Práctica 2: Limpieza y validación de los datos

Adrián Quijada Gomariz

7 de enero, 2019

Contents

1 Descripción del dataset.

```
df <- fread("..\\data\\dataset.csv", encoding = 'UTF-8')
clase <- sapply(df, class)
kable(data.frame(variable=names(clase), class=as.vector(clase)))
```

variable	class
date	character
floor	character
garage	integer
link	character
m2	numeric
off	integer
price	integer
rooms	integer
title	character

El dataset contiene información sobre los 16837 inmuebles en venta dentro del municipio de Barcelona a fecha 27-12-2018. Para crearlo se ha usado una versión extendida del *crawler* desarrollado en la *Práctica* 1[1].

Como se puede observar, contiene información sobre:

- La fecha de descarga del anuncio.

- Planta del inmueble (en caso de ser un piso).
- Si tiene garaje asociado.
- Link de la publicación.
- Metros cuadrados del inmueble.
- Descuento sobre el precio publicado inicialmente.
- Precio del inmueble.
- Número de habitaciones.
- Titulo del anuncio, de donde se puede extraer la zona.

Con esta información nos podríamos preguntar: **¿es el momento idóneo para comprar un inmueble? ¿es posible que baje de precio?** Precisamente, a este tipo de preguntas es a las que el modelo pretende dar respuesta.

Para ello se genera la variable respuesta dicotómica que tomará un valor positivo para los inmuebles que hayan sufrido un descuento y un valor negativo para aquellos que no:

```
df$target <- ifelse(df$off > 0, 1, 0)
df$target <- as.factor(df$target)
kable(table(df$target))
```

Var1	Freq
0	15157
1	1680

Por lo consiguiente, se genera una variable categórica que representa que a fecha 27-12-2018, el **10%** de los anuncios publicados han sufrido un descuento en su precio inicial propuesto.

Aunque nuestra variable respuesta esté claramente desbalanceada, se procederá a generar un modelo de aprendizaje automático que intentará discriminar entre las distintas propiedades de un inmueble para dar respuesta a las preguntas planteadas.

2 Integración y selección de los datos de interés a analizar.

Para el desarrollo del modelo usaremos todas las variables disponibles salvo aquellas que no tienen una relación lógica con la variable respuesta. Será el caso de la fecha, el link, el descuento y el titulo.

3 Limpieza de los datos.

A continuación se realiza el tratamiento adecuado para cada una de las variables seleccionadas en función de su naturaleza y tipología:

- Variable *garage*:

```
df$garage <- as.factor(df$garage)
kable(table(df$garage))
```

Var1	Freq
0	14419
1	2418

Se transforma a categórica. El **14%** de los inmuebles tiene garaje.

- Variable *zone*:

```r
barris <- read_xlsx("..\\data\\barrios barcelona.xlsx")
df$zone <- ''

for(i in barris$barris){
  df$zone[grepl(i, df$title)] <- i
}

tz <- table(df$zone)
kable(tz[order(tz, decreasing = T)])
```

Var1	Freq
Sant Gervasi	1099
Dreta de l'Eixample	1038
El Raval	999
Nova Esquerra de l'Eixample	999
El Gòtic	754
Sant Pere	709
El Poble Sec	606
Sant Antoni	606
Antiga Esquerra de l'Eixample	579
La Sagrada Família	563
Sants	481
Vila de Gràcia	434
El Fort Pienc	376
El Putxet	345
Les Corts	330
Camp de l'Arpa	323
Diagonal Mar	310
El Guinardó	303
El Poblenou	303
Sant Andreu	284
El Camp d'En Grassot i Gràcia Nova	249
El Clot	248
Pedralbes	228
Sarrià	222
La Barceloneta	213
El Carmel	201
Provençals del Poblenou	199
Hostafrancs	182
La Maternitat i Sant Ramon	174
Navas	173
El Besòs	169
El Baix Guinardó	163
Les Tres Torres	158
Porta	144
El Parc i la Llacuna del Poblenou	143
Vallcarca	142
La Sagrera	138
La Prosperitat	133
La Salut	133
Vilapicina i la Torre Llobeta	128
Sant Martí	119
Les Roquetes	111

Var1	Freq
Horta	109
Vallvidrera	103
La Bordeta	101
La Verneda i la Pau	101
La Teixonera	98
Montbau	95
La Trinitat Vella	93
Vila Olímpica	91
La Marina del Port	89
La Font d'En Fargues	88
Verdun	81
La Font de la Guatlla	80
Can Baró	76
El Coll	76
El Bon Pastor	68
El Congrés i els Indians	65
Torre Baró	49
La Trinitat Nova	48
El Turó de la Peira	46
La Guineueta	29
Zona Franca	25
Baró de Viver	8
La Clota	5
La Marina del Prat Vermell	2

```
df$zone[df$zone == 'Provençals del Poblenou'] <- 'El Poblenou'
df$zone[df$zone == 'Hostafrancs'] <- 'Sants'
df$zone[df$zone == 'La Maternitat i Sant Ramon'] <- 'Les Corts'
df$zone[df$zone == 'El Camp d\'En Grassot i Gràcia Nova'] <- 'Vila de Gràcia'
df$zone[df$zone == 'El Besòs'] <- 'El Poblenou'
df$zone[df$zone == 'El Baix Guinardó'] <- 'El Guinardó'
df$zone[df$zone == 'Les Tres Torres'] <- 'Sant Gervasi'
df$zone[df$zone == 'Porta'] <- 'Nou Barris'
df$zone[df$zone == 'El Parc i la Llacuna del Poblenou'] <- 'El Poblenou'
df$zone[df$zone == 'El Congrés i els Indians'] <- 'Navas'
df$zone[df$zone == 'La Sagrera'] <- 'Navas'
df$zone[df$zone == 'La Prosperitat'] <- 'Nou Barris'
df$zone[df$zone == 'La Salut'] <- 'El Carmel'
df$zone[df$zone == 'Vilapicina i la Torre Llobeta'] <- 'Nou Barris'
df$zone[df$zone == 'Sant Martí'] <- 'El Clot'
df$zone[df$zone == 'Les Roquetes'] <- 'Nou Barris'
df$zone[df$zone == 'Horta'] <- 'Nou Barris'
df$zone[df$zone == 'Vallvidrera'] <- 'Sarrià'
df$zone[df$zone == 'La Bordeta'] <- 'Sants'
df$zone[df$zone == 'La Verneda i la Pau'] <- 'El Clot'
df$zone[df$zone == 'La Teixonera'] <- 'El Carmel'
df$zone[df$zone == 'Montbau'] <- 'El Carmel'
df$zone[df$zone == 'La Trinitat Vella'] <- 'Nou Barris'
df$zone[df$zone == 'Vila Olímpica'] <- 'El Poblenou'
df$zone[df$zone == 'La Marina del Port'] <- 'El Poble Sec'
df$zone[df$zone == 'La Font d\'En Fargues'] <- 'El Guinardó'
```

```
df$zone[df$zone == 'Verdun'] <- 'Nou Barris'
df$zone[df$zone == 'La Font de la Guatlla'] <- 'El Poble Sec'
df$zone[df$zone == 'Can Baró'] <- 'El Guinardó'
df$zone[df$zone == 'El Coll'] <- 'Vallcarca'
df$zone[df$zone == 'El Bon Pastor'] <- 'Sant Andreu'
df$zone[df$zone == 'Torre Baró'] <- 'Nou Barris'
df$zone[df$zone == 'La Trinitat Nova'] <- 'Nou Barris'
df$zone[df$zone == 'El Turó de la Peira'] <- 'Nou Barris'
df$zone[df$zone == 'La Guineueta'] <- 'Nou Barris'
df$zone[df$zone == 'Zona Franca'] <- 'El Poble Sec'
df$zone[df$zone == 'Baró de Viver'] <- 'Nou Barris'
df$zone[df$zone == 'La Clota'] <- 'El Carmel'
df$zone[df$zone == 'La Marina del Prat Vermell'] <- 'El Poble Sec'

df$zone <- factor(df$zone)
```

Para la creación de la variable *zone* nos ayudamos de un diccionario que recoge los barrios de Barcelona. De esta forma, del título de la publicación extraemos el barrio en el que se encuentra el inmueble. Después, las zonas menos representativas se unen a los barrios vecinos con el fin de que el contenido de la variable sea más representativo. Se observa que los barrios con menos de 200 inmuebles en venta se pueden unir a barrios vecinos. Y finalmente, asignamos a la variable el tipo factor.

Se puede observar que la mayor concentración de inmuebles está en la **zona alta y centro** del municipio.

- Variable *floor*:

```
df$floor[df$floor == ''] <- 'Casa'

tf <- table(df$floor)
kable(tf[order(tf, decreasing = T)])
```

Var1	Freq
1ª	3194
2ª	2443
3ª	2249
Casa	2040
4ª	2033
Bajo	1192
5ª	1147
6ª	864
Entreplanta	573
7ª	522
8ª	275
9ª	128
10ª	60
11ª	25
18ª	17
16ª	13
15ª	12
12ª	7
13ª	7
14ª	6
20ª	5
21ª	5
22ª	4

Var1	Freq
23ª	4
24ª	3
60ª	3
19ª	2
26ª	2
17ª	1
25ª	1

```
df$floor[df$floor %in% names(tf)[tf < 500]] <- 'Atico'

df$floor <- factor(df$floor)
```

Entendemos que los inmuebles dónde no se especifica la planta se tratan de adosados e imputamos así los valores vacios. De la misma forma que el caso anterior, los valores no representativos se agrupan en una única clase. Ya que todos se corresponden a plantas "altas" la llamamos "ático". Y finalmente, asignamos a la variable el tipo factor.

Se puede observar que la mayoría de pisos del dataset se distribuyen entre la **primera y la cuarta planta**.

- El resto de variables ya tienen el tipo de dato que les corresponde.

3.1 ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Comprobamos qué variables contienen nulos.

```
colSums(is.na(df))
```

```
##   date  floor garage   link     m2    off  price  rooms  title target
##      0      0      0      0      0      0      0    299      0      0
##   zone
##      0
```

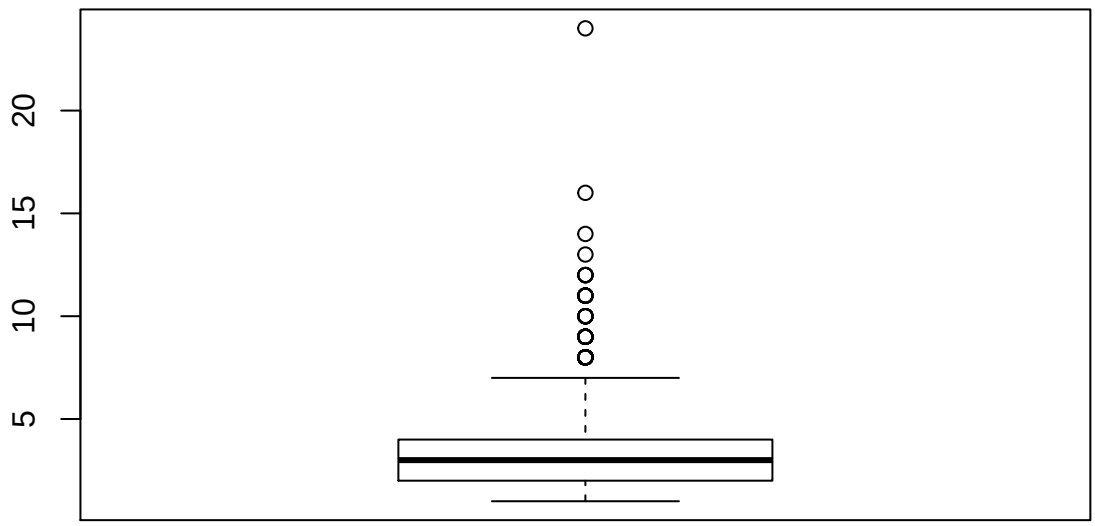
Como se puede observar, únicamente la variable *rooms* contiene nulos. Una buena alternativa, ya que son pocos casos, sería imputar los nulos usando la media de la variable. Pero antes, vamos a tratar los *outliers*.

3.2 Identificación y tratamiento de valores extremos.

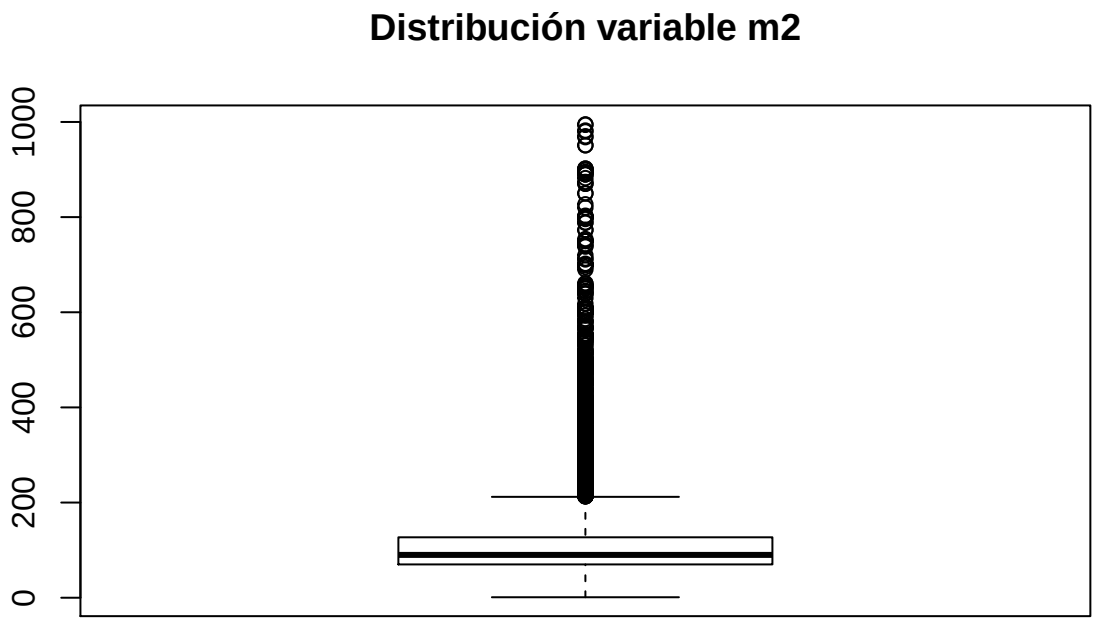
En las variables de tipo categórico que ya se han tratado, las agrupaciones realizadas podrían ser una forma de tratamiento de valores extremos.

El resto de variables se comprueban a continuación:

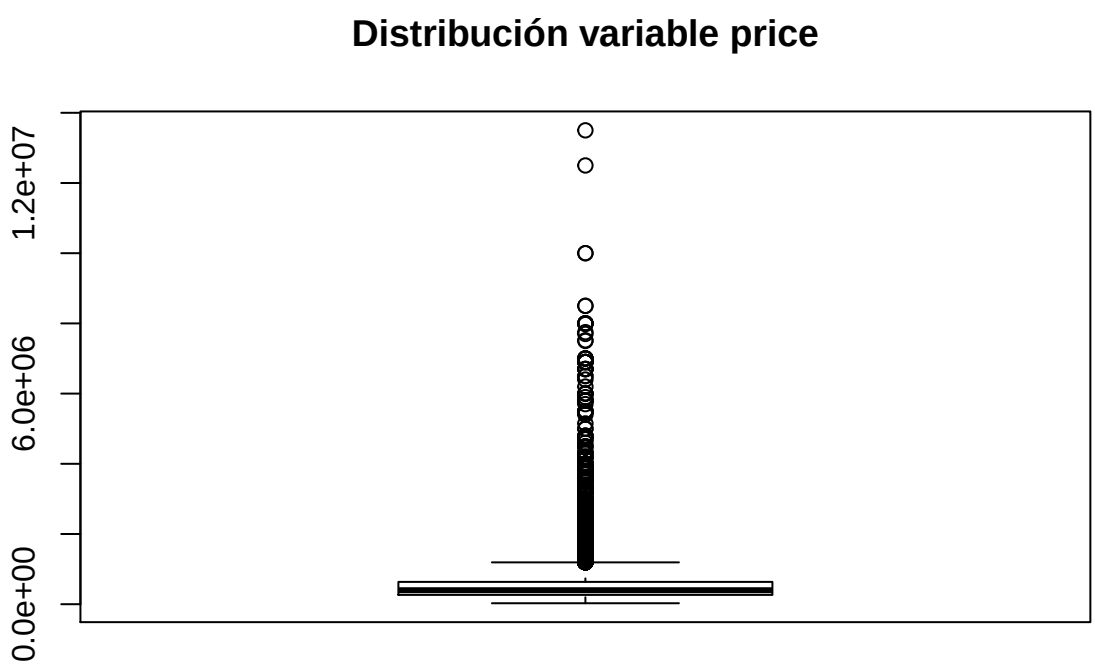
```
boxplot(df$rooms, main = "Distribución variable rooms")
```




```r
boxplot(df$m2, main = "Distribución variable m2")
```




```r
boxplot(df$price, main = "Distribución variable price")
```



Consideraremos un valor extremo x es todo aquel que se encuentre fuera del rango: $mean(x) \pm 3 * sd(x)$.

```
remove_outliers <- function(x, limit = 3) {
  mn <- mean(x, na.rm = T)
  out <- limit * sd(x, na.rm = T)
  x < (mn - out) | x > (mn + out)
}
```

Ya que sólo se trata de 62 casos, se eliminan del conjunto de datos.

```
df <- df[remove_outliers(df$m2, 3)==FALSE |
           remove_outliers(df$price, 3)==FALSE |
           remove_outliers(df$rooms, 3)==FALSE,]
```

Ahora estamos en disposición de imputar los nulos de la variable *rooms*.

```
df$rooms[is.na(df$rooms)] <- round(mean(df$rooms, na.rm = T))
```

4 Análisis de los datos.

4.1 Selección de los grupos de datos que se quieren analizar/comparar.

Deshechamos aquellas variables que no vamos a utilizar.

```
df$date <- NULL
df$link <- NULL
df$off <- NULL
df$title <- NULL
```

El resto se analizará a continuación.

4.2 Comprobación de la normalidad y homogeneidad de la varianza.

Se comprueba la normalidad de las variables númericas usando el test de Saphiro-Wilk.

```
shapiro.test(sample(df$price, 5000))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(df$price, 5000)
## W = 0.57776, p-value < 2.2e-16
```

```
shapiro.test(sample(df$m2, 5000))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(df$m2, 5000)
## W = 0.66291, p-value < 2.2e-16
```

```
shapiro.test(sample(df$rooms, 5000))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(df$rooms, 5000)
## W = 0.8588, p-value < 2.2e-16
```

Obtenemos que ninguna de las tres variables sigue una distribución normal ya que el p-value siempre es inferior a 0.05, por lo que se rechaza la hipótesis nula. Aunque sabemos que según el teorema del limite central cualquier población con más de 30 observaciones se podría aproximar a una distribución normal de media 0 y desviación estándar 1.

Se comprueba la homogeniedad de la varianza usando el test de Levene.

```
leveneTest(y = df$price, group = df$target, center = "mean")
```

```
## Levene's Test for Homogeneity of Variance (center = "mean")
##          Df F value    Pr(>F)
## group     1  104.63 < 2.2e-16 ***
##       16773
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(y = df$m2, group = df$target, center = "mean")
```

```
## Levene's Test for Homogeneity of Variance (center = "mean")
##          Df F value    Pr(>F)
## group     1  59.382 1.371e-14 ***
##       16773
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(y = df$rooms, group = df$target, center = "mean")
```

```
## Levene's Test for Homogeneity of Variance (center = "mean")
##          Df F value   Pr(>F)
```

```
## group     1  7.8333 0.005135 **
##       16773
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

El test de Levene indica que las varianzas entre grupos son significativamente distintas para las variables *price* y *m2* ya que el p-value es inferior a 0.05 en ambos casos. Por lo que podemos rechazar la hipótesis nula de homoscedasticidad. Para la variable *rooms* no es tan distinta la varianza entre grupos con un p-value inferior a 0.1 pero no a 0.05. En este caso se tendría que aceptar la hipótesis nula y asegurar la igualdad de las varianzas.

En definitiva, el test de Levene nos está indicando que hay una varianza significativa en el precio de los inmuebles que han sufrido un descuento y de los que no. Además de que hay una varianza significativa en los metros cuadrados de los inmuebles que han sufrido un descuento y de los que no.

4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos.

Antes de la aplicación de técnicas de minería de datos, vamos a comprobar las correlaciones que existen entre las variables categóricas de nuestro dataset con la variable respuesta. Para ello hacemos uso de la siguiente función que nos resume el porcentaje de casos positivos de la variable respuesta exitentes para cada categoría de la variable explicativa.

```
table.resume <- function (x, y){
  t <- table(x, y)
  N <- t[,1]+t[,2]
  return(data.frame('y' = t[,2], 'N' = N, 'var' = t[,2]/N))
}
```

- Correlación para la variable *garage*:

```
kable(table.resume(df$garage, df$target))
```

	y	N	var
0	1503	14398	0.1043895
1	175	2377	0.0736222

La diferencia entre las correlaciones es del 0.03%. **Siendo ligeramente mayor para los inmuebles sin garaje**.

- Correlación para la variable *zone*:

```
kable(table.resume(df$zone, df$target))
```

	y	N	var
Antiga Esquerra de l'Eixample	49	579	0.0846287
Camp de l'Arpa	64	323	0.1981424
Diagonal Mar	20	310	0.0645161
Dreta de l'Eixample	59	1037	0.0568949
El Carmel	62	532	0.1165414
El Clot	47	468	0.1004274
El Fort Pienc	45	376	0.1196809
El Gòtic	55	754	0.0729443
El Guinardó	51	630	0.0809524
El Poble Sec	87	802	0.1084788
El Poblenou	93	905	0.1027624

	y	N	var
El Putxet	31	345	0.0898551
El Raval	93	999	0.0930931
La Barceloneta	21	213	0.0985915
La Sagrada Família	93	563	0.1651865
Les Corts	41	504	0.0813492
Navas	40	376	0.1063830
Nou Barris	129	979	0.1317671
Nova Esquerra de l'Eixample	80	999	0.0800801
Pedralbes	7	206	0.0339806
Sant Andreu	58	352	0.1647727
Sant Antoni	65	606	0.1072607
Sant Gervasi	133	1230	0.1081301
Sant Pere	61	709	0.0860367
Sants	76	764	0.0994764
Sarrià	32	314	0.1019108
Vallcarca	21	218	0.0963303
Vila de Gràcia	65	682	0.0953079

La zona con mayor porcentaje de pisos con descuento en sus precios iniciales es **Camp de l'Arpa**. Y la zona con menor porcentaje es **Pedralbes**.

- Correlación para la variable *floor*:

```
kable(table.resume(df$floor, df$target))
```

	y	N	var
1ª	325	3194	0.1017533
2ª	253	2441	0.1036460
3ª	228	2248	0.1014235
4ª	193	2032	0.0949803
5ª	119	1145	0.1039301
6ª	100	864	0.1157407
7ª	50	519	0.0963391
Atico	61	580	0.1051724
Bajo	150	1191	0.1259446
Casa	143	1989	0.0718954
Entreplanta	56	572	0.0979021

La planta dónde se encuentran mayor porcentaje de pisos con descuento en sus precios iniciales es **Bajo**. Y con menor porcentaje es **Casa**.

Por lo que, en vista de los resultados obtenidos hasta ahora nos encontramos con base suficiente para pensar que usando un modelo de aprendizaje automático podremos predecir un posible descuento sobre el precio inicial. En ésta linea se particionan los datos en *train* y *test*, se usa la técnica *Random Forest* con todas las variables y se aplica un muestreo relativo a la clase positiva (inmuebles con descuento) para intentar obtener un resultado más robusto.

```
set.seed(0)

r <- nrow(df)
train_ind <- sample(r, 0.9 * r)
train <- df[train_ind,]
```

```
test <- df[-train_ind,]
n <- table(train$target)[[2]]

rf <- randomForest(target~garage+m2+price+rooms+zone+floor,
                   mtry = length(train)-1,
                   importance=TRUE,
                   sampsize=c(n, n),
                   data = train)
```

5 Representación de los resultados a partir de tablas y gráficas.

Comenzamos revisando el resultado del modelo ejecutado.

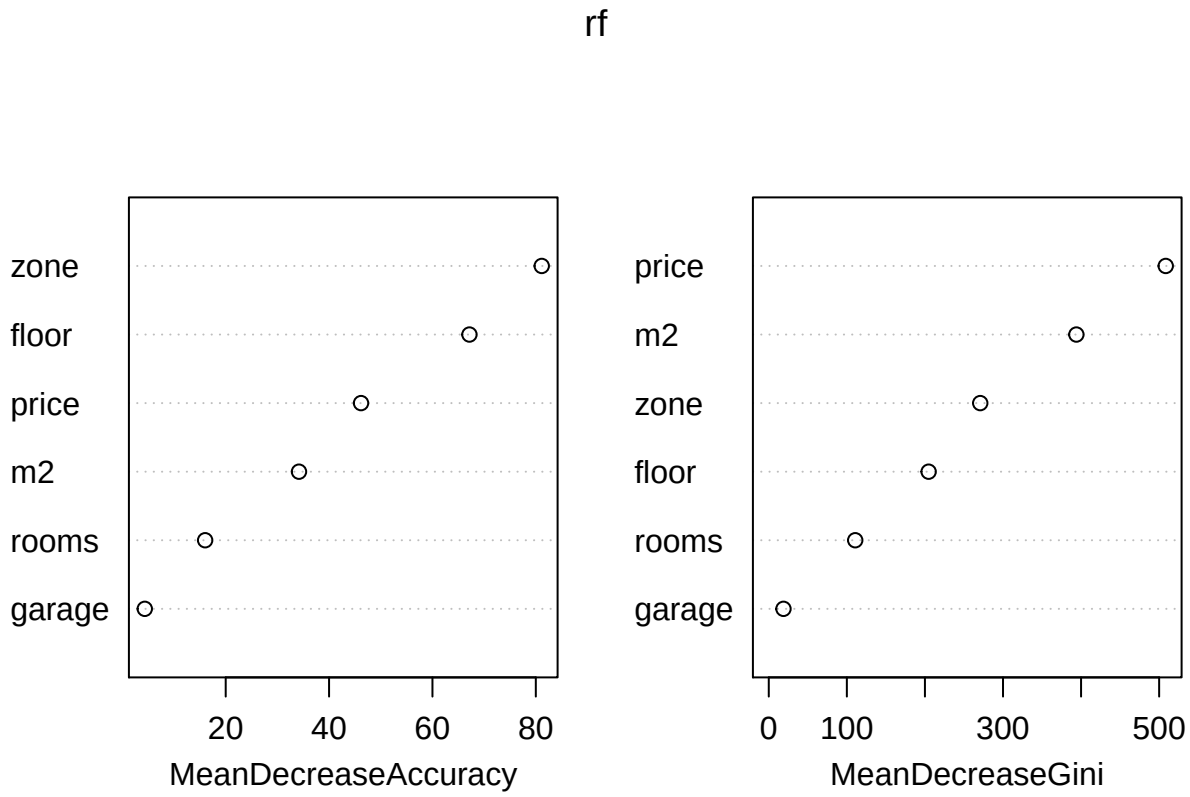
```
rf
```

```
##
## Call:
##  randomForest(formula = target ~ garage + m2 + price + rooms +      zone + floor, data = train, mtry
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##         OOB estimate of  error rate: 22.42%
## Confusion matrix:
##       0    1 class.error
## 0 10887 2680   0.1975381
## 1   704  826   0.4601307
```

En resumen, se destacan dos cosas:

- El *Out Of Bag error* es del 23%. Esta métrica se utiliza como validación del modelo, por lo que el *accuracy* del mismo debería ser cercano al 77.08%.
- La matriz de confusión. Destacar el alto error en la clase positiva: del 46%. Podría ser debido a que la variable respuesta está muy poco balanceada. También hay que tener en cuenta que, por defecto, el punto de corte está situado en 0.5 y en raras ocasiones es el punto de corte óptimo aunque ésto depende, más bien, del uso que se le quiera dar al modelo.

```
varImpPlot(rf)
```



Nos fijamos en la gráfica de la derecha: las variables que hacen decrecer el *gini*, ya que el *gini* es invariante al punto de corte y por lo tanto más estable que el *accuracy*.

Las variables con mayor influencia son *price* y *m2*, precisamente las que anteriormente habiamos destacado por su heterocedasticidad. En un segundo lugar encontramos *zone* y *floor*, en las que habiamos encontrado cierta correlación de alguno de sus grupos con la variable respuesta. Y por último encontramos *rooms* y *garage*, en el caso de la primera el test de Levene ya advertía de su homocedasticidad mientras que la segunda apenas mostraba diferencia entre la correlación de sus valores con la variable respuesta.

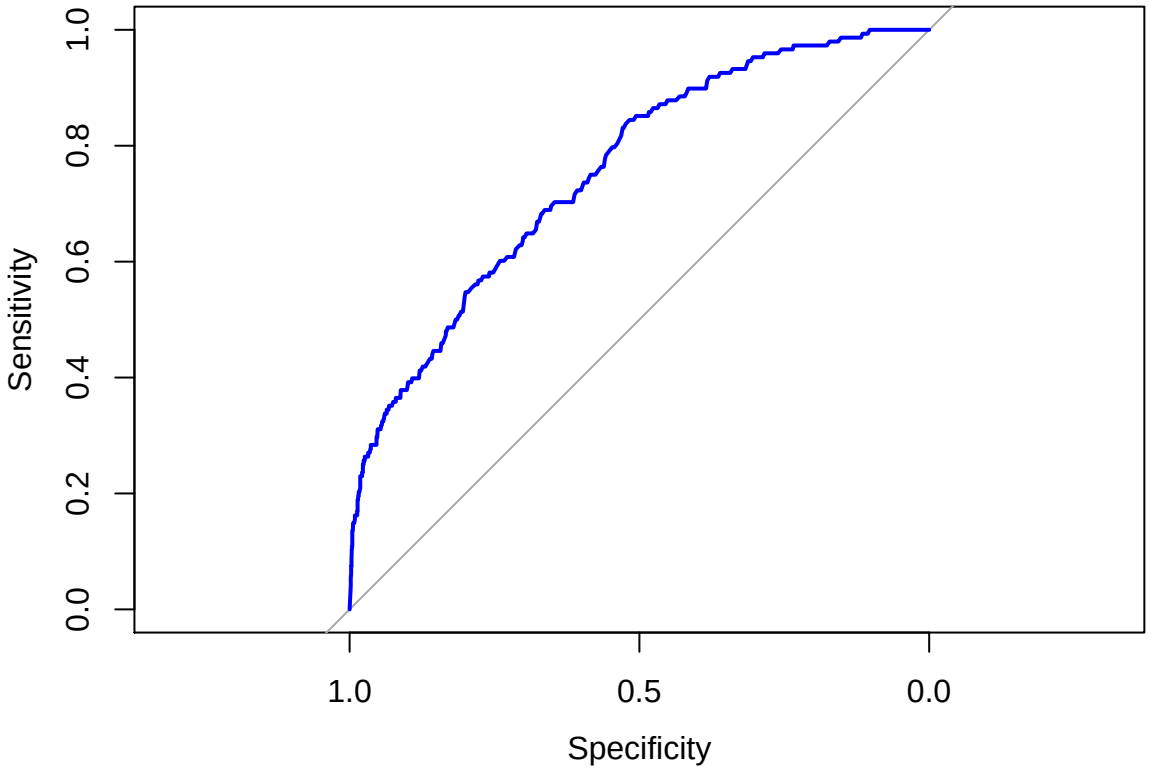
Si nos fijamos en la gráfica de la izquierda: las variables que más hacen decrecer el *accuracy* son *zone* y *floor*. Esto lo podriamos interpretar como que la información que proporcionen dichas variables será determinante para establecer que el inmueble no recibirá un descuento ya que la mayoría de las muestras de nuestra población no han recibido un descuento en su precio inicial y el *accuracy* se basa en muestras clasificadas correctamente.

Para acabar de validar si el resultado obtenido es bueno, vamos a calcular el *gini* del modelo a partir del conjunto de *test* que simularía un conjunto de datos real no usado para entrenar el modelo.

```
pred <- predict(rf, test, type = "prob")[,2]
roc_val <-roc(test$target, pred)
gini <- 2*auc(roc_val)-1
```

El modelo presenta un **Gini del 51%**. Por lo tanto se concluye, que es modelo con un buen poder discriminatorio y que clasifica bien. También lo podemos visualizar a partir de la curva ROC.

```
plot(roc_val,col='blue')
```



Por último, se presenta la población de *test* segmentada en deciles de probabilidad y el porcentaje de casos acertados para cada tramo.

```
kable(table.resume(discretize(pred, "interval", 10), test$target))
```

	y	N	var
[0.004,0.104)	1	172	0.0058140
[0.104,0.203)	5	233	0.0214592
[0.203,0.303)	12	304	0.0394737
[0.303,0.402)	26	324	0.0802469
[0.402,0.502)	29	280	0.1035714
[0.502,0.602)	19	174	0.1091954
[0.602,0.701)	14	91	0.1538462
[0.701,0.801)	15	52	0.2884615
[0.801,0.900)	7	20	0.3500000
[0.900,1.000]	20	28	0.7142857

De esta forma podriamos determinar de una forma coherente cual sería el punto de corte para considerar que el modelo está dando una respuesta positiva (el inmueble va a bajar de precio) o negativa (el inmueble no va a bajar de precio). Por ejemplo, si este punto de corte teórico lo situaramos en 0.9 podriamos decir que las respuestas positivas proporcionadas por el modelo tendrían un error del 29%, es decir, en este caso estamos minimizando el error tipo 1 (falso positivo).

6 Resolución del problema.

Cómo hemos podido observar, existe una clara relación entre los inmuebles que reciben un descuento sobre su precio inicial y sus características principales, tales como el precio, metros cuadrados, zona, habitaciones, planta (en caso de ser un piso), etc. En función de estas relaciones, a priori desconocidas, hemos podido generar un modelo *machine learning*. Debido a la cantidad de datos disponible, podemos concluir que el modelo propociona un resultado aceptable a la hora de clasificar **si un inmueble en venta en el municipio de Barcelona va a bajar de precio**. Por lo que se considera que se dá respuesta a la pregunta inicial planteada. Aunque, hay que tener en cuenta que si realizaramos este ejercicio con una cantidad de datos mayor seguramente mejoraría la calidad de la respuesta proporcionada.

Finalmente, exportamos los datos tratados junto a la respuesta del modelo (probabilidad de que el inmueble baje de precio).

```
df$prob <- predict(rf, df, type = "prob")[,2]
fwrite(df, "..\\data\\dataset_out.csv")
```

7 Código

Se adjunta el código empleado en la carpeta *code* y los datos necesarios en la carpeta *data*.

8 Referencias

[1] https://github.com/dripi/idealista-scraping