

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

MAT281

APLICACIONES DE LA MATEMÁTICA EN INGENIERÍA

---

## Adult Income

---

*Autores:*

Nicolás Boyardi Alache

Adrián López


Rodrigo Pizarro

2022-1

# Índice general

<b>1. Introducción</b>	<b>2</b>
<b>2. Análisis</b>	<b>3</b>
2.1. Limpieza y lectura de datos . . . . .	3
2.2. Otras consideraciones . . . . .	3
2.3. Software y librerías . . . . .	3
2.4. Clasificadores . . . . .	4
2.5. Análisis de los clasificadores . . . . .	5
2.5.1. Validación . . . . .	5
<b>3. Conclusiones</b>	<b>7</b>
3.1. Validación . . . . .	7
3.2. Estado del Arte y rendimiento . . . . .	7
3.3. Conclusiones particulares del problema . . . . .	7
3.4. Bibliografía . . . . .	8
<b>4. Anexo</b>	<b>9</b>
4.1. Librerías . . . . .	9
4.2. Lectura y limpieza de datos . . . . .	9
4.3. Función auxiliar para graficar matrices de confusión . . . . .	10
4.4. Creación conjunto de entrenamiento-test . . . . .	11
4.5. Implementación Naive Bayes . . . . .	11
4.6. Implementación árbol de decisión . . . . .	11
4.7. Implementación vecinos cercanos . . . . .	11
4.8. Implementación regresión logística . . . . .	12
4.9. Curva ROC . . . . .	12

# Introducción

En estudios asociados a la ciencia de datos, es muy útil poseer criterios o condiciones claras para poder predecir, evaluar y categorizar elementos (ya sean cualitativos o cuantitativos) en categorías que no parezcan estar directamente relacionadas con ellos. Debido a lo anterior, el objetivo de este proyecto es lograr clasificar de manera predictiva si una persona gana más de 50K USD al año o no, recibiendo como entradas 14 atributos de su vida personal, laboral y educacional. Para ello, implementaremos algunos algoritmos estudiados en cátedra utilizando el software .

Posterior a la implementación, desarrollaremos y explicaremos los resultados mediante indicadores que son capaces de precisar (de manera cuantitativa) la eficacia de los algoritmos. Con toda esta información, podremos concluir cuál de las opciones propuesta es la más adecuada para el trabajo indicado.

El dataset que utilizaremos se encuentra en

`https://archive.ics.uci.edu/ml/datasets/adult`

el cual recaba información sobre un censo realizado a distintas personas por el “US Census Bureau”, en 1994. En este, se encuentran 48,842 datos, los cuales tienen por atributos una variable respuesta que corresponde a si su sueldo es mayor o igual a 50K USD, mientras que como covariables tenemos la edad, tipo de educación, trabajo, estado civil, país de origen, sexo, entre otros. Es claro apreciar la variedad de las variables, pues tenemos tanto variables cuantitativas desde binarias a continuas, como variables cualitativas nominales y ordinales.

# Análisis

## 2.1. Limpieza y lectura de datos

Es usual que para alguna variable, en un dato en específico, falte información.

En nuestro caso, de los 48 mil datos, menos de 4 mil datos poseían al menos una entrada faltante. Dado que tenemos muchos datos en comparación a otros datasets, optaremos por la solución más simple, pero que suele traer problemas cuando hay pocos datos: eliminar todo dato que posea al menos 1 entrada faltante.

Por otra parte, respecto a la organización y lectura de datos, la variable de estudio en la que nos enfocamos (si gana una cantidad mayor a 50K USD, o si gana una cantidad menor o igual a 50K USD) la transformamos a numérica binaria, representando por 0 a quienes ganan menor o igual, y 1 para el resto. También, trabajamos de manera binaria el sexo indicando por 0 a hombres y 1 a mujeres.

Por otro lado, juntamos las variables “CapitalGain” y “CapitalLoss”, variables que representan la ganancia y pérdida en inversiones respectivamente, en una sola variable continua con posibilidad de ser 0 que reutiliza el nombre de “CapitalGain”. Esto a modo de no repetir variables innecesariamente.


Así mismo, la última adaptación que se realizó fue la de considerar solamente al grupo de personas de América del norte, pues el resto de personas representaban una muy reducida parte de la muestra (menor al 10 % de los datos).

## 2.2. Otras consideraciones

Se eliminaron las variables “Education”, pues esta es equivalente a su numérico “EducationNum”, el cual es una medida ordinal de la educación adquirida por individuo, y la variable “FnlWgt”, pues como indica la bibliografía este es simplemente un indicador que crearon los encuestadores que relaciona a 2 individuos si sus valores son cercanos. No nos proporciona información a-priori.

Por otro lado, consideramos también en eliminar “Relationship” y “Race”, pero análisis a posteriori mostraron una diferencia insignificante.

## 2.3. Software y librerías

Para proceder en lo computacional, como ya se dijo, se hizo uso principalmente del programa de , el cual nos hizo que implementar los modelos fuera casi inmediato. A continuación, se especifican las librerías por modelo:

Clasificador	Librería	Función
Vecinos cercanos	<code>class</code>	<code>knn</code>
Naive Bayes	<code>e1071</code>	<code>naiveBayes</code>
Regresión Logística	<code>stats</code>	<code>glm</code>
Árbol de decisión	<code>rpart</code>	<code>rpart</code>

Por otra parte, es importante destacar que se usaron librerías para poder realizar los gráficos asociados a los análisis posteriores. Todas estas librerías se podrán encontrar en el anexo al final de este documento.

## 2.4. Clasificadores

En general, dado que los problemas de clasificación binaria son relativamente sencillos, no tuvimos mayor problema en adaptar los códigos vistos en clases, lo que resultó que la implementación de cada clasificador no fuera más de un par de decenas de líneas de código.

Los clasificadores utilizados fueron:

- † Vecinos cercanos.
- † Naive Bayes.
- † Regresión logística.
- † Árbol de decisión

Todos los modelos anteriores fueron verificados en calidad a través de distintas técnicas, según la naturaleza del predictor, que se explicitarán en el siguiente punto.

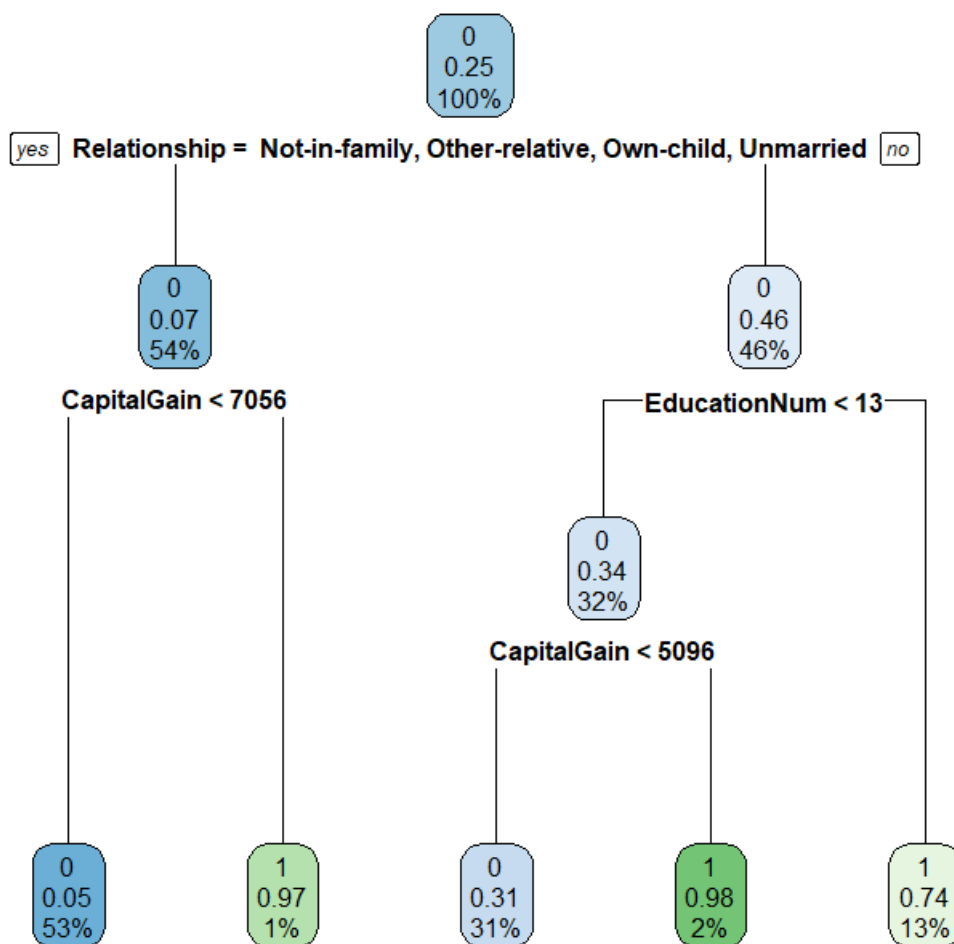
## 2.5. Análisis de los clasificadores

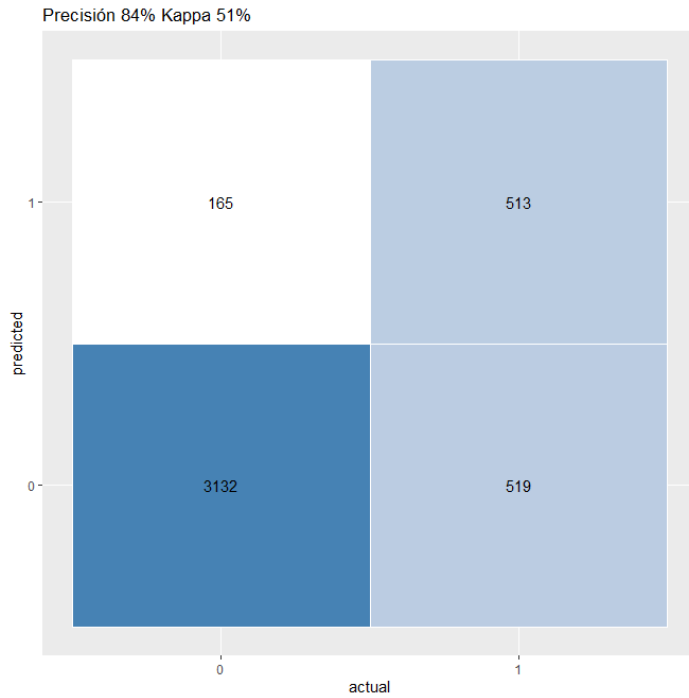
### 2.5.1. Validación

Para el análisis, se usaron 2 técnicas dependiendo de lo que el clasificador arrojaba. En particular, como los modelos del Árbol de decisión, Vecinos cercanos y Bayes Ingenuo generan un aparato predictor, podemos construirles sus respectivas matrices de confusión. Por otro lado, como regresión logística nos retorna probabilidades de que los eventos ocurran, entonces es adecuado usar una curva ROC. Las matrices de confusión las generaremos con la librería “caret”, las curvas ROC se generan con la librería “CURVAS ROC”, y ambas son graficadas usando “ggplot2”.

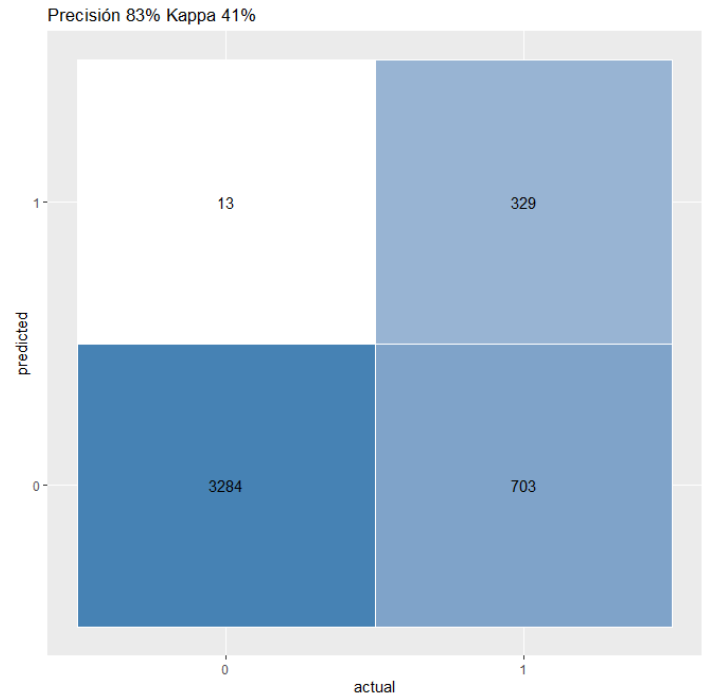
Las conclusiones extraídas de los modelos serán en base a la precisión, que suele ser un buen estadístico para lograr medir la eficacia de un algoritmo en el caso de dos etiquetas (es decir, cuando la matriz de confusión es de  $2 \times 2$ ). A continuación, se presentan tanto el árbol de decisión generado, las matrices de confusión y la curva ROC de cada modelo asociado, según lo explicado anteriormente.

El árbol de clasificación generado fue el siguiente:

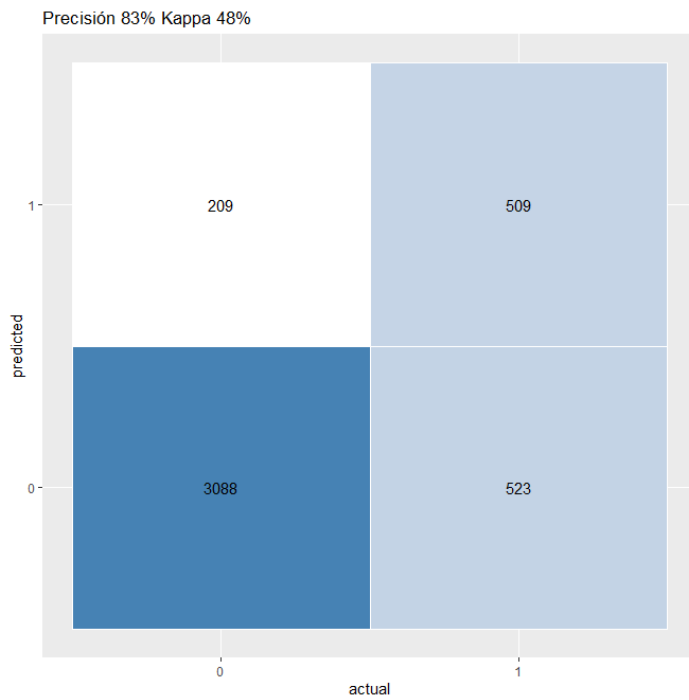




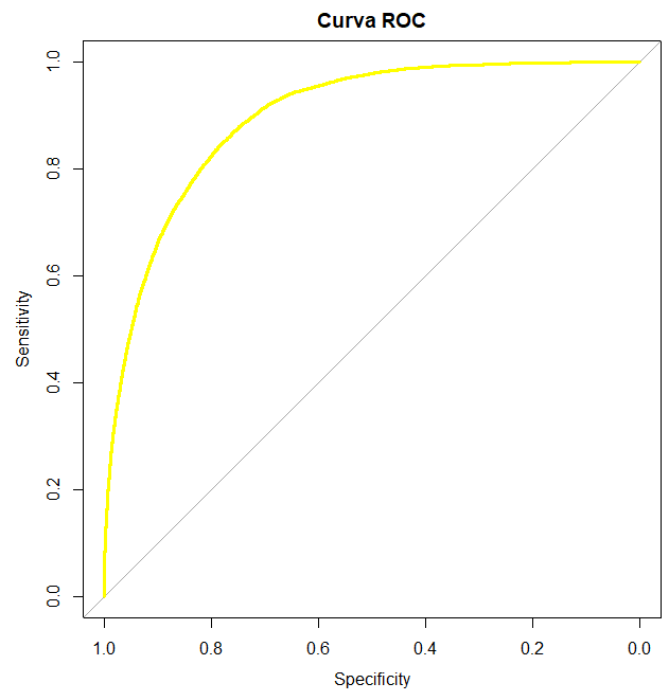
(a) Arbol de decisión



(b) Vecinos cercanos



(c) Naive Bayes



(d) Regresión Logística

## Conclusiones

### 3.1. Validación

Es claro que, con los gráficos y sus porcentajes de acierto todos mayores a 80 %, y la curva roc visiblemente por encima de la diagonal, que se puede decir con seguridad que para los datos usados, los 4 modelos fueron exitosos en su labor predictiva.

### 3.2. Estado del Arte y rendimiento

Como se puede revisar en la descripción oficial de los datos, según

<https://archive.ics.uci.edu/ml/datasets/Adult>

se sabe que los autores ya trabajaron con estos datos, obteniendo “Error de precisión de 83.88 % para Naive Bayes, (...) 85.90 % para árbol de decisión” con el método de validación cruzada. Estos porcentajes son increíblemente cercanos a lo obtenido bajo nuestra implementación, por lo que quedamos satisfechos.

### 3.3. Conclusiones particulares del problema

Ahora que ya sabemos que nuestros modelos son confiables, y que se ajustan excelentemente a la bibliografía original, entonces no nos queda más que concluir con los datos.

Como se puede apreciar en el árbol de decisión, el factor que por lejos más influenció fue el de “CapitalGain”. Esto se deduce pues cuando las personas tienen desde cierto umbral de dinero invertido hacia arriba, éstos en un casi absoluto porcentaje son personas que superan la barrera de los 50K USD al año. Este nos dice cuánto ingreso, o salida, de dinero por inversiones tiene dicha persona. A nosotros, esto nos hizo sentido claro sentido, pues, una persona que tiene capacidad de invertir es aquella que sus ingresos deben ser altos de base. Para invertir, se debe haber sobrepasado la barrera de costo mínimo de vida, y otras barreras económicas, antes de poder tomar dinero que podría perderse totalmente por las impredecibles fluctuaciones del mercado.



### 3.4. Bibliografía

Para poder complementar esta investigación se consultaron las siguientes referencias:

- [1. ] “Data Skills for Reproducible Science” <https://psyteachr.github.io/msc-data-skills/glm.html>.
- [2. ] “R for Data Science”, Hadley Wickham & Garrett Golemund

# Anexo

## 4.1. Librerías

---

```
library(readr)
library(caTools)
library(ggplot2)
library(scales)
library(e1071)
library(caret)
library(rpart)
library(class)
```

---

## 4.2. Lectura y limpieza de datos

---

```
adult1<-read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data",
header = FALSE)
adult2<-read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test",
header = FALSE, skip=1)
adult<-rbind(adult1,adult2)
```

```
names(adult)<-c("Age", "WorkClass", "Fnlwgt", "Education", "EducationNum", "MaritalStatus",
"Occupation", "Relationship", "Race", "Sex", "CapitalGain", "CapitalLoss", "HoursPerWeek",
"NativeCountry", "Income")
```

```
adult$CapitalGain <- (adult$CapitalGain - adult$CapitalLoss)
adult$CapitalGain = as.numeric(as.character(adult$CapitalGain))
adult<-adult[,!names(adult) %in% c("CapitalLoss")]
```

```
adult$Income[adult$Income==' <=50K.']<- ' <=50K'
adult$Income[adult$Income==' >50K.']<- ' >50K'
adult$Income[adult$Income==' <=50K']<-0
adult$Income[adult$Income==' >50K']<-1
adult$Income=as.numeric(as.character(adult$Income))
```

```
adult$Sex[adult$Sex == ' Male'] <- 0
adult$Sex[adult$Sex == ' Female'] <- 1
adult$Sex=as.numeric(as.character(adult$Sex))
```

```
Americadelnorte<-c(" Canada", " Cuba", " Dominican-Republic", " El-Salvador",
```

```
" Guatemala", " Haiti", " Honduras", " Jamaica", " Mexico", " Nicaragua",  
" Outlying-US(Guam-USVI-etc)", " Puerto-Rico", " Trinidad&Tobago", " United-States")  
Otro<-c(" Cambodia", " China", " Hong", " India", " Iran", " Japan", " Laos"  
," Philippines", " Taiwan", " Thailand", " Vietnam", " Columbia", " Ecuador",  
" Peru", " England", " France", " Germany", " Greece", " Holand-Netherlands",  
" Hungary", " Ireland", " Italy", " Poland", " Portugal", " Scotland", " Yugoslavia",  
" South", " ?")  
adult$NativeCountry[adult$NativeCountry %in% Americadelnorte]<-"America del norte"  
adult$NativeCountry[adult$NativeCountry %in% Otro]<-" ?"  
  
adult[adult == ' ?'] <- NA  
adult <- na.omit(adult)  
  
adult<-adult[,!names(adult) %in% c("NativeCountry")]  
adult<-adult[,!names(adult) %in% c("Education")]  
adult<-adult[,!names(adult) %in% c("Fnlwgt")]
```

---

### 4.3. Función auxiliar para graficar matrices de confusión

<sup>1</sup>

---

```
ggplotConfusionMatrix <- function(m){  
  mytitle <- paste("Precisión", percent_format()(m$overall[1]),  
                  "Kappa", percent_format()(m$overall[2]))  
  dat <- as.data.frame(m$table)  
  dat$lab <- ifelse(dat$Freq == 0, '', dat$Freq)  
  p <-  
    ggplot(data = dat ,  
           aes(x = actual, y = predicted)) +  
    geom_tile(aes(fill = log(Freq)), colour = "white") +  
    scale_fill_gradient(low = "white", high = "steelblue") +  
    geom_text(aes(x = actual, y = predicted, label = lab)) +  
    theme(legend.position = "none") +  
    ggtitle(mytitle)  
  return(p)  
}
```

---

<sup>1</sup><https://stackoverflow.com/questions/67946452/how-can-i-improve-this-confusion-matrix-in-r>

## 4.4. Creación conjunto de entrenamiento-test

---

```
set.seed(1)
data<-adult
random<-sample(1:nrow(data), 0.9*nrow(data))
train<-data[random,]
test<-data[-random,]
```

---

## 4.5. Implementación Naive Bayes

---

```
modelo_nb<-naiveBayes(train$Income~., data=train)
pred_nb<-predict(modelo_nb,test)
cm_nb<-confusionMatrix(pred_nb, as.factor(test$Income), dnn=c("predicted","actual"))
ggplotConfusionMatrix(cm_nb)
```

---

## 4.6. Implementación árbol de decisión

---

```
modelo_ad<-rpart(as.factor(train$Income) ~., data=train)
pred_ad<-predict(modelo_ad, test, type="class")
cm_ad<- confusionMatrix(pred_ad, as.factor(test$Income), dnn=c("predicted","actual"))
ggplotConfusionMatrix(cm_ad)
```

---

## 4.7. Implementación vecinos cercanos

---

```
modelo_knn<-knn ( train = cbind(train$Age,train$Sex,train$CapitalGain) ,
test = cbind (test$Age,test$Sex,test$CapitalGain) ,cl = train$Income ,k =7)
cm_knn<-confusionMatrix(modelo_knn,as.factor(test$Income), dnn=c("predicted","actual"))
ggplotConfusionMatrix(cm_knn)
```

---

## 4.8. Implementación regresión logística

---

```
w <- adult$Fnlwgt
attach(adult)
modelo <- glm(Income ~ Age+WorkClass+EducationNum+MaritalStatus+
              Ocupation+Sex+CapitalGain+HoursPerWeek, family=binomial(link="logit"))
```

---

## 4.9. Curva ROC

---

```
x <- adult[c("Age", "WorkClass", "EducationNum", "MaritalStatus",
            "Ocupation", "Sex", "CapitalGain", "HoursPerWeek")]
labels <- adult$Income
scores <- predict(modelo, x)
scores
plot(roc(labels, scores, direction="<"),
     col="yellow", lwd=3, main="Curva ROC")
```

---