Aprendizaje no supervisado: Árboles de desición

Daniela Arbeláez Montoya Jefferson Gamboa Betancur

Trabajo Arboles de Decisión - SVMs, Aprendizaje no supervizado No 1 S-02-20.

***Fecha de entrega*: Jueves 26 de Noviembre**

1. En este ejercicio se utilizarán arboles de regresión para predecir los valores de la variable `sales` en la base de datos `Carseats` de la libreria `ISLR`, tratando dicha variable como continua:

   a) Divida el conjunto de observaciones en un conjunto de entrenamiento y un conjunto de prueba. De forma aleatoria. En que proporciones dividió los datos?.

   b) Ajuste un árbol de regressión en el conjunto de entrenamiento. Gráfique el árbol e interprete los resultados. Que valor del MSE **de prueba** obtiene?.

   c) Utilice validación cruzada para determinar el grado óptimo de complijidad del árbol. Consigue la poda del árbol mejorar el EMS **de prueba**?

   d) Utilice el método **bagging** para analizar estos datos. Que valor del MSE obtiene?

   - Use la función `importance()` para determinar cúal de las variables es la más importante.

   e) Ahora utilice un **bosque aleatorio** (*Random-Forest*) para analizar estos datos.

   - Que valor del MSE de **prueba** obtiene?.
   - Use la función `importance()` para determinar cuales variables son las más importantes.
   - Describa el efecto de m el número de variables consideradas en cada subdivisión, en la tasa de error obtenida.

2. **(MSVs - aplicado)**En este ejercicio, se utilizará el enfoque de **máquinas de soporte vectorial** para predecir si un automóvil determinado posee un alto o bajo consumo de combustible basado en el conjunto de datos `Auto` (librería `ILSR`).

   a) Cree una variable binaria que tome un 1 para automóviles con millaje por galón por encima de la mediana, y un 0 para automóviles con millaje por debajo de la mediana.

b) Ajuste un **clasificador de soporte vectorial** a los datos con varios valores del parámetro `cost` para predecir si un automóvil posee millaje alto o bajo. Informe los errores de validación cruzada asociados con diferentes valores de este parámetro. **Comente sobre sus resultados**.

c) Ahora repita b), esta vez utilizando una **máquina de soporte vectorial (svm)** con una base de kernels radiales y polinomiales, con diferentes valores de los hiperparámetros `cost`, `gamma` o `degree` según el kernel y . **comente sus resultados**.

d) Realice algunos plots que sirvan de apoyo a sus afirmaciones en (b) y (c).

Recomendación :*En el lab, se utilizó la función* `plot()` *para objetos* `svm` *solo en casos con* $p = 2$. *Cuando* $p > 2$, *se puede utilizar la función* `plot()` *para crear gráficos que muestran pares de variables a la vez. Esencialmente, en lugar de escribir*

$>$ `plot(svmfit,dat)`

donde `svmfit` contiene el modelo ajustado y `dat` es un data frame que contiene los datos, se puede utilizar

$>$ `plot(svmfit,dat,x1~x4)`

para graficar solo las variables primera y cuarta. Sin embargo, se debe reemplazar `x1` *y* `x4` *con los nombres correctos de las variables. Se puede encontrar más información, escribiendo* `?plot.svm`.

3. Este ejercicio utiliza el conjunto de datos `OJ` el cual es parte de la librería `ISLR`

a) Cree un conjunto de entrenamiento con una muestra aletoria de 800 observaciones y un conjunto de prueba que conste del resto de observaciones.

b) Ajuste un **clasificador de soporte vectorial** utilizando `cost = 0.1`, con `Purchase` como la variable respuesta y las demás como predictores.

- Utlice la función `summary()` para obtener un resumen de estadísticas y describa los resultados obtenidos.

c) Que tasas de error de entrenamiento y de prueba obtiene?.

d) Utilice la función `tune()` para obtener un valor óptimo del parámetro `cost`. Considere valores en el rango de 0.01 a 10.

e) Calcule nuevamente las tasas de error de entrenamiento y de prueba usando el valor óptimo obtenido de `cost`.

f) Repita items de (b) hasta (e) ajustando esta vez una **máquina de soporte vectorial** (**svm**) con un nucleo **radial**. Utilizando el valor de default para γ.

g) Repita items (b) hasta (e) utilizando nuevamente una máquina de soporte vectorial pero esta vez con un nucleo **polinomial**, usando `degree = 2`.

h) En general cúal método parece proporcionar los mejores resultados en estos datos?.

4. **[PCA, K-medias]** En este ejercicio Ud va a generar un conjunto simulado de datos y entonces aplicará PCA y agrupamiento por k-medias sobre dichos datos.

   a) Genere un conjunto de datos simulados con 20 observaciones en cada una de tres clases (es decir, 60 observaciones en total) y 50 variables.

   Sugerencia: *hay una serie de funciones en R que puede utilizar para generar datos. Un ejemplo es la función* ***`rnorm()`***; ***`runif()`*** *es otra opción. Asegúrese de agregar un* ***cambio en la media*** *en las observaciones de cada clase a fin de obtener tres clases distintas.*

   b) Realice PCA en las 60 observaciones y grafique las observaciones en términos de las 2 primeras variables principales Z_1 y Z_2. Use un color diferente para indicar las observaciones en cada una de las tres clases. Si las tres clases aparecen separados en esta gráfica, **solo** entonces continúe con la parte (c). Si no, vuelva al inciso a) y modifique la simulación para que haya una mayor separación entre las tres clases. No continúe con la parte (c) hasta que las tres clases muestren al menos algún grado de separación en los dos primeros vectores de scores de componentes principales.

   c) Desarrolle agrupación de K-medias de las observaciones con $K = 3$. ¿Qué tan bien funcionan los clústeres que obtuvo con el algoritmo de K-medias comparado con las verdaderas etiquetas de clase?

   Sugerencia: *puede usar la función* ***`table()`*** *en R para comparar las verdaderas etiquetas de clase con las etiquetas de clase obtenidas por agrupamiento. Tener cuidado cómo se interpretan los resultados: el agrupamiento de K-medias numera los grupos arbitrariamente, por lo que no puede simplemente comprobar si las verdaderas etiquetas de clase y las etiquetas de agrupación son las mismas.*

   d) Realice agrupamiento de K-medias con $K = 2$. Describa sus resultados.

   e) Ahora realice agrupamiento de K-medias con K = 4 y describa su resultados.

   f) Ahora realice agrupamiento de K-medias con K = 3 en los dos primeros vectores de scores de componentes principales, en lugar de los datos en las variables originales. Es decir, realice la agrupación de K-medias en la matriz de 60×2, cuya primera columna es la coordenada z_{i1} en la primera componente principal Z_1 y la segunda columna es la coordenada z_{i2} en la segunda componente principal Z_2. Comente los resultados.

   g) Con la función `scale()`, realice agrupamiento de K-medias con $K = 3$ en los datos después de escalar cada variable para tener una desviación estándar de uno. ¿Cómo se comparan estos resultados con los obtenidos? en (b)? **Explique**.

5. Considere el conjunto de datos `USArrests`. En este ejercicio se agruparán los estados en `USArrests` con agrupamiento jerarquico

   a) Utilice agrupación jerárquica con enlace completo y distancia euclidiana, para agrupar los estados.

   b) Corte el **dendrograma** a una altura que dé como resultado **tres** clusters. ¿Qué estados pertenecen a qué cluster?

   c) Agrupe jerárquicamente los estados utilizando un enlace completo y distancia euclidiana, después de escalar las variables para tener una desviación estándar uno.

   d) ¿Qué efecto tiene el escalado de las variables en la estructura jerárquica del agrupamiento obtenido? En su opinión, ¿deberían las variables ser escaladas antes de que se calculen las disimilitudes entre observaciones? Proporcione una justificación para su respuesta.

Solución

```
require(ISLR)
datos <- Carseats
```

Descripción de Carseats.

Un conjunto de datos simulados que contiene las ventas de asientos de seguridad para niños en 400 tiendas diferentes.

Formato Un marco de datos con 400 observaciones sobre las siguientes 11 variables.

Descripción de las variables

Sales Ventas unitarias (en miles) en cada ubicación

CompPrice Precio cobrado por la competencia en cada ubicación

Income Nivel de ingresos de la comunidad (en miles de dólares)

Advertising Presupuesto de publicidad local para la empresa en cada ubicación (en miles de dólares)

Population Tamaño de la población en la región (en miles)

Price Precio que cobra la empresa por los asientos de seguridad en cada sitio

ShelveLoc Un factor con niveles Malo, Bueno y Medio que indica la calidad de la ubicación de las estanterías para los asientos del automóvil en cada sitio.

Age Edad media de la población local

Education Nivel de educación en cada ubicación

Urban Un factor con niveles No y Sí para indicar si la tienda está en una ubicación urbana o rural.

US Un factor con niveles No y Sí para indicar si la tienda está en EE. UU. O no

1.a)

Los datos son particionados aleatoriamente en el 70% para entrenamiento (train) y el 30% de prueba (test)

```r
set.seed(123)
muestra <- sample(1:nrow(datos), size = floor(nrow(datos) * 0.7))
train <- datos[muestra, ]; test <- datos[-muestra, ]
```

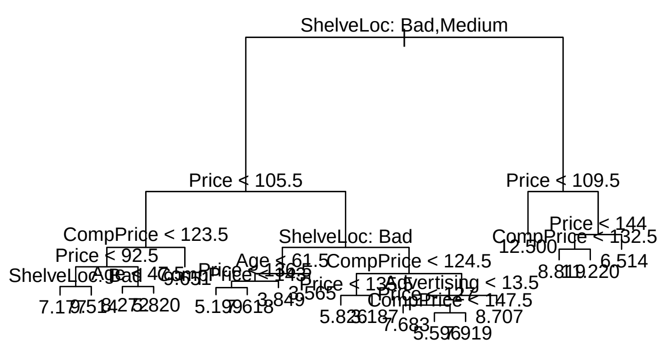
1.b)

```r
require(tree)
require(MASS)

Reg.tree <- tree(Sales ~ ., data = datos, subset = muestra)
summary(Reg.tree)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = datos, subset = muestra)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "CompPrice"   "Age"         "Advertising"
## Number of terminal nodes:  19
## Residual mean deviance:  2.373 = 619.2 / 261
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4.1570 -1.0160  0.1123  0.0000  0.8903  4.0310
```

```r
plot(Reg.tree)
text(Reg.tree, pretty = 0)
```



Observe que el árbol de regresión consideró que las variables más importantes fueron:

ShelveLoc: Un factor con niveles Malo, Bueno y Medio que indica la calidad de la ubicación de las estanterías para los asientos del automóvil en cada sitio.

Price: Precio que cobra la empresa por los asientos de seguridad en cada sitio.

CompPrice: Precio cobrado por la competencia en cada ubicación.

Age: Edad media de la población local.

Advertising: Presupuesto de publicidad local para la empresa en cada ubicación (en miles de dólares)

La calidad de la ubicación en las estanterías para los asientos es la covariable más importante para determinar la venta unitaria en cada ubicación, donde el la venta promedio del asiento es más baja cuando la calidad del asiento esta entre baja y media, mientras que la venta promedio para la calidad del asiento cuando es alta crece.

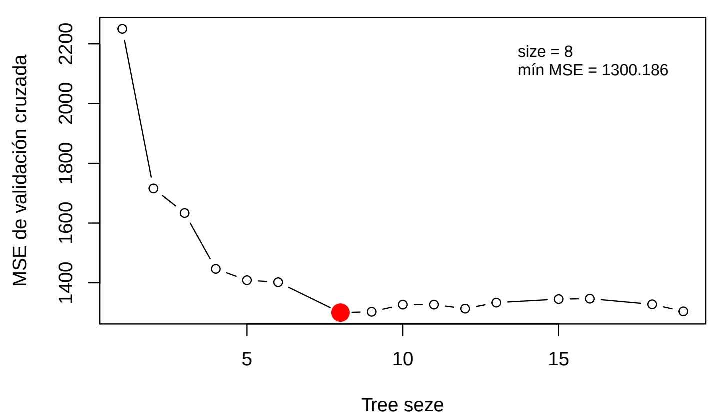
```
MSE1 <- mean((test$Sales - predict(Reg.tree, newdata = test))^2)
```

Y el MSE de prueba es de 3.602818

1.c)

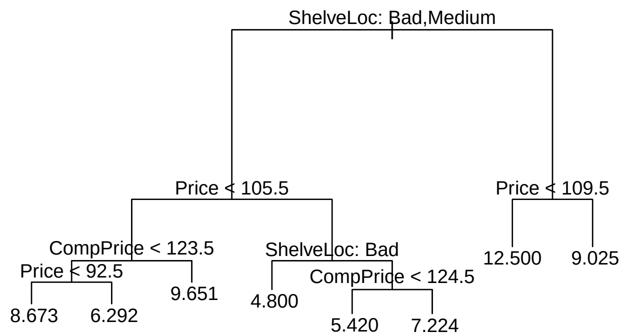
Ahora se utilizará cv.tree para ver si una poda mejora su desempeño

```
set.seed(123)
cv_Reg <- cv.tree(Reg.tree)
D.prueba <- data.frame(cv_Reg$size, cv_Reg$dev)
D.prueba <- D.prueba[which.min(D.prueba[,2]),]
plot(cv_Reg$size, cv_Reg$dev, type = "b", xlab = "Tree seze", ylab = "MSE de validación cruzada")
points(D.prueba[,1],D.prueba[,2], pch=19, cex=2, col="red")
legend("topright", inset = .05, legend=c("size = 8", "mín MSE = 1300.186"), cex=0.8, box.lty=0)
```



Luego se procede a realizar la poda con la funsión de R, prune.tree

```
prune.Reg <- prune.tree(Reg.tree, best = 8)
plot(prune.Reg)
text(prune.Reg, pretty = 0)
```




```r
MSE2 <- mean((test$Sales - predict(prune.Reg, newdata = test))^2);
```

MSE sin poda	MSE con poda
3.602818	4.353022

Como se logra detallar el MSE de prueba sin poda es mucho menor al MSE de prueba con poda, es decir que no podar el árbol con 8 nodos no ayuda a mejorar el modelo.

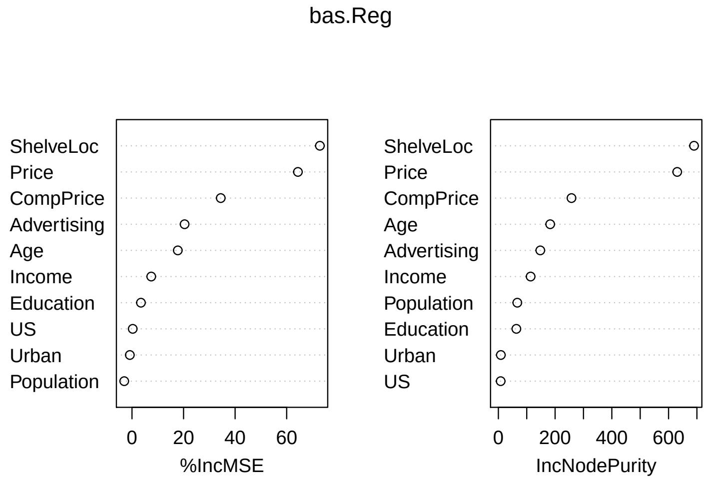
1.d)

```r
require(randomForest)
set.seed(123)

bas.Reg <- randomForest(Sales ~ . , data = datos, subset = muestra, mtry = 10, importance = TRUE)
importance(bas.Reg)
```

```
##               %IncMSE IncNodePurity
## CompPrice   34.4262904    257.760843
## Income       7.4645949    113.727916
## Advertising 20.4003141    147.788939
## Population  -3.0115418     66.776729
## Price       64.3523306    630.500603
## ShelveLoc   72.8822146    689.811838
## Age         17.7747876    182.686692
## Education    3.4693027     63.408518
## Urban       -0.8360222      8.755624
## US           0.2868891      8.107184
```

```r
varImpPlot(bas.Reg)
```




```r
MSE3 <- mean((test$Sales - predict(bas.Reg, newdata = test))^2)
```

Las covariables *ShelveLoc* y *Price* son las covariables con mayor importancia en la base de datos. El MSE de prueba que se obtuvo utilizando bagging fue de 2.2817104

1.e)

Para bosques aleatorios se utilizará por defecto en mtry = p/3 para generar el bosque.

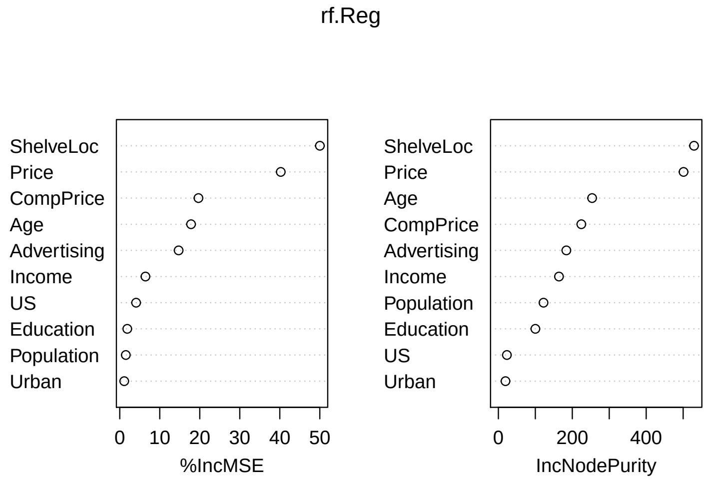
```r
set.seed(123)
rf.Reg <- randomForest(Sales ~ . , data = datos, subset = muestra, importance = TRUE)
MSE4 <- mean((test$Sales - predict(rf.Reg, newdata = test))^2)
```

El MSE de prueba para random forest es de 2.6254673

```r
importance(rf.Reg)
```

```
##               %IncMSE IncNodePurity
## CompPrice   19.669545     224.50086
## Income       6.424047     163.98134
## Advertising 14.712752     183.97789
## Population   1.513814     122.15101
## Price       40.239965     501.07261
## ShelveLoc   50.020107     529.44556
## Age         17.817908     253.67201
## Education    1.881294     100.14106
## Urban        1.118754      19.28070
## US           4.085561      23.08111
```

```r
varImpPlot(rf.Reg)
```



Para random forest se encuentra que las mismas dos covariables que se presentaron en bagging son las más importante.

Si se compara el MSE de prueba para el bagging y el de random forest es:

```
M <- data.frame(MSE3, MSE4)
names(M) <- c("MSE bagging", "MSE random forest")

kable(
  M,
  align = "c",
  booktabs = TRUE
) %>% kable_styling(position = "center")
```

MSE bagging	MSE random forest
2.28171	2.625467

Se observa que a pesar de utilizar con bagging un m = 10 y para random forest un m = 10/3 se logra evidenciar el MSE de prueba del bagging es mucho más pequeño que el de random forest, esto es devido a que random forest tiene una presición mucho más baja que el otro método utilizado.

2.a)

2.b)

2.c)

2.d)

3.a)

3.b)

3.c)

3.d)

3.e)

3.f)

3.g)

3.h)

4.a)

4.b)

4.c)

4.d)

4.e)

4.f)

4.g)

5.a)

5.b)

5.c)

5.d)