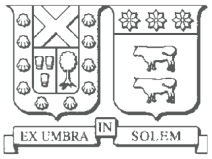
Departamento de Informática
Universidad Técnica Federico Santa María




# Informe Árboles de Clasificación
# Investigación de Operaciones
# 2019-1

Integrantes:

| Nombre | Rol USM | Email |
|---|---|---|
| Cristian Navarrete | 201573549-2 | cristian.navarreteg@sansano.usm.cl |

# 1. Describir el conjunto de datos: Cantidad de datos, Atributo Predictor/Clasificador, Tipo de Dato por atributo, Valores posibles.

```
datos = read.csv("DatosInforme19.csv", header=TRUE, sep=";")
summary(datos)
nrow(datos)
ncol(datos)
```

```
 Sexo    Horas.Estudio.Semanal      VTR           Tiempo.Libre      Carrete
 F:501   <2 hr  :266           Min.   :0.0000   Demasiado: 89   Demasiado:137
 M:369   >10 hr : 59           1st Qu.:0.0000   Mucho    :237   Mucho    :188
         2-5 hr :409           Median :0.0000   Nada     : 61   Nada     : 62
         5-10 hr:136           Mean   :0.2057   Normal   :330   Normal   :272
                               3rd Qu.:0.0000   Poco     :153   Poco     :211
                               Max.   :2.0000
       Salud     Inasistencias    Nota.Final
 Buena     :148   Min.   : 0.000   <55 :325
 Muy Buena :306   1st Qu.: 2.000   >=55:545
 Muy Mala  :121   Median : 3.000
 Normal    :190   Mean   : 3.863
 Suficiente:105   3rd Qu.: 6.000
                  Max.   :10.000

870

8
```

## 1.1. Cantidad de datos

870

## 1.2. Atributos / Clasificación / Tipo de Dato /

| Nombre | Clasificación | Tipo de dato |
|---|---|---|
| Sexo | Predictor | Discreto |
| Horas.Estudio.Semanal | Predictor | Discreto |
| VTR | Predictor | Discreto |
| Tiempo.Libre | Predictor | Discreto |
| Carrete | Predictor | Discreto |
| Salud | Predictor | Discreto |
| Inasistencias | Predictor | Discreto |
| Nota.Final | Clasificador | Discreto |

### 1.3. Valores posibles para datos

#### 1.3.1. Sexo

F/M

#### 1.3.2. Horas.Estudio.Semanal

< 2 horas, [2 - 5] horas, [5 - 10] horas,> 10 horas

#### 1.3.3. VTR

0, 1 o 2.

#### 1.3.4. Tiempo.Libre

Nada, Poco, Normal, Mucho o Demasiado

#### 1.3.5. Carrete

Nada, Poco, Normal, Mucho o Demasiado

#### 1.3.6. Salud

Muy Mala, Suficiente, Normal, Buena, Muy Buena

#### 1.3.7. Inasistencias

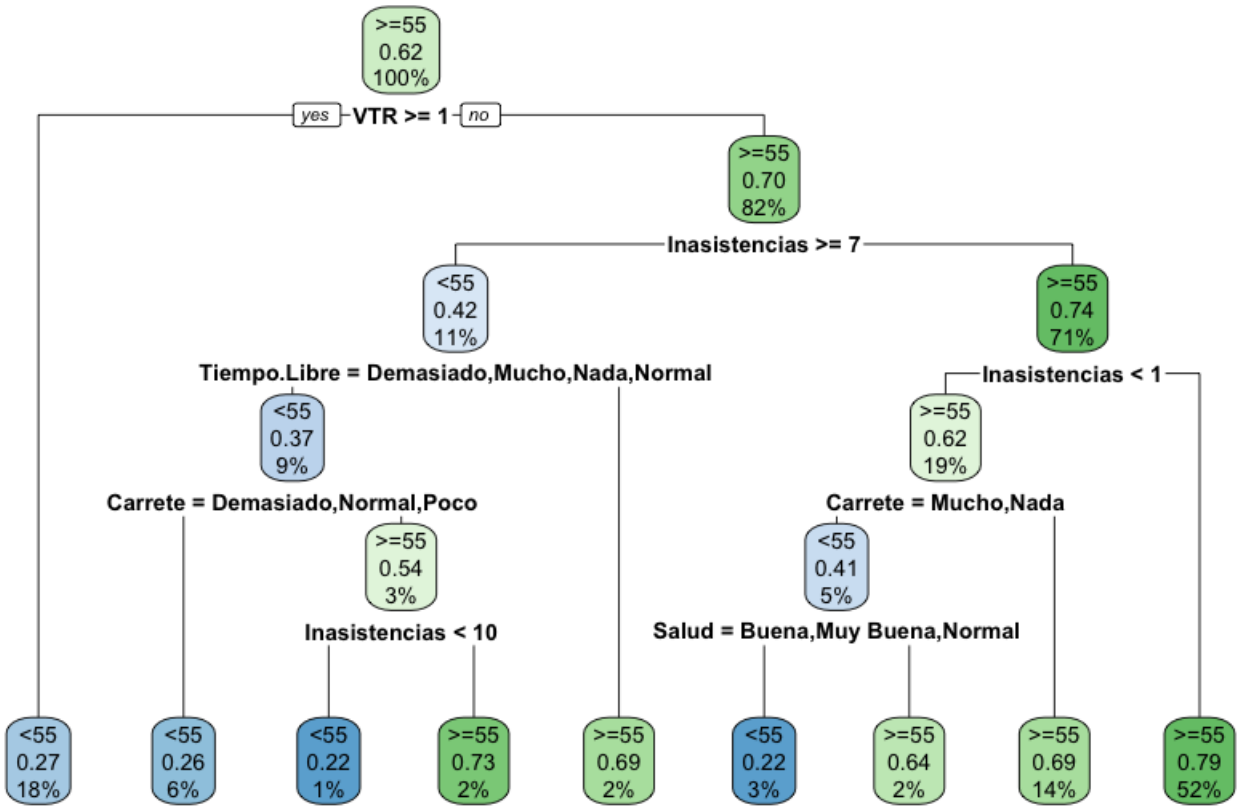
[0, 10]

#### 1.3.8. Nota.Final

<55 o >= 55

## 2. Construir e incluir un árbol de clasificación con los datos. Evalúe el árbol, interprete los resultados. Entregue el error de clasificación n

```
library(tidyverse)
library(rpart)
library(rpart.plot)
library(caret)

entrenamiento = sample_frac(datos, .8)
prueba = setdiff(datos, entrenamiento)


arbol_1 = rpart(formula = Nota.Final ~ ., data = entrenamiento)
rpart.plot(arbol_1)
```




Del árbol es posible ver que el VTR es uno de los factores más decisivos a la hora de saber si un alumno reprobara, seguido de las inasistencias, luego de esto multiples parametros empiezan a ser tomados en cuenta.

#### 2.0.1. Error de Clasificación

```
prediccion_1 = predict(arbol_1, newdata = prueba, type = "class")
confusionMatrix(prediccion_1, prueba[["Nota.Final"]])
```

```
Confusion Matrix and Statistics

          Reference
Prediction <55 >=55
      <55   33   11
      >=55  28  102

               Accuracy : 0.7759
                 95% CI : (0.7066, 0.8355)
    No Information Rate : 0.6494
    P-Value [Acc > NIR] : 0.0002089

                  Kappa : 0.474

 Mcnemar's Test P-Value : 0.0104056

            Sensitivity : 0.5410
            Specificity : 0.9027
         Pos Pred Value : 0.7500
         Neg Pred Value : 0.7846
             Prevalence : 0.3506
         Detection Rate : 0.1897
   Detection Prevalence : 0.2529
      Balanced Accuracy : 0.7218

       'Positive' Class : <55
```

El error de clasificación es de 1-0.7759 = 0.2241 o 22 %

## 2.1. Contestar las siguientes preguntas

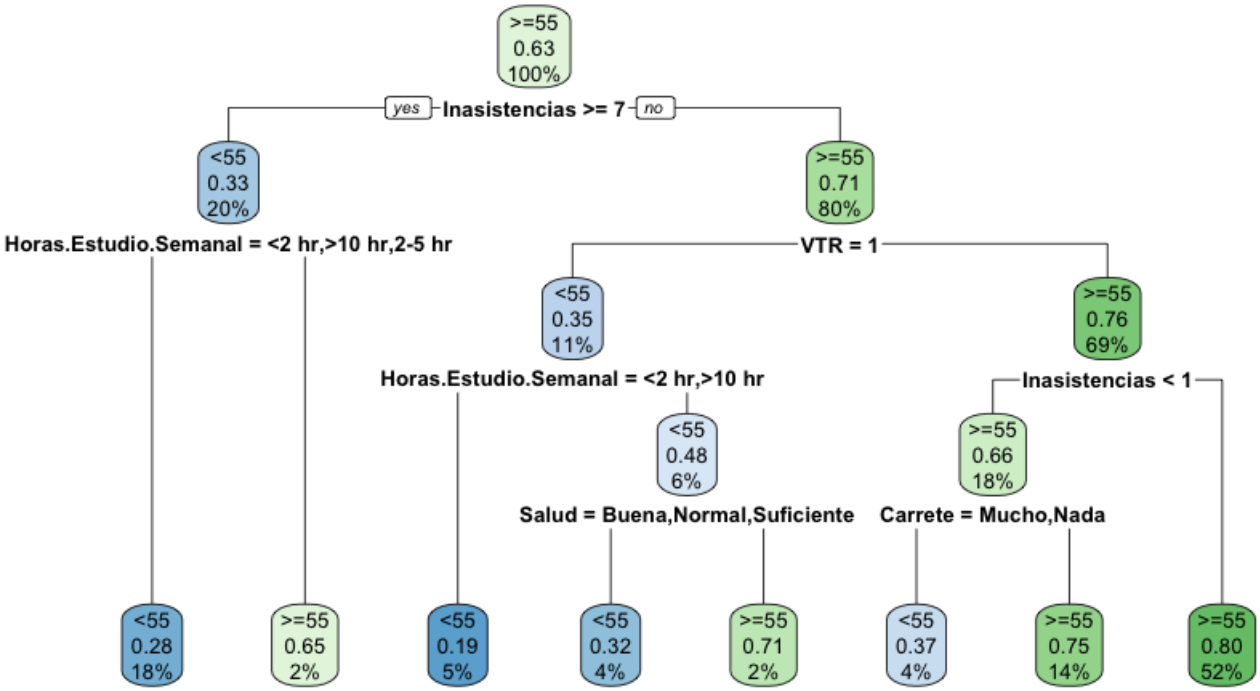
### 2.1.1. ¿Cuáles fueron las variables relevantes en la construcción del árbol? Explique por qué una variable es más relevante que otra.

VTR, Inasistencias, Tiempo Libre, una variable es más relevante que otra por que esa variable permite separar de mejor manera la población, de los datos podemos inferir que variables como el sexo no juegan un rol importante en predecir si un alumno reprobara.

### 2.1.2. ¿Qué sucede si se modifica el tipo de dato de VTR?

La librería utilizada en esta entrega no permite utilizar datos continuos, pero según lo investigado (*https* : *//discuss.analyticsvidhya.com/t/decision−tree−with−continuous−variables*/201/6) no debería provocar diferencia, ya que de igual manera los algoritmos buscan discretizar los datos (ya que un árbol no se puede armar con datos continuos, para eso utilizamos regresiones). De todos modos, se incluye un árbol realizado cuando VTR pasa a tener valor binario (Tiene o no tiene VTR).

```
datos$VTR[datos$VTR>=1] <- 1
summary(datos)
entrenamiento = sample_frac(datos, .8)
prueba = setdiff(datos, entrenamiento)
arbol2 = rpart(formula = Nota.Final ~ ., data = entrenamiento)
rpart.plot(arbol2)
prediccion = predict(arbol2, newdata = prueba, type = "class")
confusionMatrix(prediccion, prueba[["Nota.Final"]])
```

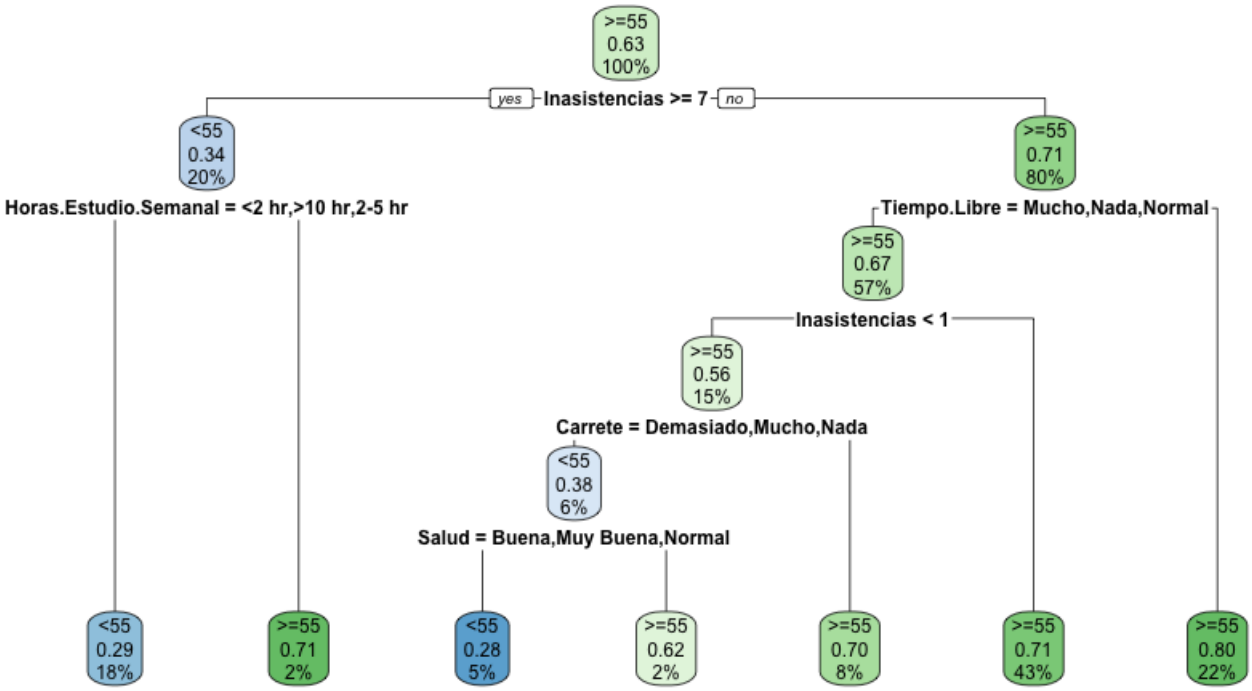

>=55
0.63
100%
yes
Inasistencias >= 7
no
<55
0.33
20%
>=55
0.71
80%
Horas.Estudio.Semanal = <2 hr,>10 hr,2-5 hr
VTR = 1
<55
0.35
11%
>=55
0.76
69%
Horas.Estudio.Semanal = <2 hr,>10 hr
Inasistencias < 1
<55
0.48
6%
>=55
0.66
18%
Salud = Buena,Normal,Suficiente
Carrete = Mucho,Nada
<55
0.28
18%
>=55
0.65
2%
<55
0.19
5%
<55
0.32
4%
>=55
0.71
2%
<55
0.37
4%
>=55
0.75
14%
>=55
0.80
52%

```
Confusion Matrix and Statistics

          Reference
Prediction <55 >=55
      <55   40    8
      >=55  41   84

               Accuracy : 0.7168
                 95% CI : (0.6434, 0.7825)
    No Information Rate : 0.5318
    P-Value [Acc > NIR] : 4.984e-07

                  Kappa : 0.417

 Mcnemar's Test P-Value : 4.844e-06

            Sensitivity : 0.4938
            Specificity : 0.9130
         Pos Pred Value : 0.8333
         Neg Pred Value : 0.6720
             Prevalence : 0.4682
         Detection Rate : 0.2312
   Detection Prevalence : 0.2775
      Balanced Accuracy : 0.7034

       'Positive' Class : <55
```

Es posible observar que la inasistencia pasa a tomar un rol más importante que el VTR, también vemos que el error aumenta.

#### 2.1.3. Si la variable VTR no es incluída en la construcción del árbol, ¿Qué sucede con el árbol? Evalué los cambios

```
datos$VTR <- NULL

summary(datos)
entrenamiento = sample_frac(datos, .8)
prueba = setdiff(datos, entrenamiento)
arbol2 = rpart(formula = Nota.Final ~ ., data = entrenamiento)
rpart.plot(arbol2)
prediccion = predict(arbol2, newdata = prueba, type = "class")
confusionMatrix(prediccion, prueba[["Nota.Final"]])
```

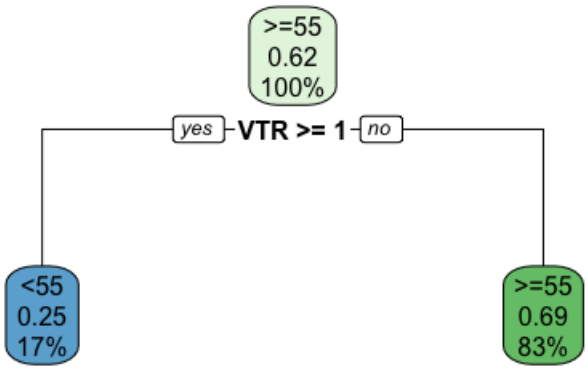



```
Confusion Matrix and Statistics

          Reference
Prediction <55 >=55
      <55   18    6
      >=55  48   95

               Accuracy : 0.6766
                 95% CI : (0.6, 0.7469)
    No Information Rate : 0.6048
    P-Value [Acc > NIR] : 0.0332

                  Kappa : 0.2398

 Mcnemar's Test P-Value : 2.414e-08

            Sensitivity : 0.2727
            Specificity : 0.9406
         Pos Pred Value : 0.7500
         Neg Pred Value : 0.6643
             Prevalence : 0.3952
         Detection Rate : 0.1078
   Detection Prevalence : 0.1437
      Balanced Accuracy : 0.6067

       'Positive' Class : <55
```

Vemos que Tiempo libre toma mucha más importancia, así mismo, el árbol crece bastante y utiliza múltiples atributos para decidir en que categoría clasificar al sujeto, ademas el error del arbol aumenta casi hasta niveles de ser aleatorio.

#### 2.1.4. Si la variable Inasistencias no es incluida en la construcción del árbol, ¿Qué sucede con el árbol? Evalué los cambios

```
# Previa carga de datos nueva, manteniendo todos los atributos originales

datos$Inasistencias <- NULL
summary(datos)
entrenamiento = sample_frac(datos, .8)
prueba = setdiff(datos, entrenamiento)
arbol2 = rpart(formula = Nota.Final ~ ., data = entrenamiento)
rpart.plot(arbol2)
prediccion = predict(arbol2, newdata = prueba, type = "class")
confusionMatrix(prediccion, prueba[["Nota.Final"]])
```



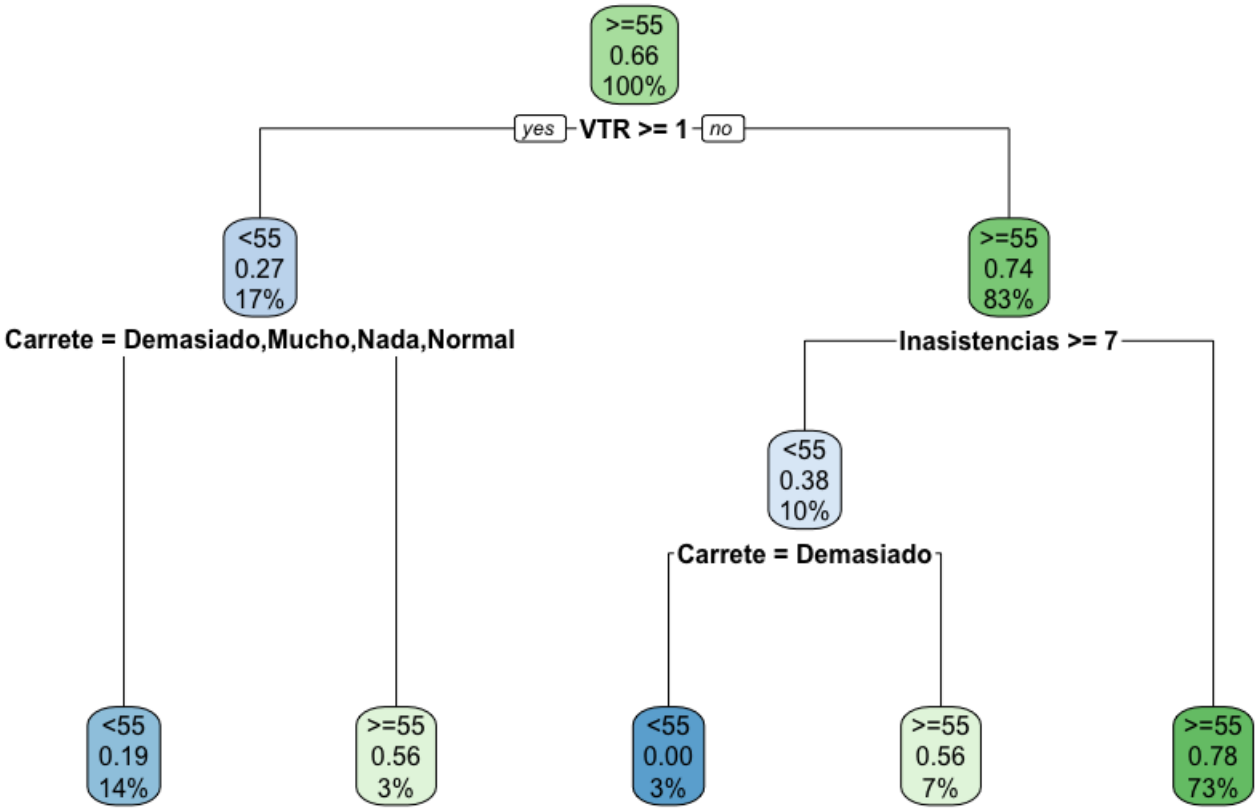
```
Confusion Matrix and Statistics

          Reference
Prediction <55 >=55
      <55   15    7
      >=55  18   24

               Accuracy : 0.6094
                 95% CI : (0.4793, 0.729)
    No Information Rate : 0.5156
    P-Value [Acc > NIR] : 0.08403

                  Kappa : 0.2263

 Mcnemar's Test P-Value : 0.04550

            Sensitivity : 0.4545
            Specificity : 0.7742
         Pos Pred Value : 0.6818
         Neg Pred Value : 0.5714
             Prevalence : 0.5156
         Detection Rate : 0.2344
   Detection Prevalence : 0.3438
      Balanced Accuracy : 0.6144

       'Positive' Class : <55
```

Vemos que el árbol de decisión pasa a ser binario, decidiendo solo en base al VTR, el error aumenta a un 40 % también.

**2.1.5. ¿Qué sucede si utiliza un 30, 50 y 70 % de los datos entregados como Training? Evaluar e imprimir los árboles obtenidos. Explique.**

**30 %**

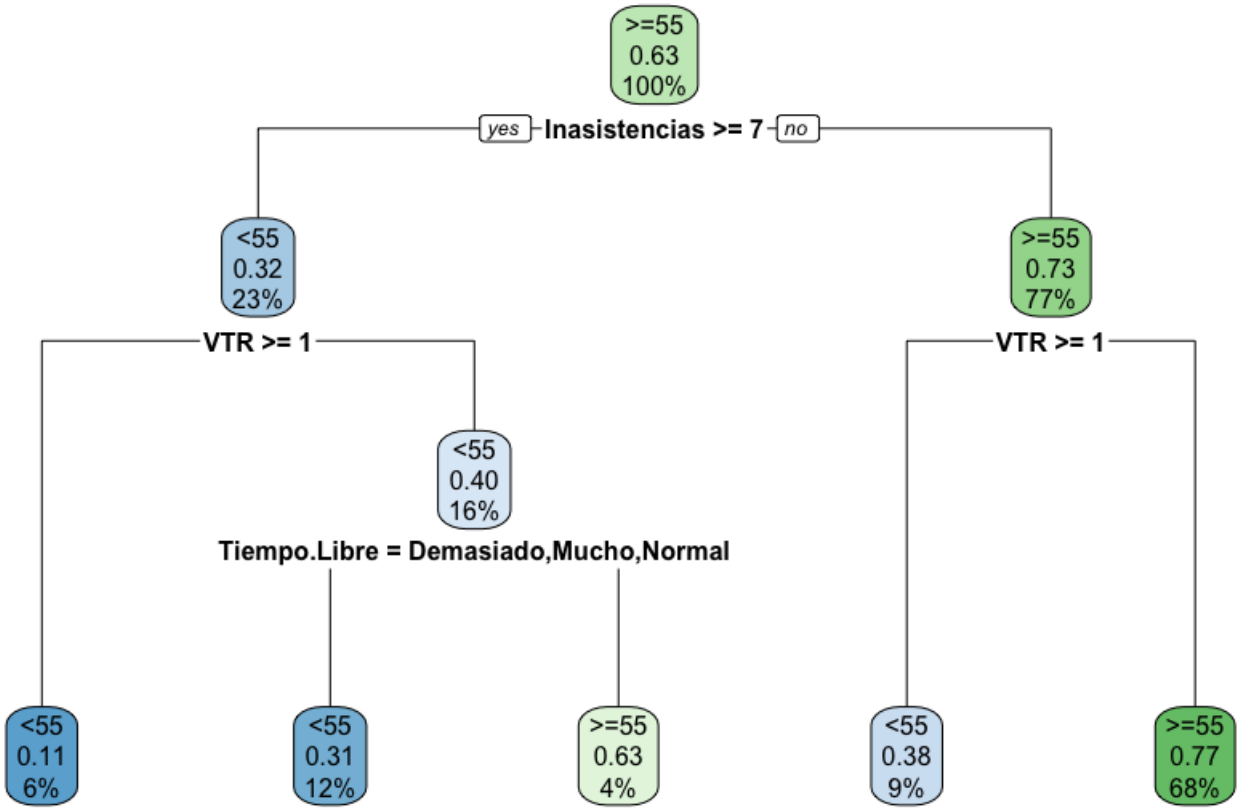
```
entrenamiento = sample_frac(datos, .3)
prueba = setdiff(datos, entrenamiento)
arbol2 = rpart(formula = Nota.Final ~ ., data = entrenamiento)
rpart.plot(arbol2)
prediccion = predict(arbol2, newdata = prueba, type = "class")
confusionMatrix(prediccion, prueba[["Nota.Final"]])
```


>=55
0.66
100%
yes
VTR >= 1
no
<55
0.27
17%
>=55
0.74
83%
Carrete = Demasiado,Mucho,Nada,Normal
Inasistencias >= 7
<55
0.38
10%
Carrete = Demasiado
<55
0.19
14%
>=55
0.56
3%
<55
0.00
3%
>=55
0.56
7%
>=55
0.78
73%

```
Confusion Matrix and Statistics

          Reference
Prediction <55 >=55
      <55   69   25
      >=55 166  349

               Accuracy : 0.6864
                 95% CI : (0.6479, 0.723
    No Information Rate : 0.6141
    P-Value [Acc > NIR] : 0.0001227

                  Kappa : 0.2552

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.2936
            Specificity : 0.9332
         Pos Pred Value : 0.7340
         Neg Pred Value : 0.6777
             Prevalence : 0.3859
         Detection Rate : 0.1133
   Detection Prevalence : 0.1544
      Balanced Accuracy : 0.6134

       'Positive' Class : <55
```

**50 %**

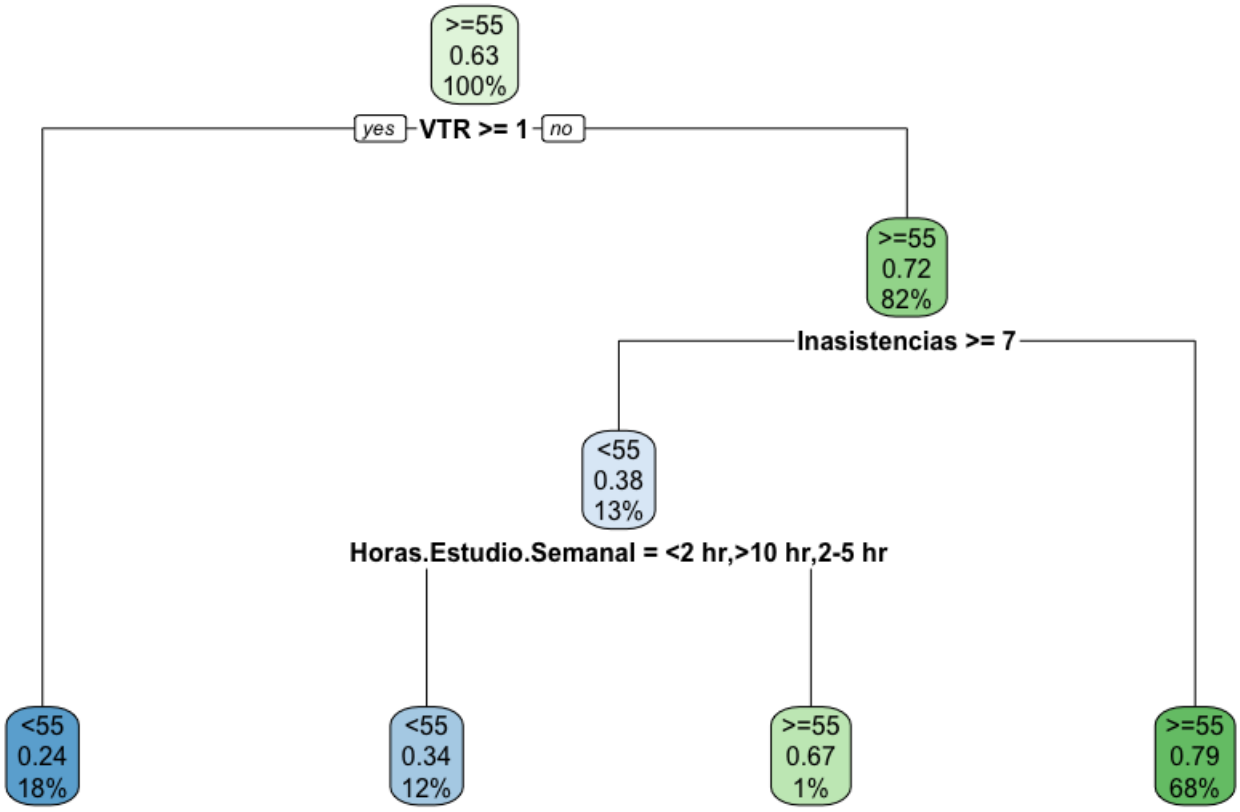
```
entrenamiento = sample_frac(datos, .5)
prueba = setdiff(datos, entrenamiento)
arbol2 = rpart(formula = Nota.Final ~ ., data = entrenamiento)
rpart.plot(arbol2)
prediccion = predict(arbol2, newdata = prueba, type = "class")
confusionMatrix(prediccion, prueba[["Nota.Final"]])
```


>=55
0.63
100%
yes
Inasistencias >= 7
no
<55
0.32
23%
>=55
0.73
77%
VTR >= 1
VTR >= 1
<55
0.40
16%
Tiempo.Libre = Demasiado,Mucho,Normal
<55
0.11
6%
<55
0.31
12%
>=55
0.63
4%
<55
0.38
9%
>=55
0.77
68%

```
Confusion Matrix and Statistics

          Reference
Prediction <55 >=55
      <55   77   30
      >=55  89  239

               Accuracy : 0.7264
                 95% CI : (0.6819, 0.7678)
    No Information Rate : 0.6184
    P-Value [Acc > NIR] : 1.354e-06

                  Kappa : 0.3781

 Mcnemar's Test P-Value : 1.056e-07

            Sensitivity : 0.4639
            Specificity : 0.8885
         Pos Pred Value : 0.7196
         Neg Pred Value : 0.7287
             Prevalence : 0.3816
         Detection Rate : 0.1770
   Detection Prevalence : 0.2460
      Balanced Accuracy : 0.6762

       'Positive' Class : <55
```

**70%**

```
entrenamiento = sample_frac(datos, .7)
prueba = setdiff(datos, entrenamiento)
arbol2 = rpart(formula = Nota.Final ~ ., data = entrenamiento)
rpart.plot(arbol2)
prediccion = predict(arbol2, newdata = prueba, type = "class")
confusionMatrix(prediccion, prueba[["Nota.Final"]])
```


>=55
0.63
100%
yes
VTR >= 1
no
>=55
0.72
82%
Inasistencias >= 7
<55
0.38
13%
Horas.Estudio.Semanal = <2 hr,>10 hr,2-5 hr
<55
0.24
18%
<55
0.34
12%
>=55
0.67
1%
>=55
0.79
68%

```
Confusion Matrix and Statistics

          Reference
Prediction <55 >=55
      <55   39   20
      >=55  62  140

               Accuracy : 0.6858
                 95% CI : (0.6257, 0.7417)
    No Information Rate : 0.613
    P-Value [Acc > NIR] : 0.008722

                  Kappa : 0.2828

 Mcnemar's Test P-Value : 5.963e-06

            Sensitivity : 0.3861
            Specificity : 0.8750
         Pos Pred Value : 0.6610
         Neg Pred Value : 0.6931
             Prevalence : 0.3870
         Detection Rate : 0.1494
   Detection Prevalence : 0.2261
      Balanced Accuracy : 0.6306

       'Positive' Class : <55
```

Cuando utilizamos el 50 % de los datos para training, encontramos el menor error en el árbol de decisión, también es interesante ver que en el 50 % la asistencia es más importante que el VTR, algo que no ocurre con los otros gráficos.