Predicción de ingresos

Eduardo Chamizo

2025-06-22

Resumen

Este proyecto desarrolla un modelo predictivo para clasificar si una persona gana más de $50,000 al año basándose en características demográficas y laborales del Census Income Dataset (Adult). Utilizamos XGBoost, un algoritmo de gradient boosting, alcanzando una precisión del 82% en el conjunto de prueba.

Objetivo del problema

Predecir si el ingreso anual de una persona supera los $50,000.

-Segmentación de mercado para productos financieros -Políticas públicas de asistencia social -Análisis de equidad salarial

Instalar y cargar librerías necesarias

```
library(OpenML)
library(xgboost)
library(dplyr)
```

Descargar el dataset “adult” de OpenML (ID = 1590)

```
adult_data <- getOMLDataSet(data.id = 1590)
adult <- adult_data$data
```

Preprocesamiento básico

Convertir target y quitar columnas con 0 en la mayoría de entradas

```
adult$class <- ifelse(adult$class == ">50K", 1, 0)
sapply(adult, unique)
```

###Las variables “fnlwgt”, “capital.gain”, “capital.loss” se quitarán; la de nacionalidad, se reajustará como procede:

```
sort(table(adult$native.country))
```

```
##
##         Holand-Netherlands                     Hungary
##                          1                          19
##                   Honduras                    Scotland
##                         20                          21
## Outlying-US(Guam-USVI-etc)                        Laos
##                         23                          23
##                  Yugoslavia             Trinadad&Tobago
##                         23                          27
##                   Cambodia                    Thailand
##                         28                          30
##                       Hong                     Ireland
##                         30                          37
##                     France                     Ecuador
##                         38                          45
##                       Peru                      Greece
##                         46                          49
##                  Nicaragua                        Iran
##                         49                          59
##                     Taiwan                    Portugal
##                         65                          67
##                      Haiti                    Columbia
##                         75                          85
##                    Vietnam                      Poland
##                         86                          87
##                  Guatemala                       Japan
##                         88                          92
##         Dominican-Republic                       Italy
##                        103                         105
##                    Jamaica                       South
##                        106                         115
##                      China                     England
##                        122                         127
##                       Cuba                       India
##                        138                         151
##                El-Salvador                      Canada
##                        155                         182
##                Puerto-Rico                     Germany
##                        184                         206
##                Philippines                      Mexico
##                        295                         951
##              United-States
##                      43832
```

```
adult$native.country<-ifelse(adult$native.country=="United-States","United States","Rest of the world")
adult<-adult[,c(-3,-11,-12)]
```

Convertir el conjunto de variables explicativas en matriz

```
adultmatrix <- as.matrix(adult[,-12])
adultclass<-adult$class
```

Partición train-test

```
set.seed(123)
n = nrow(adult)
ind = sample(n,n*2/3)
train.adultmatrix = adultmatrix[ind,]
train.adultclass = adultclass[ind]
test.adultmatrix = adultmatrix[-ind,]
test.adultclass = adultclass[-ind]
nrow(train.adultmatrix)==length(train.adultclass)
```

```
## [1] TRUE
```

Limpiar datos

```
train_data <- adultmatrix[ind, ]
train.adultclass <- adultclass[ind]
complete_rows <- complete.cases(train_data)
train_data_clean <- train_data[complete_rows, ]
train.adultclass <- train.adultclass[complete_rows]

test_data <- adultmatrix[-ind, ]
test.adultclass <- adultclass[-ind]
complete_rows_test <- complete.cases(test_data)
test_data_clean <- test_data[complete_rows_test, ]
test.adultclass <- test.adultclass[complete_rows_test]
```

Transformar de tabla de datos a matrices

```
train.adultmatrix <- model.matrix(~ . - 1, data = as.data.frame(train_data_clean))
test.adultmatrix <- model.matrix(~ . - 1, data = as.data.frame(test_data_clean))
```

Limpiar etiquetas de entrenamiento y de prueba

```
sum(is.na(train.adultclass))
```

```
## [1] 0
```

```
sum(is.infinite(train.adultclass))
```

```
## [1] 0
```

```
valid_labels <- !is.na(train.adultclass) &
               !is.infinite(train.adultclass) &
               train.adultclass %in% c(0, 1)
train.adultmatrix <- train.adultmatrix[valid_labels, ]
train.adultclass <- train.adultclass[valid_labels]

valid_labels_test <- !is.na(test.adultclass) &
                     !is.infinite(test.adultclass) &
                     test.adultclass %in% c(0, 1)
test.adultmatrix <- test.adultmatrix[valid_labels_test, ]
test.adultclass <- test.adultclass[valid_labels_test]
```

Convertir etiquetas a numérico

```
train.adultclass <- as.numeric(train.adultclass)
test.adultclass <- as.numeric(test.adultclass)
```

Alinear características entre conjuntos

```
common_features <- intersect(colnames(train.adultmatrix), colnames(test.adultmatrix))
train.adultmatrix <- train.adultmatrix[, common_features, drop = FALSE]
test.adultmatrix <- test.adultmatrix[, common_features, drop = FALSE]
```

Convertir a matriz

```
train.adultmatrix <- as.matrix(train.adultmatrix)
test.adultmatrix <- as.matrix(test.adultmatrix)
```

Análisis exploratorio de datos

Estructura de datos

```
str(adult)
```

```
## 'data.frame':    48842 obs. of  12 variables:
##  $ age           : num  25 38 28 44 18 34 29 63 24 55 ...
##  $ workclass     : Factor w/ 8 levels "Private","Self-emp-not-inc",..: 1 1 5 1 NA 1 NA 2 1 1 ...
##  $ education     : Factor w/ 16 levels "Bachelors","Some-college",..: 3 4 6 2 2 13 4 5 2 9 ...
##  $ education.num : num  7 9 12 10 10 6 9 15 10 4 ...
```

```
##  $ marital.status: Factor w/ 7 levels "Married-civ-spouse",..: 3 1 1 1 3 3 3 1 3 1 ...
##  $ occupation    : Factor w/ 14 levels "Tech-support",..: 8 10 13 8 NA 3 NA 6 3 2 ...
##  $ relationship  : Factor w/ 6 levels "Wife","Own-child",..: 2 3 3 3 2 4 6 3 6 3 ...
##  $ race          : Factor w/ 5 levels "White","Asian-Pac-Islander",..: 5 1 1 5 1 1 5 1 1 1 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 2 2 1 2 ...
##  $ hours.per.week: num  40 50 40 40 30 30 40 32 40 10 ...
##  $ native.country: chr  "United States" "United States" "United States" "United States" ...
##  $ class         : num  0 0 1 1 0 0 0 1 0 0 ...
```

```
summary(adult)
```

```
##       age                     workclass            education     education.num
##  Min.   :17.00   Private         :33906   HS-grad     :15784   Min.   : 1.00
##  1st Qu.:28.00   Self-emp-not-inc: 3862   Some-college:10878   1st Qu.: 9.00
##  Median :37.00   Local-gov       : 3136   Bachelors   : 8025   Median :10.00
##  Mean   :38.64   State-gov       : 1981   Masters     : 2657   Mean   :10.08
##  3rd Qu.:48.00   Self-emp-inc    : 1695   Assoc-voc   : 2061   3rd Qu.:12.00
##  Max.   :90.00   (Other)         : 1463   11th        : 1812   Max.   :16.00
##                  NA's            : 2799   (Other)     : 7625
##                marital.status           occupation            relationship
##  Married-civ-spouse   :22379   Prof-specialty : 6172   Wife          : 2331
##  Divorced             : 6633   Craft-repair   : 6112   Own-child     : 7581
##  Never-married        :16117   Exec-managerial: 6086   Husband       :19716
##  Separated            : 1530   Adm-clerical   : 5611   Not-in-family :12583
##  Widowed              : 1518   Sales          : 5504   Other-relative: 1506
##  Married-spouse-absent:  628   (Other)        :16548   Unmarried     : 5125
##  Married-AF-spouse    :   37   NA's           : 2809
##                  race           sex        hours.per.week  native.country
##  White             :41762   Female:16192   Min.   : 1.00   Length:48842
##  Asian-Pac-Islander: 1519   Male  :32650   1st Qu.:40.00   Class :character
##  Amer-Indian-Eskimo:  470                  Median :40.00   Mode  :character
##  Other             :  406                  Mean   :40.42
##  Black             : 4685                  3rd Qu.:45.00
##                                            Max.   :99.00
##
##      class
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2393
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

Distribución de la variable objetivo

```
prop.table(table(adultclass))
```

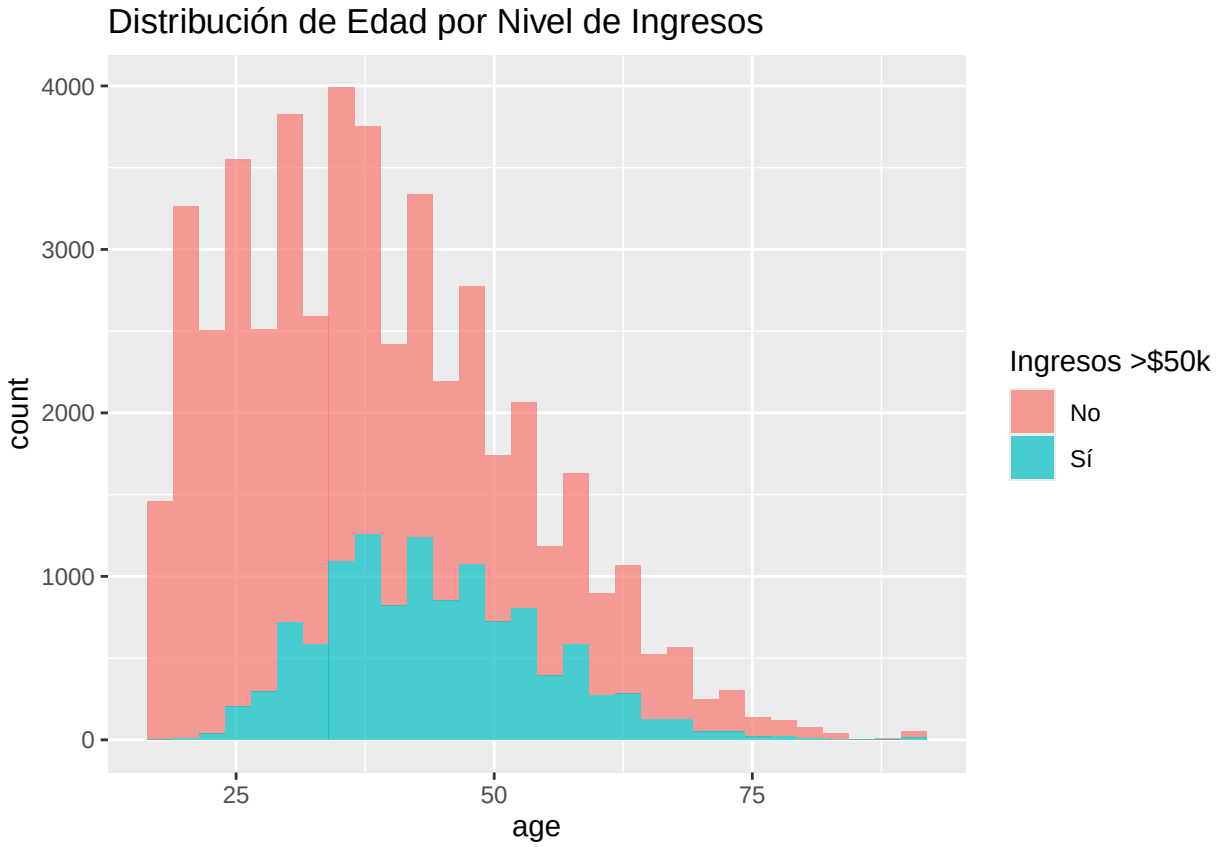
```
## adultclass
##         0         1
## 0.7607182 0.2392818
```

Gráficos para la visualización

Distribución por edad

```
library(ggplot2)
library(dplyr)

ggplot(data.frame(age = adult$age, target = adultclass),
       aes(x = age, fill = factor(target))) +
  geom_histogram(bins = 30, alpha = 0.7) +
  labs(title = "Distribución de Edad por Nivel de Ingresos",
       fill = "Ingresos >$50k") +
  scale_fill_discrete(labels = c("No", "Sí"))
```

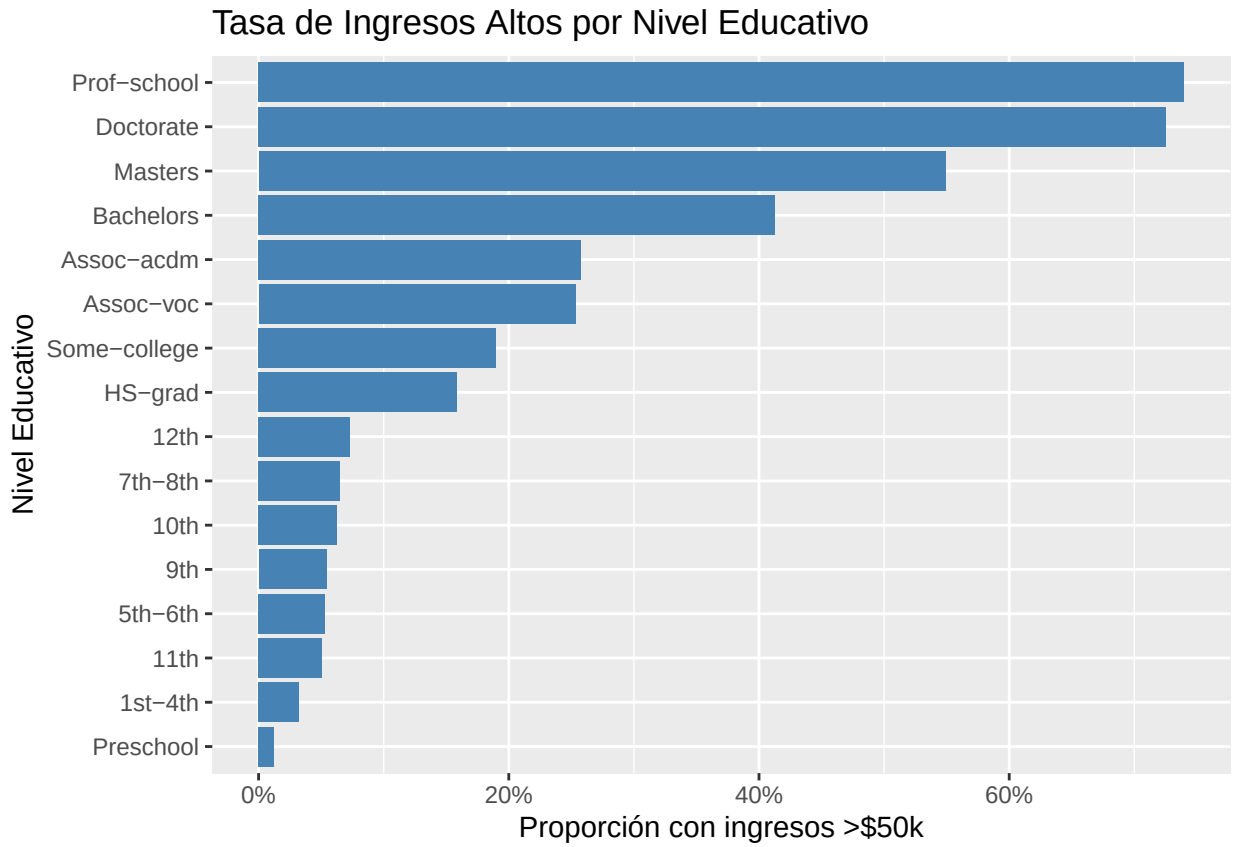


Conforme se avanza en edad, menor es el ingreso. La franja de edad más común en el estudio es entre 25 y 50.

Ingresos por nivel educativo

```
adult_analysis <- adult
adult_analysis$target <- adultclass
adult_analysis %>%
  group_by(education) %>%
  summarise(high_income_rate = mean(target == 1), .groups = 'drop') %>%
```

```r
ggplot(aes(x = reorder(education, high_income_rate), y = high_income_rate)) +
geom_col(fill = "steelblue") +
coord_flip() +
labs(title = "Tasa de Ingresos Altos por Nivel Educativo",
     x = "Nivel Educativo",
     y = "Proporción con ingresos >$50k") +
scale_y_continuous(labels = scales::percent)
```



La educación superior está fuertemente correlacionada con mayor salario. En etapas superiores, el salto salarial es más notable, con excepción de los dos últimos niveles.

Modelo

```r
modelo <- xgboost(data = train.adultmatrix,
                  label = train.adultclass,
                  max.depth = 4,
                  eta = 0.3,
                  nthread = 2,
                  nround = 20,
                  objective = "binary:logistic")
```

- Excelente rendimiento en datos tabulares.
- Manejo automático de valores faltantes.

- Resistente al overfitting.
- Proporciona importancia de variables.

Predicciones

```
pred_train <- predict(modelo, train.adultmatrix)
pred_test <- predict(modelo, test.adultmatrix)
```

Convertir probabilidades a clases

```
pred_train_class <- ifelse(pred_train > 0.5, 1, 0)
pred_test_class <- ifelse(pred_test > 0.5, 1, 0)
```

Evaluar rendimiento

```
train_accuracy <- mean(pred_train_class == train.adultclass)
test_accuracy <- mean(pred_test_class == test.adultclass)
cat("Train Accuracy:", round(train_accuracy, 4), "\n")
```

```
## Train Accuracy: 0.8313
```

```
cat("Test Accuracy:", round(test_accuracy, 4), "\n")
```

```
## Test Accuracy: 0.8288
```

Matriz de confusión

```
table(pred_test_class,test.adultclass,dnn = c("Predicted class","Actual class"))
```

```
##                Actual class
## Predicted class     0     1
##               0 10590  1796
##               1   789  1922
```

Curva ROC

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.3.3
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

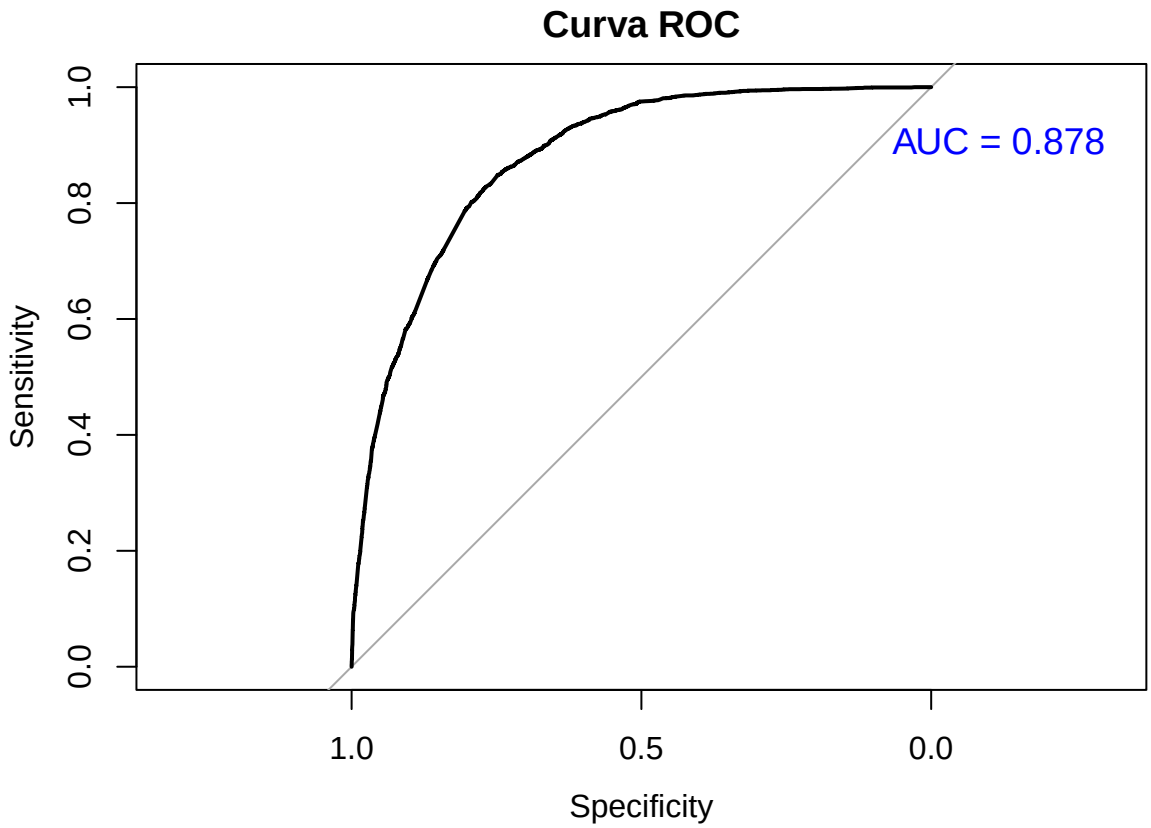
```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
roc_curve <- roc(test.adultclass, pred_test)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

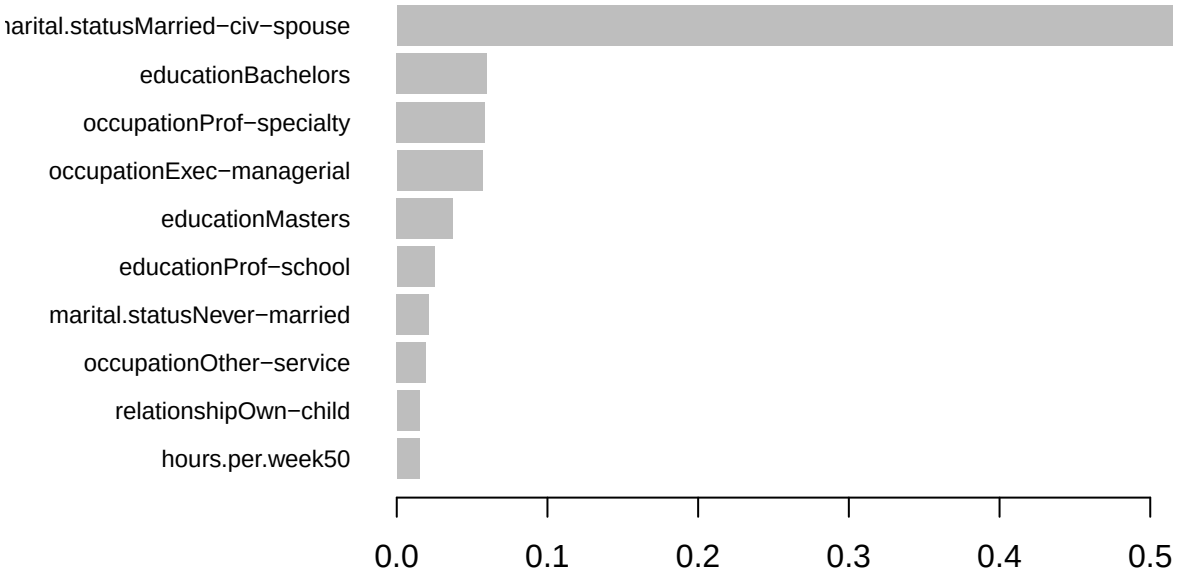
```
plot(roc_curve, main = "Curva ROC")
auc_score <- auc(roc_curve)
text(x = 0.1, y = 0.9,
     labels = paste("AUC =", round(auc_score, 3)),
     pos = 4, col = "blue", cex = 1.2)
```



La curva es relativamente buena. El área bajo la curva tiene un valor de 0.881, lo cual es buen indicativo porque al acercarse a 1, discrimina con cierto éxito entre ambas clases.

Intepretabilidad del modelo

```
importance <- xgb.importance(model = modelo)
xgb.plot.importance(importance, top_n = 10)
```




```
print(importance[1:5])
```

```
##                              Feature       Gain      Cover  Frequency Importance
##                                <char>      <num>      <num>      <num>      <num>
## 1: marital.statusMarried-civ-spouse 0.51467176 0.13425540 0.05035971 0.51467176
## 2:               educationBachelors 0.05978177 0.05287016 0.06115108 0.05978177
## 3:         occupationProf-specialty 0.05863173 0.04788896 0.03597122 0.05863173
## 4:        occupationExec-managerial 0.05685556 0.04910965 0.03956835 0.05685556
## 5:                 educationMasters 0.03717275 0.04676640 0.05035971 0.03717275
```

- Vemos que el factor más determinante con diferencia a la hora de ganar más o menos de $50,000 es el hecho de estar casado o no.
- Otros motivos que influencian la variable objetivo son los niveles superiores de estudio y la ocupación.

Conclusiones

- El modelo logra una exactitud satisfactoria para asignar el atributo buscado a cada nuevo individuo, de 82%

- El estado civil es el predictor con más peso para la estimación.
- Como limitación, mencionar que se obvian variables que podrían ser fundamentales como la localización del puesto de trabajo de cada muestra de la tabla. También mencionar el desbalanceo de datos pertenecientes a una clase con respecto a la otra (sólo un 24% ganan más de $50,000).
- De cara al futuro, sería interesante probar otros algoritmos como redes neuronales o árboles de decisión.