Fundamentos de Análisis de Datos

Unidad 7

Predicción y clasificación con técnicas numéricas

Referencias

- García, J., Molina, J., Berlanga, A., Patricio, M., Bustamante, Á., & Padilla, W. (2018). Ciencia de datos: técnicas analíticas y aprendizaje estadísticos en un enfoque práctico (1st ed.). Alfaomega Altaria. https://visorweb.utpl.edu.ec/reader/ciencia-de-datos-tecnicas-analiticas-y-aprendizaje-estadisticos-en-un-enfoque-practico?location=118 (apartado 4, pág 117)

Agenda

- Predicción y clasificación con técnicas numéricas
  - Concepto
  - Técnicas
  - Pasos
- Regresión logística
- Arboles de decisión
- Random Forest (bosques aleatorios)
- Clasificación bayesiana

Predicción y clasificación con técnicas numéricas

- El análisis predictivo utiliza diversas técnicas y modelos estadísticos para predecir resultados futuros y tomar decisiones estratégicas. Dos de los tipos más comunes de análisis predictivo son la **clasificación** y la **regresión**.
- **Clasificación**: asigna observaciones a categorías específicas. Se basa en algoritmos que analizan las características y propiedades de los datos disponibles para realizar predicciones precisas.
- **Regresión**: se utiliza para predecir valores numéricos o continuos. A diferencia del análisis de clasificación, que asigna categorías, el análisis de regresión estima el valor de una variable dependiente en función de una o más variables independientes.

Predicción y clasificación con técnicas numéricas

Algunas de las técnicas más utilizadas para predicción y clasificación son:

Técnica	Usos
Regresión Lineal: asume una relación lineal entre la variable dependiente y una variable independiente.	Previsión de ventas, precios de viviendas, análisis de tendencias económicas
Regresión Lineal Múltiple: extensión de la regresión lineal que utiliza múltiples variables independientes.	Predicción de ingresos, análisis financiero
Regresión Logística: utilizada para modelar una variable dependiente binaria	Diagnóstico médico, predicción de fraude, análisis de comportamiento del consumidor.
Árboles de decisión: Modelos que dividen los datos en ramas basadas en reglas de decisión	Clasificación de clientes, análisis de riesgo, diagnóstico médico.
Bosques Aleatorios (Random Forests): Combina múltiples árboles de decisión para mejorar la precisión y reducir el sobreajuste.	Clasificación de texto, análisis de imágenes, predicción de ventas
Máquinas de Soporte Vectorial (SVM): Encuentran el hiperplano que maximiza el margen entre clases	Clasificación de imágenes, detección de fraude, análisis de texto
K-Vecinos Más Cercanos (K-NN): Clasifica una observación basada en las etiquetas de sus vecinos más cercanos	Reconocimiento de patrones, análisis de imagen, clasificación de texto
Series Temporales (ARIMA): modelos que capturan la autocorrelación en los datos de series temporales	Previsión de demanda, análisis financiero, meteorología
Regresión Logística Multinomial: Extensión de la regresión logística para variables dependientes categóricas con más de dos niveles	Análisis de mercado, clasificación de documentos, diagnóstico médico
Redes neuronales: Modelos inspirados en la estructura del cerebro humano, utilizados para capturar patrones complejos	Reconocimiento de voz, imágenes, predicción de series temporales complejas
Naive Bayes: Se basa en el teorema de Bayes para calcular la probabilidad de pertenencia a una clase	Clasificación de texto y análisis de sentimientos.

Predicción y clasificación con técnicas numéricas

- Estas técnicas numéricas de predicción y clasificación son herramientas poderosas en el análisis de datos y el aprendizaje automático.
- La elección de la técnica adecuada depende del tipo de problema, la naturaleza de los datos y los objetivos del análisis.
- Es común probar varias técnicas y comparar sus rendimientos utilizando métricas de evaluación para seleccionar el modelo más adecuado.

Regresión logística

- **Propósito**: Modelar la relación entre una variable dependiente BINARIA y múltiples variables independientes.
- **Descripción**: Es un tipo de modelo lineal generalizado (GLM) que se utiliza para clasificación binaria. A diferencia de la regresión lineal, que predice un valor continuo, la regresión logística predice una probabilidad que luego se mapea a una de dos posibles categorías.
- **Aplicaciones**: En cualquier ámbito en el que necesitemos predecir la ocurrencia o no de un evento (Ej: deserción de un estudiante, otorgar un crédito, compra de un articulo por un cliente, presencia de una enfermedad)
- **Condiciones**:
  - Aplica para la predicción de **variables categóricas BINARIAS**
  - Si los predictores (variables independientes) son variables categóricas, se las debe **factorizar**.

Regresión logística

- Entre dos categorías A y B, la regresión logística estima la probabilidad de que una observación pertenezca a la 2da categoría. Si la probabilidad es mayor a 0.5 se asume que pertenece a B.
- **Ejemplo** (dataset "**titanic.csv**"):
  - **Objetivo**: Mediante regresión logística construir un modelo que permita predecir el status de supervivencia de cada pasajero, a partir de las variables independientes: *Pclass*, *Sex*, *Age*, *SibSp*, *Parch*, *Fare*, y *Embarked*.
  - **Librería a utilizar**: **caret**
    - Incluye un conjunto de funciones que facilitan el uso de métodos complejos de clasificación y regresión. Para regresión logística se usa la función **glm()**

Regresión logística

```r
# Instalar y cargar las librerías necesarias
install.packages("caret")
library(dplyr)
library(ggplot2)
library(caret)

# Cargar y explorar datos
df <- read.csv("titanic.csv")
head(df)
names(df)
str(df)

# Seleccionar variables relevantes y eliminar valores faltantes
df_clean <- df %>%
  select(Survived, Pclass, Sex, Age, SibSp, Parch, Fare, Embarked)

sapply(df_clean,function(x) mean(is.na(x) | !nzchar(x)))

df_clean <- df_clean %>%
  filter(!is.na(Age) & nzchar(Embarked))

# Convertir variables categóricas a factores
df_clean$Survived <- as.factor(df_clean$Survived)
df_clean$Pclass <- as.factor(df_clean$Pclass)
df_clean$Sex <- as.factor(df_clean$Sex)
df_clean$Embarked <- as.factor(df_clean$Embarked)
str(df_clean)
```

Regresión logística

```r
# Dividir los Datos en Conjuntos de Entrenamiento y de Prueba
# 80% entrenamiento, 20% prueba, y generar un vector con
# los índices de las obseraciones seleccionadas
set.seed(88)
train_index <- createDataPartition(df_clean$Survived,
                                   p = 0.8,
                                   list = FALSE)
train_data <- df_clean[train_index, ]   #Datos de entrenamiento
test_data <- df_clean[-train_index, ]   #Datos de prueba

# Ajustar el Modelo de Regresión Logística
# Variables dependiente (a predecir): Survivded
model <- glm(Survived ~ Pclass + Sex + Age + SibSp +
               Parch + Fare + Embarked,
             data = train_data,
             family = binomial)

# Resumen del modelo
summary(model)
model$coefficients
```

Para que se ajuste a un modelo de regresión logística (tipo de variable binaria)

Coeficientes β (logaritmo de la razón de probabilidades) del modelo. Indica el sentido del cambio de valor de la variable dependiente, por cada aumento del valor del predictor, mientras el resto de valores se mantiene constante

```
  (Intercept)       Pclass2       Pclass3       Sexmale           Age         SibSp         Parch
4.2769382253 -1.1208987508 -2.4797172142 -2.5819978513 -0.0432734526 -0.3731713769  0.0462173295
        Fare     EmbarkedQ     EmbarkedS
0.0001232415 -0.6217113936 -0.2232999545
```

Regresión logística

```
Call:
glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
    Fare + Embarked, family = binomial, data = train_data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.2769382  0.5813403   7.357 1.88e-13 ***
Pclass2     -1.1208988  0.3658516  -3.064  0.00219 **
Pclass3     -2.4797172  0.3781737  -6.557 5.49e-11 ***
Sexmale     -2.5819979  0.2488561 -10.375  < 2e-16 ***
Age         -0.0432735  0.0090663  -4.773 1.82e-06 ***
SibSp       -0.3731714  0.1416637  -2.634  0.00843 **
Parch        0.0462173  0.1466590   0.315  0.75266
Fare         0.0001232  0.0029058   0.042  0.96617
EmbarkedQ   -0.6217114  0.6500200  -0.956  0.33885
EmbarkedS   -0.2233000  0.3141234  -0.711  0.47717
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 770.64  on 570  degrees of freedom
Residual deviance: 508.74  on 561  degrees of freedom
AIC: 528.74

Number of Fisher Scoring iterations: 5
```

Medida del coeficiente β (logaritmo de la razón de probabilidades) asociado a la variable dependiente cuando todos los predictores con cero

P-value < 0.05 => la incidencia de la variable en el modelo de regresión es significativa

Coeficientes del modelo. Indica el sentido del cambio de valor de la variable dependiente, por cada aumento del valor del predictor, mientras el resto de valores se mantiene constante

Criterio de información de Akaike: mide la calidad del modelo, sirve para comparar modelos, mientras más bajo, mejor se ajusta el modelo.

Regresión logística

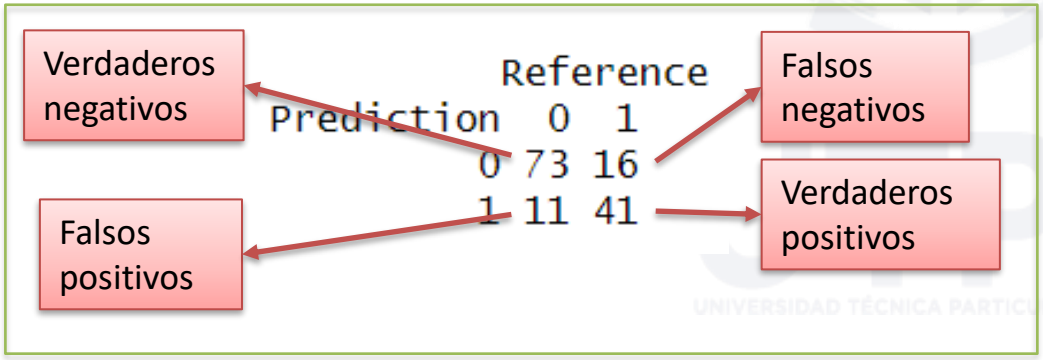
```r
# EVALUAR EL MODELO
# Predecir en el conjunto de prueba
predictions <- predict(model, test_data, type = "response")

# Convertir probabilidades a etiquetas binarias
predicted_classes <- ifelse(predictions > 0.5, 1, 0)

# Matriz de confusión
conf_matrix <- confusionMatrix(factor(predicted_classes),
                               test_data$Survived,
                               positive = "1")

# Ver la matriz de confusión
# y las métricas de evaluación
conf_matrix
```

Devuelve valores en una escala de probabilidades (probabilidad de que se cumpla la hipótesis alternativa)(que Survived = 1)

Exactitud: Proporción de predicciones correctas.

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 73 16
         1 11 41

               Accuracy : 0.8085
                 95% CI : (0.7338, 0.8699)
    No Information Rate : 0.5957
    P-Value [Acc > NIR] : 5.647e-08

                  Kappa : 0.5968

 Mcnemar's Test P-Value : 0.4414

            Sensitivity : 0.7193
            Specificity : 0.8690
```

'Positive' Class : 1



Regresión logística

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 73 16
         1 11 41

               Accuracy : 0.8085
                 95% CI : (0.7338, 0.8699)
    No Information Rate : 0.5957
    P-Value [Acc > NIR] : 5.647e-08

                  Kappa : 0.5968

 Mcnemar's Test P-Value : 0.4414

            Sensitivity : 0.7193
            Specificity : 0.8690
         Pos Pred Value : 0.7885
         Neg Pred Value : 0.8202
             Prevalence : 0.4043
         Detection Rate : 0.2908
   Detection Prevalence : 0.3688
      Balanced Accuracy : 0.7942

       'Positive' Class : 1
```

Exactitud: Proporción de predicciones correctas.

Intervalo de confianza. La precisión estará entre 73.38% y 86.99% el 95% de las veces

Probabilidad de llegar a los mismos resultados por el azar

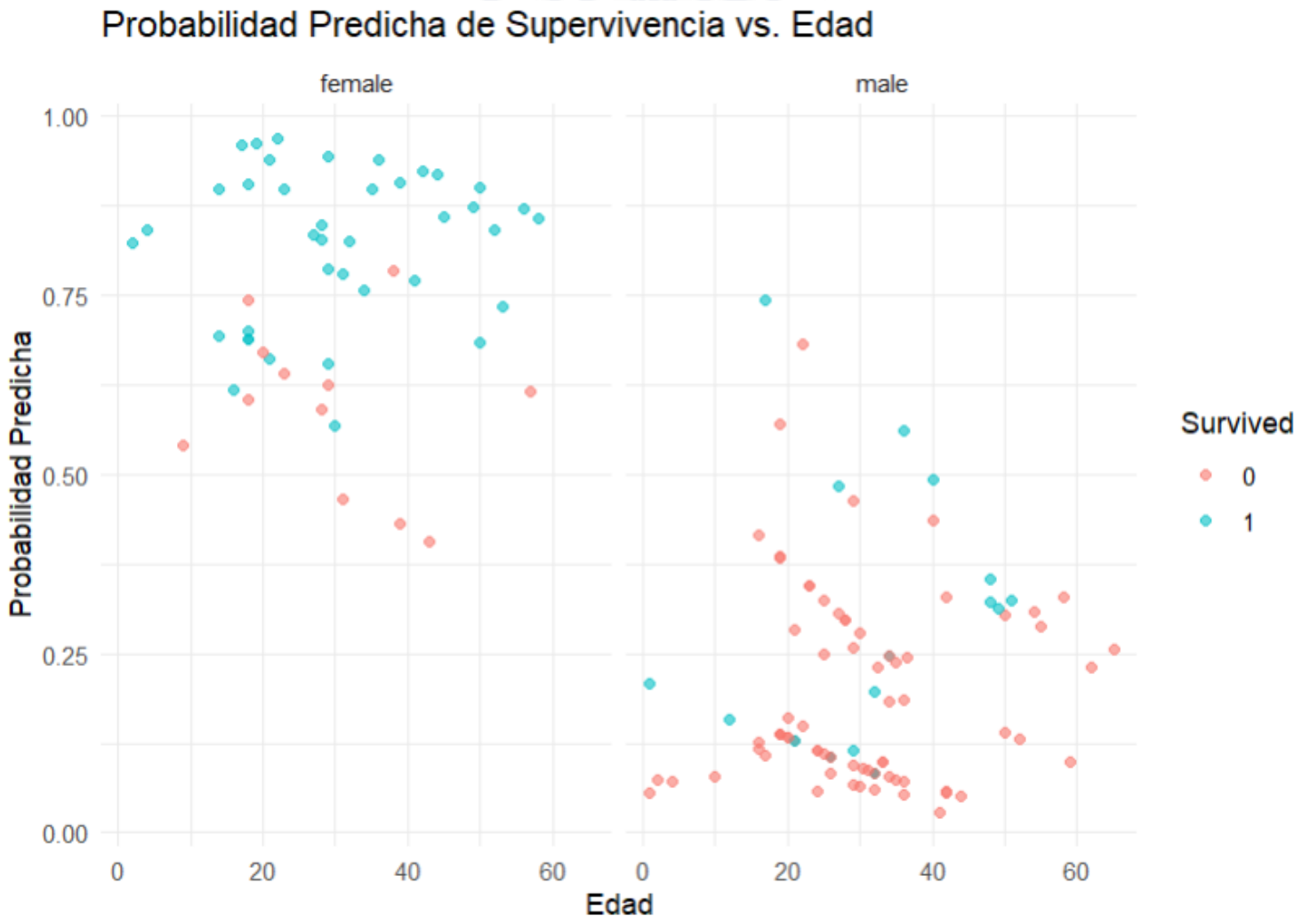
Proporción de verdaderos positivos correctamente detectados (41/57)

Proporción de verdaderos negativos correctamente identificados(73/84)

Regresión logística

```r
# Visualizar los Resultados

# Añadir las predicciones al conjunto de prueba
test_data$predicted_prob <- predictions

# Gráfico de las probabilidades predichas vs. Edad,
# separado por Sexo y Supervivencia
ggplot(test_data, aes(x = Age, y = predicted_prob,
                      color = Survived)) +
  geom_point(alpha = 0.6) +
  facet_wrap(~ Sex) +
  labs(title = "Probabilidad Predicha de Supervivencia vs. Edad",
       x = "Edad",
       y = "Probabilidad Predicha") +
  theme_minimal()
```

Regresión logística



Ejercicio 1

- Desarrollar tarea en clase sobre Regresión Logística

Árboles de decisión

- **Propósito**: Predecir el valor de una variable objetivo (dependiente) basándose en varias variables independiente. Son utilizados tanto para problemas de clasificación como de regresión
- **Descripción**: Es una estructura de árbol donde los nodos internos representan decisiones basadas en las variables independientes, y los nodos terminales (hojas) representan los resultados finales (predicciones).
- **Aplicaciones**: Clasificación de enfermedades a partir de síntomas, segmentación de clientes, pronóstico de ventas, predicción de precios, decisiones de crédito etc.
- **Condiciones**:
  - Aplica para la predicción de **variables categóricas** o **continuas**
  - Si los predictores (variables independientes) son variables categóricas, se las debe **factorizar**.

Árboles de decisión

- Se debe propender a que el árbol resultante guarde equilibrio entre precisión y complejidad. Esto es, el árbol no debe estar:
  - **Sobreajustado**: aumentar complejidad sin mejora significativa de precisión.
  - **Subajustado**: aumentar simplicidad disminuyendo drásticamente la precisión.
- Si es necesario se debe "podar" el árbol para mejorar el modelo.
- **Ejemplo** (dataset "**titanic.csv**"):
  - **Objetivo**: Mediante árboles de decisión construir un modelo que permita predecir el status de supervivencia de cada pasajero, a partir de las variables independientes: *Pclass*, *Sex*, *Age*, *SibSp*, *Parch*, *Fare*, y *Embarked*.
  - **Librería a utilizar**: **rpart** y **rpart.plot**

Árboles de decisión

Clasificación

```r
# Instalar y cargar las librerías necesarias
install.packages("rpart")
install.packages("rpart.plot")
library(dplyr)
library(caret)
library(rpart)
library(rpart.plot)

# Cargar y explorar datos
df <- read.csv("titanic.csv")
head(df)
names(df)
str(df)

# Seleccionar variables relevantes y eliminar valores faltantes
df_clean <- df %>%
  select(Survived, Pclass, Sex, Age, SibSp, Parch, Fare, Embarked)

sapply(df_clean,function(x) mean(is.na(x) | !nzchar(x)))

df_clean <- df_clean %>%
  filter(!is.na(Age) & nzchar(Embarked))

# Convertir variables categóricas a factores
df_clean$Survived <- as.factor(df_clean$Survived)
df_clean$Pclass <- as.factor(df_clean$Pclass)
df_clean$Sex <- as.factor(df_clean$Sex)
df_clean$Embarked <- as.factor(df_clean$Embarked)
str(df_clean)
```

Árboles de decisión

Clasificación

```r
# Dividir los datos de entrenamiento (80%) y de Prueba (20%)
set.seed(88)
train_index <- createDataPartition(df_clean$Survived,
                                   p = 0.8,
                                   list = FALSE)
train_data <- df_clean[train_index, ]   #Datos de entrenamiento
test_data <- df_clean[-train_index, ]   #Datos de prueba

# Ajustar el Modelo de Árbol de Decisión
model <- rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch +
                 Fare + Embarked,
               data = train_data,
               method = "class")
summary(model)

# Visualizar el árbol de decisión
rpart.plot(model,extra = 104)

# EVALUAR EL MODELO
# Predicciones en el conjunto de prueba
predictions <- predict(model, test_data, type = "class")

# Matriz de confusión
conf_matrix <- confusionMatrix(predictions,
                               test_data$Survived,
                               positive = "1")
print(conf_matrix)
```

Indica que es clasificación y no regresión

Árboles de decisión

Clasificación

```
Call:
rpart(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
    Fare + Embarked, data = train_data, method = "class")
  n= 571

          CP nsplit rel error    xerror       xstd
1 0.45887446      0 1.0000000 1.0000000 0.05077096
2 0.03030303      1 0.5411255 0.5411255 0.04277523
3 0.02597403      2 0.5108225 0.5757576 0.04372530
4 0.02164502      3 0.4848485 0.5367965 0.04265153
5 0.01731602      4 0.4632035 0.5281385 0.04240070
6 0.01298701      5 0.4458874 0.5238095 0.04227357
7 0.01000000      7 0.4199134 0.5411255 0.04277523

Variable importance
     Sex   Pclass     Fare      Age    Parch    SibSp Embarked
      50       15       13       11        7        3        1

Node number 1: 571 observations,    complexity param=0.4588745
  predicted class=0  expected loss=0.4045534  P(node) =1
    class counts:   340   231
   probabilities: 0.595 0.405
  left son=2 (363 obs) right son=3 (208 obs)
  Primary splits:
      Sex      splits as  RL,           improve=80.276740, (0 missing)
      Pclass   splits as  RRL,          improve=30.558450, (0 missing)
      Fare     < 52.2771  to the left,  improve=30.223080, (0 missing)
      Embarked splits as  RLL,          improve=13.516790, (0 missing)
      Age      < 5.5      to the right, improve= 8.729757, (0 missing)
  Surrogate splits:
      Parch < 0.5      to the left,  agree=0.669, adj=0.091, (0 split)
      Fare  < 64.17915 to the left,  agree=0.664, adj=0.077, (0 split)
      Age   < 15.5     to the right, agree=0.639, adj=0.010, (0 split)
```

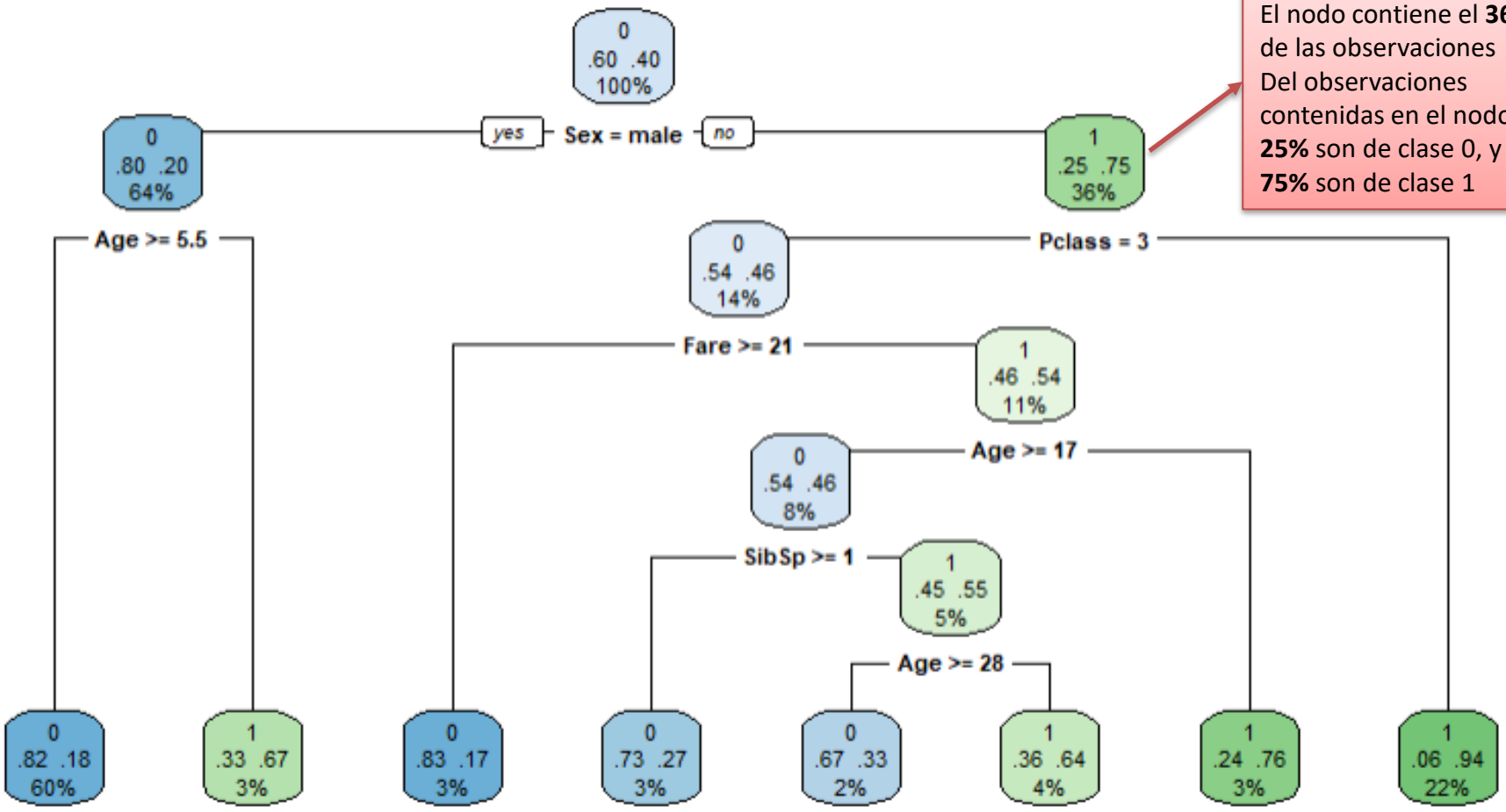
Parámetro de complejidad. Costo de añadir complejidad al modelo. Mientras más bajo, el árbol es más grande. Ayuda a prevenir el sobreajuste.

Error de validación cruzada comparado con el error del árbol sin ninguna división. El valor óptimo de CP es el que minimiza este error. Es donde hay mayor equilibrio entre complejidad y precisión

Importancia relativa de cada variable predictora en el modelo.

Árboles de decisión

Clasificación



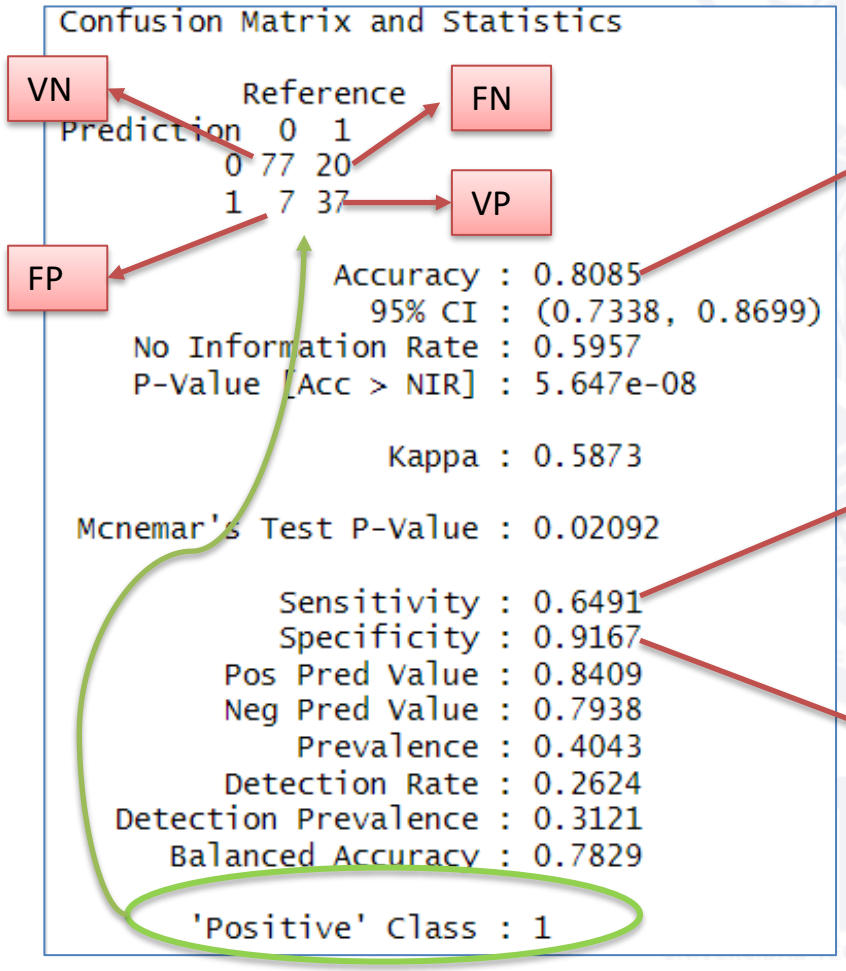
El **color** indica la clase predicha. Y la **saturación** el nivel de pureza del nodo (confianza)

Árboles de decisión

- Respecto a la condiciones establecidas en cada nodo
  - El árbol elige la variable y el punto de corte (threshold) para dividir los datos, que reduce más impureza al dividir el nodo.
  - Impureza significa que hay una mezcla de clases dentro de un nodo. Cuanto más mezcladas estén mayor la impureza**.**
  - El algoritmo explora todos los puntos posibles de corte (basados en los valores que existen en los datos), y elige el que mejor separa las clases en ese nodo. No es un valor arbitrario: es el valor óptimo que reduce más la impureza en ese momento del árbol.

Árboles de decisión

Clasificación



Precisión o exactitud: Proporción de predicciones correctas

Sensibilidad: Proporción de verdaderos positivos sobre todos los casos reales positivos VP / (VP + FN).
En este caso: El modelo es capaz de identificar correctamente el 65% de las instancias positivas (supervivientes).

Especificidad: Proporción de verdaderos negativos sobre todos los casos reales negativos VN / (VN + FP).
En este caso: El modelo es capaz de identificar correctamente el 92% de las instancias negativas (fallecidos)

Árboles de decisión

Clasificación

```r
# PODAR EL ÁRBOL BASADO EN EL VALOR ÓPTIMO DE CP

cp_optimo <- model$cptable[which.min(model$cptable[,"xerror"]), "CP"]

p_model <- prune(model, cp = cp_optimo)

summary(p_model)

# Visualizar el árbol podado
rpart.plot(p_model)


# Evaluar el Modelo Podado
# Predicciones en el conjunto de prueba con el modelo podado
predictions <- predict(p_model, test_data, type = "class")

# Matriz de confusión
conf_matrix <- confusionMatrix(predictions, test_data$Survived)
print(conf_matrix)
```

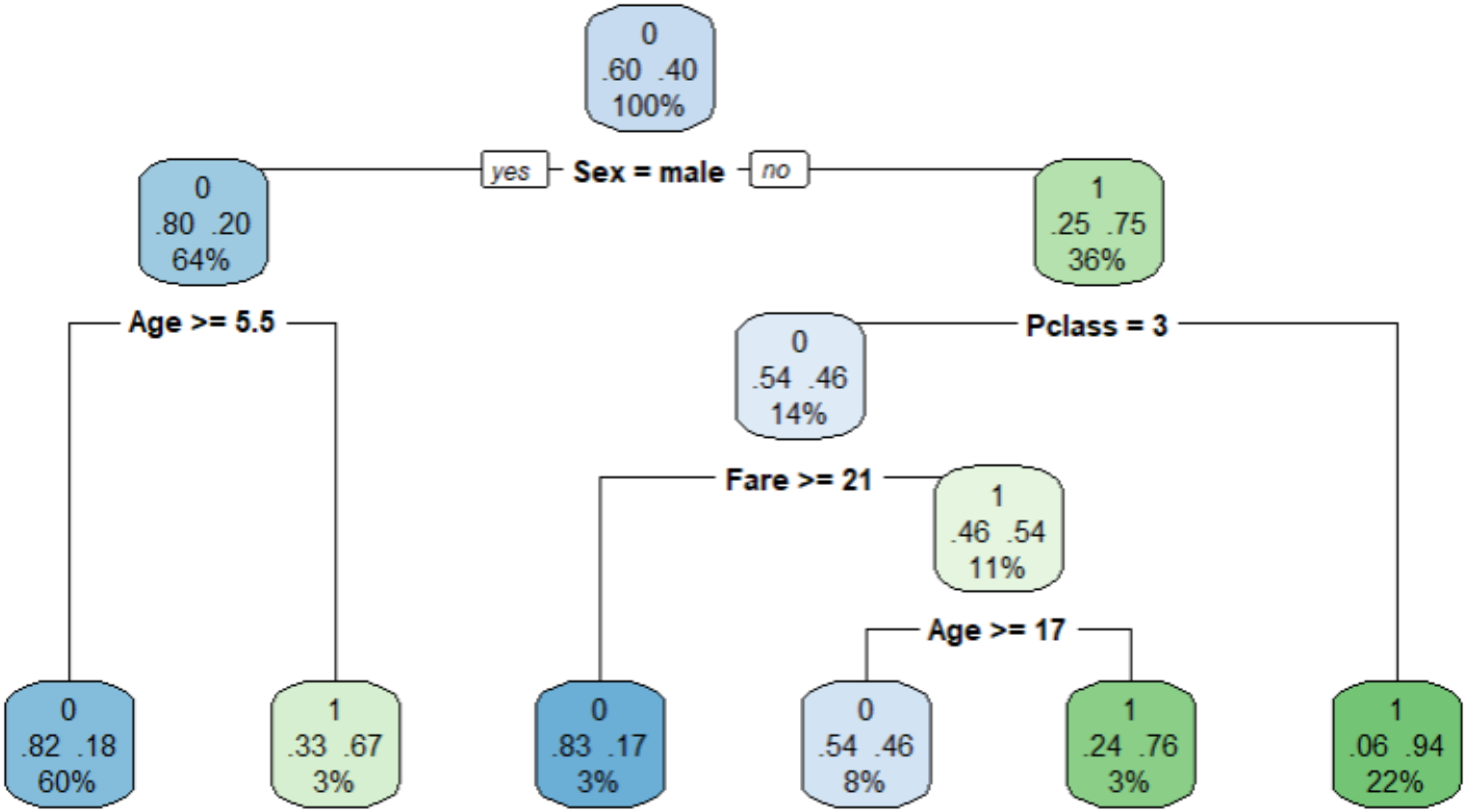
Árboles de decisión

Clasificación

```
Call:
rpart(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
    Fare + Embarked, data = train_data, method = "class")
  n= 571

          CP nsplit rel error    xerror       xstd
1 0.45887446      0 1.0000000 1.0000000 0.05077096
2 0.03030303      1 0.5411255 0.5411255 0.04277523
3 0.02597403      2 0.5108225 0.5757576 0.04372530
4 0.02164502      3 0.4848485 0.5367965 0.04265153
5 0.01731602      4 0.4632035 0.5281385 0.04240070
6 0.01298701      5 0.4458874 0.5238095 0.04227357

Variable importance
     Sex   Pclass     Fare      Age    Parch    SibSp Embarked
      51       15       13       10        7        2        1

Node number 1: 571 observations,    complexity param=0.4588745
  predicted class=0  expected loss=0.4045534  P(node) =1
    class counts:   340   231
   probabilities: 0.595 0.405
  left son=2 (363 obs) right son=3 (208 obs)
  Primary splits:
      Sex      splits as  RL,           improve=80.276740, (0 missing)
      Pclass   splits as  RRL,          improve=30.558450, (0 missing)
      Fare     < 52.2771  to the left,  improve=30.223080, (0 missing)
      Embarked splits as  RLL,          improve=13.516790, (0 missing)
      Age      < 5.5      to the right, improve= 8.729757, (0 missing)
  Surrogate splits:
      Parch < 0.5      to the left,  agree=0.669, adj=0.091, (0 split)
      Fare  < 64.17915 to the left,  agree=0.664, adj=0.077, (0 split)
      Age   < 15.5     to the right, agree=0.639, adj=0.010, (0 split)
```

Árboles de decisión

Clasificación



Árboles de decisión

Clasificación

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 80 23
         1  4 34

               Accuracy : 0.8085
                 95% CI : (0.7338, 0.8699)
    No Information Rate : 0.5957
    P-Value [Acc > NIR] : 5.647e-08

                  Kappa : 0.5799

 Mcnemar's Test P-Value : 0.000532

            Sensitivity : 0.5965
            Specificity : 0.9524
         Pos Pred Value : 0.8947
         Neg Pred Value : 0.7767
             Prevalence : 0.4043
         Detection Rate : 0.2411
   Detection Prevalence : 0.2695
      Balanced Accuracy : 0.7744

       'Positive' Class : 1
```

No se logra una mejora en la precisión pero si se logra reducir la complejidad del árbol

Ejercicio 2

- Desarrollar la PARTE 1 de tarea en clase sobre Árboles de decisión.