

# Aprendizaje no supervisado: Árboles de decisión

Daniela Arbeláez Montoya

Jefferson Gamboa Betancur

## Trabajo Árboles de Decisión - SVMs, Aprendizaje no supervisado No 1 S-02-20.

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

1. En este ejercicio se utilizarán árboles de regresión para predecir los valores de la variable **sales** en la base de datos **Carseats** de la librería **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 regresión 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 complejidad 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 cuál 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 **ISLR**).
  - 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 aleatoria 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.
    - Utilice 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  $\gamma$ .
  - 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 cuál 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()`; `rnorm()` 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 sus 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  $60 \times 2$ , 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 jerárquico
- 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

1.a)

1.b)

1.c)

1.d)

1.e)

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)