¿Cómo realizar la limpieza y análisis de datos?

Autores: Eduardo Mora González y Diego Sánchez De La Fuente

Enero 2023

1. Dataset

1.1. Motivación

En Europa, el paro cardiaco es una de las primeras causas de mortalidad y en España fallecen en torno a 100 personas al día por este suceso [https://fundaciondelcorazon.com/prensa/notas-de-prensa/2900-solo-el-30-de-espanoles-sabe-realizar-la-reanimacion-cardio-pulmonar-rcp-.html], esto representa aproximadamente el 31% de las muertes a nivel mundial.

1.2. Descripción del dataset

El conjunto de datos ha sido extraído de Kaggle: https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset, está compuesto de 12 variables y 918 registros. Que correlacionan una serie de características recogidas de varios pacientes con la posibilidad de sufrir un ataque al corazón.

Explicación de cada variable:

- **Age**: Edad del paciente
- **Sex**: Sexo del paciente
- **ChestPainType**: Tipo de dolor torácico: Angina Típica Angina Atípica Dolor no debido a una angina Asintomático
- **RestingBP**: Presión arterial en reposo (en mm Hg)
- **Cholesterol**: Colesterol en sangre (mg/dL)
- **FastingBS**: Tiene Glucemia en ayunas > 120 mg/dl -> (1: True, 0: False)
- **RestingECG**: Resultados electrocardiográficos en reposo Value 0: normal Value 1: Tiene anormalidad de la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST > 0.05 mV) Value 2 Muestra hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes
- **MaxHR**: Frecuencia cardíaca máxima alcanzada
- **ExerciseAngina**: Angina inducida por el ejercicio (1 = sí, 0 = no)
- **Oldpeak**: Descenso del segmento ST inducido por el ejercicio en relación con el reposo ('Segmento ST' se relaciona con las posiciones en el gráfico de Electrocardiograma).
- **ST_Slope**: La pendiente del segmento ST de ejercicio máximo: 0: pendiente descendente 1: plano 2: pendiente ascendente

- **HeartDisease**: Variable Objetivo: 0= menos posibilidades de infarto 1= más posibilidades de infarto.

1.3. Objetivo buscado

Se puede decir que el objetivo buscado es predecir la posibilidad de que una persona tenga un alto riesgo de ser diagnosticado como un paciente cardíaco a través de las diversas características. Para llegar a al objetivo se tiene pensado realizar diversos métodos de análisis para así relacionar las diversas características para obtener unos parámetros finales y así concluir la posibilidad de que una persona tenga o no una enfermedad cardiaca.

2. Integración y selección

En nuestro caso se ha seleccionado todos los datos del fichero.

3. Preprocesado, gestión de características y exploración de los datos.

3.1. Valores nulos del conjunto de los datos

De tipo numérico

```
colSums(is.na(datos))
```

```
##            Age            Sex  ChestPainType      RestingBP    Cholesterol
##              0              0              0              0              0
##      FastingBS     RestingECG          MaxHR ExerciseAngina        Oldpeak
##              0              0              0              0              0
##       ST_Slope   HeartDisease
##              0              0
```

De tipo cadena

```
colSums(datos=="")
```

```
##            Age            Sex  ChestPainType      RestingBP    Cholesterol
##              0              0              0              0              0
##      FastingBS     RestingECG          MaxHR ExerciseAngina        Oldpeak
##              0              0              0              0              0
##       ST_Slope   HeartDisease
##              0              0
```

Como se puede comprobar, tenemos la “suerte” de no tener ningún valor nulo o vacío en los dos juegos de datos.

3.2. Normalización del conjunto de los datos

3.2.1. PRESIÓN ARTERIAL EN REPOSO (RestingBP)

Como se muestran en las estadísticas esta característica es de tipo numérico y en el conjunto de datos va desde 0 hasta 200. Como se puede apreciar, tener una presión arterial de 0 es estar considerado muerto, por lo que considero que el valor 0 es un valor nulo.

Lo primero que se va a hacer es obtener el número de casos que la presión arterial es 0, y se consideraran las diversas formas de tratar estos datos:

```
#Veces que aparece el valor cero en la presion arterial
length(datos$RestingBP[datos$RestingBP == 0])

## [1] 1
```

Como solo aparece una vez, se le asignará un valor por defecto. El valor por defecto será el más común.

```
#Función para calcular el valor más común
common_value <- function(x) {
uniqx <- unique(na.omit(x))
uniqx[which.max(tabulate(match(x, uniqx)))]
}

#Calculamos el valor más comun
BP_comun <- common_value(datos$RestingBP)

#Asignamos el valor
datos$RestingBP[datos$RestingBP == 0] <- BP_comun

#vemos las estaditicas del dato
summary(datos$RestingBP)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    80.0   120.0   130.0   132.5   140.0   200.0
```

3.2.2. COLESTEROL (Cholesterol)

La siguiente característica es de tipo numérico. Al igual que en la presión arterial en reposo, que tenemos valores 0 que debemos analizar. Lo primero que se va a hacer es obtener el número de casos que el colesterol es 0, y se consideraran las diversas formas de tratar estos datos.

```
#Veces que aparece el valor cero en la presion arterial
length(datos$RestingBP[datos$Cholesterol == 0])

## [1] 172
```

Esta vez tenemos 172 casos en lo que ocurre esto (equivale a un 18% de los casos totales).

```
#Calculamos el valor más comun
colesterol_media <- mean(datos$Cholesterol)

#Asignamos el valor truncado para evitar decimales
```

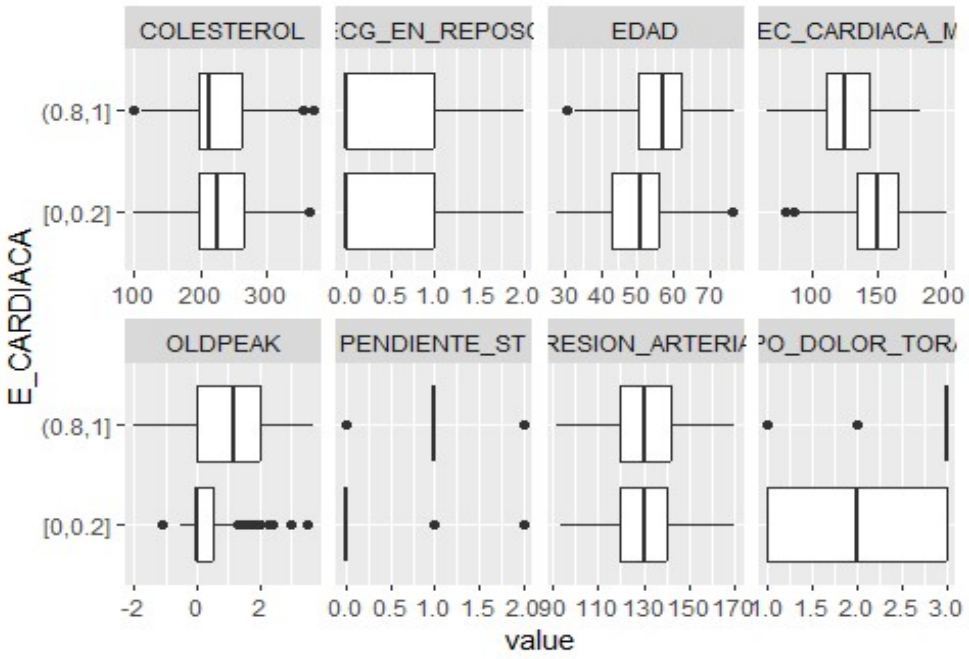
```
datos$Cholesterol[datos$Cholesterol == 0] <- trunc(colesterol_media)

#vemos las estaditicas del dato
summary(datos$RestingBP)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    80.0   120.0   130.0   132.5   140.0   200.0
```

3.3. Eliminamos outliers

Ahora miraremos los outliers

```
datos_bp.colesterol.out <- datos_bp.colesterol$out
print("Eliminamos Outliers de la variable COLESTEROL con valores: ")

## [1] "Eliminamos Outliers de la variable COLESTEROL con valores: "

datos_bp.colesterol.out

##  [1] 468 518 412 529  85 392 466 393 388 603 404 491 394 458 384 385 564 407 417
## [20] 409 394

datos_final <- datos_final %>% filter(!(COLESTEROL %in% datos_bp.colesterol.out))
dev.off()

## null device
##           1

datos_frec.cardiaca.max <- boxplot(datos_final$FREC_CARDIACA_MAX)
datos_frec.cardiaca.max.out <- datos_frec.cardiaca.max$out
print("Eliminamos Outliers de la variable FREC CARDIACA MAX con valores: ")

## [1] "Eliminamos Outliers de la variable FREC CARDIACA MAX con valores: "

datos_frec.cardiaca.max.out

## [1] 63 60

datos_final <- datos_final %>% filter(!(FREC_CARDIACA_MAX %in% datos_frec.cardiaca.max.out))
dev.off()

## null device
##           1

datos_oldpeak <- boxplot(datos_final$OLDPEAK)
datos_oldpeak.out <- datos_oldpeak$out
print("Eliminamos Outliers de la variable OLDPEAK con valores: ")

## [1] "Eliminamos Outliers de la variable OLDPEAK con valores: "

datos_oldpeak.out

##  [1]  4.0  5.0 -2.6  4.0  4.0  4.0  4.0  4.2  4.0  5.6  3.8  4.2  6.2  4.4  4.0

datos_final <- datos_final %>% filter(!(OLDPEAK %in% datos_oldpeak.out))
dev.off()

## null device
##           1

datos_bp.presion_arterial <- boxplot(datos_final$PRESION_ARTERIAL)
datos_bp.presion_arterial.out <- datos_bp.presion_arterial$out
print("Eliminamos Outliers de la variable PRESIÓN ARTERIAL ST con valores: ")

## [1] "Eliminamos Outliers de la variable PRESIÓN ARTERIAL ST con valores: "
```

```
datos_bp.presion_arterial.out

##  [1] 190 180 180 200 180 180 180  80 200 185 200 180 180 178 172 180 190 174 180
## [20] 192 178 180 180 172

datos_final <- datos_final %>% filter(!(PRESION_ARTERIAL %in% datos_bp.presion_arterial.out))
dev.off()

## null device
##           1

datos_bp.tipo_dolor_torax <- boxplot(datos_final$TIPO_DOLOR_TORAX)
datos_bp.tipo_dolor_torax.out <- datos_bp.tipo_dolor_torax$out
print("Eliminamos Outliers de la variable TIPO DOLOR TORAX con valores: ")

## [1] "Eliminamos Outliers de la variable TIPO DOLOR TORAX con valores: "

datos_bp.tipo_dolor_torax.out

##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [39] 0 0 0 0 0 0

datos_final <- datos_final %>% filter(!(TIPO_DOLOR_TORAX %in% datos_bp.tipo_dolor_torax.out))
dev.off()

## null device
##           1

#Diagrama de caja de todas las características enfrentadas a si un paciente tiene enfermedad cardiac
a
plot_boxplot(datos_final, by = "E_CARDIACA")
```



3.4. Proceso de limpieza de datos

Se ha realizado en el código un proceso más elaborado de limpieza de datos, para no alargar la memoria se ha mencionado lo más significativo.

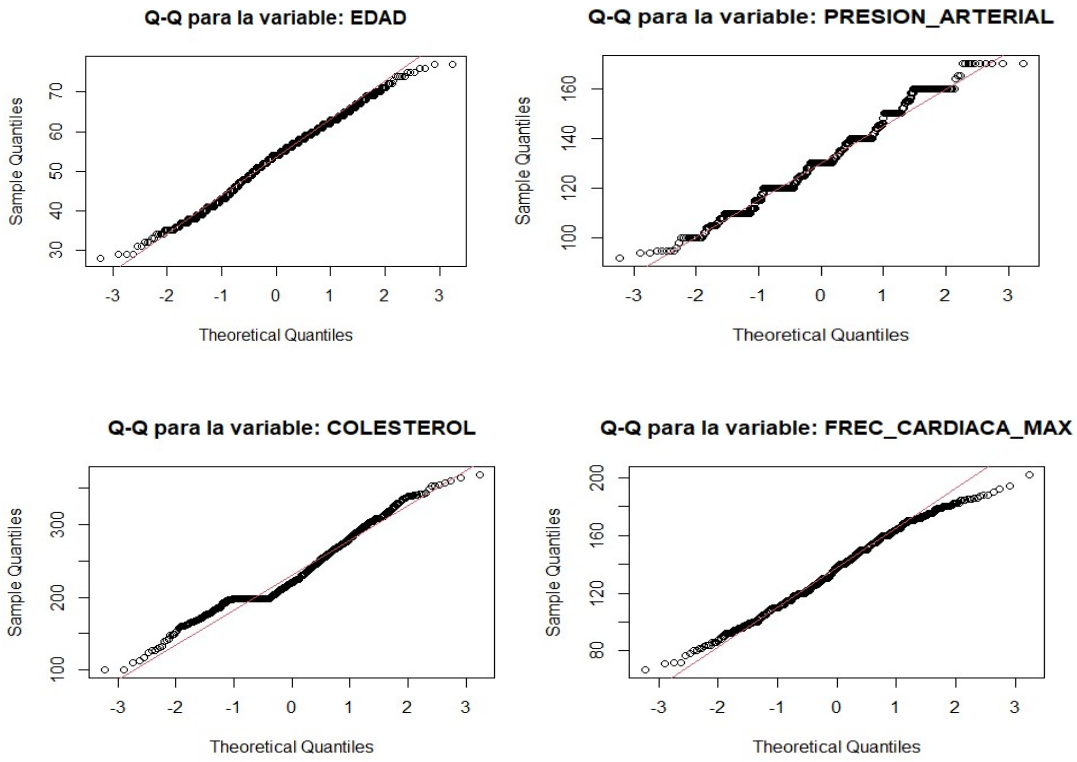
4. Análisis de los datos

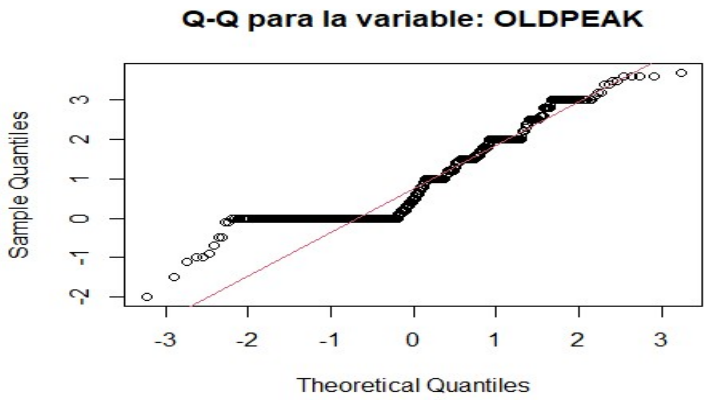
4.1. Comprobación de la normalidad y la homogeneidad de la varianza de las variables numéricas

Para la comprobación de que los valores que toman nuestra variables cuantitativa provienen de una distribución normal vamos a utilizar la prueba de normalidad de Anderson-Darling.

Podemos comprobar que para cada prueba se obtiene un p-valor superior al nivel de significancia estadistica prefijado en alpha = 0,05. Si esto se cumple, entonces se considera que variable en cuestion sigue la distribución normal.

```
variables <- c("EDAD", "PRESION_ARTERIAL", "COLESTEROL",
               "FREC_CARDIACA_MAX", "OLDPEAK")


for(i in(variables)){
  qqnorm(unlist(datos_final[i]),  main = paste0("Q-Q para la variable: ", i));qqline(unlist(datos_fi
nal[i]), col = 2)
}
```





Vemos que tanto la distribución de los valores de las características de EDAD y de la Frecuencia Cardiaca máxima se acercan mucho a la normalidad, por otro lado la distribución de los valores de la característica Presión Arterial, Colesterol y Old distan de la normal.

4.2. Test estadísticos de significancia

Antes de proceder a la clasificación de los parámetros de pacientes con más probabilidades de sufrir enfermedad cardiaca, deberemos de hacer una selección previa de las características a utilizar en nuestro modelo.

Para ello nos vamos a ayudar de una matriz de correlación con el objetivo de confirmar las conclusiones en cuanto a correlación de variables obtenidas en el apartado anterior.

Para aplicar los modelos de arbol de decision debemos de discretizar las variables COLESTEROL y FREC CARDIACA MAXIMA.

```
# Backup dataset inicial:
datos_final_orig <- datos_final

datos_final <- datos_final %>% mutate(COLESTEROL = case_when(
  COLESTEROL < 100 ~ 0,
  (COLESTEROL >= 100 & COLESTEROL < 200) ~ 1,
  (COLESTEROL >= 200 & COLESTEROL < 300) ~ 2,
  (COLESTEROL >= 300 & COLESTEROL < 400) ~ 3,
  (COLESTEROL >= 400 & COLESTEROL < 500) ~ 4,
  COLESTEROL >= 600 ~ 5,
  ))


datos_final <- datos_final %>% mutate(FREC_CARDIACA_MAX = case_when(
  FREC_CARDIACA_MAX < 50 ~ 0,
  (FREC_CARDIACA_MAX >= 50 & FREC_CARDIACA_MAX < 80) ~ 1,
  (FREC_CARDIACA_MAX >= 80 & FREC_CARDIACA_MAX < 110) ~ 2,
  (FREC_CARDIACA_MAX >= 110 & FREC_CARDIACA_MAX < 140) ~ 3,
  (FREC_CARDIACA_MAX >= 140 & FREC_CARDIACA_MAX < 170) ~ 4,
  (FREC_CARDIACA_MAX >= 170 & FREC_CARDIACA_MAX < 200) ~ 5,
  FREC_CARDIACA_MAX >= 200 ~ 6,
  ))


# Convertimos todas las variables a tipo factor
datos_final[] <- lapply(datos_final, factor)
```

```
# Analizamos las correlaciones de todos las caracteristicas de tipo categoricas con "E. CARDIACA"
# Lo añadimos a una tabla

datos_corr.Phi <- list()
datos_corr.CramerV <- list()
datos_corr.nombre <- list()

vector_tipos[8] <- "character"
vector_tipos[5] <- "character"

ind <- 1
for (i in colnames(datos_final))
{

  if(i != "E_CARDIACA")
  {
    tabla_cruzada <- table(as.numeric(unlist(datos_final[i])), datos_final$E_CARDIACA)
    datos_corr.CramerV <- append(datos_corr.CramerV, CramerV(tabla_cruzada))
    datos_corr.Phi <- append(datos_corr.Phi, Phi(tabla_cruzada))
    datos_corr.nombre <- append(datos_corr.nombre, i)
  }

  if (vector_tipos[ind] != 'numeric')
  {
    # Solo pintamos las variables categoricas ya que con las de tipo numerico no se aprecian los val
ores
    plot(tabla_cruzada, col = c("black","#008000"), main = paste0(i, " vs. E. CARDIACA"))
  }

  ind <- ind + 1

}

n_list <- list(nombre=as.character(datos_corr.nombre),
               CamerV=as.numeric(datos_corr.CramerV),
               Phi=as.numeric(datos_corr.Phi))

df_CramerV <- (as.data.frame(do.call(cbind, n_list)))
print(df_CramerV[order(df_CramerV$Phi, decreasing = TRUE),])

##                nombre            CamerV               Phi
## 11       PENDIENTE_ST 0.633914979247456 0.633914979247456
## 3    TIPO_DOLOR_TORAX 0.560309860170942 0.560309860170942
## 10            OLDPEAK 0.520727759286187 0.520727759286187
## 9  ANGINA_x_EJERCICIO 0.513768132124416 0.513768132124416
## 1               EDAD 0.404380273871888 0.404380273871888
## 8   FREC_CARDIACA_MAX 0.398095648097636 0.398095648097636
## 4    PRESION_ARTERIAL 0.331282489356957 0.331282489356957
## 2                SEXO 0.317570321818606 0.317570321818606
## 6     NIVEL_DE_AZUCAR 0.283413646865563 0.283413646865563
## 5         COLESTEROL  0.17862988864165  0.17862988864165
## 7       ECG_EN_REPOSO 0.115604279173622 0.115604279173622
```

Obtenemos las siguientes conclusiones del dataset:

- En cuanto al Sexo, las Mujeres tienen menos probabilidad de sufrir una enfermedad cardiaca.
- En cuanto al tipo de dolor de Torax, los pacientes que sufren de dolor tipo Asintomáticos son los que pese a lo que se pdoría pensar tienen más probabilidades de sufrir enfermedad cardiaca.

- COLESTEROL: Vemos que cuando el colesterol está en valores comprendidos entre 100-200 y 300-400 hay más posibilidades de enfermedad cardiaca que en valores entre 200-300, probablemente porque existan medicación para pacientes con dichas enfermedades que se enfocan en reducir el colesterol
- FREC CARDIACA MAXIMA hay correlación negativa con la enfermedad cardiaca es decir contra menor frecuencia más posibilidades de sufrir enfermedad cardiaca.
- En cuanto al ECG en Reposo, tenemos que hay más probabilidades de Enfermedad Cardiaca cuando esta variable toma el valor 1 (ST -> Tiene anormalidad de la onda ST-T (inversiones de la onda T y/o elevación o depresión del ST > 0.05 mV))
- Angina por Ejercicio -> Cuando esta toma el valor 1 (es decir hay angina inducida por ejercicio) hay más probabilidades de Enfermedad Cardiaca.
- En cuanto a la variable Pendiente ST, si esta indica valor 1 y 2, hay una alta probabilidad de sufrir enfermedad cardíaca.

4.2.1. Prueba de la C de Crámer

Valores de la V de Cramer (https://en.wikipedia.org/wiki/Cramér%27s_V) y Phi (https://en.wikipedia.org/wiki/Phi_coefficient) entre 0.1 y 0.3 nos indican que la asociación estadística es baja, y entre 0.3 y 0.5 se puede considerar una asociación media. Finalmente, si los valores fueran superiores a 0.5 la asociación estadística entre las variables sería alta.

Podemos observar dentro del dataframe: df_cramerV que las variables PENDIENTES ST, COLESTEROL, TIPO DOLOR TORAX, FREC CARDIACA MAX y OLDPEAK tienen correlación alta con E. CARDIACA.

Usaremos dichas variables para la obtención del árbol de decisión, se podría utilizar un número mayor de variables, pero podría hacerse mucho complejo (con muchas reglas de decisión)

```
# caracteristicas con significancia estadistica:
nombres_columnas <- df_CramerV$nombre[df_CramerV$Phi > 0.5]
nombres_columnas

## [1] "TIPO_DOLOR_TORAX"   "ANGINA_x_EJERCICIO" "OLDPEAK"
## [4] "PENDIENTE_ST"
```

4.3. ¿Cuáles son los factores o parámetros que más influyen a la hora de tener una enfermedad cardiaca? (Correlaciones y Regresión logística)

4.3.1. Test de Spearman

```
corr_matrix <- matrix(nc=2, nr=0)
colnames(corr_matrix) <- c("estimate", "p-value")

# Calculamos el coficiente de correlacion para cada variable cuantitativa
# con respecto al campo E. CARDICA

for(i in 1:(ncol(datos_final) -1 ))
{
  if(vector_tipos[i] == "numeric")
  {
    spearman_test = cor.test(unlist(datos_final[,i]),
                             unlist(datos_final[,length(datos_final)]),
                             method = "spearman")
    corr_coef <- spearman_test$estimate
    p_val <- spearman_test$p.value

    # Aañade a la matriz
    pair <- matrix(ncol = 2, nrow = 1)
    pair[1][1] = corr_coef
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(datos_final)[i]
  }
}
print(corr_matrix)

##                       estimate      p-value
## EDAD                0.32133836 5.839139e-21
## PRESION_ARTERIAL    0.11079517 1.566612e-03
## COLESTEROL         -0.06760546 5.414249e-02
## NIVEL_DE_AZUCAR     0.28341365 1.825875e-16
## FREC_CARDIACA_MAX -0.42471804 6.745047e-37
## OLDPEAK             0.44538146 8.078414e-41
```

Podemos identificar cual es la variable más correlacionada con la variable Enfermedad Cardiaca, viendo cuales de los valores de la columna estimate se acercan más al valor +1 o -1, en este caso los más cercanos y por lo tanto los que más correlacionados están con la variable objetivo son: OLDPEAK y FREC CARDÍACA MAX.

Por otro lado, en la columna p-value, tenemos el indicador del peso estadístico de diga variable, en este caso las variables que tienen peso estadístico más alto son: COLESTEROL y PRESION ARTERIAL.

4.3.2. Regresión logística

Generación de los conjuntos de entrenamiento y de test

```
set.seed(123)
ind <- sample(seq_len(nrow(datos_final)), size = round(.8 * dim(datos_final)[1]))
training <- datos_final[ind, ]
testing <- datos_final[-ind, ]
prop.table(table(datos_final$E_CARDIACA))

##
##         0         1
## 0.4507389 0.5492611

prop.table(table(training$E_CARDIACA))

##
##         0         1
## 0.4569231 0.5430769

prop.table(table(testing$E_CARDIACA))
```

```
##
##         0         1
## 0.4259259 0.5740741
```

Estimación del modelo con el conjunto de entrenamiento e interpretación

```
#Estimación del modelo
mod1<- glm(E_CARDIACA~.,data=training[,-1], family=binomial)
summary(mod1)
```

```
##
## Call:
## glm(formula = E_CARDIACA ~ ., family = binomial, data = training[,
##     -1])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5113  -0.4332   0.1592   0.4636   2.4653
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -3.7391233  1.6052230  -2.329 0.019841 *
## SEXO                1.3672239  0.3170565   4.312 1.62e-05 ***
## TIPO_DOLOR_TORAX    1.0365159  0.1755062   5.906 3.51e-09 ***
## PRESION_ARTERIAL    0.0010092  0.0081799   0.123 0.901809
## COLESTEROL          0.0001034  0.0027892   0.037 0.970441
## NIVEL_DE_AZUCAR     1.4315653  0.3213654   4.455 8.40e-06 ***
## ECG_EN_REPOSO       0.1695613  0.1510045   1.123 0.261484
## FREC_CARDIACA_MAX  -0.0132493  0.0053744  -2.465 0.013691 *
## ANGINA_x_EJERCICIO  0.9887369  0.2889107   3.422 0.000621 ***
## OLDPEAK             0.3031768  0.1508919   2.009 0.044513 *
## PENDIENTE_ST        1.8443446  0.2498882   7.381 1.57e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 896.26  on 649  degrees of freedom
## Residual deviance: 442.58  on 639  degrees of freedom
## AIC: 464.58
##
## Number of Fisher Scoring iterations: 5
```

Existe colinealidad en todas la variables menos en: FREC_CARDIACA_MAX, ANGINA_x_EJERCICIO, OLDPEAK y PENDIENTE_ST (En Cramer no aparece FREC_CARDIACA_MAX pero esta nos damos cuenta con VIF que no tiene colinealidad)

```
training2 = training %>%
select(-SEXO,-TIPO_DOLOR_TORAX, -PRESION_ARTERIAL, -COLESTEROL, -NIVEL_DE_AZUCAR, -ECG_EN_REPOSO)
ModlgF<- glm(E_CARDIACA~.,data=training2, family=binomial)
summary(ModlgF)
```

```
##
## Call:
## glm(formula = E_CARDIACA ~ ., family = binomial, data = training2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2307  -0.5666   0.2488   0.5422   2.2897
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.788651   1.161330  -0.679 0.497079
## EDAD                0.029939   0.013153   2.276 0.022831 *
## FREC_CARDIACA_MAX  -0.017371   0.005019  -3.461 0.000538 ***
## ANGINA_x_EJERCICIO  1.242050   0.254904   4.873  1.1e-06 ***
## OLDPEAK             0.351152   0.140126   2.506 0.012212 *
## PENDIENTE_ST        1.878777   0.219396   8.563  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 896.26  on 649  degrees of freedom
## Residual deviance: 533.76  on 644  degrees of freedom
## AIC: 545.76
##
## Number of Fisher Scoring iterations: 5
```

Se observa que todas las variables explicativas son significativas con un nivel de significación del 5%

Predicciones con con casos del dataframe

```
pred_1 <-predict (ModlgF, datos_final[1,], type = "response")
cat("La probabilidad de que el primer usuario tenga una enfermedad cardiaca es del: ", pred_1*100)

## La probabilidad de que el primer usuario tenga una enfermedad cardiaca es del:  7.051109

pred_14 <-predict (ModlgF, datos_final[14,], type = "response")
cat("La probabilidad de que el usuario 14 tenga una enfermedad cardiaca es del: ", pred_14*100)

## La probabilidad de que el usuario 14 tenga una enfermedad cardiaca es del:  84.79263
```

Nos damos cuenta de que la probabilidad obtenida es muy acertada a si esos usuarios han tenido o no enfermedad cardiaca.

4.4. ¿Influye el sexo en tener una enfermedad cardiaca? (Contraste de hipótesis)

Vamos a realizar una prueba estadística para establecer un contraste de hipótesis sobre dos muestras (una con presión arterial alta y otra con presión arterial normal) y ver cuál de ellas tiene mayor probabilidades de sufrir enfermedad cardiaca.

4.4.1. Hipótesis nula y la alternativa

H0 : Enf.Cardiaca_Mujer = Enf.Cardiaca_Hombre

H1 : Enf.Cardiaca_Mujer != Enf.Cardiaca_Hombre

4.4.2. Preparación datos

```
hombres <- datos_final[datos_final$SEXO==1,]
mujeres <- datos_final[datos_final$SEXO==0,]

var.test( as.numeric(hombres$E_CARDIACA),  as.numeric(mujeres$E_CARDIACA) )

##
##  F test to compare two variances
##
## data:  as.numeric(hombres$E_CARDIACA) and as.numeric(mujeres$E_CARDIACA)
## F = 1.2869, num df = 646, denom df = 164, p-value = 0.04968
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.000355 1.626469
## sample estimates:
## ratio of variances
##           1.286949
```

El resultado del test muestra diferencias significativas entre varianzas (p=0.04968). Por tanto, aplicaremos un test de dos muestras independientes sobre la media con varianzas desconocidas diferentes.

4.4.3. Cálculo

```
t.test( as.numeric(datos_final$SEXO), as.numeric(datos_final$E_CARDIACA), alternative="greater", conf.level=0.95)

##
##  Welch Two Sample t-test
##
## data:  as.numeric(datos_final$SEXO) and as.numeric(datos_final$E_CARDIACA)
## t = 11.016, df = 1554, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.2105546       Inf
## sample estimates:
## mean of x mean of y
## 0.7967980 0.5492611
```

4.4.4. Interpretación del test

Existen diferencias significativas en tener enfermedad cardiaca entre los hombres y la mujeres (p=2.2e-16) por lo que se encuentra fuera de la zona de aceptación de la hipótesis nula, es decir, se acepta la hipótesis alternativa.

4.5. Modelo árbol de decisión

Podemos elaborar un árbol de decisión, para ver que variables tienen más influencia en la enfermedad cardiaca y establecer las reglas que definen dicha variable.

4.5.1. Aplicación del modelo decisión Tree (Arbol de decisión)

Aplicamos el modelo Decision Tree sobre las 3 características que hemos obtenido en el test de significancia estadística de Cramer V.

```
# Reducimos el datset
for(i in colnames(datos_final))
{
  if(!i %in% nombres_columnas)
  {
    if (!i == "E_CARDIACA")
 {
      datos_final$i <- NULL
    }
  }
}
```

Separamos conjunto de test y de entrenamiento con una proporción 33% Test 66% Training.

```
set.seed(666)
y <- datos_final$E_CARDIACA # Variable objetivo
X <- datos_final[nombres_columnas] # Variables predictoras

split_prop <- 3
indexes = sample(1:nrow(X), size=floor(((split_prop-1)/split_prop)*nrow(X)))
train_X <- X[indexes,]
```

```
train_y <- y[indexes]
test_X <- X[-indexes,]
test_y <- y[-indexes]
```

Comprobamos que las variables train_y y test_y estén balanceadas reflejo de la variable y.

```
print("y:")

## [1] "y:"

summary(y)

##   0   1
## 366 446

print("train_y:")

## [1] "train_y:"

summary(train_y)

##   0   1
## 249 292

print("test_y:")

## [1] "test_y:"

summary(test_y)

##   0   1
## 117 154
```

Hacemos lo mismo con las variales train_crX y test_crX y X

```
print("X:")

## [1] "X:"

summary(X)

##  TIPO_DOLOR_TORAX ANGINA_x_EJERCICIO    OLDPEAK    PENDIENTE_ST
##  1:166            0:479              0      :336   0:359
##  2:195            1:333              1      : 78   1:405
##  3:451                               2      : 67   2: 48
##                                      1.5    : 49
##                                      3      : 27
##                                      1.2    : 23
##                                      (Other):232

print("train_X:")

## [1] "train_X:"

summary(train_X)

##  TIPO_DOLOR_TORAX ANGINA_x_EJERCICIO    OLDPEAK    PENDIENTE_ST
##  1:114            0:327              0      :227   0:246
##  2:129            1:214              1      : 49   1:260
##  3:298                               2      : 44   2: 35
##                                      1.5    : 35
##                                      3      : 18
##                                      1.2    : 15
##                                      (Other):153
```

```
print("test_X:")
```

```
## [1] "test_X:"
```

```
summary(test_X)
```

```
##  TIPO_DOLOR_TORAX ANGINA_x_EJERCICIO    OLDPEAK    PENDIENTE_ST
##  1: 52            0:152              0      :109   0:113
##  2: 66            1:119              1      : 29   1:145
##  3:153                               2      : 23   2: 13
##                                      1.5    : 14
##                                      3      :  9
##                                      1.2    :  8
##                                      (Other): 79
```

Se crea el árbol de decisión usando los datos de entrenamiento (no hay que olvidar que la variable outcome es de tipo factor):

```
tree <- C50::C5.0(train_X, train_y, rules=TRUE )
summary(tree)
```

```
##
## Call:
## C5.0.default(x = train_X, y = train_y, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Sat Jan 07 09:44:25 2023
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 541 cases (5 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (160/7, lift 2.1)
##  TIPO_DOLOR_TORAX in {1, 2}
##  PENDIENTE_ST = 0
##  ->  class 0  [0.951]
##
## Rule 2: (193/20, lift 1.9)
##  OLDPEAK in {-1.1, -0.5, -0.1, 0, 0.2, 0.3, 0.4, 0.6, 0.7, 1.1, 1.2, 1.9,
##                  2.3, 3}
##  PENDIENTE_ST = 0
##  ->  class 0  [0.892]
##
## Rule 3: (140/15, lift 1.6)
##  TIPO_DOLOR_TORAX = 3
##  OLDPEAK in {-1, -0.7, 0.1, 0.5, 0.8, 0.9, 1, 1.4, 1.5, 1.6, 1.8, 2, 2.8}
##  ->  class 1  [0.887]
##
## Rule 4: (295/50, lift 1.5)
##  PENDIENTE_ST in {1, 2}
##  ->  class 1  [0.828]
##
## Default class: 1
##
##
## Evaluation on training data (541 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       4   77(14.2%)   <<
##
##
##     (a)   (b)    <-classified as
```

```
##    ----  ----
##     193    56    (a): class 0
##      21   271    (b): class 1
##
##
##  Attribute usage:
##
##   94.09% PENDIENTE_ST
##   61.55% OLDPEAK
##   55.45% TIPO_DOLOR_TORAX
##
##
## Time: 0.0 secs
```

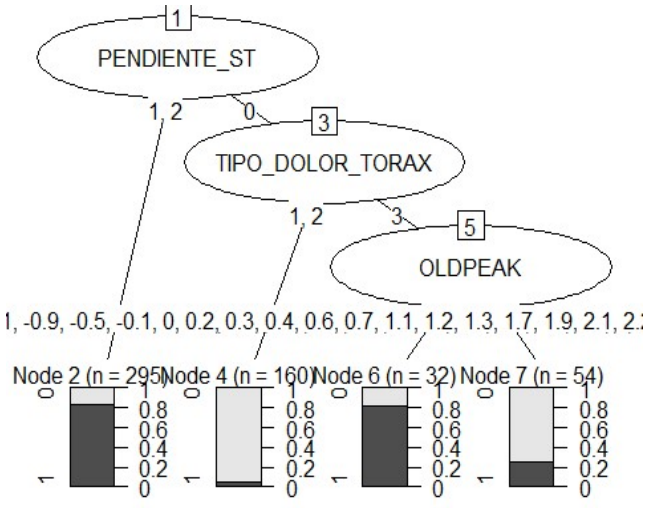
El modelo decision tree explica con dos reglas la probabilidad de sufrir una enfermedad cardiaca en función de las variables: TIPO DOLOR TORAX, COLESTEROL FREC CARDÍACA MÁX, OLDPEAK, PENDIENTE ST:

- Regla: 1 -> TIPO DE DOLOR DE TORAX con valores entre {1, 2} y PENDIENTE_ST <- 0 No tienen probabilidad de sufrir enfermedad cardiaca.
- Regla: 2 -> OLDPEAK entre {-1.1 y 3} con PENDIENTE_ST <- 0 No tienen probabilidad de sufrir Enfermedad Cardiaca.
- Regla: 3 -> TIPO_DOLOR_TORAX <- 3 y OLDPEAK entre {-1, 2.8} Tienen probabilidad de padecer enfermedad cardiaca.
- Regla 4 -> entre {1,2} -> Tienen probabilidad de tener enfermedad cardiaca.

El modelo solo usa la variable predictora PENDIENTE ST, y tiene una tasa de error de 14.2 % es decir es capaz de explicar el 82.6 % de los casos.

De manera más gráfica:

```
model <- C50::C5.0(train_X, train_y)
plot(model)
```



Como podemos observar de manera visual el modelo basado en árbol de decisión, solo tiene en cuenta la variable "PENDIENTE_ST", para decidir entre si un paciente es propenso a sufrir una enfermedad cardiaca o no.

4.5.2. Evaluación del modelo arbol de decision

Una vez tenemos el modelo, podemos comprobar su calidad prediciendo la clase para los datos de prueba que nos hemos reservado al principio.

```
predicted_model <- predict( tree, test_X, type="class" )
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(predicted_model == test_y) / length(predi
cted_model)))

## [1] "La precisión del árbol es: 83.7638 %"
```

Cuando hay pocas clases, la calidad de la predicción se puede analizar mediante una matriz de confusión que identifica los tipos de errores cometidos.

```
mat_conf <- table(test_y, Predicted=predicted_model)
mat_conf

##       Predicted
## test_y   0   1
##      0  85  32
##      1  12 142
```

De la matriz de confusión observamos los siguientes valores:

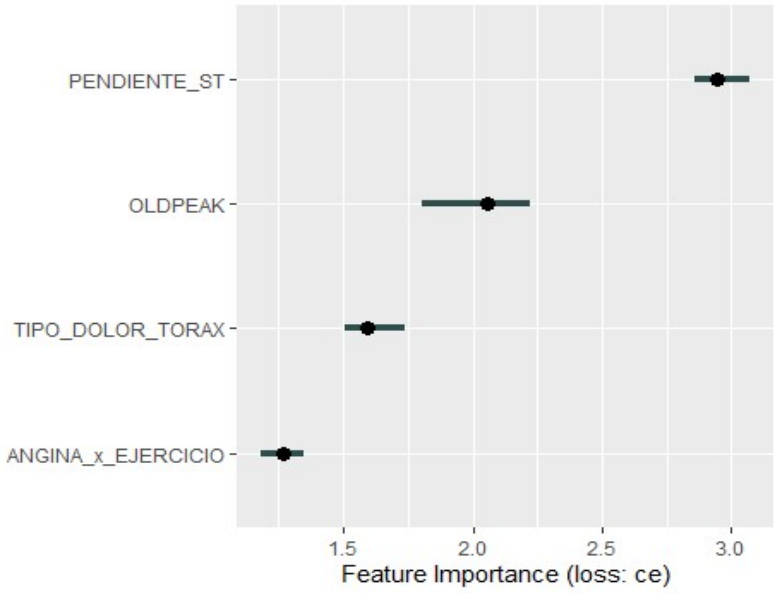
- Verdaderos Negativos (E. CARDIACA): 85
- Verdaderos Positivos (E. CARDIACA): 142
- Falsos Negativos (E. CARDIACA): 32
- Falsos Positivos (E. CARDIACA): 12

El modelo podría mejorarse sesgando a minimizar los falsos negativos, ya que no queremos que se nos escapen del diagnóstico pacientes que puedan desarrollar una enfermedad cardiaca.

4.6. Random Forest

Nos interesa saber para las predicciones que variable son las que tienen más influencia. Así, probaremos con un enfoque algorítmico de Random Forest y obtendremos métricas de interpretabilidad con la librería IML (https://cran.r-project.org/web/packages/iml/iml.pdf). As:

```
colnames(train_X)

## [1] "TIPO_DOLOR_TORAX"   "ANGINA_x_EJERCICIO" "OLDPEAK"
## [4] "PENDIENTE_ST"

train.data <- as.data.frame(cbind(train_X ,train_y))
colnames(train.data)[5] <- "E_CARDIACA"
rf <-  randomForest(E_CARDIACA ~ ., data = train.data, ntree = 50)

X <- train.data[which(names(train.data) != "E_CARDIACA")]
predictor <- Predictor$new(rf, data = X, y = train.data$E_CARDIACA)
imp <- FeatureImp$new(predictor, loss = "ce")
plot(imp)
```

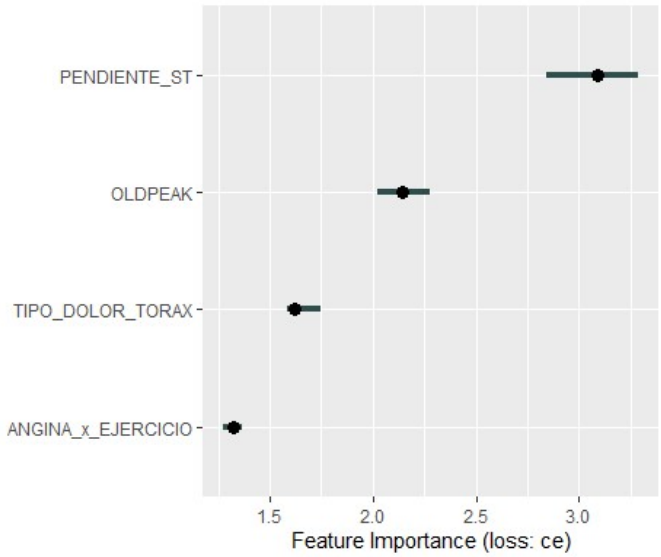



```
imp$results
```

```
##              feature importance.05 importance importance.95 permutation.error
## 1       PENDIENTE_ST      2.860714   2.946429      3.071429         0.3049908
## 2            OLDPEAK      1.807143   2.053571      2.225000         0.2125693
## 3   TIPO_DOLOR_TORAX      1.503571   1.589286      1.739286         0.1645102
## 4 ANGINA_x_EJERCICIO      1.182143   1.267857      1.346429         0.1312384
```

Podemos medir y graficar la importancia de cada variable para las predicciones del random forest con *FeatureImp*. La medida se basa funciones de pérdida de rendimiento que en nuestro caso será con el objetivo de clasificación ("ce").

```
X <- train.data[which(names(train.data) != "E_CARDIACA")]
predictor <- Predictor$new(rf, data = X, y = train.data$E_CARDIACA)
imp <- FeatureImp$new(predictor, loss = "ce")
plot(imp)
```



```
imp$results
```

```
##              feature importance.05 importance importance.95 permutation.error
## 1       PENDIENTE_ST      2.843636   3.090909      3.287273         0.3142329
## 2            OLDPEAK      2.021818   2.145455      2.276364         0.2181146
## 3   TIPO_DOLOR_TORAX      1.581818   1.618182      1.749091         0.1645102
## 4 ANGINA_x_EJERCICIO      1.276364   1.327273      1.363636         0.1349353
```

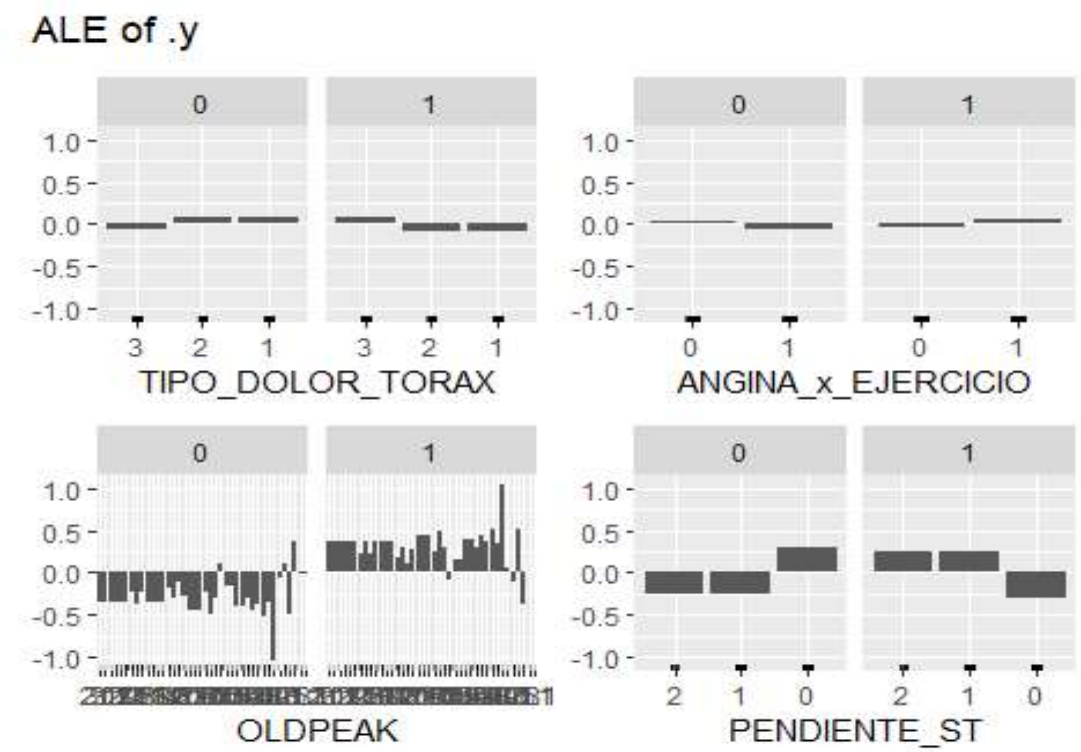
Precisión del modelo Random Forest

```
# Extraido de la matriz de confusion
print(paste0("La precsión del modelo randomforest es: ",
             (as.numeric(rf$confusion[1,][1]) / (as.numeric(rf$confusion[1,][1]) +
                                                   as.numeric(rf$confusion[1,][2]))) * 100))
```

```
## [1] "La precsión del modelo randomforest es: 75.5020080321285"
```

Podemos observar el grado de importancia de las variables:

```
X <- train.data[which(names(train.data) != "E_CARDIACA")]
predictor_cr <- Predictor$new(rf, data = X, y = train.data$E_CARDIACA)


effs <- FeatureEffects$new(predictor_cr)
plot(effs)
```



Parece ser que para el modelo de clasificación Random Forest la variable que toma mayor importancia es OLDPEAK.

Podemos verlo de manera textual:

```
rf$importance
```

```
##                     MeanDecreaseGini
## TIPO_DOLOR_TORAX           42.92313
## ANGINA_x_EJERCICIO         21.40851
## OLDPEAK                    47.57807
## PENDIENTE_ST               79.24776
```

5. Conclusiones

Tras realizar con el conjunto de datos distintas pruebas estadísticas y distintos modelos, nos damos cuenta de que todas las variables son necesarias para detectar enfermedades cardiacas. No obstante, dependiendo del modelo, no todas hay que usarlas.

Con los diversos modelos, hemos explorado distintas formas de predicción, como, por ejemplo, en las pruebas del modelo de regresión logística, los resultados de las pruebas han estado muy bien para comprobar las eficacia de dicho modelo. Por otro lado, los árboles de decisión la precisión también es muy buena.

En conclusión, con este análisis de datos se contestan las preguntas planteadas inicialmente, y se da unos modelos predictivos para detectar futuros casos.