Universitat
de les Illes Balears

# Aprendizaje Estadístico y Toma de Decisiones

## Clasificación de Tumores según Datos de Mamografías

Maria del Mar Bibiloni

MADM, UIB
Curso 2017-2018

# Índice

Éste documento se complementa con el archivo R−markdown `TrabajoFinal_codigo_MBF.Rmd` y el HTML que genera. En ambos aparece todo el código usado para generar las gráficas, tablas y los resultados de los métodos de clasificación aplicados. Aquí sólo se presenta la parte del código que se ha considerado más relevante.

# 1. Sobre los datos

Los datos que se han analizado contienen información de mamografías de mujeres que tienen un tumor. Éstos datos se pueden descargar del repositorio UCI, aquí.

Tal como dice la documentación del data set, la información sobre mamografías de 961 mujeres ha sido recogida por el Instituto de Radiología University Erlangen-Nuremberg, entre los años 2003 y 2006. Además de los datos de las mamografías, también aparecen los resultados de biopsias realizadas a cada una de las pacientes, que determinan si la masa tumoral corresponde a un tumor benigno o maligno. Muchas de estas biopsias resultan innecesarias, ya que su resultado es que el tumor es benigno. Así, el objetivo es usar toda la información de la mamografía para predecir si el tumor es benigno o maligno, reduciendo el número de casos detectados cómo maligno que no lo sean [1].

Por tanto, el **objetivo** de éste trabajo es encontrar un modelo de predicción que consiga clasificar correctamente las masas tumorales, sin realizar la biopsia y con una alta precisión en detectar tumores malignos; todo esto sin descuidar la precisión con los casos benignos, ya que no detectar una masa tumoral maligna es peligroso.

Concretamente, las variables independientes y dependiente que conforman el data set son las siguientes.

- **BI-RADS assessment**. Variable cualitativa ordinal que toma valores en $\{1, 2, 3, 4, 5\}$. BI-RADS es una puntuación obtenida como resultado de una mamografía [2]. Según la puntuación se realiza la biopsia y, por tanto, no es suficiente. Ésto no quiere decir que no deba usarse para hacer la predicción, sino que los resultados BI-RADS, por ahora, no presentan la precisión buscada. A continuación, se describe cada puntuación.

  1. Negativo. No se ha encontrado ninguna masa sospechosa.
  2. Tumor Benigno.
  3. Tumor probablemente Benigno.
  4. Anormalidad sospechosa.
  5. Alta sospecha de maligno.

- **Age**. Variable cuantitativa con la edad de la paciente en años.
- **Shape**. Variable cualitativa nominal que representa la forma de la masa tumoral. Toma valores en $\{1, 2, 3, 4, 5\}$ con el siguiente significado.

  1. Redondo.
  2. Ovalado.
  3. Lobular.
  4. Irregular.

- **Margin**. Variable cualitativa nominal con valores en $\{1, 2, 3, 4, 5\}$. Representa el margen (borde) de la masa según las características que siguen.

    1. Circunscrito.
    2. Microlobulado.
    3. Oscurecido.
    4. Confuso.
    5. Espiculado.

- **Density**. Variable cualitativa ordinal con valores en $\{1, 2, 3, 4\}$, que representan la densidad del tumor según los siguientes niveles.

    1. Alta.
    2. *Iso*.
    3. Baja.
    4. Contiene grasa.

- **Severity**. Valiable cualitativa binomial. Es la variable dependiente de las anteriores y el resultado de la biopsia. Toma valores en $\{0, 1\}$, y como suele ser común, representan lo siguiente.

    0. Tumor Benigno.
    1. Tumor Maligno.

Una vez limpiado el data set, como se explica más adelante en la sección 2.1, cada variable queda distribuida como se muestra en la Tabla 1. Ésta distribución, también se puede observar en la última fila del pairplot en la Figura 1, dónde el color verde se identifica con los casos de tumor benigno y el azul con los casos de maligno. Según se observa no hay ninguna variable BI-RADS de tipo 1, ésto no es grave porque el 1 quiere decir que no se ha detectado ningún tumor y el objetivo es clasificar tumores. Por otra parte, hay muy pocas observaciones con BI-RADS 2 o 3, pero en estos casos no se realiza biopsia porque no hay sospecha de que el tumor sea maligno, por tanto, como el objetivo es predecir bien los tumores benignos que el BI-RADS clasifica como sospechosos tampoco es grave. De hecho, hay muchas observaciones en los casos que nos interesan: 4 y 5.

Por otra parte, notemos que hay un número parecido de casos de tumor benigno y casos de tumor maligno, por tanto, no se trata de un problema no balanceado (fijándonos sólo en ésta variable). Ésta proporción es importante para evitar que los métodos de clasificación generen un modelo con una predicción constante en un valor.

| BI-RADS | | | | | Shape | | | | Severity | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 0 | 1 |
| 0 | 7 | 24 | 468 | 316 | 189 | 178 | 79 | 369 | 423 | 392 |

| Margin | | | | | Density | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 |
| 318 | 23 | 101 | 249 | 124 | 11 | 55 | 741 | 8 |

**Tabla 1:** Distribución de las variables cualitativas del *Mammographic Mass Data Set*; después de limpiar los datos.

Ahora, si observamos la Figura 1, tenemos que la mayoría de observaciones con un 5 en BI-RADS son de una paciente con un tumor maligno, mientras que con un 4 son de una paciente con una masa benigna. Éste hecho motiva el estudio, ya que una anormalidad sospechosa puede llevar a una biopsia innecesaria si no se analizan también los demás resultados de la mamografía. Según las gráficas, también parece que la forma del tumor va a ser importante en la clasificación.

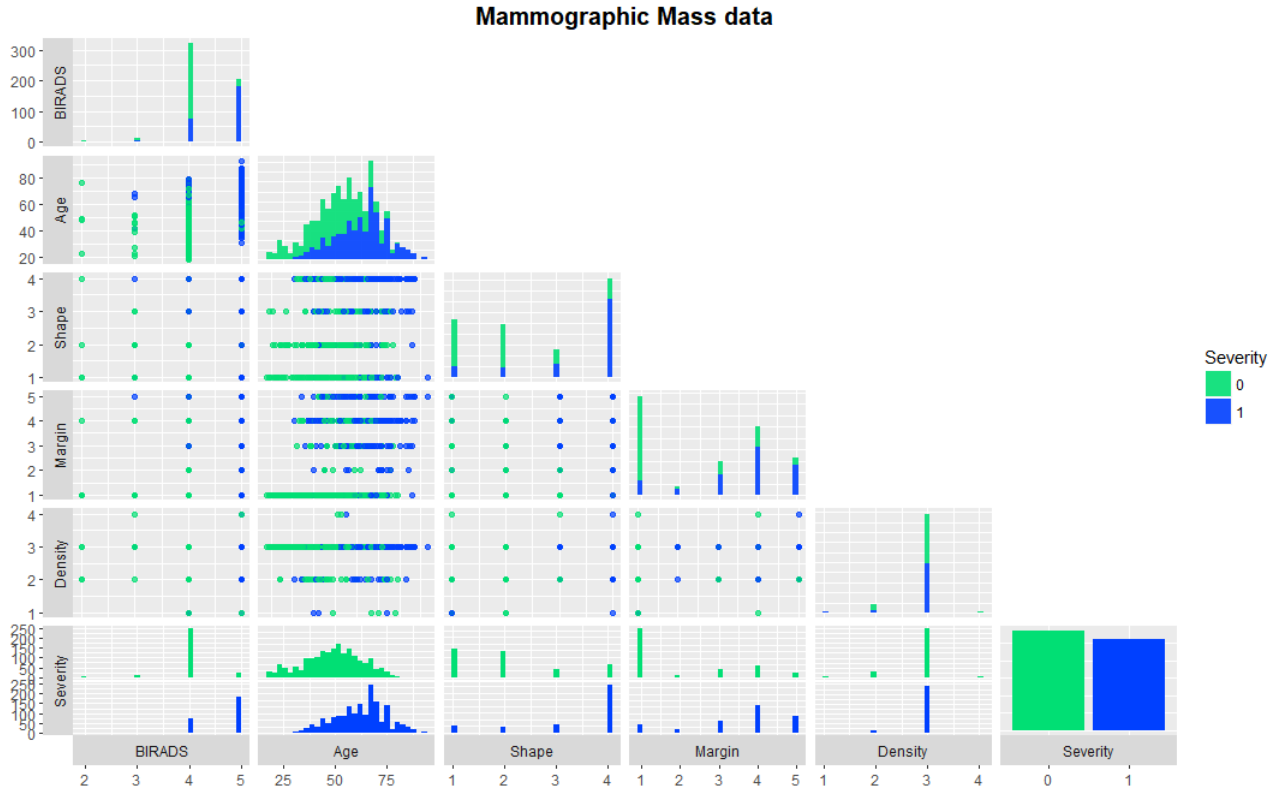


**Figura 1:** Pairplot de las variables del *Mammographic Mass Data Set* después de limpiar los datos; todas las variables menos Severity se han tratado como continuas para obtener los puntos sobre el plano.

## 2. Preparación del data set

Una vez conocemos cuál es el objetivo del estudio y como son los datos que vamos a analizar, el siguiente paso es limpiar el data set y prepararlo para poder aplicar los métodos de clasificación

que se escojan. Así, en esta sección se explican los pasos que se han seguido y parte del código en R que se ha usado.

## 2.1. Limpieza del data set

Después de leer el data set en R y nombrar las columnas, se han obtenido sus dimensiones: 6 predictores y 961 observaciones. Algunas de estas observaciones presentaban caracteres vacíos o información que no se correspondía con la documentación del data set. Concretamente, los valores en las columnas estaban originalmente distribuidos como en la tabla 2.

Lo primero que destaca es un valor BI-RADS 55. Problablemente se trate de un error tipográfico y sea un 5, pero aún así se ha decidido eliminar ésta observación para no contaminar la muestra con un posible error. Lo mismo ocurre con los valores 6 y 0, en este caso porque en la documentación se indica que BI-RADS es una variable con valores en $\{1, 2, 3, 4, 5\}$. En la clasificación BI-RADS real si existen estos valores, un 0 indica un valor incompleto, es decir, que los resultados no son concluyentes, mientras que un 6 indica que se ha realizado una biopsia y se ha demostrado que es maligno [2]. Sin embargo, si observamos las observaciones de esta categoría encontramos casos de tumor benigno en la biopsia.

```
data[which(data$BIRADS==6),]
```

```
     BIRADS  Age  Shape  Margin  Density  Severity
551       6   80      4       5        3         1
665       6   60      3       5        3         1
693       6   51      4       4        3         1
708       6   41      2       1        3         0
718       6   71      4       4        3         1
719       6   68      4       3        3         1
746       6   76      3       ?        3         0
785       6   63      1       1        3         0
825       6   40      ?       3        4         1
833       6   72      4       3        3         1
890       6   41      3       3        2         1
```

Así, estas dudas deberían de ser contrastadas para decidir si deben permanecer o no en el data set, o para saber cómo modificarlas, pero ésto no ha sido posible. Por tanto, también se han eliminado las filas con BI-RADS 0 o 6 para evitar información errónea.

La siguiente modificación ha sido eliminar todas las observaciones con algún valor nulo en alguna de las columnas. Otra opción podría haber sido sustituir éstos valores por la media o mediana, por ejemplo, pero al ser variables cualitativas sobre características de masas tumorales, menos Age, se ha descartado esta opción.

Finalmente, R reconoce los valores en todas las columnas como factores menos Severity, que detecta como valores numéricos. Por esto, inicialmente se ha modificado el conjunto de datos para que todos los valores sean numéricos menos los de la variable dependiente, Severity. El

| BI-RADS | | | | | | | | | Shape | | | | | Severity | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ? | 0 | 1 | 2 | 3 | 4 | 5 | 55 | 6 | ? | 1 | 2 | 3 | 4 | 0 | 1 |
| 2 | 5 | 0 | 14 | 36 | 547 | 345 | 1 | 11 | 31 | 224 | 211 | 95 | 400 | 516 | 445 |

| Margin | | | | | | Density | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ? | 1 | 2 | 3 | 4 | 5 | ? | 1 | 2 | 3 | 4 |
| 48 | 357 | 24 | 116 | 280 | 136 | 76 | 16 | 59 | 798 | 12 |

**Tabla 2:** Distribución de las variables cualitativas del *Mammographic Mass Data Set*; antes de limpiar los datos.

código utilizado es el que sigue a continuación. Cabe destacar que en algunos métodos de clasificación que se han usado, se ha tenido que modificar de nuevo ésta característica.

```
data <- as.data.frame(apply(data, 2, as.numeric))
data$Severity <- as.factor(data$Severity)
```

## 2.2. Conjuntos de Train y Test

El siguiente paso para preparar los datos es tomar un conjunto de validación, test, que no se va a usar para entrenar ninguno de los modelos. Así, el objetivo de este conjunto es simular el error fuera de la muestra, que se comete al hacer una predicción. Además, también se consigue reducir el overfitting; ya que un modelo muy ajustado puede generar un error de validación elevado.

En este caso, se ha decidido dividir aleatóriamente la muestra en dos conjuntos; el conjunto de entrenamiento con 2/3 de observaciones y el de testeo con el 1/3 restante. Para hacerlo se han usado las siguientes instrucciones.

```
set.seed(1123)
train=sample(nrow(data), size = ceiling(2/3*nrow(data)) )
test=c(1:nrow(data))[-train]
```

Notemos que no se ha impuesto que se mantenga la proporción de casos de tumor maligno y benigno originales en los conjuntos y, por tanto, uno de los sets podría resultar no balanceado. Así, se ha obtenido la Tabla 3 , dónde se observa que se mantienen las proporciones, aproximadamente, y no es necesario cambiar los conjuntos.

## 2.3. Variables Dummy

Uno de los métodos que se han aplicado para crear un modelo de clasificación de las masas tumorales es la regresión logística. Dado que la mayoría de variables independientes son cualitativas, se ha creado un nuevo conjunto de datos, `data_dum`, con variables dummy para cada una de ellas: BI-RADS, Shape, Margin y Density. Para hacerlo, se ha usado la librería **dummies**

| | Benign (0) | Malign (1) |
|---|---|---|
| Original set | 0.5369407 | 0.4630593 |
| Train set | 0.5202206 | 0.4797794 |
| Test set | 0.5166052 | 0.4833948 |

**Tabla 3:** Proporción de observaciones de tumor maligno y benigno en el *Mammographic Mass Data Set*; antes de limpiar los datos (Original set), en el conjunto de entrenamiento (Train set) y de validación (Test set).

de R tal como sigue.

```
library("dummies")
data_dum <-dummy.data.frame(data, names=c("BIRADS","Shape", "
   Margin", "Density"), sep="_")
```

La instrucción `dummy.data.frame()` crea un data frame con una columna de valores en $\{0, 1\}$ para cada una de las características de cada predictor; 1 presenta la característica, 0 no. Así, en este caso tenemos las siguientes columnas: `BIRADS_2, BIRADS_3, BIRADS_4, BIRADS_5, Age, ,Shape_1, Shape_2, Shape_3, Shape_4, Margin_1, Margin_2, Margin_3, Margin_4, Margin_5, Density_1, Density_2, Density_3, Density_4, Severity.`

Una vez llegados a este punto, nos podemos preguntar ¿qué significa un cero en todas las dummies? Que una paciente no presenta ninguna característica y esto, de hecho, no ocurre. Así, se debe establecer una categoria referencia para cada predictor original, que vendría representado por "tener un cero en todas las dummies de ésta predictora". Un ejemplo lo tenemos en la variable dependiente, Severity, ya que no hay una variable *benigno* y una variable *maligno*, sino una única variable donde el cero indica un tumor benigno. En este caso, la categoría benigno sería la categoría referencia [3].

Para este problema, se han seleccionado las siguientes categorías referencia, por los motivos que se expone; aunque no es la única elección posible.

- **BI-RADS_2**. Es una variable cualitativa ordinal, por lo que se toma el valor *menor*: 2-tumor clasificado como benigno por la mamografía.
- **Shape_4**. Es una variable cualitativa nominal. En este caso, tomamos el valor más común: 4-irregular.
- **Margin_1**. Variable cualitativa nominal, tomamos el más común: 1-cincunscrito.
- **Density_3**. Variable cualitativa ordinal. Tomamos la categoría que indica una densidad menor: 3-bajo.

Así, eliminamos las categorías referencia con las siguientes instrucciones.

```
refCategory=names(data_dum) %in% c("BIRADS_2", "Shape_4", "Margin_
   1", "Density_3")
data_dum = data_dum[, !refCategory]
```

Finalmente, tenemos un data set con las siguientes variables dummies: `BIRADS_3, BIRADS_4, BIRADS_5, Age, ,Shape_1, Shape_2, Shape_3, Margin_2, Margin_3, Margin_4, Margin_5, Density_1, Density_2, Density_4, Severity.`

# 3. Métodos de clasificación

Una vez tenemos los datos preparados, vamos a aplicar varios métodos de clasificación para crear diferentes modelos que permitan hacer predicciones sobre futuras mamografías. Primero, probaremos dos métodos simples de clasificación: K-nearest neighbors y regresión logística. Estos métodos tienen la ventaja de ser fáciles de interpretar, como los árboles de clasificación, que también aplicaremos. Otros métodos que se van a usar para crear un modelo son Random Forests y Boosting.

Los pasos a seguir para cada uno de los métodos serán: aplicar validación cruzada en train para determinar los mejores parámetros del método (si tiene), usar esos parámetros para crear un modelo y usar el conjunto de test para calcular el error que se comete en datos con los que no se ha entrenado el modelo. Una vez calculado el error de todos los métodos, en el mismo conjunto de validación, vamos a comparar los errores y escoger el mejor o mejores modelos según éste. Ya que se trata de un problema de clasificación, vamos a usar *accuracy* para medir el error en cada modelo, es decir,

$$error = 1 - accuracy = 1 - \frac{\text{Número de aciertos}}{\text{Total}}.$$

## 3.1. K-nearest neighbors

El método de K-nearest neighbors decide como se clasifica una observación tomando la clase más común entre sus k-vecinos. En otras palabras, el método busca k pacientes con características similares al tumor que se quiere clasificar y predice si es benigno o maligno en función de si lo es para la mayoría de las k pacientes. Así, la elección de K afecta directamente al resultado de la predicción.

Para elegir K correctamente hacemos validación cruzada 10-veces en train con el siguiente código en R, que nos devuelve el mejor parámetro según *accuracy*. Indicamos `tuneLength = 30` para que pruebe con 30 valores de K.

```
library(ISLR)
library(caret)
set.seed(5813)
cross_val <- trainControl(method="cv", number=10)
knn_fit <- train(Severity ~ ., data = data[train,], method = "knn"
   , metric="Accuracy",  trControl = cross_val, tuneLength = 30)
```

El modelo escogido por la función `train()` es K= 31, con *accuracy* = 0,7698990, pero el segundo mejor valor de precisión es *accuracy* = 0,7645791 para K= 5. Dada la poca diferencia y que un valor de K pequeño ayuda a detectar *zonas* pequeñas con un tipo de clasificación, es decir, un número reducido de observaciones con características similares y una clasificación definida,

tomamos los dos modelos como buenos y comparamos su error para escoger el mejor.

Para aplicar knn y predecir las clasificaciones de las observaciones en test se pueden ejecutar las siguientes instrucciones. En este ejemplo, se ha tomado k=31. La cuarta línea de código crea una tabla con número de clasificaciones correctas e incorrectas de cada una de las clases, por tanto, se ha usado para calcular las tasas de error y precisión de la Tabla 4. El código para k=5 es análogo.

```
library(class)
library('gmodels')
knn_k31_fit=knn(data[train,], data[test,], cl=data[train,]$
   Severity, k = 31)
ct_knn=CrossTable(x=data[test,]$Severity, y=knn_k31_fit)
```

Si nos fijamos en la columna de error de la Tabla 4, tenemos que el modelo con K= 5 es mejor. Además, tiene una alta tasa de aciertos de tumores benignos, hecho que es importante porque quiere decir que hay pocas masas malignas que no se detectan, un hecho importante, ya que tratamos con datos de tumores. Y no sólo eso, se consigue el objetivo del estudio, que es encontrar un modelo que reduzca el número de biopsias innecesarias, ya la precisión en los casos malignos es muy alta. En otras palabras, hay un número reducido de tumores benignos que de declaran como malignos.

| | Error | Accuracy | Accuracy-Benign | Accuracy-Malign |
|---|---|---|---|---|
| Knn (K=31) | 0.1845018 | 0.8154982 | 0.7500000 | 0.8854962 |
| Knn (K=5) | 0.1143911 | 0.8856089 | 0.8571429 | 0.9160305 |

**Tabla 4:** Error y precisión cometidos por el método de K-nearest neighbors para k=31 y k=5 en el conjunto de validación, en total y para los casos de tumor benigno y maligno del *Mammographic Mass Data Set*.

Ahora, veamos si podemos mejorar la precisión usando otros métodos de clasificación.

## 3.2. Logistic regression

El siguiente método que vamos a probar es el método de regresión logística, que consiste en suponer linealidad del logaritmo de odds y aplicar el método de máxima verosimilitud para estimar los coeficientes.

Para aplicar el método de regresión logística es necesario usar el conjunto de datos con las variables dummy que se ha creado en la sección 2.3. Así, basta aplicar la función `lm()` de R tal como sigue para estimar el modelo.

```
logReg <- glm(Severity ~.,family=binomial(link='logit'),data=data_
   dum[train,])
logReg$coefficients
```

```
##  (Intercept)      BIRADS_3      BIRADS_4      BIRADS_5           Age
## -17.83915110  14.98758698  14.18573252  16.47738735 0.05608061
##      Shape_1       Shape_2       Shape_3      Margin_2    Margin_3
##   -1.19351332  -1.40851600  -0.76915994    0.70441571 0.27308474
##     Margin_4      Margin_5     Density_1     Density_2  Density_4
##    0.70573668   0.52628792   0.23960310  -0.27047293 -1.98147844
```

Los coeficientes positivos indican un aumento en la probabilidad de que el tumor sea maligno si se da la característica correspondiente en lugar de la referencia (y las de fuera del grupo se mantienen constantes). Así mismo, los coeficientes negativos se asocian con el aumento de la probabilidad de que la masa tumoral sea benigna.

Para cada uno de los coeficientes $\beta_i$, podemos realizar el siguiente contraste de significación

$$\begin{cases} H_0: & \beta_i = 0, \\ H_1: & \beta_i \neq 0, \end{cases}$$

al nivel de significación $\alpha = 0{,}05$. En R el código utilizado es el siguiente.

```
summary(logReg)
```

```
[...]
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -17.83915  639.30081  -0.028 0.977739
## BIRADS_3     14.98759  639.30092   0.023 0.981296
## BIRADS_4     14.18573  639.30046   0.022 0.982297
## BIRADS_5     16.47739  639.30050   0.026 0.979438
## Age           0.05608    0.01089   5.149 2.62e-07 ***
## Shape_1      -1.19351    0.43498  -2.744 0.006073 **
## Shape_2      -1.40852    0.39517  -3.564 0.000365 ***
## Shape_3      -0.76916    0.39998  -1.923 0.054478 .
## Margin_2      0.70442    0.71611   0.984 0.325279
## Margin_3      0.27308    0.46315   0.590 0.555445
## Margin_4      0.70574    0.39507   1.786 0.074038 .
## Margin_5      0.52629    0.50684   1.038 0.299100
## Density_1     0.23960    0.94960   0.252 0.800793
## Density_2    -0.27047    0.47646  -0.568 0.570257
## Density_4    -1.98148    1.41175  -1.404 0.160450
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
[...]
```

De esta manera, se obtiene que ninguna de las variables dummy de los grupos Margin, Density y BI-RADS es significativa al nivel de significación escogido. Un motivo podría ser que algún grupo de dummies esté altamente relacionado con otro, por tanto, se ha realizado la matriz de correlaciones de los datos (sin dummies) de la Figura 2.

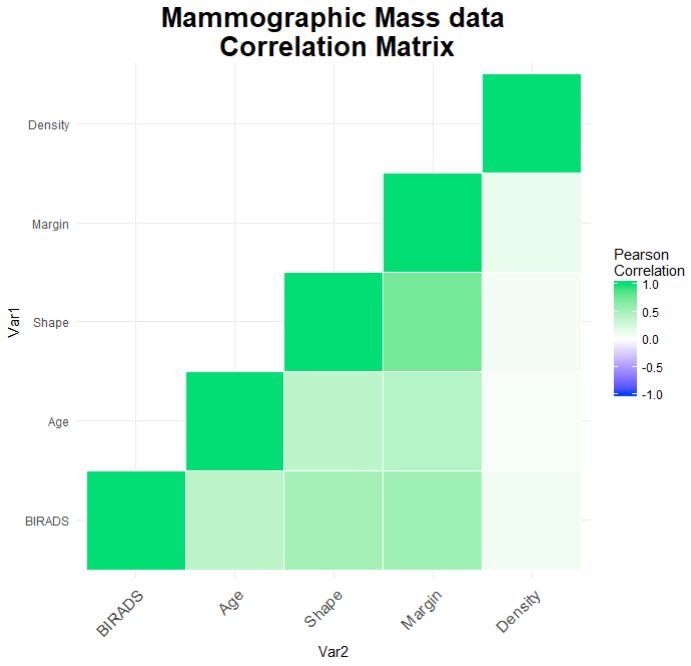


**Figura 2:** Matriz de correlaciones de las variables independientes del *Mammographic Mass Data Set* después de limpiar los datos.

La variable Density tiene una correlación muy baja con todas las demás variables, pot tanto, la mantenemos en el modelo. En cuanto a las variables BIRADS y Margin, tomamos las variables BIRADS y las eliminamos del modelo, con el fin de ver si es mejor.

Del mismo modo que al apartando anterior, estimamos los coeficientes mediante la regresión logística y contrastamos la significancia de cada uno de los coeficientes al mismo nivel de significación, $\alpha = 0{,}05$. En este caso, todas las variables son significativas menos density (incluso Margin). De esta manera, queremos comparar los dos modelos para ver cuál explica mejor nuestros datos. Para hacerlo, vamos a hacer el contraste *Likelihood ratio test* [4]. Tenemos,

$$\begin{cases} H_0: & \text{El modelo sin variables BI-RADS es verdadero,} \\ H_1: & \text{El modelo completo es verdadero.} \end{cases}$$

En R se puede realizar este contraste con la función `anova`, especificando el contraste que realizamos: `test="LRT"`.

```
anova(logReg_notBR, logReg, test="LRT")
```

El p-valor obtenido es $1{,}024 \cdot 10^{-14}$, por tanto, rechazamos la hipótesis nula y nos quedamos con el modelo con todas las variables dummy. Además, si comparamos el error de predicción en

el conjunto de validación, es menor para el modelo completo. Estos resultados aparecen en la Tabla 5 y se han calculado ejecutando las siguientes instrucciones para obtener una tabla con las clasificaciones correctas e incorrectas en la predicción; tal como se ha hecho en la sección anterior.

```
logReg_fit_0 <- predict(logReg,newdata=data_dum[test,-15], type='
   response')
#>=0.5 --> 1; <0.5 --> 0
logReg_fit_0  <- ifelse(logReg_fit_0  >= 0.5,1,0)
ct_logReg_0=CrossTable(x=data[test,]$Severity, y=logReg_fit_0)
```

Notemos en el código anterior, que `predict()` devuelve la probabilidad de que cada una de las observaciones en train pertenezca a la classe 1 (maligno). Así, se ha impuesto que si la probabilidad es mayor o igual que 0,5, entonces el tumor se clasifica como maligno, en caso contrario se clasifica como benigno.

| | Error | Accuracy | Accuracy-Benign | Accuracy-Malign |
|---|---|---|---|---|
| Log. Reg. (All) | 0.1439114 | 0.8560886 | 0.8214286 | 0.8931298 |
| Log. Reg. (Not BR) | 0.1660517 | 0.8339483 | 0.7714286 | 0.9007634 |

**Tabla 5:** Error y precisión del método de represión logística en el conjunto de validación, con todas las variables dummy en el modelo (Log. Reg. (All) ) y sin las variables BIRADS (Log. Reg. (Not BR) ). Valores totales y para los casos de tumor benigno y maligno del *Mammographic Mass Data Set*.

En la Tabla 5 también cabe destacar que la precisión de los casos benignos es menor para el modelo sin la variable BIRADS, mientras que para los casos malignos es muy similar. Éste hecho, hace que no quepa duda de que es mejor usar el modelo completo, ya que detectar menos casos de tumor maligno pone en riesgo la salud de las pacientes. En otras palabras, si tenemos dos modelos con una tasa similar de falsos positivos, nuestro problema requiere tomar el modelo con menos falsos negativos para hacer predicciones.

## 3.3. Árboles de clasificación

Los árboles de clasificación son uno de los modelos más simples de interpretar. El método para crear los árboles se basa en *cortar* el espacio en varias regiones simples y la predicción se hace según la clasificación predominante en cada región. Una vez el método ha escogido la mejor división del espacio, el resultado se devuelve en forma de árbol. Más adelante veremos algún ejemplo.

Los métodos basados en árboles no requieren usar variables dummies, por tanto, se pueden aplicar al data set con 5 predictores. El parámetro que afecta al resultado es la profundidad, es decir, el punto donde *podamos* en árbol. Demasiada profundidad puede llevar a overfitting i por eso se debe estudiar dónde acaba el árbol. Aún así, se ha querido ver el árbol que genera R por defecto. A continuación se presenta el código en R que lo genera y en la Figura 3 su representación.

```
require(ISLR)
require(tree)
tree=tree(Severity ~. ,data=data[train,])
```

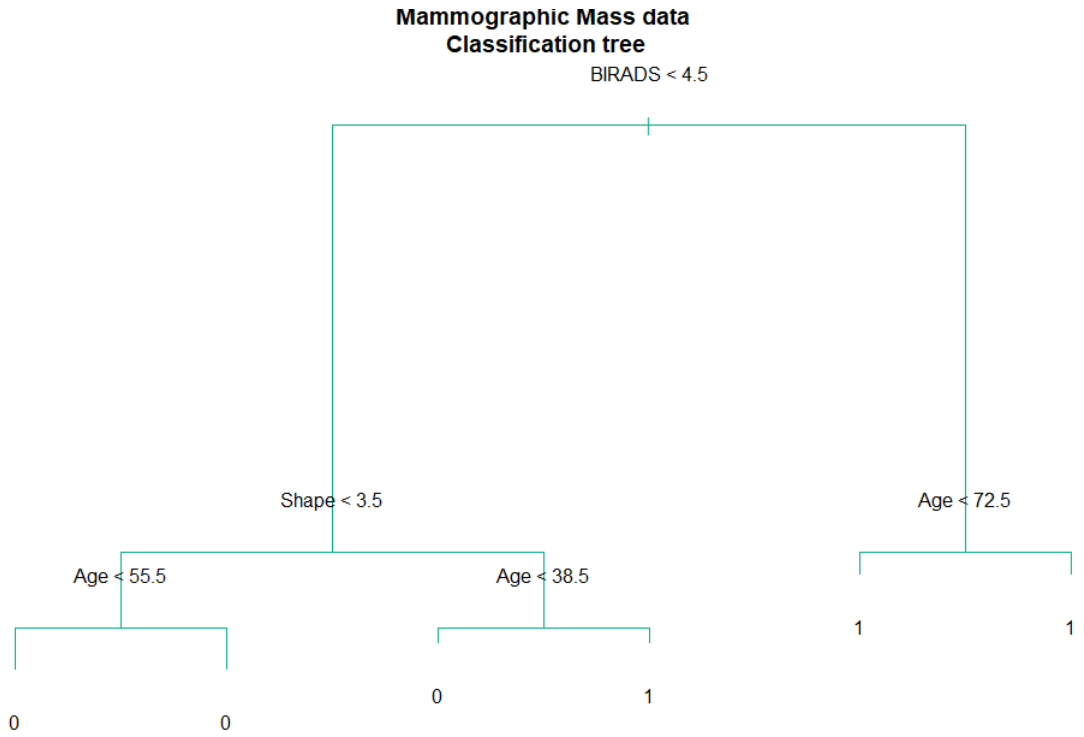


**Figura 3:** Árbol de clasificación del *Mammographic Mass Data Set*; después de limpiar los datos.

Como se ha comentado anteriormente, un árbol ayuda a la interpretación. En este caso, en el primer corte tenemos que si BIRADS es 5, el tumor se clasifica como maligno. Éste hecho es coherente con la intuición: Si BIRADS es 5, hay una alta sospecha de que la masa sea maligna, por tanto, la biopsia es necesaria. En caso contrario, se analizan las demás características.

Por otra parte, es interesante notar que el árbol no usa las variables Margin ni Density para definir las regiones. De hecho el árbol resultante tiene muy poca profundidad. Ahora, aplicamos el método al conjunto de entrenamiento tal como sigue.

```
set.seed(555)
tree_cv=cv.tree(tree,FUN=prune.misclass, K=10)
```

El resultado de validación cruzada 10 veces aparece en la Figura 4, que sugiere una profundidad de $d = 3$, aunque la diferencia es poca comparada con el árbol completo.

Así, tomamos 3 de profundidad y calculamos el error en test. Los resultados aparecen en la Tabla 6 y, a pesar de la ventaja en la interpretación, el error es mayor que en los anteriores

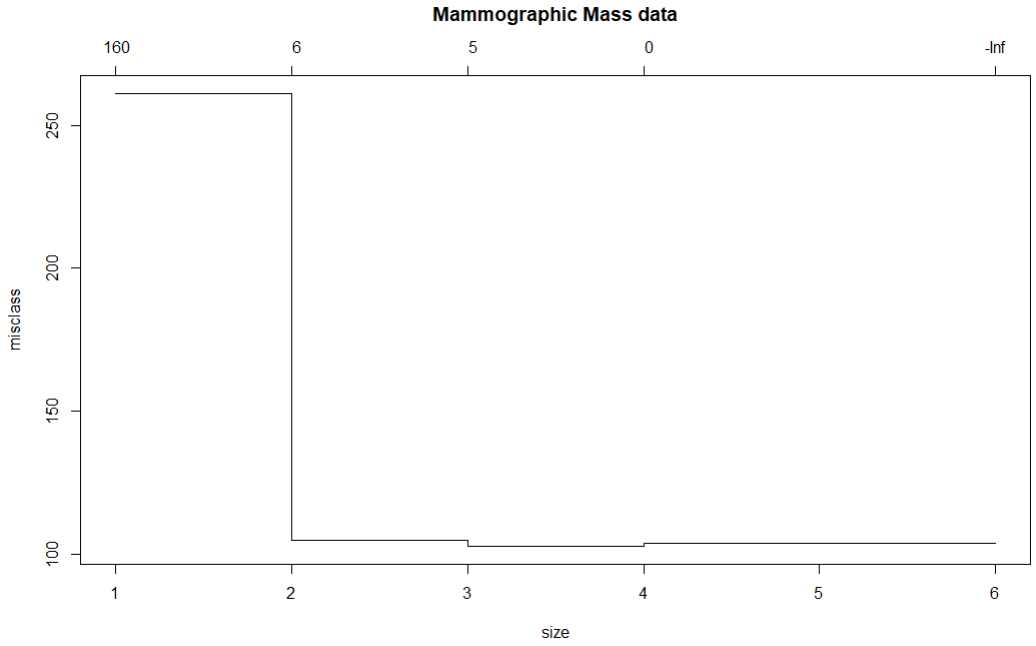


**Figura 4:** Clasificaciones erroneas cometidas por un árbol de clasificación en función de la profundidad. Datos: *Mammographic Mass Data Set*, después de la limpieza.

modelos. Veremos que ocurre con los siguientes.

Para podar el árbol usamos la siguiente instrucción, cuyo resultado aparece en la Figura 5.

```
prune_tree=prune.misclass(tree,best=3)
```

Para calcular la precisión y error en el conjunto de validación, también se ha obtenido una tabla con las clasificaciones correctas e incorrectas, pero en este caso el código usado es el siguiente.

```
tree_fit=predict(prune_tree,data[test,],type="class")
ct_tree=with(data[test,],table(tree_fit,Severity))
```

Aunque el error total en la Tabla 6 no sea tan pequeño como en otros métodos, podría ser un modelo útil si sólo buscáramos no hacer biopsias innecesarias, ya que la precisión para los casos malignos es muy alta. Aún así, para nuestro problema los casos benignos mal clasificados son un problema de igual importancia.

## 3.4. Random Forests

Con el objetivo de mejorar los resultados obtenidos con los métodos anteriores, vamos a aplicar los siguientes métodos basados en árboles: Random Forests y Boosting. En esta sección nos centraremos en árboles aleatorios.

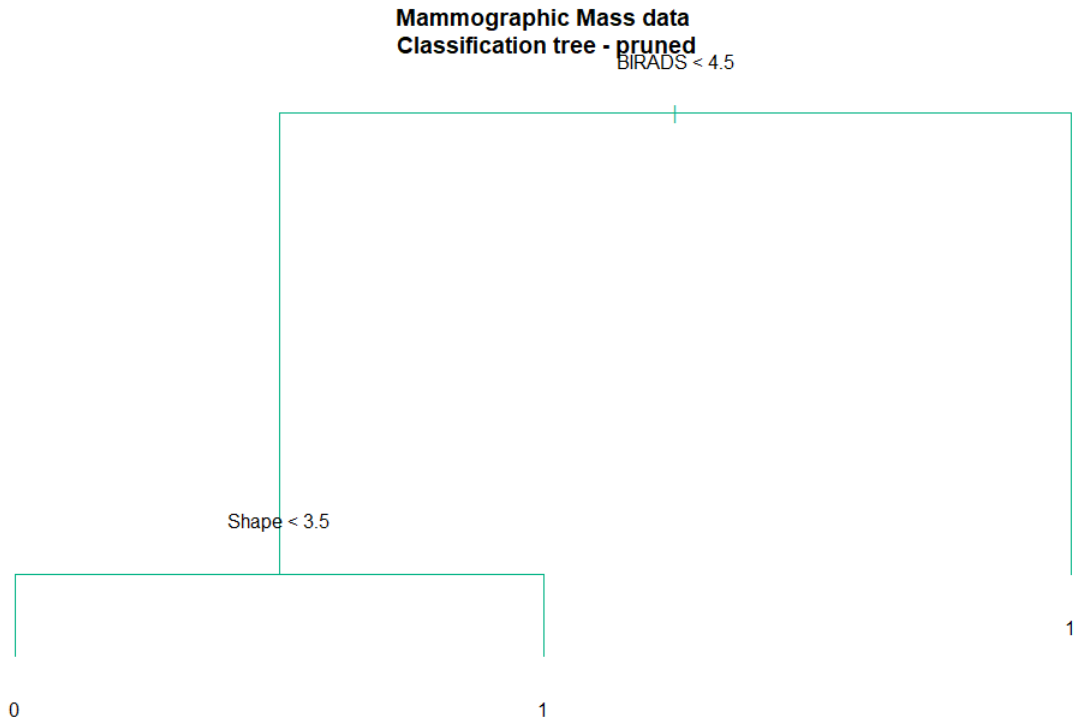


**Figura 5:** Árbol de clasificación con profundidad 3 del *Mammographic Mass Data Set*; después de limpiar los datos.

| | Error | Accuracy | Accuracy-Benign | Accuracy-Malign |
|---|---|---|---|---|
| Tree (d=3) | 0.1439114 | 0.8560886 | 0.7642857 | 0.9541985 |

**Tabla 6:** Error y precisión del árbol de clasificación de profundidad 3 en el conjunto de validación; valores totales y para los casos de tumor benigno y maligno del *Mammographic Mass Data Set*.

El método de Random Forests crea B muestras de Bootstrap. y calcula un árbol de clasificación para cada una de las muestras. Además, cada división en el árbol tiene en cuenta sólo un subconjunto de $m$ de variables del conjunto de predictores original, que se escogen aleatoriamente a cada paso.

Un valor típico de m es $\sqrt{p}$, donde $p$ es el número de predictores. Éste es el valor que utiliza R por defecto y, por tanto, hemos querido observar los resultados con éste valor. Un parámetro que sí se ha indicado es el numero B de muestras bootstrap, `ntree=1000`. El código usado es el siguiente.

```
set.seed(1515)
rForest=randomForest(Severity~.,data=data[train,],ntree=1000)
```

Tal como se ha hecho en la sección anterior, se han usado las funciones `predict()` y `table()`

para calcular el error de validación. Luego, se ha querido contrastar éste error con el que se obtiene usando los parámetros obtenidos de realizar validación cruzada 10-veces en train.

R nos permite introducir los diferentes modelos que queremos comparar, es decir, los diferentes parámetros. En nuestro caso, especificamos que pruebe con tomar $1, 2, 3, 4$ y $5$ variables a cada paso (*mtry*), tal como sigue.

```
set.seed(1717)
cross_val_randForest<- trainControl(method="cv", number=10)

df_mtry=as.data.frame(matrix(c(1,2,3,4,5), nrow=5, ncol=1))
names(df_mtry)=c("mtry")

randForest_fit_cv <- train(Severity ~ ., data = data[train,],
   method = "rf", metric="Accuracy", trControl = cross_val_
   randForest, tuneGrid = df_mtry )
```

El valor escogido comparando los valores de accuracy es 1. Por tanto, aplicamos el método de Random Forests con `mtry=1` y `ntree=1000`, y calculamos los errores de predicción en test. Los resultados de los errores y precisión para cada uno de los modelos de Random Forests aparecen en la tabla 7. Como se puede observar la diferencia no es significativa, por lo que es mejor tomar `mtry=2`. Ésta poca diferencia de debe a que en validación cruzada la precisión (*accuracy*) obtenida era similar en ambos casos: 0.8179125 para `mtry=1` y 0.8031987 para `mtry=1`.

| | Error | Accuracy | Accuracy-Benign | Accuracy-Malign |
|---|---|---|---|---|
| Random Forests (mtry=2) | 0.1328413 | 0.8671587 | 0.8285714 | 0.9083969 |
| Random Forests (mtry=1) | 0.1365314 | 0.8634686 | 0.8071429 | 0.9236641 |

**Tabla 7:** Error y precisión del método Random Forests en el conjunto de validación, tomando dos variable aleatoriamente a cada paso (mtry=1) y una (mtry=2). Valores totales y para los casos de tumor benigno y maligno del *Mammographic Mass Data Set*.

## 3.5. Boosting

El método de Boosting, al igual que Random Forests, se basta en crear B muestras de bootstrap y crear un árbol de clasificación según cada una de ellas. La diferencia principal es que ahora no se eliminan variables a cada paso y que cada árbol se crea usando información del creado anteriormente; sumando el antiguo árbol al nuevo por un parámetro $\lambda$. Éste parámetro se conoce por parámetro de *shrinkage* y hace que el método resuelva mejor zonas conflictivas (pequeño). Otros parámetros del método son el número de muestras bootstrap (`n.trees` en R), la profundidad de cada árbol (`interaction.depth`) y el número mínimo de observaciones en los nodos finales (`n.minobsinnode`), es decir, sirve para evitar que tome muy pocas observaciones por nodo, una incluso, y se llegue a sobreajustar el modelo. A continuación, se expone el código para realizar validación cruzada 10-veces en train para encontrar los mejores parámetros. Las primeras filas son para cambiar la variable Severity como factor y tomarla como caracter, ya que el método en R lo requiere.

```
#hay problemas con as.factor y gbm
data_gbm=data
data_gbm$Severity = as.character(data_gbm$Severity)

set.seed(8799)
cross_val_boost <- trainControl(method="cv", number=10)
boost_fit <- train(Severity ~ ., data = data_gbm[train,], method =
    "gbm", distribution = "bernoulli", metric="Accuracy", trControl
    = cross_val_boost, tuneLength = 10)
boost_fit$bestTune
```

El resultado ha sido `n.trees`=50, `interaction.depth`=2, `shrinkage`=0.1 y `n.minobsinnode`=10. Así, aplicamos el método de Boosting con estos parámetros y predecimos las clasificaciones que hace en test tal comos sigue.

```
set.seed(3573)
boost = gbm(Severity~. , distribution = "bernoulli", data = data_
   gbm[train,],
               n.trees = 50, shrinkage = 0.1, interaction.depth =
                  2, n.minobsinnode = 10)
boost_pred = predict(object = boost, newdata = data_gbm[test,], n.
   trees = 50, type = "response")
boost_pred   <- ifelse(boost_pred  >= 0.5,1,0)
ct_boost=with(data[test,],table(boost_pred,Severity))
```

En este caso, la función `train()` devuelve, de nuevo, la probabilidad de que cada observación sea de la classe tumor maligno. Así, se ha clasificado como maligno si ésta supera o iguala el 0.5 y como benigno en caso contrario. Calculamos la tabla de clasificaciones correctas e incorrectas y obtenemos los valores de la Tabla 8.

| | Error | Accuracy | Accuracy-Benign | Accuracy-Malign |
|---|---|---|---|---|
| Boosting (nt=50, d=2, shr=0.1, n.mino=10) | 0.1217712 | 0.8782288 | 0.8642857 | 0.8931298 |

**Tabla 8:** Error y precisión del método Random Forests en el conjunto de validación, tomando 50 muestras bootstrap (nt), la profundidad de cada árbol (d), el parámetro de contracción $\lambda = 0{,}1$ (shr) y el número mínimo de observaciones en los nodos finales (n.mino). Valores totales y para los casos de tumor benigno y maligno del *Mammographic Mass Data Set*.

Los resultados de la Tabla 8 son muy buenos, ya que el error total es pequeño y, además, hay una precisión alta tanto en los casos de tumor benigno como de tumor maligno.

## 3.6. Clustering para clasificación

En esta sección se ha pretendido utilizar el algoritmo de K-means sobre los datos, aunque es un método para aprendizaje no supervisado. La idea era ver si éstos se distribuyen de forma

natural en el espacio según tumores benignos y malignos. Así, aplicamos el algoritmo a todos los datos, sin la columna Severiry y especificando que buscamos dos clases, `centers=2`. Además se han escalado las variables, aunque éstas sean cualitativas.

Por otra parte, aunque tenemos la clasificación real de todas las observaciones, tomamos sólo las de test para poder comparar los errores con los de los otros métodos. A continuación tenemos el códido y en la Tabla 9 el error y la precisión.

```
data_clust=data
data_clust[,-6]=apply(data[,-6], 2, function(x) scale(x))

set.seed(2222)
kmeans_sc = kmeans(data_clust[,-6], centers=2)
ct_kmeans=table(kmeans_sc$cluster[test], data_clust[test,]$
   Severity)
```

Tal como cabía esperar el error total en la Tabla 9 no es bueno. Sí sorprende la precisión en detectar casos malignos. Según el problema que se estuviera atacando, se podría usar k-means y ver en que cluster cae una nueva observación para predecir su clasificación. En nuestro caso, hay métodos que nos sirven mejor.

| | Error | Accuracy | Accuracy-Benign | Accuracy-Malign |
|---|---|---|---|---|
| kmeans (scaled) | 0.1549815 | 0.8450185 | 0.75 | 0.9465649 |

**Tabla 9:** Error y precisión de aplicar el método de k-means para crear dos clusters e identificarlos con las clases tumor benigno y maligno. Valores totales y para los casos de tumor benigno y maligno del *Mammographic Mass Data Set*. El algoritmo se ha aplicado a todo el data set menos la variable dependiente y los errores en el conjunto de validación.

## 3.7. Comparación

En esta sección se han recogido todos los errores y precisión de los métodos estudiados en una única tabla, con el objetivo de poder comparar todos los métodos y escoger los mejores para nuestro problema. La Tabla 10 contiene todos éstos valores.

Según el error y precisión totales, los mejores métodos són K-nearest neighbors con 5 vecinos y Boosting, con los parámetros que se exponen en la tabla. Además, la diferencia de precisión en los casos benignos y malignos no es significativa, por lo que no destaca uno por encima del otro. Lo bueno de ambos métodos es la alta precisión en ambas clasificaciones. Por otra parte, si lo único importante en los datos fuera reducir las biopsias innecesarias, una buena idea seria usar el árbol de clasificación con profundidad 3, o incluso podríamos dividir los datos en dos clusters para hacer la clasificación. En cambio, si sólo quisiéramos dejar un mínimo número de masas tumorales sin detectar, es decir, no cometer el error de clasificar un tumor como benigno si no lo es, también seria recomendable usar Knn o Boosting con los parámetros aquí descritos.

En resumen, dado que el objetivo del estudio era hallar un modelo de clasificación que permita reducir el número de biopsias que declaran un tumor benigno y, además, en enfermedades graves no se pueden descuidar los falsos benignos (negativos), los modelos recomendados són K-nearest neighbors con K=3 y Boosting con `n.trees`=50, `interaction.depth`=2, `shrinkage`=0.1

y `n.minobsinnode=10`.

| | Error | Accuracy | Accuracy-Benign | Accuracy-Malign |
|---|---|---|---|---|
| Knn (K=31) | 0.1845018 | 0.8154982 | 0.7500000 | 0.8854962 |
| Knn (K=5) | 0.1143911 | 0.8856089 | 0.8571429 | 0.9160305 |
| Log. Reg. (All) | 0.1439114 | 0.8560886 | 0.8214286 | 0.8931298 |
| Log. Reg. (Not BR) | 0.1660517 | 0.8339483 | 0.7714286 | 0.9007634 |
| Tree (d=3) | 0.1439114 | 0.8560886 | 0.7642857 | 0.9541985 |
| Random Forests (mtry=2) | 0.1328413 | 0.8671587 | 0.8285714 | 0.9083969 |
| Random Forests (mtry=1) | 0.1365314 | 0.8634686 | 0.8071429 | 0.9236641 |
| Boosting (nt=50, d=2, shr=0.1, n.mino=10) | 0.1217712 | 0.8782288 | 0.8642857 | 0.8931298 |
| kmeans (scaled) | 0.1549815 | 0.8450185 | 0.75 | 0.9465649 |

**Tabla 10:** Tabla resumen del error y precisión de aplicar cada uno de los métodos estudiados en este documento. Aparecen los valores totales y para los casos de tumor benigno y maligno del *Mammographic Mass Data Set*. Los métodos se han aplicado en el conjunto de entrenamiento (menos k-means) y los errores se han calculado en el conjunto de validación.

# 4. Modificación sobre Age

Una vez seleccionados los mejores modelos, nos podemos preguntar si se puede mejorar aún más la precisión modificando el conjunto de datos. Así, si observamos los valores que toman los predictores, tenemos que la variable Age tiene valores mucho mayores que las demás y, por tanto, puede que le estemos dando demasiado peso a la edad. Por este motivo, nos hemos propuesto cambiar la columna Age por su logaritmo y así reducir su valor. La instrucción en R es la siguiente.

```
data_logA=data
data_logA$Age=log(data$Age)
```

De esta manera, se han aplicado los mejores métodos seleccionados, K-nearest neighbors y boosting, a este nuevo conjunto de datos. Los parámetros se han vuelto a estimar por el método de validación cruzada 10 veces en train y los resultados se resumen en la tabla 11. En este caso, no se presenta nada del código porque es análogo a los casos anteriores, pero se puede encontrar en el archivo R-markdown complementario.

En el caso de k-nearet neighbors el K asociado con la mayor precisión, después de hacer validación cruzada, es 35, con *accuracy* = 0,8181145. Se ha considerado que eran demasiados vecinos y, por esta razón, se ha tomado el segundo valor mayor, *accuracy* = 0,8163973, que corresponde a K=27. Los resultados son altamente sorprendentes, ya que que la precisión supera el 93 % en todos los casos. Sin embargo, para Boosting el cambio no ha sido favorable, ya que no se ha conseguido disminuir el error, menos para los casos benignos, y la diferencia no es significativa.

| | Error | Accuracy | Accuracy-Benign | Accuracy-Malign |
|---|---|---|---|---|
| Knn (K=27) | 0.04059041 | 0.9594096 | 0.9428571 | 0.9770992 |
| Boosting (nt=50, d=1, shr=0.1, n.mino=10) | 0.1328413 | 0.8671587 | 0.8785714 | 0.8549618 |

**Tabla 11:** Tabla del error y precisión de aplicar los métodos de k-nearest neighbors y boosting al *Mammographic Mass Data Set* con el logaritmo de Age. Aparecen los valores totales y para los casos de tumor benigno y maligno. Los métodos se han aplicado en el conjunto de entrenamiento y los errores se han calculado en el conjunto de validación.

# 5. Conclusiones

Establecer el objetivo del estudio y conocer que representan los datos que se van a analizar es fundamentar para saber como comparar los modelos y escoger el más adecuado. Además, tener información sobre las observaciones y las variables puede ayudar a mejorar la preparación del dataset. Por ejemplo, en este trabajo se han eliminado varias filas que se podrían haber usado para el estudio si los conocimientos sobre mamografías fueran mayores. Puede que algunas de éstas columnas tuvieran infromación relevante para futuras predicciones, ya que se ha supuesto que no se van a obtener nuevas observaciones que no se puedan resumir en ninguna de las características usadas.

Con este trabajo, se ha visto que hay métodos que pueden dar muy buenos resultados, y que vale la pena usarlos aunque se pierda la interpretación del modelo final, como Boosting en este trabajo. Esto no quiere decir que los métodos más simples, comunes e interpretables no vayan bien, ya que, el método que ha alcanzado una mayor precisión con diferencia ha sido un k-nearest neighbors (con $\ln(Age)$). Relacionado con este modelo, cabe destacar que a veces es una buena idea hacer transformaciones en algunas de las columnas, por ejemplo aplicando logarítmos, para reducir el peso de algunas variables.

Finalmente, es importante resaltar la importancia de tener un conjunto de validación que no sea usado por ninguno de los modelos, ni para encontrar sus mejores parámetros, ni para generar el modelo, ya que así no se puede estimar cuál será el error de una nueva observación. Esto puede llevar a un alto sobreajuste y un futuro error de predicción alto si no se tiene en cuenta. Una vez seleccionado el modelo que se va a usar para hacer predicciones, sí se pueden usar todos los datos para entrenarlo, ya que un mayor número de datos permite afinar las predicciones; según como sean los datos. Aún así, se podrían recoger nuevos datos para calcular nuevos errores y comprobar de manera más fina el error fuera de la muestra.

# Referencias


[1] UCI Machine Learning Repository, "Mammographic Mass Data Set." [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Mammographic+Mass

[2] Steven Halls, "BIRADS," *Moose and Doc. Breast Cancer*, January 11, 2018, [Blog]. [Online]. Available: http://breast-cancer.ca/bi-rads/

[3] Manfred te Grotenhuis and Paula Thijs, "Dummy variables and their interactions in regression analysis: examples from research on body mass index," *arXiv:1511.05728*, 2015. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1511/1511.05728.pdf

[4] The Pennsylvania State University. Eberly College of Science Site Administrator, "6.2.3 - More on Goodness-of-Fit and Likelihood ratio tests," in STAT 504 – Analysis of Discrete Data, Lesson 6: Logistic Regression, Seccion 6.2 - Binary Logistic Regression with a Single Categorical Predictor. [Course]. [Online]. Available: https://onlinecourses.science.psu.edu/stat504/node/220