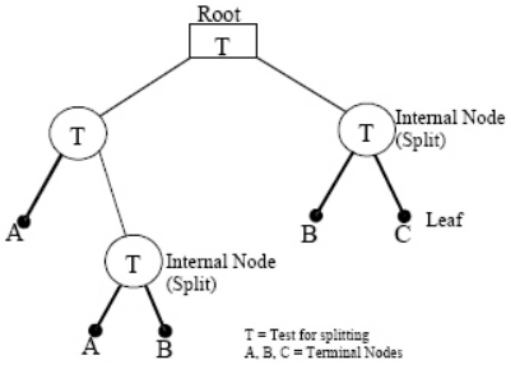
Árboles de Regresión y Clasificación

Motivación

Los árboles de regresión y clasificación (CART), son un método que permite modelar la relación de una variable dependiente y con un conjunto de variables explicativas X, introduciendo interacciones entre los predictores. Utilizando un algoritmo de particiones recursivas, se ajustan modelos simples (con una sola variable de decisión), gracias a la cual los CART consiguen generar reglas similares a las que se realizarían en el proceso previo a la toma de decisiones.

Descripción de la Técnica

Los Árboles de Regresión y Clasificación permiten dividir un nodo raíz en diferentes nodos hijos mediante particionamiento recursivo de los predictores. El objetivo de estas particiones es crear nodos que sean homogéneos (minimizando la impureza), de modo que se estén agrupando observaciones similares para entender el comportamiento promedio del grupo.



A continuación se describe el algoritmo:

1. Comience en el nodo raíz.
2. Para cada predictor X, encuentre el conjunto S que minimice la suma de las impurezas de los dos nodos hijos resultantes para cada partición.
3. Escoja el X^* y S^* tal que se minimice la impureza sobre todo el conjunto de X y S. Es decir, escoja la partición con menor impureza.
4. Repita desde 2 hasta alcanzar un criterio para detenerse (e.g. longitud del árbol).

La operación del algoritmo es similar para los problemas de regresión y clasificación. Su única diferencia es en el criterio de impureza seleccionado para realizar las particiones.

En el caso de un problema de regresión, la impureza de un nodo s, con $s =$

$\{izquierda,\ derecha\}$, se mide como la suma de residuales al cuadrado de todos los valores de y que pertenecen al nodo s, comparado con el valor promedio de y:

$$\bar{y}_{js} = \sum_{i \in s} y_{ijs}$$

$$\phi_{js} = \sum_{i \in s} (y_{ijs} - \bar{y}_{js})^2$$

Así, la impureza de la partición j sería la suma de la impureza de ambos nodos de la partición:

$$\phi_j = \phi_{j,izq} + \phi_{j,der}$$

Cuando se trata de un problema de clasificación binario, la impureza se mide utilizando el Índice de Gini. De modo que si p es la proporción de observaciones de clase positiva ($y_i = 1$), entonces:

$$Gini = p(1-p)$$

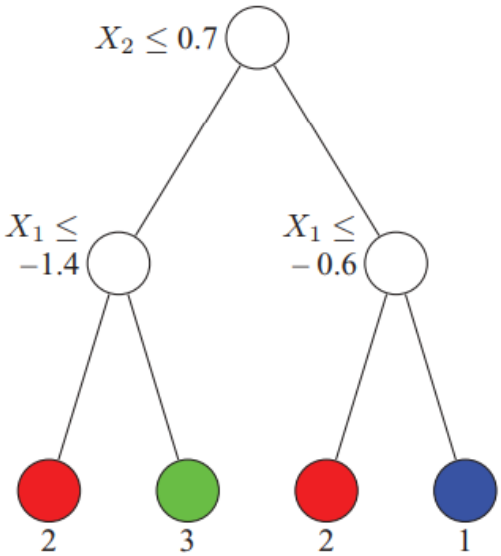
Así, la impureza del nodo s en la partición j estaría dada por:

$$\phi_{js} = p_{js}(1 - p_{js})$$

Y la impureza de la partición:

$$\phi_j = \phi_{j,izq} + \phi_{j,der}$$

Si $x \in X$ es una variable continua, los grupos $s = \{izquierda,\ derecha\}$ estarán caracterizados por encontrarse a la izquierda o derecha de un valor. De esta forma se crean reglas de la forma *"Si x es menor a τ, entonces el nodo pertenece al grupo g"*.



Ejemplo 1 - Titanic

El Titanic naufragó el 15 de abril de 1912 tras chocar con un iceberg. Este evento le arrebató la vida a 1502 de 2224 personas (entre pasajeros y tripulación). Según la historia, el Titanic no contaba con suficientes botes salvavidas para emergencias.

Naturalmente surge la pregunta... ¿por qué se salvaron esos 722 pasajeros? ¿Tenían algo en especial? ¿Es cierto la frase de las películas: *Mujeres y niños primero*? Mediante un árbol de clasificación veremos qué factores influyeron en la supervivencia de los pasajeros del Titanic.

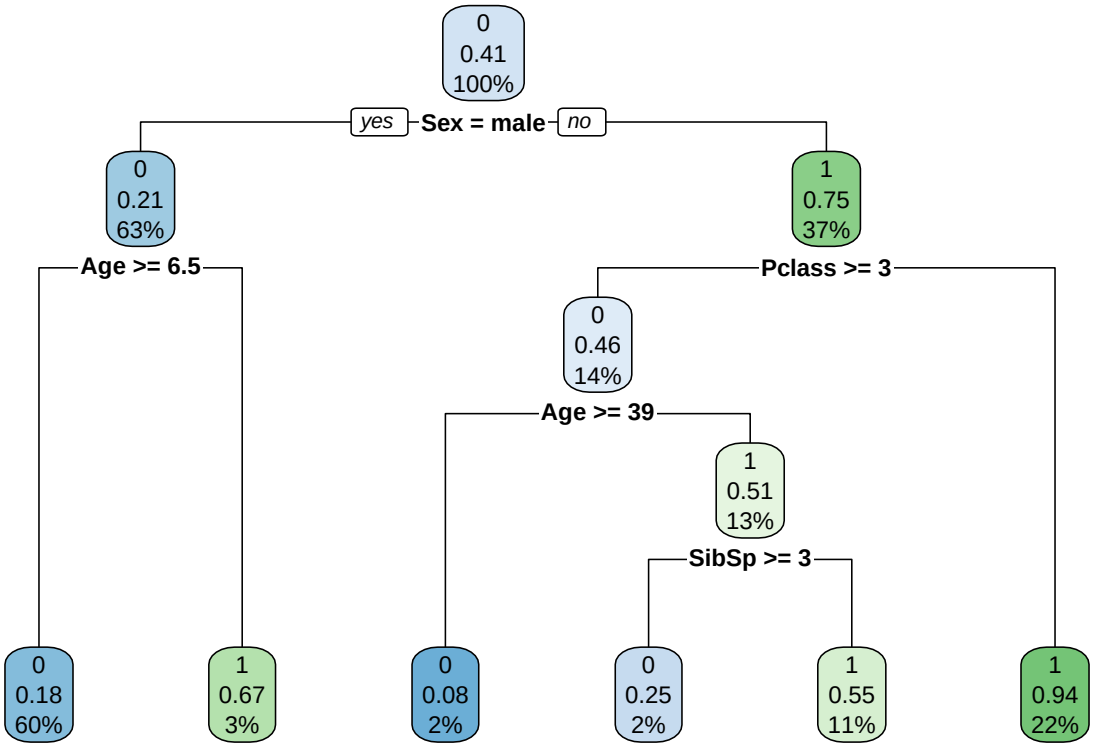
Para ello, utilizaremos la base de datos "Titanic", que trae una muestra de las personas abordo de la nave y de sus características. En este caso, se trabajará con las variables de edad, clase, sexo, el costo del ticket y número de familiares abordo, para predecir la supervivencia de los individuos.

```
titanic = read.csv("titanic.txt")

# Selección de variables de interés
titanic = titanic[, c("Survived", "Pclass", "Sex", "Age", "SibSp", "Fare")]

# Eliminar valores nulos
titanic = titanic[complete.cases(titanic),]
titanic$Pclass = as.numeric(titanic$Pclass)
```

De esta forma, queremos modelar la supervivencia mediante un conjunto de interacciones entre las variables explicativas:

$$Survived_i = f(Pclass_i,\ Sex_i,\ Age_i,\ SibSp_i,\ Fare_i)$$

```
library("rpart")
library("rpart.plot")
tree = rpart(Survived~Pclass+Sex+Age+SibSp,
             data = titanic, method = "class", control = list(minsplit=90))
rpart.plot(tree)
```



Con el árbol de clasificación se puede ver que en general, las mujeres tienen mayor tasa de supervivencia que los hombres (primer nodo). Para las mujeres (derecha), si su clase es 1 o 2, la probabilidad de supervivencia es de 0.94, de lo contrario es poco probable que sobrevivan. Adicionalmente, se puede observar que sólo los varones menores de 6 años y medio se salvaban. De esta forma, no es descabellado afirmar que en el Naufragio del titanic se siguó la regla de *"Mujeres y niños primero"*.

Para evaluar la eficacia del modelo se utiliza la matriz de confusión (de igual manera que se vió en la Regresión Logística).

```
# Se crea la predicción del modelo

matriz_confusion = function(data, model, dependent){
  Predicted = predict(model, type="class")
  Real = data[, dependent]
  conf = table(Real, Predicted)
  accuracy = mean(ifelse(Predicted == Real, 1, 0))
  return(list("ConfusionMatrix"= conf, "Accuracy" = accuracy))
}

matriz_confusion(titanic, tree, "Survived")
```

```
## $ConfusionMatrix
```

```
##     Predicted
## Real   0   1
##    0 372  52
##    1  81 209
##
## $Accuracy
## [1] 0.8137255
```

Se observa que el árbol de clasificación determinó correctamente la supervivencia de los individuos en 81.3% de los casos. Asimismo, se observa que la proporción de individuos que el modelo erróneamente predijo que sobrevivían es bastante pequeña.

Ejemplo 2 - Consumo

Anteriormente se trabajó con datos de gasto de los hogares en Colombia. Para efectos de comparación entre un modelo lineal y un modelo no lineal, se utilizará la misma base de datos con la misma especificación. Así, utilizando la Encuesta Multipropósito para Bogotá de 2011 se ajustará el siguiente modelo:

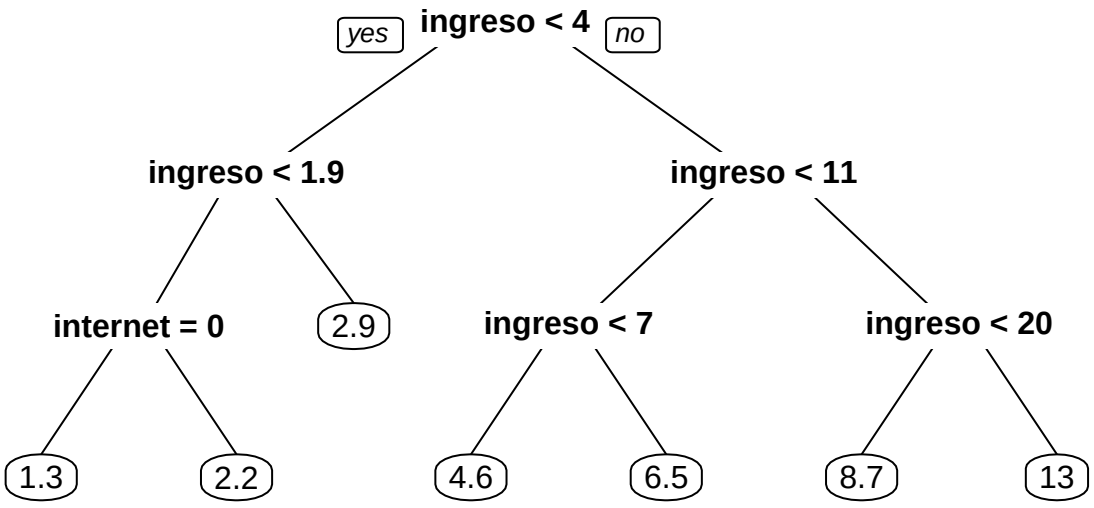
$$gasto_i = \beta_1 + \beta_2 ingreso_i + \beta_3 internet_i + \beta_4 hacin_i + \beta_{5k} estrato_{ki} + \epsilon_i$$

Donde, $gasto_i$ e $ingreso_i$ son los valores en pesos del gasto y del ingreso del i-ésimo hogar,respectivamente. $internet_i$ es una variable indicadora que toma el valor de 1 si el hogar tiene conexión a internet, y 0 si no. $hacin_i$ es el índice de hacinamiento para el hogar. $estrato_{ki}$ es una variabe dummy que toma el valor de 1 si el hogar pertenece a estrato k y 0 si no, $k \in \{2, 3, 4, 5, 6\}$. ϵ_i es el término de perturbación.

```r
# Creación de Conjuntos de Entrenamiento y Validación
set.seed(1234)
indices_entrenamiento <- sample(1:nrow(datos), 0.7 * nrow(datos))
indices_prueba <- setdiff(1:nrow(datos), indices_entrenamiento)

datos_entrenamiento <- datos[indices_entrenamiento,]
datos_prueba <- datos[indices_prueba,]
```

Ya con los datos listos se procede al ajuste del *Árbol de Regresión*, utilizando nuevamente la librería `rpart`. La diferencia con la implementación en clasificación es el parámetro `method`, el cual tendrá como valor `anova`.

```r
reg_tree = rpart(gasto ~ ingreso + internet + hacinamiento + estrato,
            data = datos_entrenamiento, method = "anova")

prp(reg_tree)
```



Se puede observar que según el árbol de regresión, la variable que ayuda a determinar el gasto de los hogares es $ingreso_i$, y que sólo cuando los hogares tienen un ingreso inferior a 1.9 millones de pesos, se revisa el gasto en internet (un servicio que hoy en día se da por sentado).

Con esta información se procede a revisar el RMSE para el modelo sobre los datos de entrenamiento como los de prueba.

```
rmse = function(model, data, y){

  pred = predict(model, type=, newdata=data)
  res = pred - data[, y]
  res_squared = res^2
  mean = mean(res_squared)
  rmse = sqrt(mean)
  print(paste("El RMSE del modelo es: ", round(rmse, 2)))
  return(rmse)
}

# Error Cuadrático Medio en Datos de Entrenamiento

rmse_train = rmse(reg_tree, datos_entrenamiento, "gasto")
```

```
## [1] "El RMSE del modelo es:  1.53"
```

```
# Error Cuadrático Medio en Datos de Prueba
rmse_test = rmse(reg_tree, datos_prueba, "gasto")
```

```
## [1] "El RMSE del modelo es:  1.5"
```

En el caso de la regresión lineal, el RMSE en entrenamiento y pruebas fue, respectivamente, 1.45 y 1.40. De esta forma, se puede evidenciar que el ingreso no es la única variable que aporta información para predecir el gasto de los hogares, y que una regresión lineal tiene mejor ajuste.

Ejemplo 3 - Encuesta Nacional de Lectura

Leer es una de las actividades más importantes que pueden realizar las personas puesto que las ayuda no solo a mejorar la ortografía y el dominio del lenguaje, sino que aporta a la creatividad, a la tranquilidad y al libre pensamiento. Es por esto que el Ministerio de Educación decidió adelantar un estudio para determinar cómo se encuentran los hábitos de lectura de los niños de 0 a 4 años en Colombia para así poder implementar políticas públicas que incremente la población lectora.

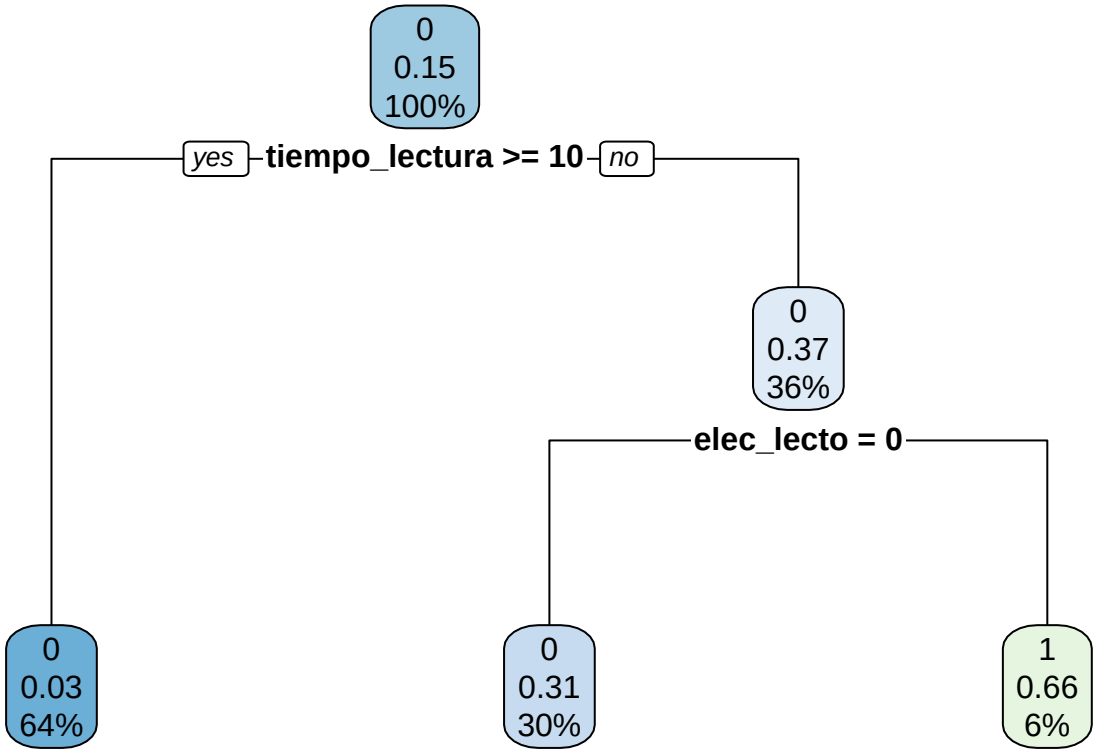
Utilizando la Encuesta Nacional de Lectura (ENLEC) de 2017, ud debe realizar un modelo que le permita responder alguna pregunta que usted considere relevante respecto a los hábitos de lectura.

```
ENLEC = read.csv("ENLEC_CERO_A_CUATRO.csv", sep=";")

# Horas Promedio de Lectura a la semana
ENLEC$tiempo_lectura = ENLEC$P1876S2A1

# ¿Le gusta leer? 1 = Sí, 0 = No
ENLEC$gusta = 2 - ENLEC$P1704S1

# ¿Visitó Bibliotecas? 1 = Sí, 0 = No
ENLEC$biblio = 2 - ENLEC$P5436

# ¿Acceso a computador? 1 = Sí, 0 = No
ENLEC$comp = ifelse(ENLEC$P1872S1A1 == 1 | ENLEC$P1872S1A2 == 1, 1, 0)

# ¿Acceso a Celular o Tablet?  1 = Sí, 0 = No
ENLEC$tablet = ifelse(ENLEC$P1872S1A3 == 1 | ENLEC$P1872S1A4 == 1, 1, 0)

# ¿Utiliza dispositivos electrónicos para leer? 1 = Sí, 0 = No
ENLEC$elec_lecto = 2 - ENLEC$P1873S2A3

# En promedio, ¿cuántas horas juega videojugos?
```

```
ENLEC$games = ENLEC$P1876S6A1

# En promedio, ¿cuántas horas ve películas?
ENLEC$pelis = ENLEC$P1876S7A1

# En promedio, ¿cuántas horas juega?
ENLEC$juego = ENLEC$P1876S3A1

ENLEC = ENLEC[, c("juego", "pelis", "elec_lecto", "tablet", "comp",
                  "biblio", "gusta", "tiempo_lectura")]

ENLEC = ENLEC[complete.cases(ENLEC), ]
```

A continuación se plantea un modelo de clasificación que busca determinar qué factores influyen en que a un niño de 0 a 4 años le guste la lectura.

```
library("rpart")
library("rpart.plot")
indices_entrenamiento <- sample(1:nrow(ENLEC), 0.7 * nrow(ENLEC))
indices_prueba <- setdiff(1:nrow(ENLEC), indices_entrenamiento)

datos_entrenamiento <- ENLEC[indices_entrenamiento,]
datos_prueba <- ENLEC[indices_prueba,]

tree = rpart(gusta~.,
             data = datos_entrenamiento, method = "class")
rpart.plot(tree)
```



El árbol de clasificación da un resultado curioso que a primera vista parece poco intuitivo: *Los niños entre 0 y 4 años cuya dedicación a la lectura semanalmente es, en promedio, de 10 horas o más, ¡no les gusta leer!.* Esto puede ser una señal de que están siendo obligados a leer, y le tienen desprecio. Asimismo, el árbol indica que los niños que dedican menos de 10 horas de lectura a la semana utilizan dispositivos electrónicos. Si algunos de los usos consisten en desarrollar lectura, es porque les gusta leer.

```
matriz_confusion = function(data, model, dependent){
  Predicted = predict(model, type=, newdata=data)[, "1"] > 0.5
  Real = as.matrix(data[, dependent])
  conf = table(Real, Predicted)
  accuracy = mean(ifelse(Predicted == Real, 1, 0))
  return(list("ConfusionMatrix"= conf, "Accuracy" = accuracy))
}

matriz_confusion(datos_prueba, tree, "gusta")
```

```
## $ConfusionMatrix
##     Predicted
## Real FALSE TRUE
##    0  1101   30
##    1   153   60
```

```
##
## $Accuracy
## [1] 0.8638393
```