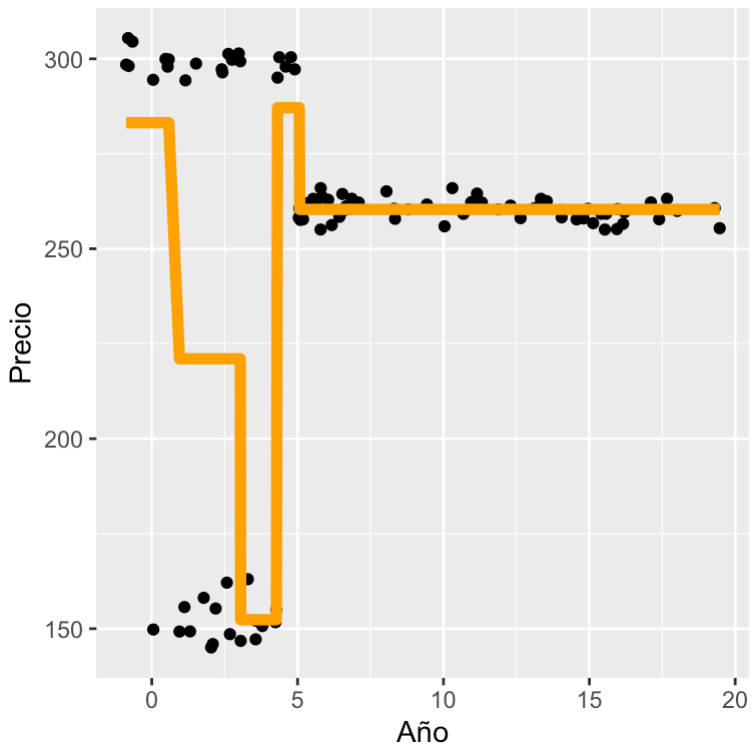
Regresión lineal:

$$f(X) = \sum_{i=1}^{n} \beta X_i$$



Árbol de regresión:

$$f(X) = \sum_{m=1}^{M} c_m I\{(X) \in R_m\}$$



Idea principal

Idea básica es particionar el espacio de los predictores en un número de regiones rectangulares, donde se ajusta un modelo simple en cada una.

El conjunto de reglas para particionar el epacio de los predictores puede ser resumido en un **Árbol de decisión**

En cada región, se predice la variable respuesta como:

- **Regresión**: Promedio de la respuesta en la hoja.
- **Clasificación**: Clase más frecuente de la respuesta en la hoja.

Atributos relevantes

Selección de atributos relevantes

- Las regiones se construyen en forma *recursiva*.
- Los datos se dividen en subconjuntos de acuerdo a una *variable explicativa*.

¿Cómo seleccionar que variable explicativa es usada para dividir los datos?

Selección de atributos relevantes

- Nodo padre: los datos sin dividir
- Nodos hijos: los datos subdivididos
- Regla de partición: condición que verificamos en cada observación para saber a que nodo-hijo pertenece
- Medida de impureza: evaluar la *variabilidad* de Y en cada nodo

En términos generales: se prueban todas las posibles particiones y se elige aquella que produce una mayor reducción de la impureza o *ganancia de información*.

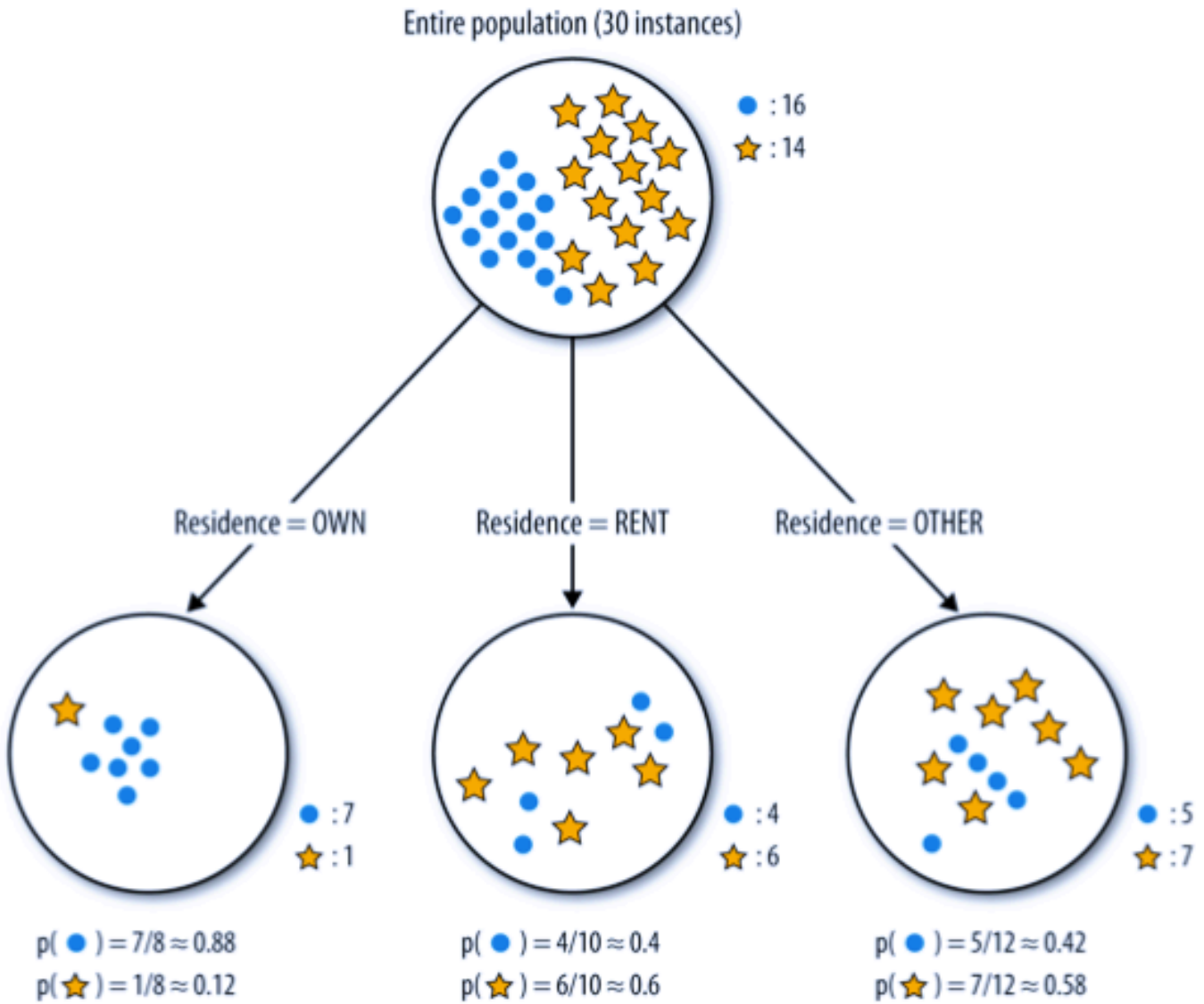
Ejemplo en Clasificación

Predecir abandono de clientes: $\mathrm{Y_i} = \mathrm{I(i\ abandona)}$

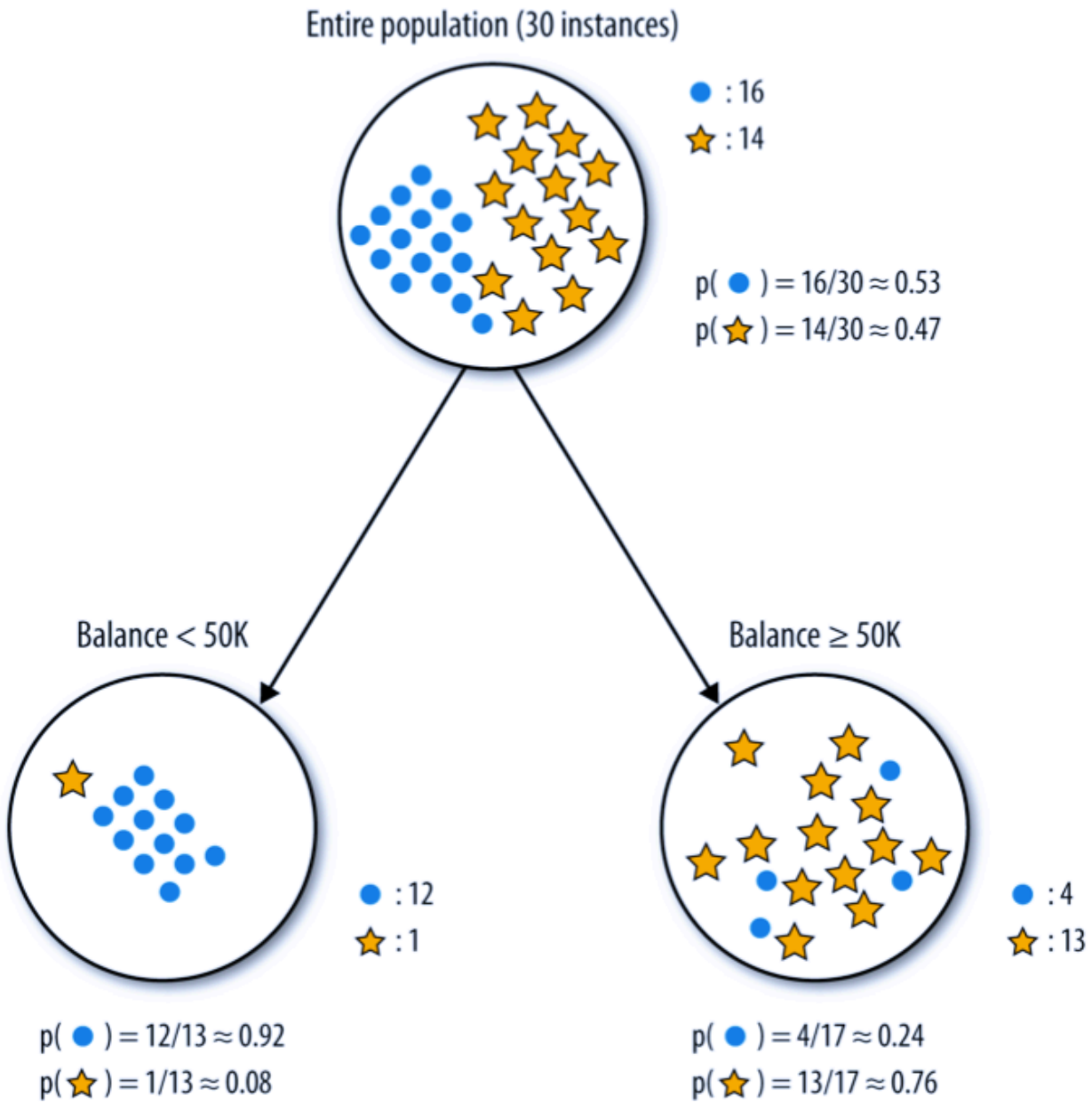
- 30 clientes en total, 16 abandonan
- 2 variables: Propiedad de la casa, Balance en el banco

División según Propieddad

Estrellas y puntitos !!!!



División según Balance



Medida de impureza: Entropía

Para seleccionar que variable es mejor para la primera partición, uso una medida de impureza.

Entropía: Medida de ‘’desorden” en un conjunto. Permite evaluar que tan homogeneo es un grupo en términos de una variable de interés.

La entropía del grupo G, según una variable categórica $Y \in \{1, 2, \ldots, K\}$, se define como:

$$E_G = -\sum_{k=1}^{K} p_k \log_2(p_k)$$

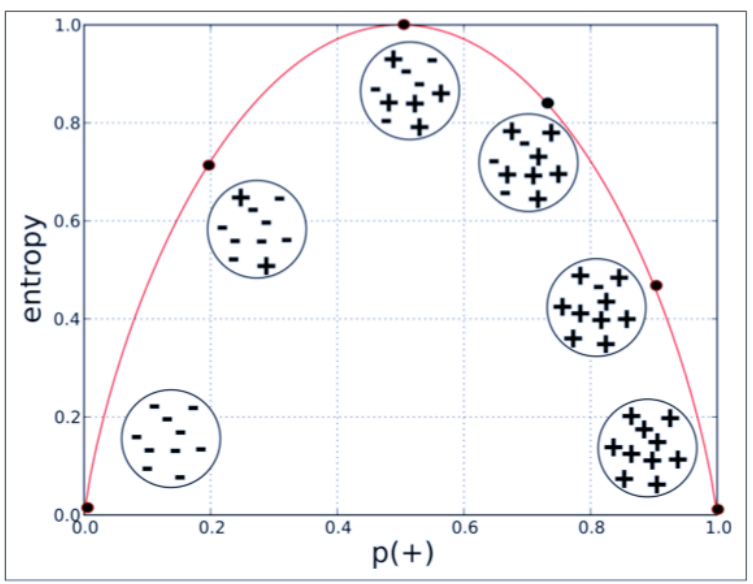
Medida de impureza: Entropía

Si la variable de interés tiene 2 clases (posibles valores):

$$E = -(p_1 \log_2(p_1) + p_2 \log_2(p_2))$$

El máximo *desorden* se da en $p_1 = p_2 = 1/2$.

Hay mucha información (mínimo desorden) cuando el nodo es **puro**



Ganancia de información

¿Cuanta información se gana al dividir los datos?

La diferencia entre la entropía en el nodo padre y el promedio en los nodos hijos:

$$IG_r = E_P - \sum_j p_j E_{H_j}$$

$$IG_{balance} = 0.99 - (0.43 \times 0.39 + 0.57 \times 0.79) = 0.37$$

$$IG_{casa} = 0.99 - (0.26 \times .53 + .3 \times 0.97 + 0.4 \times 0/98) = 0.13$$

Es más informativo dividir los datos según la variable Balance.

Atributos para el caso de regresión

Cuando $\mathbf{Y} \in \mathbf{R}$, la medida de impureza utilizada es la suma de cuadrados de *residuos* (SCR).

Ganancia de información se asocia a reducir SCR.

Seleccionamos el atributo que genera la mayor reducción de SCR al subdividir los datos.

CART: Construcción y poda

Classification and regression trees

CART: Classification and regression trees (Breiman et al. 1984), uno de los algoritmos de árboles más populares.

- Divisiones binarias
- Particiones paralelas a los ejes (2D)
- El árbol es podado para mejorar su performance

Aparte de ser muy aplicado en forma directa, sirve de base para métodos más avanzados.

Construcción de un árbol de regresión

El objetivo sería definir las regiones R_1 , R_2 … R_M de forma de minimizar la SCR

$$SCR = \sum_{m=1}^{M} \sum_{i \in R_m} (y_i - \hat{y}_{R_m})^2$$

- $\hat{y}_{R_m}$ es el promedio de la variable de respuesta en la región R_m.
- NO es viable computacionalmente, habría que considerar todas las posibles particiones del espacio de los predictores en M cajas.
- Se propone un procedimiento iterativo que optimiza en cada paso.

Construcción: algoritmo

Esquema del procedimiento iterativo (algoritmo):

1. Comienza con todos los datos de entrenamiento en un nodo (grupo).
2. Seleccionar la regla $\{X_j < s\}$, que maximiza la ganancia de información (algoritmos "glotones": mejor partición en cada paso)
3. Dividir los datos en 2 nodos hijos:

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

4. Evaluar si se cumple el *criterio de parada* (ej: menos 5 observaciones en un nodo).
5. En cada nodo hijo, repetir los pasos 2 y 3.

Los dos pasos clave son hallar la mejor partición y determinar el final.

Construcción: hallar partición

Encontrar la mejor partición en 1 paso:

- $\{X_j < s\}$ divide los datos en $R_1(j, s) = \{X | X_j \leq s\}$ y $R_2(j, s) = \{X | X_j > s\}$
- Elige la variable j y el punto de corte s tal que minimice:

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

- $\bar{y}_{R_1}$ y $\bar{y}_{R_2}$ la media de la respuesta cada región.

Construcción: hallar partición

- El algoritmo decide automáticamente la variable para partir y el punto en el que se da la partición y la topología del árbol.
- Restringir la búsqueda a particiones paralelas $\{X_j < s\}$ reduce sustancialmente el costo computacional.
- La mejor partición se define mediante algoritmos “glotones”, que miran sólo ese paso.
- Se recorren todas las variable j y, en cada variable, todos los valores observados s.
- Notar que si Y es categórica hay que evaluar de otra forma (ej: en base a la entropía).

¿Hasta cuanto dejo crecer el árbol?

El tamaño del árbol (cantidad de nodos) determina la complejidad del modelo.

- Muy grande, sobreajusta (más complejo, mayor error por varianza)
- Muy chico, subajusta (más simple, mayor error por sesgo)

Es necesario detener el crecimiento del modelo,

- El error test no disminuye
- Hay pocos datos en la hoja

¿Hasta cuanto dejo crecer el árbol?

En CART:

1. Crecer un árbol *maximal* (T_o) con algún criterio de parada muy básico.
2. **Podar** T_o usando una penalidad por costo-complejidad.

(es una *estimación penalizada* !!!)

Criterio de costo-complejidad

Se define:

- Sea T_o el árbol completo, sin podar.
- Definimos un sub árbol $T \subset T_o$, algún árbol obtenido luego de podar T_o
- $|T|$: cantidad de nodos terminales del árbol T.
- $N_m = \#\{x_i \in R_m\}$ (Cantidad de datos en R_m)
- $\hat{c_m} = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$ (Predicción en R_m).
- $Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c_m})^2$ (Medida de impureza).

Criterio de costo-complejidad

En base a las definiciones anteriores, se propone un Costo a minimizar

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$

- $\alpha \geq 0$ controla el balance entre el tamaño del árbol y la bondad del ajuste a los datos.
- Valores grandes de α resultan en árboles pequeños, mucha penalidad al tamaño.
- Si $\alpha = 0$ obtenemos T_0, no penaliza el tamaño

Podar - Selección del árbol

Podar implica colapsar los nodos internos

- para cada α, encontrar el sub-árbol $T_\alpha \subseteq T_o$ que minimiza $C_\alpha(T)$
- queda una secuencia $T_{\alpha_1} \subseteq T_{\alpha_2} \subseteq \ldots \subseteq T_o$

Selección del árbol:

- El árbol final, T_{α^*} es el que minimiza el error de predicción
- El tamaño óptimo debería ser elegido adaptativamente de los datos.

Precios de casas en Montevideo

Árboles de Regresión, Ejemplo

Tesis Maestría en Economía: **Predicción de precios de la vivienda**. *Aprendizaje estadístico con datos de ofertas y transacciones para Montevideo, 2019*. Pablo Picardo

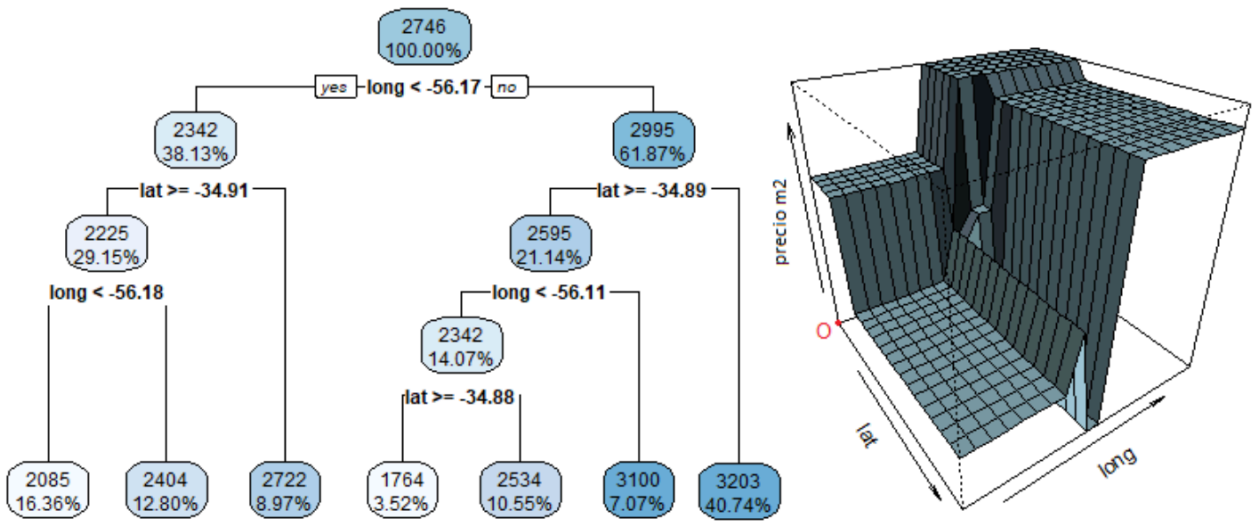
- **Objetivo:**Predecir el precio de oferta de las viviendas en Montevideo
- **Datos:** Oferta de casas y apartamentos de mercado libre febrero 2018- Enero 2019.

Ajuste con `rpart`

Los datos están en el objeto *dat_apt*, ajustamos un árbol SIN podar.

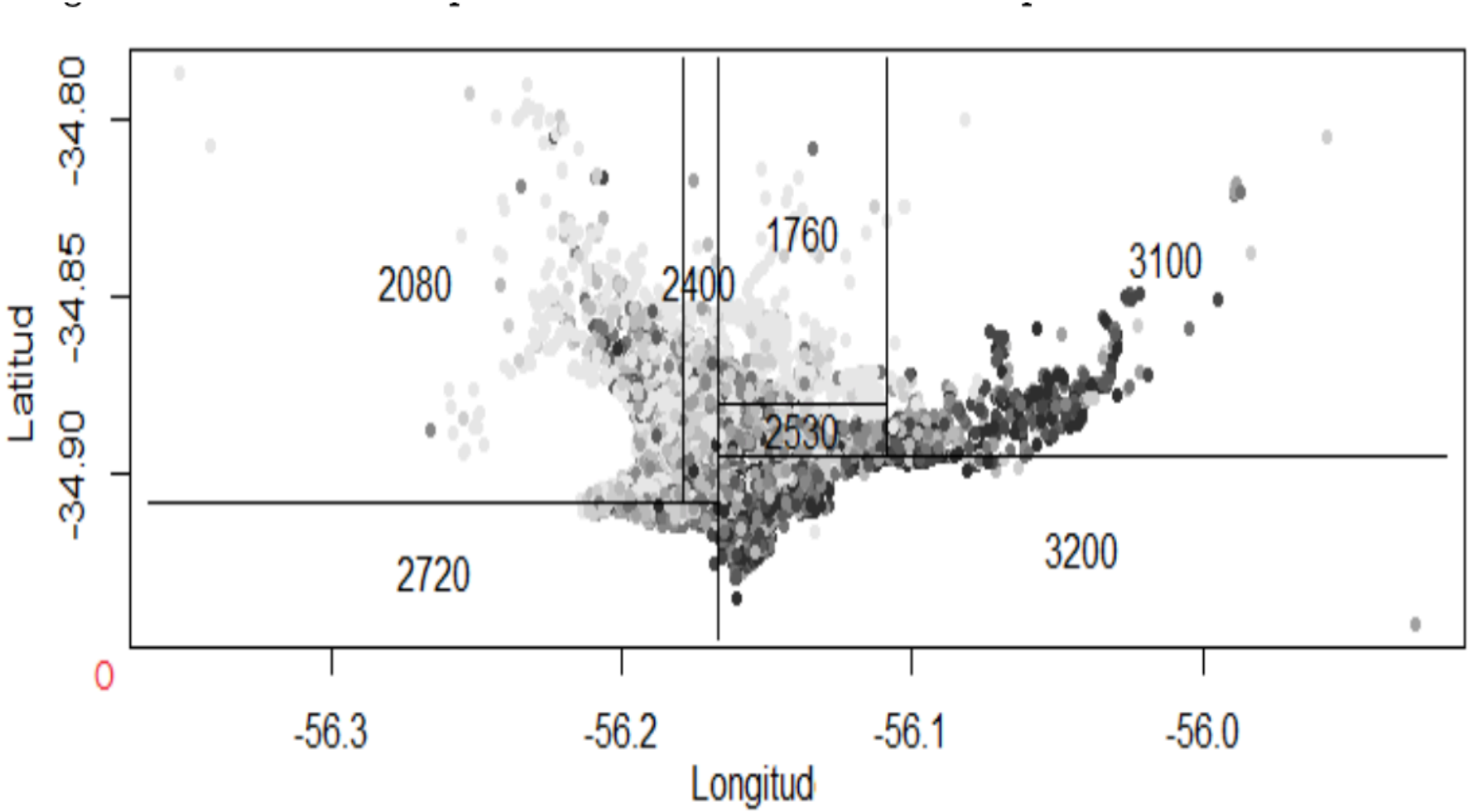
```
 1 # Divide muestra en subconjunto de entrenamiento y validac
 2 intrain <- sample(x = 1:nrow(dat_apt), size = nrow(dat_apt
 3 training <- dat_apt[intrain,]
 4 testing <- dat_apt[-intrain,]
 5 
 6 # Usamos rpart para estimar el árbol
 7 # dos argumentos básicos: fórmula y datos
 8 # usamos unicamente lat + long como predictores
 9 tree_apto <- rpart(precio_apt ~ lat + long, data = trainin
10 
11 # dibujamos el alrbol que resulta
12 rpart.plot(tree_apt, digits = (-5))
```

Árboles de Regresión, para el precio de los apartamento en MVD



- Presenta: precio promedio y el porcentaje de observaciones.
- Al inicio, el precio promedio de toda la muestra es USD 2.746.
- Primera divide entre oeste y este, `long < -56,17`
- En oeste precio promedio de USD 2.342 y al este de USD 2.995.

Árboles de Regresión



Algunas características

Propiedades atractivas:

- Son simples de usar e interpretar.
- Puede ser usado con predictores mixtos, categóricos y cuantitativos.
- Incorpora interacciones y transformaciones monótonas de forma automática.
- Pueden ser aplicados con datos faltantes en los predictores.

Problemas

- Pueden ser inestables (en particular si no se podan).
- En general presentan mayor error que otros métodos de AE.
- Predice una cantidad finita de valores, problema cuando Y sea continua.

Árboles de clasificación

- La repuesta toma valores en $\{1, \ldots, K\}$.
- La construcción del árbol es similar al de regresión.

Hay que modificar:

- Estimación de $f(X)$
- Criterio para partir el nodo y para podar el árbol

Estimo $f(X)$

Sea $\hat{p}_{mk}$ la proporción de observaciones de la clase k en el nodo m,

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

Las observaciones en el nodo m se clasifican basado en **voto mayoritario**,

$$k(m) = \text{argmax}_k \hat{p}_{mk}$$

la clase mayoritaria en el nodo m.

Criterio de partición del nodo m

Varias medidas $Q_m(T)$ para evaluar una partición.

Medidas de impureza del nodo m:

- **Error de clasificación:** $\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \max_k(\hat{p}_{mk}(m))$
- **Gini index:** $\sum_{k \neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$
- **Cross-entropy or deviance:** $-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

Cualquiera de estas medidas puede ser usada para guiar la poda teniendo en cuenta el criterio de costo-complejidad, en general se usa el error de clasificación.

Ejemplo:Abandono 1ero 2016-2017

Modelos de clasificación para el abandono

El primer año de secundaria es el que presenta mayores niveles de abandono.

Objetivo: explorar y predecir el abandono de los alumnos pertenecientes a primer año de educación secundaria pública en Uruguay.

Datos: Estudiantes que cursan primero de secundaria en 2016. Transiciones de dichos alumnos entre 2016 y 2017, 40.233 alumnos en 254 centros educativos.

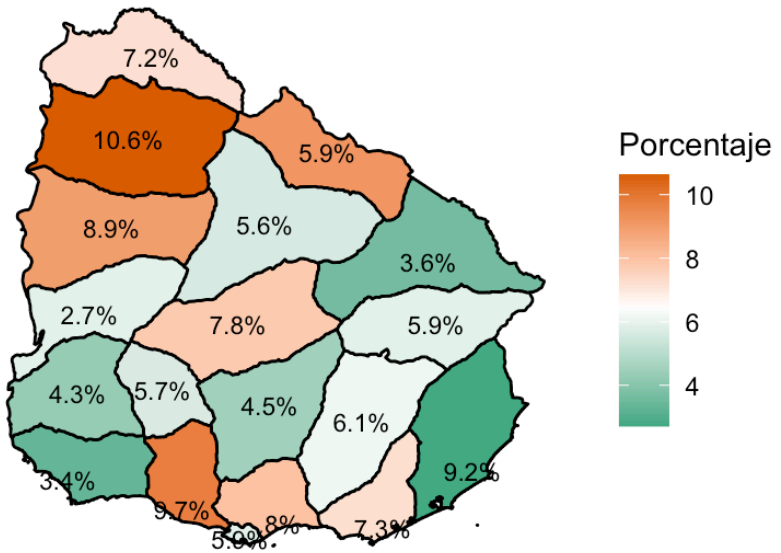
Modelos de clasificación para el abandono

- Variable de respuesta categórica con dos niveles (abandona, no abandona).
- Variables explicativas: Contexto sociocultural, sexo, extra edad fuerte, extra edad leve, inasistencias relativas, centro educativo, departamento.
- Problema: los datos están muy desbalanceados sólo un 7% abandonan

Ejemplo abandono

- Abandono: si cursó primero de CES en 2016 y en 2017 no se anotó en el sistema de educación pública.
- Limitante no hay datos de educación privada.
- Usando esta definición hay 7% de abandono en 2016-2017

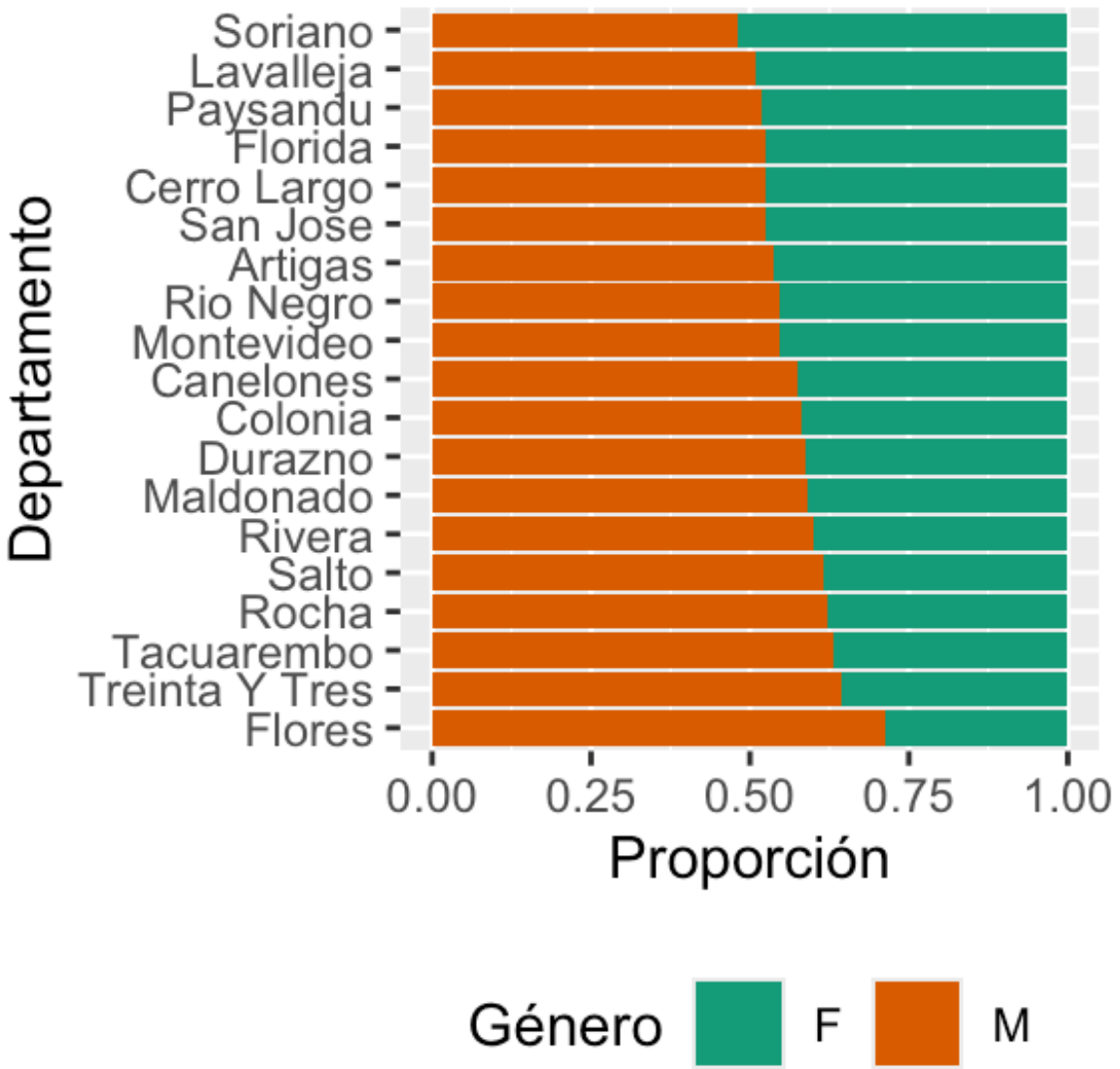
Abandono 1ero 2016-2017



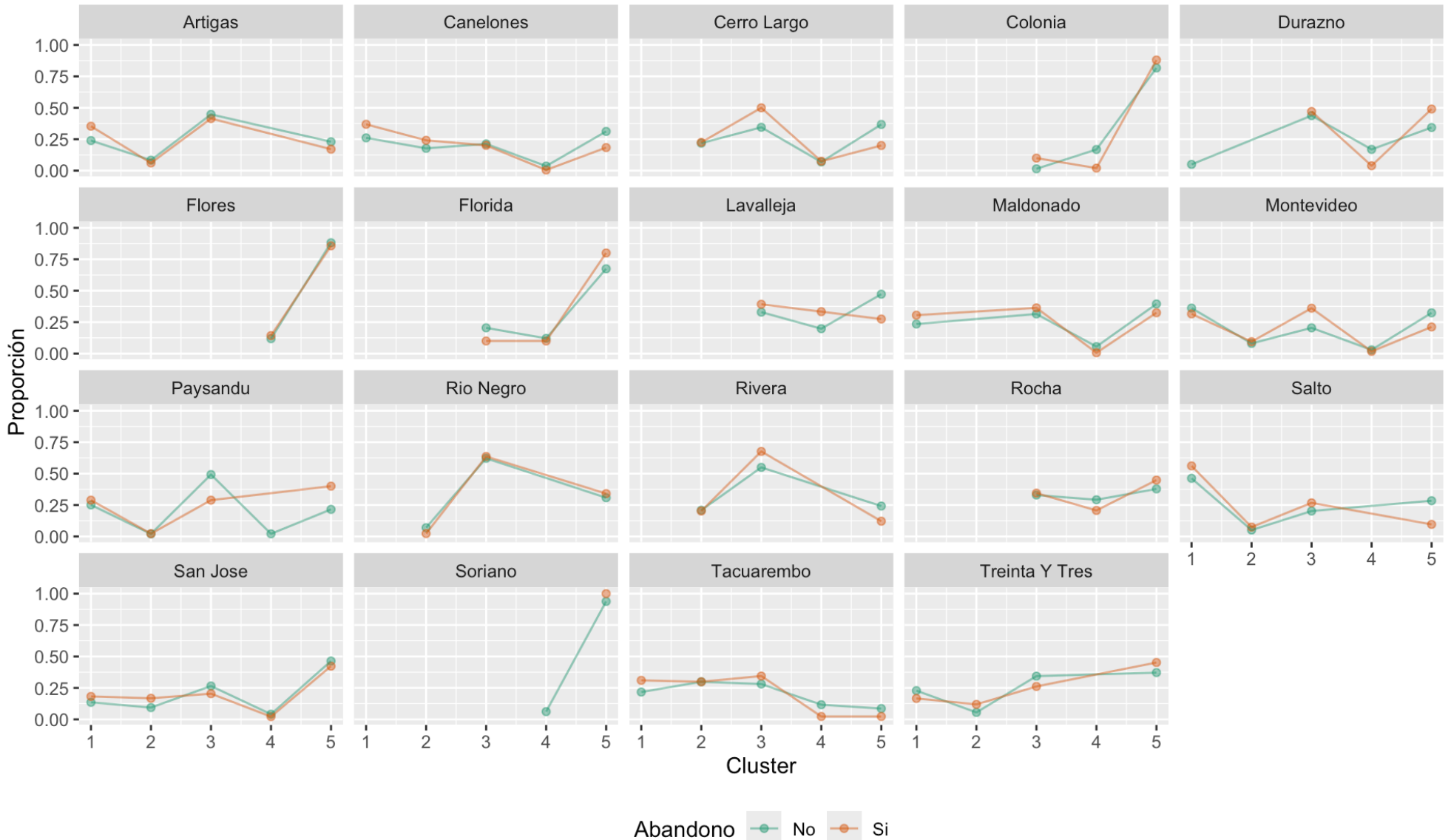
Abandono 1ero 2016-2017

- La distribución de los alumnos que no abandonan es similar para mujeres y hombres (50.45% mujeres y 49.55 % hombres).
- Para los que abandonan, hay un porcentaje mayor de alumnos de sexo masculino (57 %) que femeninos (43 %).
- ¿Existen diferencias en la distribución del abandono según género a nivel departamental?

Abandono por departamento y género



Contextos socioculturales y abandono



Modelos predictivos

Modelos de clasificación para el abandono

- Variable de respuesta categórica con dos niveles (abandona, no abandona).
- Variables explicativas: Contexto sociocultural, sexo, extra edad fuerte, extra edad leve, inasistencias relativas, centro educativo, departamento.
- Problema: los datos están muy desbalanceados sólo un 7% abandonan
- Distintas estrategias, comenzaremos con árbol de clasificación y más adelante con bosques aleatorios (RF)

```r
# Separo la muestra en entrenamiento y testeo
inTrain   <- sample(1:nrow(BC), nrow(BC)*.7)
train.set <- BC[inTrain,]
test.set  <- BC[-inTrain,]

# Ajusto un árbol de clasificación sin podar

tree_ab <- rpart(
  Abandono~ nro_doc_centro_educ + inas_rel + cl + Sexo + E
  data = train.set)
```

```r
pred <- predict( tree_ab, test.set, "class")

t1 <- table( pred, test.set$Abandono)
t1
```

```
pred    0    1
   0 7968  502
   1    0    0
```

```r
#Error classificación global

1 - sum(diag(t1))/nrow(test.set)
```

```
[1] 0.059268
```

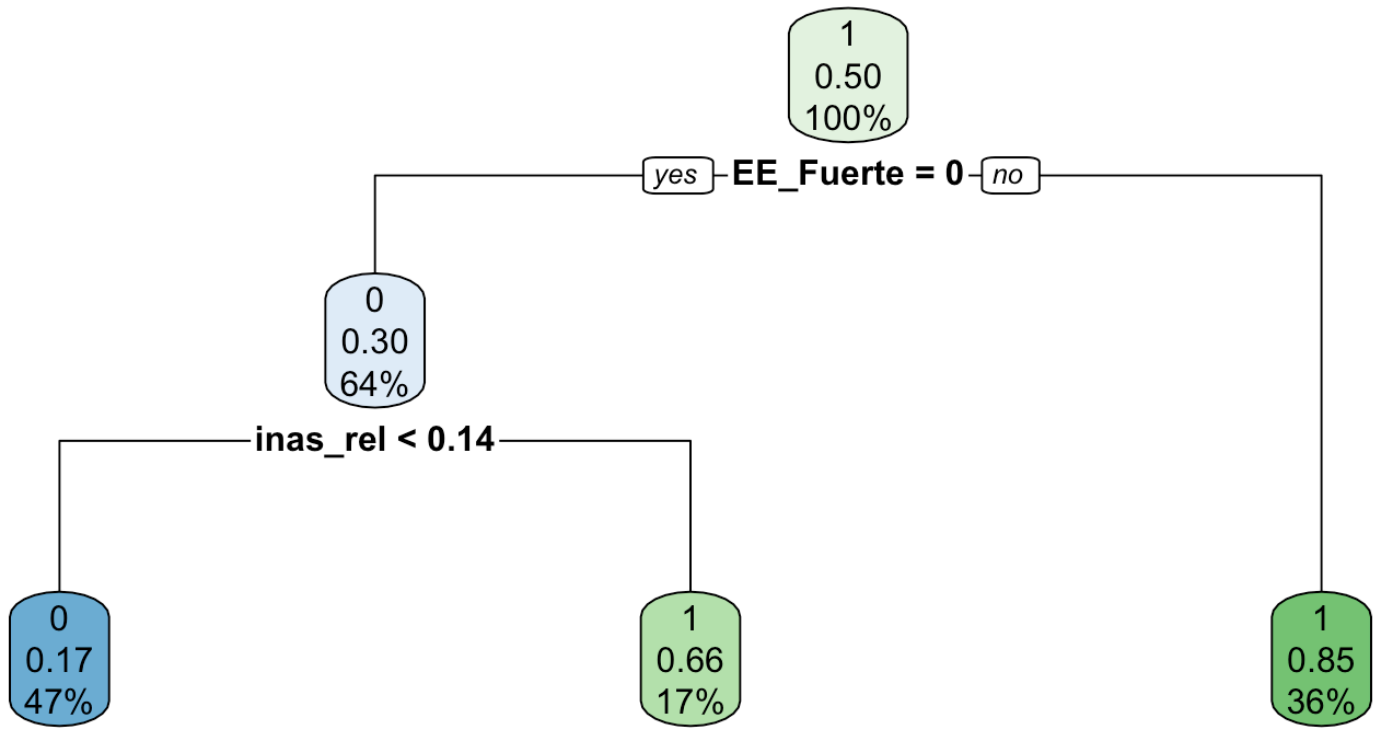
```r
#Sensibilidad TP/(TP+FN) (Error de clasificación para aban
t1[1,2]/sum(t1[,2])
```

```
[1] 1
```

```
1  rpart.plot(tree_ab)
```

```r
# Uso ponderadores para tratar el desbalance
t2 <- table(train.set$Abandono)

train.set$peso <- with(train.set, ifelse(Abandono == 1,
                                1/t2[2],
                                1/t2[1]))
tree_ab_w <- rpart(
  Abandono~ nro_doc_centro_educ + inas_rel + cl + Sexo +
  data = train.set, weight = peso )
```


1
0.50
100%
yes
EE_Fuerte = 0
no
0
0.30
64%
inas_rel < 0.14
0
0.17
47%
1
0.66
17%
1
0.85
36%

```r
pred_w <- predict( tree_ab_w, test.set, "class")
t2 <- table(pred_w, test.set$Abandono )
t2
```

```
pred_w    0    1
     0 6157   96
     1 1811  406
```

```r
#Error de clasificación
1 - sum(diag(t2))/nrow(test.set)
```

```
[1] 0.2251476
```

```r
# Sensibilidad (Error de clasificación para abandono)
t2[1,2]/sum(t2[,2])
```

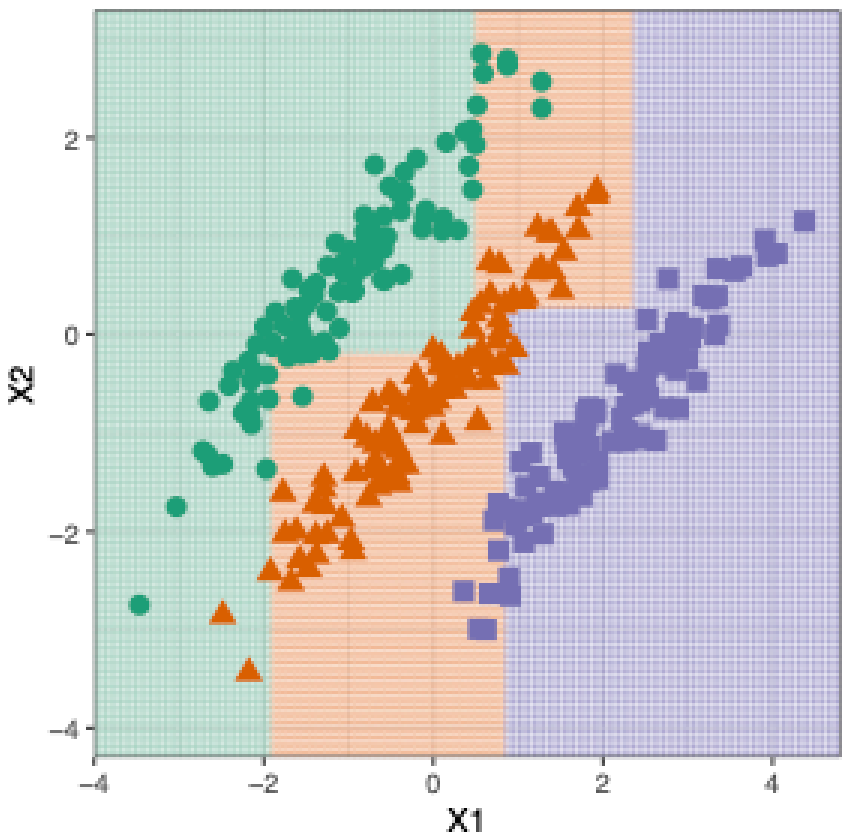
```
[1] 0.1912351
```

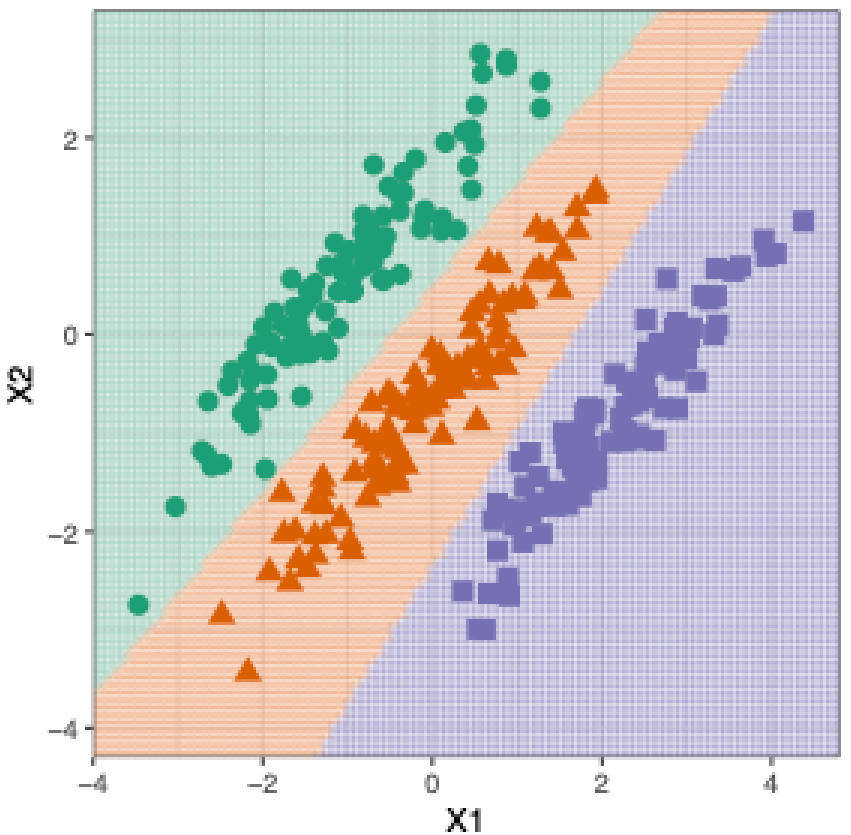
Variantes de CART

- CHAID, C4.5, FACT, QUEST, CRUISE, GUIDE, CTREE, PPtree, etc.
- Principales diferencias entre los distintos métodos son la forma en que se parte el nodo.
- Algunos usan métodos de kernel, vecino más cercano, particiones lineales en un subconjunto de variables seleccionadas.
- Pueden ser árboles binarios o con particiones múltiples.
- Los árboles que usan una sola variable en la particion del nodo generan particiones paralelas a los ejes, cundo se usan más variables las particiones tienden a ser oblicuas.

Projection pursuit Trees

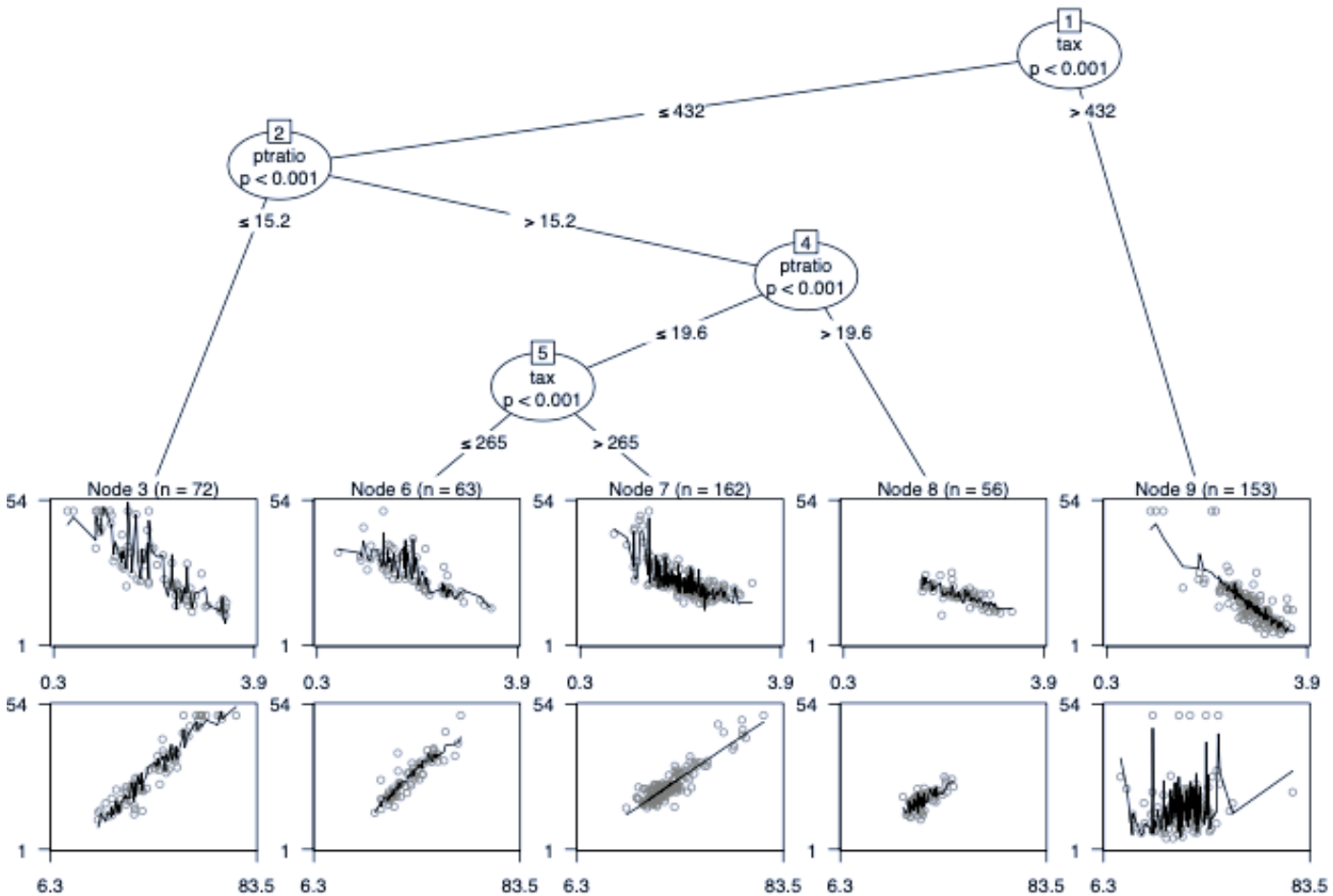
(Lee et al. 2013)





Model-based Recursive Partitioning

(Zeileis, Hothorn, and Hornik 2008)



Referencias

Breiman, Leo, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and Regression Trees*. CRC press.

Lee, Yoon Dong, Dianne Cook, Ji-won Park, and Eun-Kyung Lee. 2013. "PPtree: Projection Pursuit Classification Tree." *Electronic Journal of Statistics* 7: 1369–86.

Loh, Wei-Yin. 2014. "Fifty Years of Classification and Regression Trees." *International Statistical Review* 82 (3): 329–48.

Morgan, James N, and John A Sonquist. 1963. "Problems in the Analysis of Survey Data, and a Proposal." *Journal of the American Statistical Association* 58 (302): 415–34.

Zeileis, Achim, Torsten Hothorn, and Kurt Hornik. 2008. "Model-Based Recursive Partitioning." *Journal of Computational and Graphical Statistics* 17 (2): 492–514.