**EXAMEN FINAL**

Análisis Estadístico con R Studio

**Alumno:** Renzo J. Chan R.

**Profesor:** Juan J. Romero A.

1. **Elegir data (clasificación)**

El dataset escogido corresponde a “Heart Attack Analysis & Prediction”, el cual tiene los siguientes parámetros importantes:

* Age : Edad del paciente
* Sex : Sexo del paciente
* exang: Angina inducida por ejercicio (1 = yes; 0 = no)
* ca: Número de vasos sanguíneos mayores (0-3)
* cp : Tipo de dolor en el pecho
  + Value 1: typical angina
  + Value 2: atypical angina
  + Value 3: non-anginal pain
  + Value 4: asymptomatic
* trtbps : Presión arterial en reposo (en mm Hg)
* chol : Colesterol en mg/dl obtenido a través de un sensor BMI
* fbs : (azúcar en sangre en ayunas > 120 mg/dl) (1 = true; 0 = false)
* rest\_ecg : resultados electrocardiográficos en reposo
  + Value 0: normal
  + Value 1: anomalías en la onda ST-T (inversiones en la onda T y/o elevación o depresión del ST>0.05 mV)
  + Value 2: mostrando hipertrofia ventricular izquierda probable o definitiva según los criterios de Este
* thalach : frecuencia cardíaca máxima alcanzada

Siendo finalmente la salida (output) de 1 o 0, dependiendo de lo siguiente:

* **output:** 0= menos posibilidades de ataque al corazón, 1= más posibilidades de ataque al corazón.

1. **Partición de la data**

Particionamos la data usando un 80% para el entrenamiento y un 20% para el test.

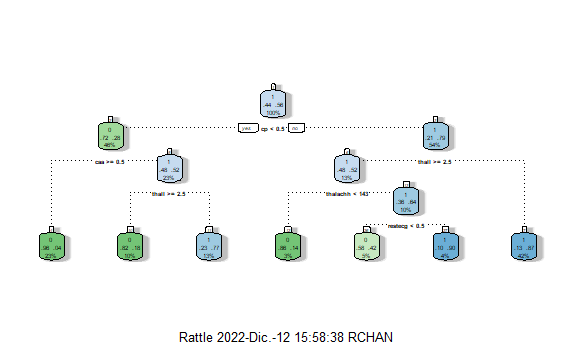
# Elegimos nuestra data para clasificación  
#choose.files()  
*setwd*("C:\\Users\\RCHAN\\Documents\\RStudio\\Examen Final\\DATA\\")  
*getwd*()  
  
myData<-*read.csv*('heart.csv')  
myData<-*data.frame*(myData)  
  
# Creando particiones para el entrenamiento y test  
particion<-createDataPartition(myData$output,p=0.75,list=*FALSE*)  
  
train<-myData[particion,]  
test<-myData[-particion,]  
  
# convertimos la variable de respuesta (output) en factor  
# para poder realizar una correcta clasificación  
train$output<-*as.factor*(train$output)

1. **Elegir dos modelos**
   1. **Modelo 1: Árbol de decisión**
      1. **Crear modelo 1**

# creación y entrenamiento del modelo  
modelo1<-rpart(output~.,data=train,  
 method="class",  
 maxdepth = 4,   
 minsplit = 2,   
 minbucket = 1)

* + 1. **Analizar modelo**

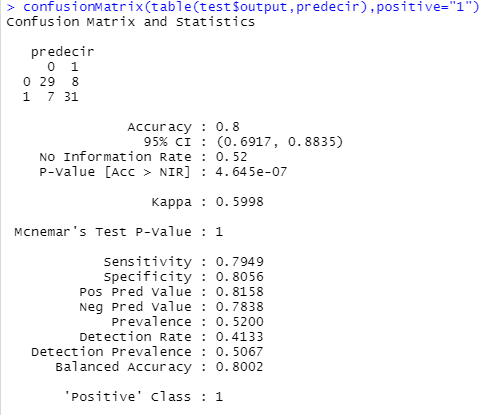
# viendo las reglas de forma gráfica  
fancyRpartPlot(modelo1)

****

* + 1. **Predecir test**

A continuación, mostramos el código para probar nuestro modulo con los datos del test.

# predecir modelo  
predecir<-*predict*(modelo1,test,type = "class")  
*table*(test$output,predecir)  
confusionMatrix(*table*(test$output,predecir),positive="1")



* + 1. **Enviar data nueva**

# enviar nueva data  
#myData[187,]  
#head(test,1)  
df<-*data.frame*(age=93, sex=1, cp=3, trtbps=145, chol=230, fbs=1, restecg=0,   
 thalachh=150, exng=0, oldpeak=2.3, slp=0, caa=0, thall=1)  
  
(predecir<-*predict*(modelo1,df))

Resultado Obtenido:

**Texto, Carta

Descripción generada automáticamente**

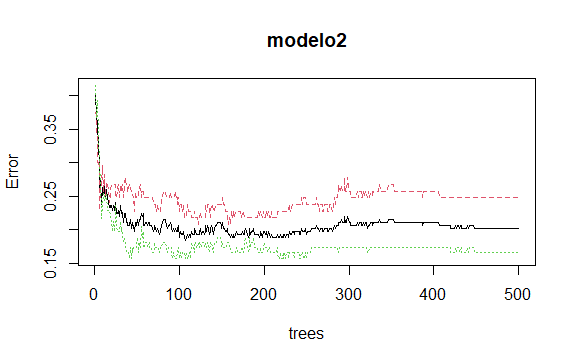
Como se puede observar tras ingresar los datos de una persona de 93 años, con tipo de dolor en el pecho 3, con una Depresión del ST inducida por el ejercicio en relación con el reposo (old Peak)=2.3 y una Pendiente del segmento ST de ejercicio máximo (slp)=0, el resultado obtenido es una gran probabilidad de sufrir un ataque al corazón.

* 1. **Modelo 2**
     1. **Crear modelo 2**

# -------------------------- MODELO RANDOM FOREST ------------------------------  
# creación y entrenamiento del modelo  
modelo2<-randomForest(output~.,data=train,importance = *TRUE*)

* + 1. **Analizar modelo**

# graficamos el error vs # de árboles  
*plot*(modelo2)

****

* + 1. **Predecir test**

**Texto

Descripción generada automáticamente**

* + 1. **Enviar data nueva**

Texto

Descripción generada automáticamente

1. **Seleccionar uno de los modelos. Explicar ¿Por qué?**

Texto

Descripción generada automáticamente