INSTITUTO TECNOLÓGICO DE BUENOS AIRES - ITBA
ESCUELA DE INNOVACIÓN

TRABAJO PRÁCTICO 1

AUTORES: Lambertucci, Guido Enrique (Leg. Nº 58009) DNI: 40397224

DOCENTES: Riccillo, Marcela Leticia

IA.03 - Inteligencia Artificial

BUENOS AIRES

Presentado: 28/02/24

Índice

1. Ejercicio 1

1.1. Consigna

- Primera Parte Ejercicio 1 – Regresión Los casos de Regresión se caracterizan por tener una variable cuantitativa para predecir.
  - Seleccione un dataset con un caso de Regresión. El dataset debe ser obtenido de alguna librería de R o de una página web pública (no incluir datos confidenciales).
  - Por ejemplo, se podría utilizar:
    $\Longrightarrow$ Datasets de R como: mtcars de base, iris de base, cheddar de faraway, etc.
    $\Longrightarrow$ datasets de UCI (Universidad de California) `https://archive.ics.uci.edu`
    $\Longrightarrow$ datasets de Kaggle `https://www.kaggle.com/`
    $\Longrightarrow$ datasets de ISLR `https://www.statlearning.com/resources-second-edition`
  - El dataset debe contener al menos 3 variables y una de ellas debe ser numérica. (Nota: este dataset es solamente para este ejercicio y no se espera ser utilizado en otros ejercicios).
    1. Indique el nombre del dataset, y la librería de R o la página web fuente del mismo.
    2. ¿De qué trata la base?
    3. ¿Cuántos registros tiene la base? ¿Cuántas variables? ¿De qué tipo son las variables? Podría utilizar `dim(base)`, `str(base)`, `summary(base)`
    4. Realice un histograma de la variable numérica seleccionada. ¿En qué rango se encuentran los valores? `hist(variable, main="Título", col="color")`
       a) Para el título ingrese su nombre, como “Histograma de Marcela”.
       b) Elija un color para el gráfico. Tenga en cuenta que ingresando `colors()` en R verá que hay más de 500 colores posibles.
       c) Indique el código R utilizado.

1.2. Resolución

La base de datos elegida es Summer Olympics Weightlifting records 2000 to 2020 de la página web Kaggle. Esta base registra los pesos máximos levantados por cada atleta olímpico en las dos movimientos correspondientes al levantamiento de pesas (Clean & Jerk y Snatch), ademas se provee el genero del atleta, su peso corporal, entre otras cosas. Cuenta con 716 registros, con 11 variables (1).

Nombre de la variable	Tipo	Comentario
x	Integer	Indice
Athlete	String	Nombre del Atleta
Bodyweight..kg	Double	Peso corporal en kilogramos
Clean...Jerk..kg	Double	Peso del Clean & Jerk
Snatch..kg	Double	Peso del Snatch
Total..kg	Double	Peso total
Ranking	Integer	Clasificaci\'on final del atleta
Url	String	Link a wikipedia
Title	String	Titulo obtenido
Year	Integer	Año de competición
Gender	String	Género

Tabla 1: Tabla de variables.

De estas variables, dependiendo que es lo que se quiere predecir, hay varias que no aportan información alguna. Por ejemplo si uno quisiera predecir el valor máximo de Snatch la variable x no aporta informacion, al igual que la Url. En contraposición el Clean & Jerk y el peso corporal son variables significativas para esta predicción. Se optó por la variable "Snatch" para graficar (1).

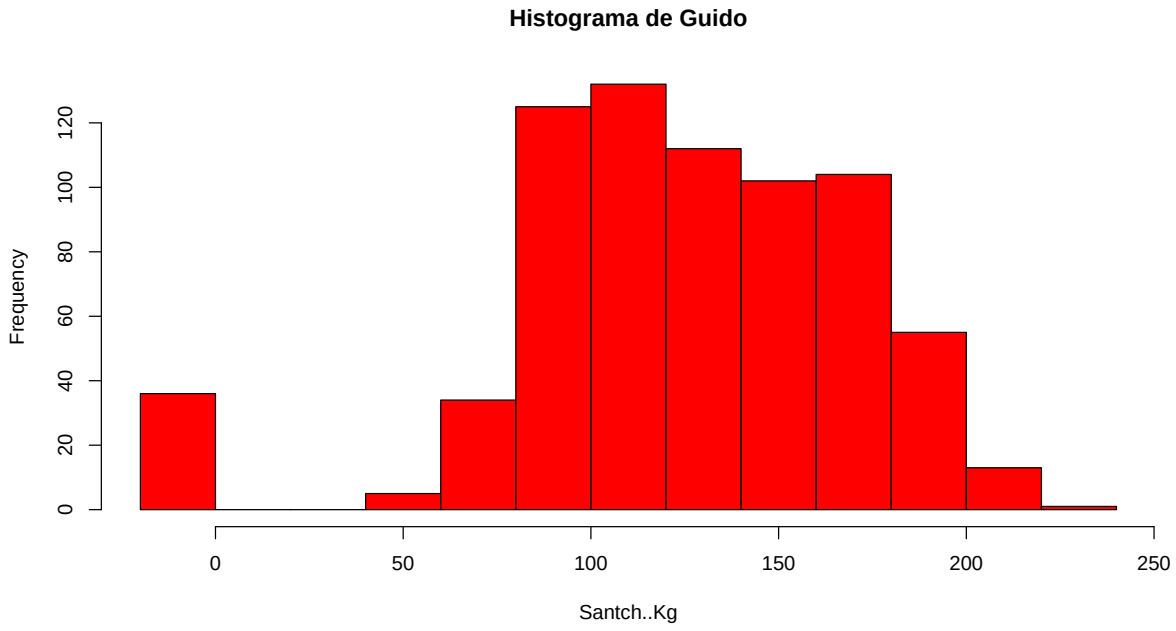


Figura 1: Histograma de la variable "Snatch..Kg".

Es interesante observar el histograma, ya que revela dos aspectos distintos. Por un lado, se puede apreciar una distribución centrada alrededor de los 125 Kg, la cual muestra una forma de campana similar a una distribución normal. Por otro lado, se destacan una serie de valores en 0 Kg. Estos valores atípicos, conocidos como "outliers", no son representativos del rendimiento del atleta ni contribuyen a su predicción. Más bien, reflejan casos de atletas que no compitieron en esa categoría y se les asignó un valor de 0.

1.3. Código utilizado

```
#install.packages("rstudioapi") #Install rstudioapi for the automatic set
    working directory feature
library(caret)

directory <- dirname(rstudioapi::getActiveDocumentContext()$path)
#just use the setwd if not using rstudio
setwd(directory)
base=read.table("./base.csv",sep=",",header=TRUE)#Loading dataset
#Some usefull information for getting to know my dataset
dim(base)
str(base)
summary(base)
View(base)

hist(base$Snatch..kg.,main = "Histograma de Guido",xlab = "Santch..Kg",
     ylab = "Frequency",col = "red")
```

2. Ejercicio 2

2.1. Consigna

Parte A – Análisis Exploratorio de Datos

1. Abra en R la base Glass de la librería mlbench.

   ```
   library(mlbench)
   data(Glass)
   ```

   Muestre `dim(Glass)`. ¿Cuántos ejemplos de vidrio tiene la base?

2. Escriba en R `?Glass` y copie aquí la Descripción (*Description*) que aparece en la página web.

3. Abra la página web de Glass en UCI (Universidad de California) https://archive.ics.uci.edu/ml/datasets/glass+identification. Busque la sección "Additional Variable Information" y luego *Class Labels*. Copie aquí los tipos de vidrio de la base. ¿Cuál tipo de vidrio no está representado en la base (none in this database)?

4. La variable a predecir es `Type` que representa los tipos de vidrios. Muestre un *summary* de la base `summary(Glass)`.

5. Indique cuántos vidrios hay de cada clase. `summary(Glass$Type)`

6. A grandes rasgos, los tipos de vidrio serían (hay 2 tipos de ventana de edificio):

   - Ventana de edificio
   - Ventana de automóvil
   - Recipiente
   - Vajilla
   - Faro de Auto

   Busque y muestre una imagen de cada tipo de vidrio. Indique para cada imagen la página web origen de la misma.

7. Renombre cada categoría y transforme la variable `Type` a factor. Luego renombre `Type` como `TipoDeVidrio`.

```
Glass$Type=as.character(Glass$Type)
Glass$Type[Glass$Type=="1"]="VentanaTipo1"
Glass$Type[Glass$Type=="2"]="VentanaEdificio"
Glass$Type[Glass$Type=="3"]="VentanaAuto"
Glass$Type[Glass$Type=="5"]="Recipiente"
Glass$Type[Glass$Type=="6"]="Vajilla"
Glass$Type[Glass$Type=="7"]="FaroAuto"
Glass$Type=factor(Glass$Type)
names(Glass)[names(Glass)=="Type"]="TipoDeVidrio"
```

Muestre un `summary(Glass)` para ver cómo quedó.

8. Realice un gráfico de barras de la variable `Glass$TipoDeVidrio`. Amplíe el gráfico para ver los nombres de los tipos de vidrio.

```
plot(Glass$TipoDeVidrio, main="Gráfico de barras de Título", col="COLOR")
```

   a) Para el título ingrese su nombre, como “Gráfico de barras de Marcela”.
   b) Elija un color para el gráfico. Tenga en cuenta que si ingresa `colors()` en R, verá que hay más de 500 colores posibles.
   c) Indique el código R utilizado.

Parte B – Conjuntos

2. Muestre un `head` y un `summary` del conjunto de entrenamiento y del conjunto de testeo.

```
head(entreno)
summary(entreno)
head(testeo)
summary(testeo)
```

3. ¿Cuántos registros quedaron en cada conjunto (entrenamiento y testeo) en total?

```
dim(Glass)
dim(entreno)
dim(testeo)
```

4. ¿Cuántos registros de cada número quedaron en cada conjunto?

```
table(Glass$TipoDeVidrio)
table(entreno$TipoDeVidrio)
table(testeo$TipoDeVidrio)
```

Parte C – Árbol de Decisión

1. Cargue la librería `rpart` y cree un Árbol de Decisión para modelar el problema planteado.

```
arbol=rpart(TipoDeVidrio~., entreno, method="class")
```

Cargue la librería `rpart.plot` y grafique el Árbol de Decisión resultante (utilice el parámetro `cex=0.8` para una mejor visualización).

```
rpart.plot(arbol, extra=1, type=5, cex=0.8)
```

2. ¿Cuántas hojas tiene el Árbol de Decisión?
3. Según el Árbol de Decisión creado, ¿cuándo un vidrio es del tipo “FaroAuto”? (Indique las reglas siguiendo las ramas desde el nodo raíz hasta las hojas “FaroAuto”.)
4. Calcule la matriz de confusión utilizando la instrucción `confusionMatrix` de la librería `caret`. Muestre una captura de pantalla de los resultados completos (la matriz de confusión, *accuracy* y tablas).

```
pred=predict(arbol, testeo, type="class")
confusionMatrix(pred, testeo$TipoDeVidrio)
```

5. La cantidad de elementos de la matriz de confusión es igual a la cantidad de elementos de testeo (o sea `dim(testeo)`) Sume la cantidad de elementos de la diagonal de la matriz de confusión y divida el resultado por dim(testeo). Muestre la cuenta con números y muestre que es igual al accuracy.
6. Vea la tabla Statistics by Class debajo de la matriz de confusión e indique cuál clase presenta menor sensibilidad.
7. Con la instrucción Glass[numVidrio,] podemos obtener los datos de un vidrio de la base base[filas,columnas]Considere los 2 últimos números de su DNI (2numDNI) y busque el vidrio de esa fila vidrioAsignado=Glass[2numDNI,] vidrioAsignado Transcriba los datos del vidrio de ese número. ¿Qué tipo de vidrio es?
8. Prediga con el Árbol de Decisión el vidrio de los 2 últimos números de su DNI predict(arbol,vidrioAsignado,type=”class”) ¿Coincide la predicción con lo esperado?

Parte D – Optimización del modelo

1. Creación de un Árbol de Decisión sin algunas restricciones

   Con la instrucción `arbol$control` se pueden ver las restricciones en la creación del Árbol de Decisión. Cree un Árbol de Decisión sin algunas de las restricciones con la siguiente instrucción:

   ```
   arbolGrande=rpart(TipoDeVidrio~., entreno, method="class", cp=0, minsplit=0)
   ```

2. Dibujar el Árbol de Decisión

   Dibuje el Árbol de Decisión utilizando la siguiente instrucción:

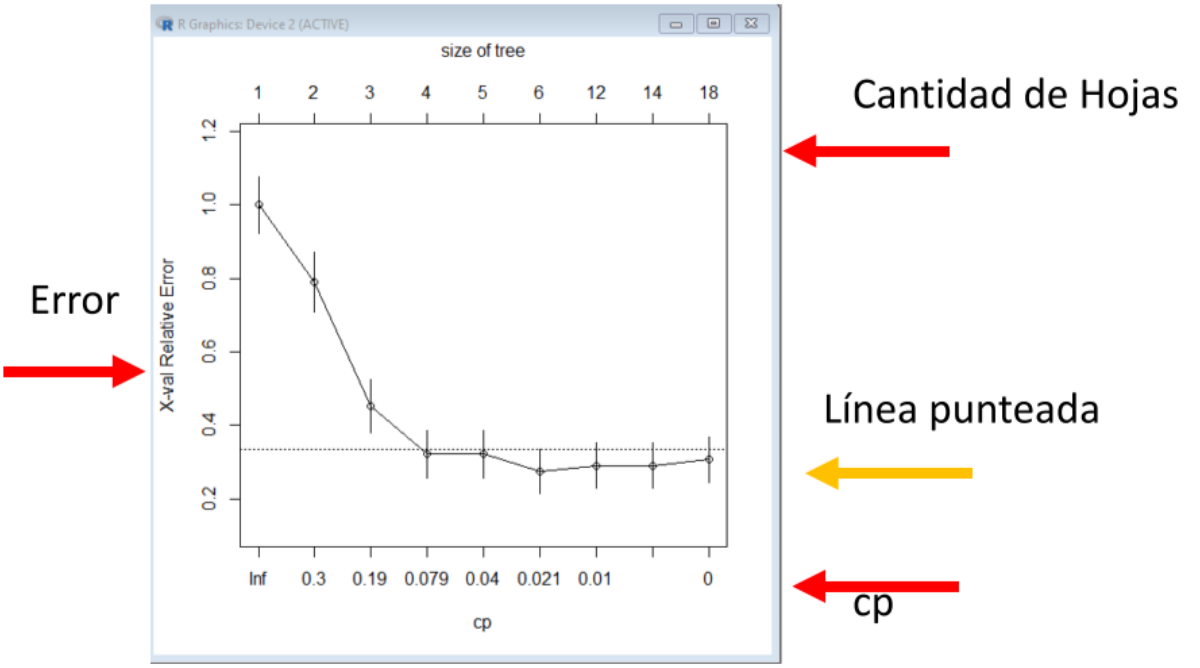
   ```
   rpart.plot(arbolGrande, extra=1, type=5, cex=0.6)
   ```

3. Mostrar el gráfico plotcp

   Muestre el gráfico `plotcp(arbolGrande)`. Amplíe el gráfico para ver más cps.

   Cada parte de este gráfico es la siguiente:

   - En la parte inferior del gráfico figura el cp (parámetro de complejidad) de cada Árbol de Decisión.
   - En la parte superior la cantidad de hojas.
   - En el eje vertical el error de cada modelo (calculado por cross validation).



4. Elija un `cp` del gráfico, cuyo error (eje vertical) esté por debajo de la línea punteada (el `cp` elegido debe figurar en el gráfico).

```
arbolPodado=prune(arbolGrande, cp=cpElegido)
```

   Indique el código R utilizado.

5. Muestre una captura de pantalla de los resultados completos (la matriz de confusión, *accuracy* y tablas) del Árbol de Decisión podado. ¿Mejoró el modelo?

```
pred=predict(arbolPodado, testeo, type="class")
confusionMatrix(pred, testeo$TipoDeVidrio)
```

   Verifique *accuracy*, y sensibilidades y especificidades de cada categoría. Tal vez el *accuracy* no mejoró, pero alguna categoría mejoró la sensibilidad.

Parte E – Parte Teórica

Resuma en un párrafo lo realizado en este TP.

2.2. Análisis Exploratorio de Datos

2.2.1. Desarrollo

Lo primero que se hace es el análisis exploratorio de datos, se comienza por leer la descripción del dataset a utilizar, la cual es: ”A data frame with 214 observation containing examples of the chemical analysis of 7 different types of glass. The problem is to forecast the type of class on basis of the chemical analysis. The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence (if it is correctly identified!).”
A continuación se obtiene la cantidad de observaciones y variables con las que se cuenta, son 214 muestras y 10 variables. Este dataset trata acerca de vidrios, en particular se busca predecir el tipo de vidrio en función de la composición quimica de los mismos. Los tipos de vidrio son:

Valor	Descripción
1	building_windows_float_processed
2	building_windows_non_float_processed
3	vehicle_windows_float_processed
4	vehicle_windows_non_float_processed (none in this database)
5	containers
6	tableware
7	headlamps

Tabla 2: Tipos de vidrios.

Cabe mencionar que hay un vidrio que no está representado, este es el 4 (la ventana del vehículo que no está procesada). Continuando con el análisis exploratorio de datos se procede a ver un summary de la base.

RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
Min.	1.511	10.73	0.000	0.290	69.81	0.000	5.430	0.000
1st Qu.	1.517	12.91	2.115	1.190	72.28	0.000	8.240	0.000
Median	1.518	13.30	3.480	1.360	72.79	0.000	8.600	0.000
Mean	1.518	13.41	2.685	1.445	72.65	0.175	8.957	0.057
3rd Qu.	1.519	13.82	3.600	1.630	73.09	0.000	9.172	0.100
Max.	1.534	17.38	4.490	3.500	75.41	3.150	16.190	0.510

Tabla 3: Resumen de la base de datos Glass, variables cuantitativas.

Type	Cantidad
1	70
2	76
3	17
5	13
6	9
7	29

Tabla 4: Resumen de la base de datos Glass, variables cualitativas.

Utilizando las tablas (4) y (2) se puede interpretar cuantos vidrios hay de cada tipo. A manera ilustrativa se procede a mostrar un ejemplo de cada vidrio, ignorando los repetidos (Edificio y faros de auto).

Figura 2: Ventana de edificio

Figura 3: Ventana de automóvil

Figura 4: Recipiente

Figura 5: Vajilla

Figura 6: Faro de auto

Luego se renombró la variable 'type' a 'TipoDeVidrio' y se asigno por cada numero una etiqueta en particular.

Valor previo	Descripción
1	VentanaTipo1
2	VentanaEdificio
3	VentanaAuto
5	Recipiente
6	Vajilla
7	FaroAuto

Tabla 5: Tabla de variables.

Finalmente se grafica un histograma de la variable TipoDeVidrio.

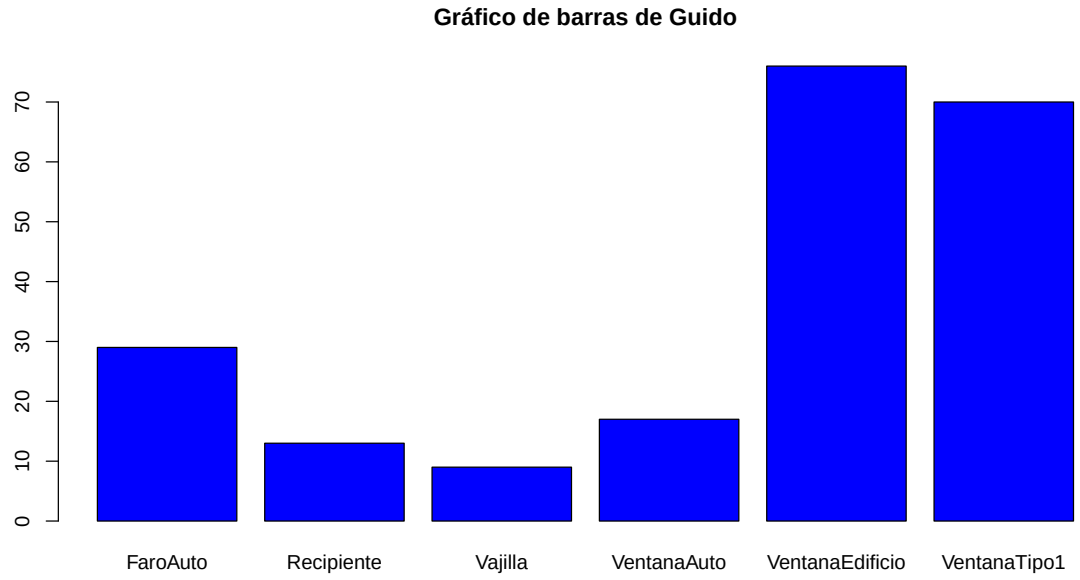


Figura 7: Histograma de la variable ”TipoDeVidrio”.

2.2.2. Código utilizado

```
library(caret)
library(MASS)
library(rpart)
library(rpart.plot)
library(mlbench)
directory <- dirname(rstudioapi::getActiveDocumentContext()$path)
#just use the setwd if not using rstudio
setwd(directory)
#EDA
data(Glass)
dim(Glass)
?Glass
summary(Glass)
summary(Glass$Type)
Glass$Type=as.character(Glass$Type)
Glass$Type[Glass$Type=="1"]="VentanaTipo1"
Glass$Type[Glass$Type=="2"]="VentanaEdificio"
Glass$Type[Glass$Type=="3"]="VentanaAuto"
Glass$Type[Glass$Type=="5"]="Recipiente"
Glass$Type[Glass$Type=="6"]="Vajilla"
Glass$Type[Glass$Type=="7"]="FaroAuto"
Glass$Type=factor(Glass$Type)
names(Glass)[names(Glass)=="Type"]="TipoDeVidrio"
summary(Glass)
plot(Glass$TipoDeVidrio,main="Gráfico de barras de Guido",col="blue")
```

2.3. Conjuntos

Lo siguiente es separar en dos conjuntos el dataset, siendo estos el conjunto de entrenamiento y testeo. Esto es fundamental para corroborar que el modelo efectivamente logro generalizar información y no ”aprenderse” el dataset (Overfitting). La semilla utilizada es 40397224, y la proporción de particion es p=0.75.

```
#Partition
```

```
set.seed(40397224)
partition=createDataPartition(y=Glass$TipoDeVidrio,p=0.75,list=FALSE)
entreno=Glass[partition,]
testeo=Glass[-partition,]
summary(entreno)
head(entreno)
summary(testeo)
head(testeo)
dim(Glass);dim(entreno);dim(testeo)
table(Glass$TipoDeVidrio);
table(entreno$TipoDeVidrio);
table(testeo$TipoDeVidrio)
```

Luego de generar la partición se hace un head() y un summary() de estos. Al hacer un head() a entreno queda:

RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	TipoDeVidrio
1.52101	13.64	4.49	1.10	71.78	0.06	8.75	0	0.00	VentanaTipo1
1.51761	13.89	3.60	1.36	72.73	0.48	7.83	0	0.00	VentanaTipo1
1.51742	13.27	3.62	1.24	73.08	0.55	8.07	0	0.00	VentanaTipo1
1.51596	12.79	3.61	1.62	72.97	0.64	8.07	0	0.26	VentanaTipo1
1.51756	13.15	3.61	1.05	73.24	0.57	8.24	0	0.00	VentanaTipo1
1.51755	13.00	3.60	1.36	72.99	0.57	8.40	0	0.11	VentanaTipo1

Tabla 6: Head de entreno

Si se hace un summary() :

	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	TipoDeVidrio
Min.	1.511	10.73	0.000	0.340	69.81	0.0000	5.430	0.000	0.00000	FaroAuto:22
1st Qu.	1.517	12.91	2.277	1.202	72.23	0.1225	8.225	0.000	0.00000	Recipiente:10
Median	1.518	13.30	3.480	1.410	72.77	0.5500	8.565	0.000	0.00000	Vajilla:7
Mean	1.518	13.44	2.760	1.467	72.61	0.5165	8.823	0.197	0.05173	VentanaAuto:13
3rd Qu.	1.519	13.82	3.610	1.667	73.10	0.6100	9.070	0.000	0.07750	VentanaEdificio:57
Max.	1.531	17.38	4.490	3.500	75.41	6.2100	14.680	3.150	0.51000	VentanaTipo1:53

Tabla 7: Summary de entreno

Y lo mismo correspondiente a testeo:

RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	TipoDeVidrio
3	1.51618	13.53	3.55	1.54	72.99	0.39	7.78	0	VentanaTipo1
4	1.51766	13.21	3.69	1.29	72.61	0.57	8.22	0	VentanaTipo1
7	1.51743	13.30	3.60	1.14	73.09	0.58	8.17	0	VentanaTipo1
9	1.51918	14.04	3.58	1.37	72.08	0.56	8.30	0	VentanaTipo1
18	1.52196	14.36	3.85	0.89	71.36	0.15	9.15	0	VentanaTipo1
22	1.51966	14.77	3.75	0.29	72.02	0.03	9.00	0	VentanaTipo1

Tabla 8: Head de testeo

	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	TipoDeVidrio
Min.	1.512	11.02	0.000	0.290	70.16	0.0000	7.780	0.0000	0.00000	FaroAuto:7
1st Qu.	1.517	12.92	0.000	1.117	72.36	0.1350	8.293	0.0000	0.00000	Recipiente:3
Median	1.518	13.27	3.490	1.300	72.86	0.5600	8.800	0.0000	0.00000	Vajilla:2
Mean	1.519	13.31	2.449	1.376	72.77	0.4365	9.373	0.1067	0.07346	VentanaAuto:4
3rd Qu.	1.519	13.81	3.580	1.545	73.07	0.6100	9.783	0.0000	0.14000	VentanaEdificio:19
Max.	1.534	14.94	3.850	2.790	75.18	2.7000	16.190	1.5700	0.34000	VentanaTipo1:17

Tabla 9: Summary de testeo

Las dimensiones finales son (~75 % entreno y ~25 % testeo):

	FaroAuto	Recipiente	Vajilla	VentanaAuto	VentanaEdificio	VentanaTipo1	Total
Glass	29	13	9	17	76	70	214
entreno	22	10	7	13	57	53	162
testeo	7	3	2	4	19	17	52

Tabla 10: Descripción de la tabla

2.4. Árbol de Decisión

Utilizando la librería rpart se arma un arbol de decisión.

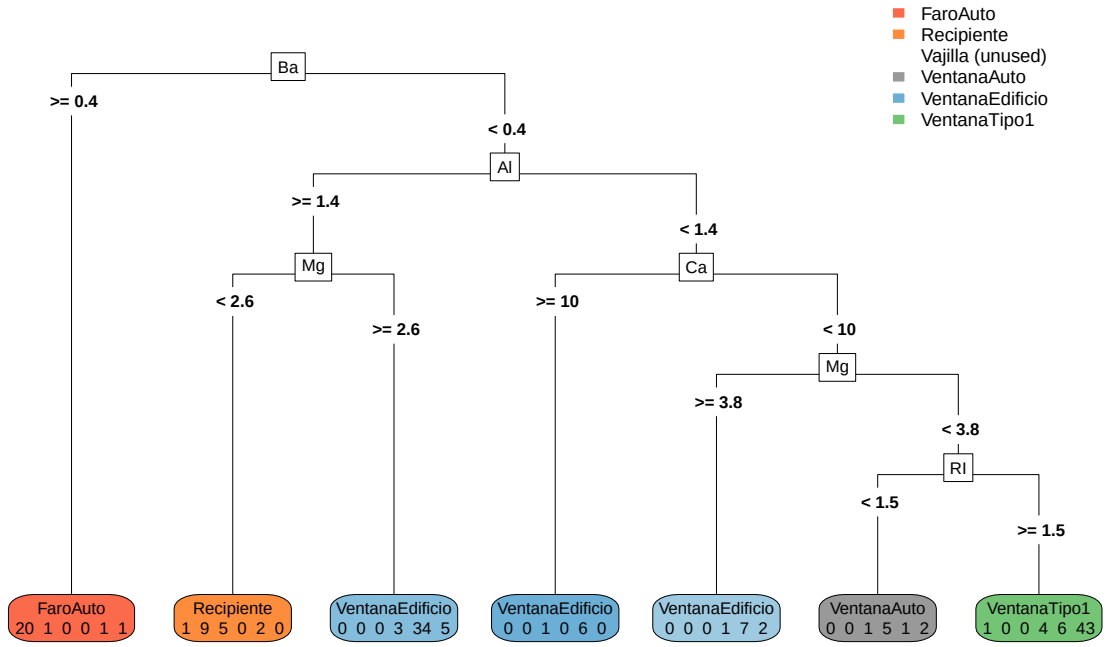


Figura 8: Primer árbol de decisión

Cuenta con 7 hojas. Para que el árbol decida que una muestra es un faro de auto basta con que el contenido de Bario sea mayor igual a 0.4. A continuación se calcula la matriz de confusión.

		Reference			
	FaroAuto	Recipiente	Vajilla	VentanaAuto	VentanaEdificio
Prediction	6	0	0	0	0
	0	2	1	0	2
	0	0	0	0	0
	1	0	0	1	3
	0	1	0	1	10
	0	0	1	2	4

Tabla 11: Matriz de confusión

Accuracy	0.6346
95 % CI	(0.4896, 0.7638)
No Information Rate	0.3654
P-Value [Acc > NIR]	7.296e-05
Kappa	0.506
Mcnemar's Test P-Value	NA

Tabla 12: Estadísticas generales

	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Prevalence	Detection Rate
Class: FaroAuto	0.8571	1.0000	1.0000	0.9783	0.1346	0.1154
Class: Recipiente	0.66667	0.93878	0.40000	0.97872	0.05769	0.03846
Class: Vajilla	0.00000	1.00000	NaN	0.96154	0.03846	0.00000
Class: VentanaAuto	0.25000	0.89583	0.16667	0.93478	0.07692	0.01923
Class: VentanaEdificio	0.5263	0.8788	0.7143	0.7632	0.3654	0.1923
Class: VentanaTipo1	0.8235	0.8000	0.6667	0.9032	0.3269	0.2692

Tabla 13: Estadísticas por clase

Cabe notar que la cantidad de elementos de la matriz de confusión es igual a la cantidad de elementos de testeo. Además se procede a calcular la accuracy utilizando la matriz (11).

$$\text{Accuracy} = \frac{\sum TP + TN}{\sum TP + TN + FP + FN} = \frac{6+2+0+1+10+14}{52} = \frac{33}{52} \approx 0.6346 \quad (1)$$

Donde TP es True positive, TN True negative, FP False positive y FN Flase negative. Cabe mencionar que el valor calculado coincide con obtenido por R. Por otro lado se observa que la categoría con menor sensibilidad corresponde a ”Vajilla”.
Se elije el vidrio en la posicion 24 como muestra, el cual es tipo ’VentanaTipo1’. Se prueba el modelo con este

RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	TipoDeVidrio
1.51	12.81	3.57	1.35	73.02	0.62	8.59	0	0	VentanaTipo1

vidrio en particular y efectivamente lo predice de manera correcta.

2.5. Optimización del modelo

Se crea un árbol de decisión pero obligando a que llegue a la maxima longitud posible.

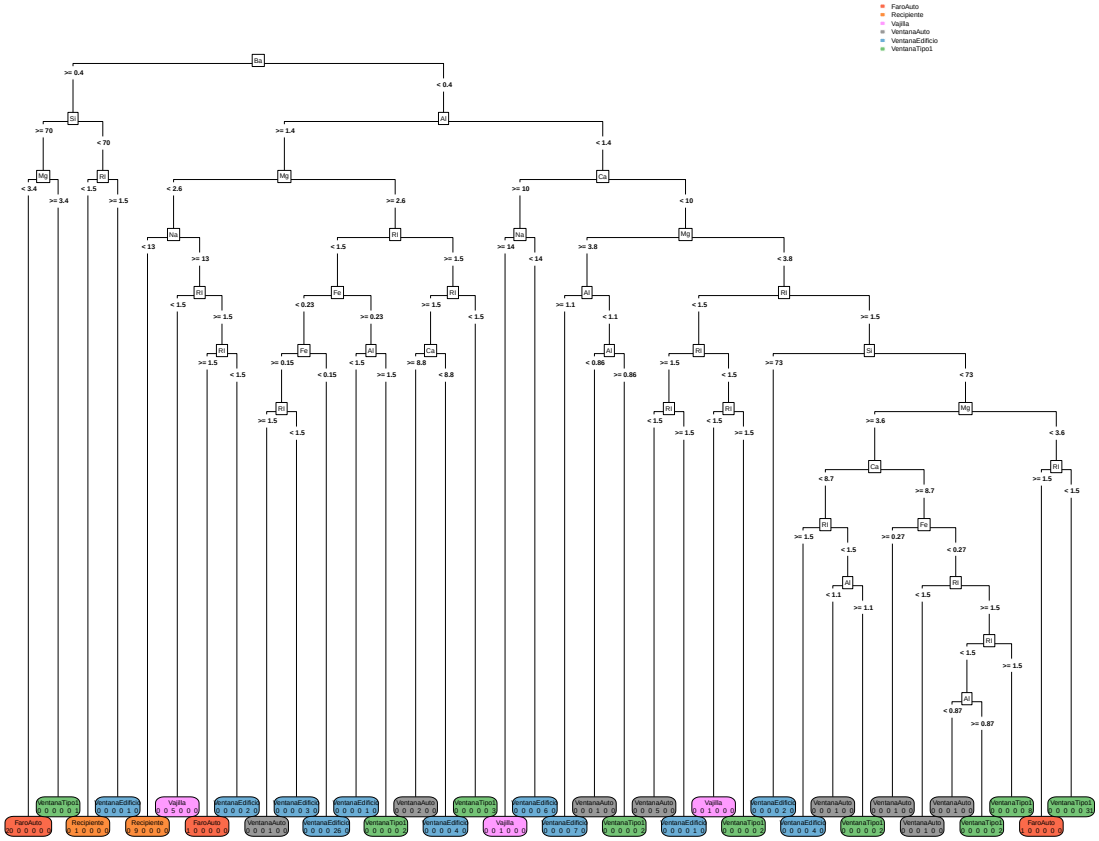


Figura 9: Árbol de decisión grande

Y se grafica el error de cada modelo (calculado por cross validation) en función del parámetro de complejidad (cp).

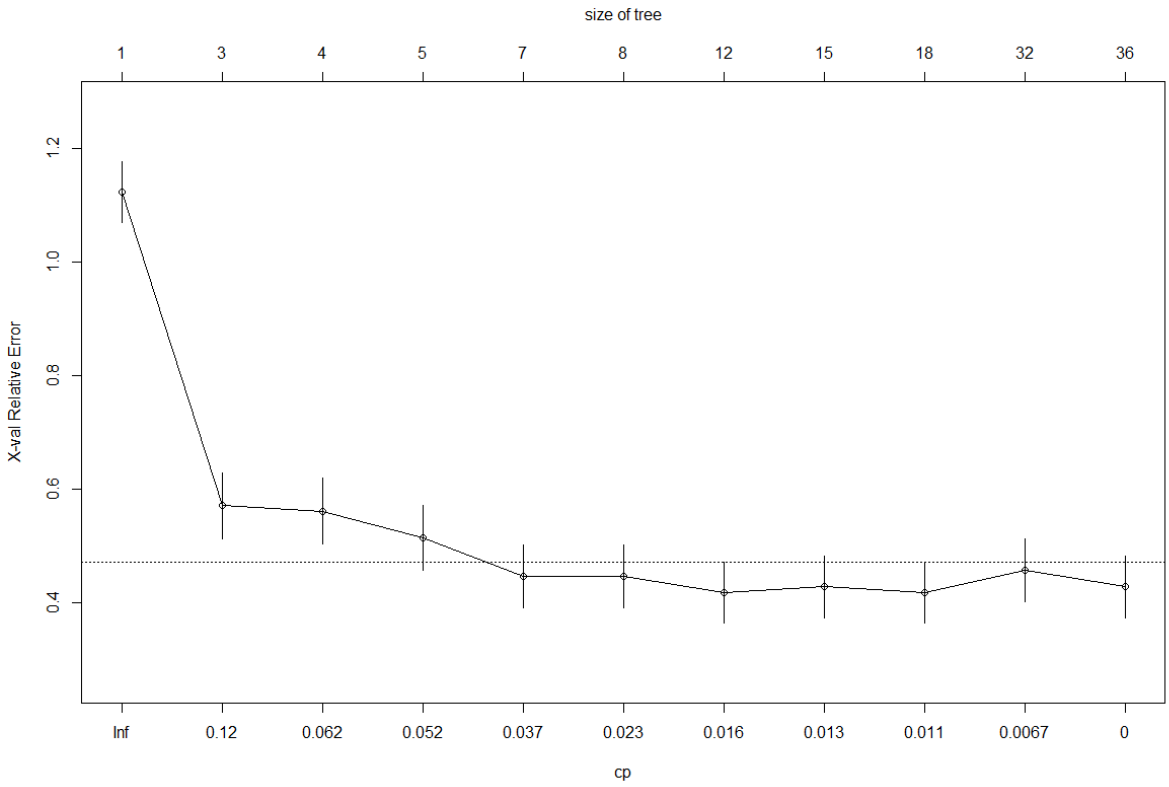


Figura 10: Error en función del parámetro de complejidad

Se elije como cp aquel que con el menor número de hojas obtenga el menor error, 12 hojas, cp=0.016. A continuación se grafica el árbol de decisión final.

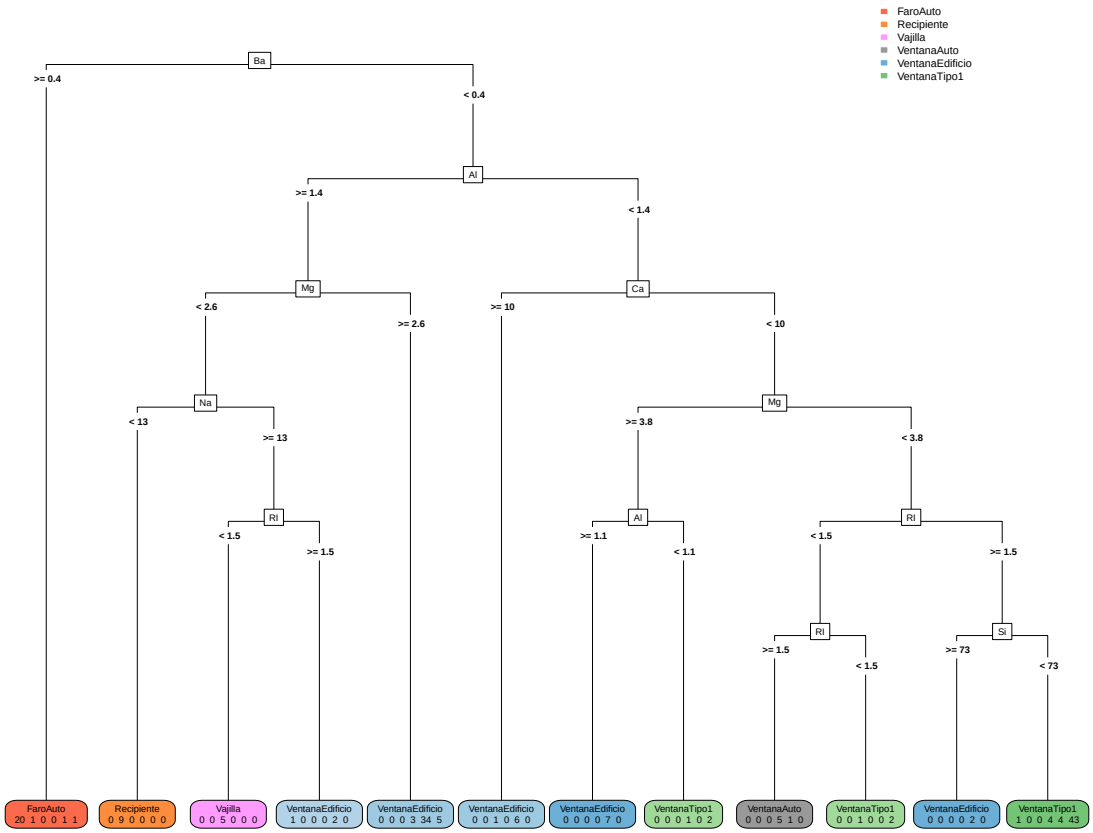


Figura 11: Árbol de decisión podado

Utilizando el metodo confusionMatrix se obtiene:

		Reference			
	FaroAuto	Recipiente	Vajilla	VentanaAuto	VentanaEdificio
Prediction	6	0	0	0	0
	0	2	0	0	1
	0	0	1	0	0
	1	0	0	1	3
	0	1	0	1	11
	0	0	1	2	4

Tabla 14: Matriz de confusión

Accuracy	0.6923
95 % CI	(0.549, 0.8128)
No Information Rate	0.3654
P-Value [Acc > NIR]	1.719e-06
Kappa	0.5781
Mcnemar's Test P-Value	NA

Tabla 15: Estadísticas generales

	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Prevalence	Detection Rate
Class: FaroAuto	0.8571	1.0000	1.0000	0.9783	0.1346	0.1154
Class: Recipiente	0.66667	0.97959	0.66667	0.97959	0.05769	0.03846
Class: Vajilla	0.50000	1.00000	1.00000	0.98039	0.03846	0.01923
Class: VentanaAuto	0.25000	0.91667	0.20000	0.93617	0.07692	0.01923
Class: VentanaEdificio	0.5789	0.8788	0.7333	0.7838	0.3654	0.2115
Class: VentanaTipo1	0.8824	0.8000	0.6818	0.9333	0.3269	0.2885

Tabla 16: Estadísticas por clase

Se puede ver como la accuracy aumento en un 5.77 % y se puede notar como la sensibilidad para la vajilla aumento de 0 a 0.5, el aumento en sensibilidad para los dos tipos de ventanas de edificio, la variable specificity practicamente quedo igual, teniendo un aumento tanto para VentanaAuto como Recipiente.

2.6. Parte Teórica

En este trabajo se realizó un árbol de decisión, capaz de predecir que tipo de vidrio es una muestra, en base a su composición química. Para esto se realizo un análisis exploratorio de datos, para entender el dataset, ver que variables resultan relevantes y cuales no, al igual que un acondicionamiento de datos. Por otro lado se hicieron varios arboles de decisión, el primero es el que la librería rpart entrega por defecto dado el dataset. El segundo fue un árbol ampliado a su mayor longitud y el último es el optimizado en función del error de cada modelo. Finalmente se observaron las mejoras de del modelo optimizado versus el original.

2.7. Código utilizado

A continuación se muestra la totalidad del código utilizado.

```
library(caret)
library(MASS)
library(rpart)
library(rpart.plot)
library(mlbench)
directory <- dirname(rstudioapi::getActiveDocumentContext()$path)
#just use the setwd if not using rstudio
setwd(directory)
#EDA
data(Glass)
dim(Glass)
?Glass
summary(Glass)
summary(Glass$Type)
Glass$Type=as.character(Glass$Type)
Glass$Type[Glass$Type=="1"]="VentanaTipo1"
Glass$Type[Glass$Type=="2"]="VentanaEdificio"
Glass$Type[Glass$Type=="3"]="VentanaAuto"
Glass$Type[Glass$Type=="5"]="Recipiente"
Glass$Type[Glass$Type=="6"]="Vajilla"
Glass$Type[Glass$Type=="7"]="FaroAuto"
Glass$Type=factor(Glass$Type)
names(Glass)[names(Glass)=="Type"]="TipoDeVidrio"
summary(Glass)
plot(Glass$TipoDeVidrio,main="Gráfico de barras de Guido",col="blue")


#Partition
set.seed(40397224)
partition=createDataPartition(y=Glass$TipoDeVidrio,p=0.75,list=FALSE)
entreno=Glass[partition,]
testeo=Glass[-partition,]
```

```
summary(entreno)
head(entreno)
summary(testeo)
head(testeo)
dim(Glass);dim(entreno);dim(testeo)
# decision tree
arbol=rpart(TipoDeVidrio~.,entreno,method="class")
rpart.plot(arbol,extra=1,type=5,cex=0.8)
sum(arbol$frame$var == "<leaf>")
#es FaroAuto si el contenido de Bario es mayor igual a 0.4
confusionMatrix(pred,testeo$TipoDeVidrio)
vidrioAsignado=Glass[24,]
vidrioAsignado
predict(arbol,vidrioAsignado,type="class")
#Optimizacion
arbol$control
arbolGrande=rpart(TipoDeVidrio~.,entreno,method="class",cp=0,minsplit=0)
rpart.plot(arbolGrande,extra=1,type=5,cex=0.4)
plotcp(arbolGrande)
arbolPodado=prune(arbolGrande,cp=0.016)
pred=predict(arbolPodado,testeo,type="class")
confusionMatrix(pred,testeo$TipoDeVidrio)
rpart.plot(arbolPodado,extra=1,type=5,cex=0.6)
```