Taller 2 Introducción analítica

```
#Lectura de librerías
library(ISLR)
library(corrplot)
library(pls)
library(gam)
library(knitr)
library(ggplot2)
library(pander)
```

Ejercicio 1

Se busca predecir el número de solicitudes recibidas en una universidad.

Validacion cruzada:

Se particionan los datos en un conjunto de entrenamiento y otro de validación:

```
#Validacion cruzada
set.seed(123)
size = ceiling(nrow(College)*0.8)
training = sample(1:nrow(College), size)
test = which(!1:nrow(College) %in% training)
```

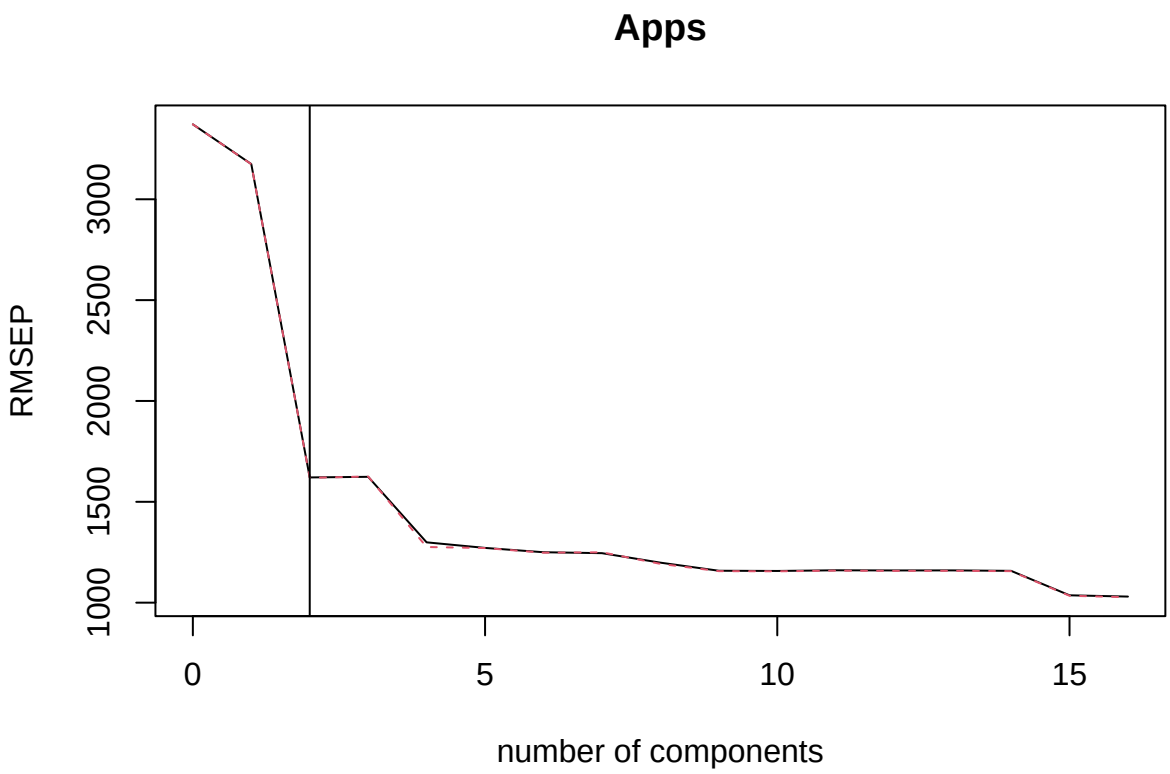
Ajuste de modelo PCR:

Se plantea un modelo de regresión por componentes principales en donde se incluyen todas las variables númericas de la base de datos:

```
set.seed(123)
#Ajuste modelo PCR
pcr.fit <- pcr(formula = Apps ~ .,
               data = College[2:18],
               subset = training,
               scale = TRUE,
               validation = 'CV')
```

Se realiza una gráfica de validación para seleccionar mediante validación cruzada el número M de componentes principales a incluir en el modelo:

```
#Grafica de validacion
validationplot(pcr.fit, val.type = 'RMSEP')
abline(v=2)
```



A partir de la anterior gráfica se elije el modelo con dos componentes principales.

Se cálcula el error de test obtenido con el M seleccionado de 2:

```
#Modelo PCR
ajuste_pcr <- predict(pcr.fit, College[test, 2:18], ncomp = 2)
PCR <- mean((ajuste_pcr-College[test, 'Apps'])^2)
```

Se obtiene que el error de test para el modelo con 2 componentes principales es de 8.2237916×10^6.

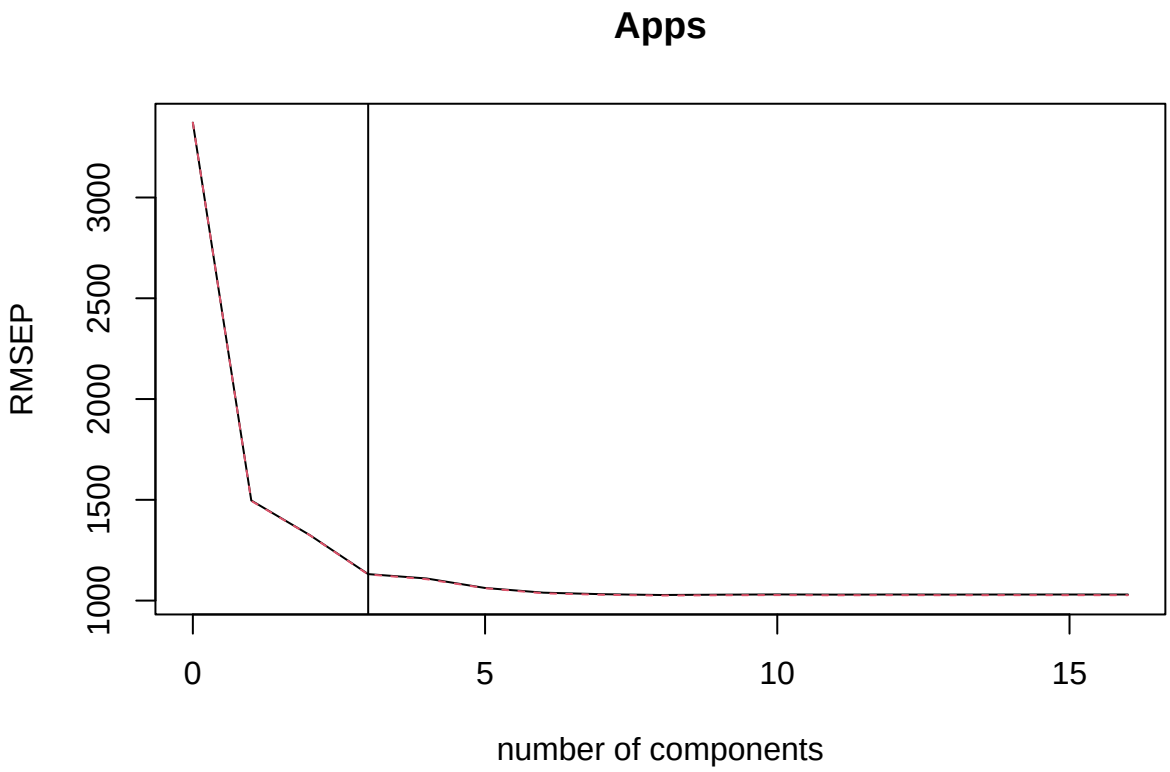
Ajuste de modelo PLS:

En este caso se plantea el modelo de regresión con todas las variables que incluye la base de datos, realizando además validación cruzada:

```
set.seed(123)
#Ajuste modelo PLS
pls.fit <- plsr(formula = Apps ~ .,
                data = College[2:18],
                subset = training,
                scale = TRUE,
                validation = 'CV')
```

Se selecciona el nùmero de componentes a través de validación cruzada:

```
#Grafica de validacion
validationplot(pls.fit, val.type = 'RMSEP')
abline(v=3)
```



En base a la gráfica nos quedamos con el modelo con 3 componentes.

Se ajusta el modelo a los datos de prueba y se cálcula su error:

```
#Modelo PLS
ajuste_pls <- predict(pls.fit, College[test, 2:18], ncomp = 3)
PLS <- mean((ajuste_pls-College[test, 'Apps'])^2)
```

En este caso se obtiene que el error de test para el modelo con 3 componentes es de 4.9149881×10^6.

```
kable(t(c(PCR, PLS)), col.names = c('PCR', 'PLS'))
```

PCR	PLS
8223792	4914988

De acuerdo a la validación cruzada el modelo que muestra mejores resultados es aquel compuesto por la regresión lineal parcial.

Ejercicio 4:

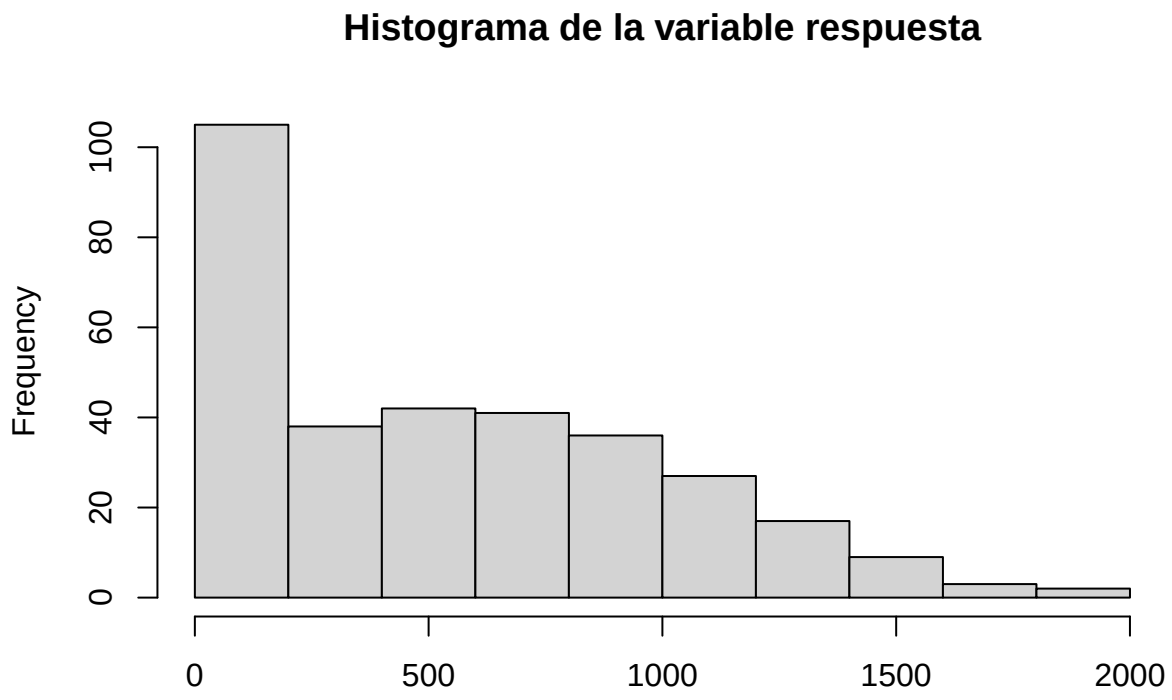
Se realiza un análisis descriptivo ligero para hallar las posibles variables regresoras de los modelos a plantear:

Se crean los conjuntos de entrenamiento y prueba para la validación cruzada:

```
#Validacion cruzada
set.seed(123)
size = ceiling(nrow(Credit)*0.8)
training = sample(1:nrow(Credit), size)
test = which(!1:nrow(Credit) %in% training)
```

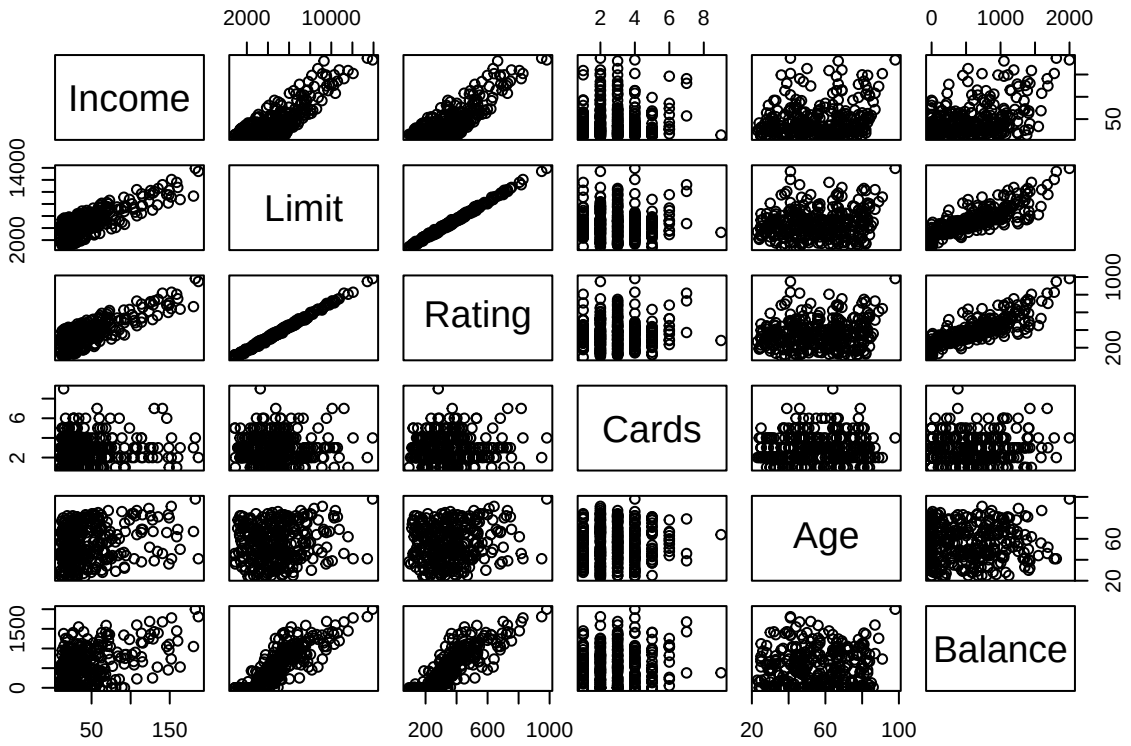
Se observa ligeramente el comportamiento de la variable respuesta:

```
#Adicionalmente se separa la variable respuesta:
y_train <- Credit[training, 'Balance']
#Distribucion variable respuesta:
hist(y_train, main ='Histograma de la variable respuesta', xlab = '')
```



Se plantean histogramas mediante la función pairs para observar el cambio en la variable respuesta respecto a las variables númericas en los datos de prueba.

```
#Diagramas de dispersion:
pairs(Credit[training, c(2:6, 12)])
```



De la gráfica anterior se observa que hay un par de variables altamente correlacionadas con el balance.

Se plantea una gráfica de correlaciones para observar la correlación entre el Balance y las demás variables númericas del conjunto de datos de prueba:

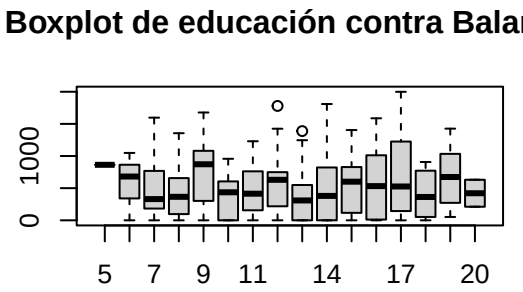
```
#Grafica de correlacion:
cor <- cor(Credit[training, c(2:6, 12)])
corrplot::corrplot(cor, method = c('number'))
```

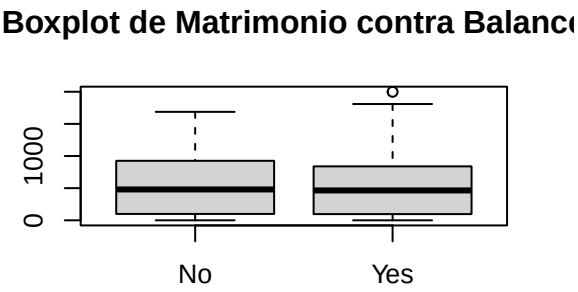
De manera rápida se eligen aquellas variables cuya correlación con la variable respuesta es de 0.8 o más cómo las variables regresoras del modelo, estas variables son el límite crediticio y el puntaje crediticio.

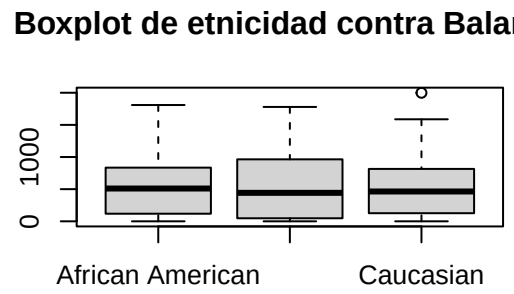
Se plantean diagramas de cajas y bigotes para encontrar relaciones entre la variable respuesta y las variables categóricas en el conjunto de datos de prueba:

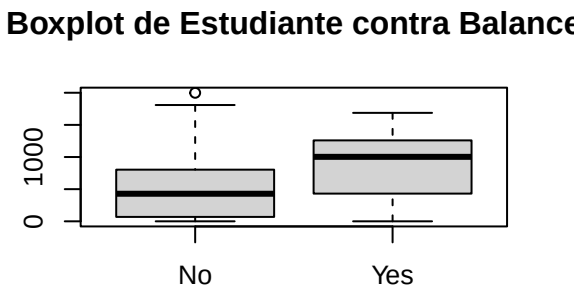
```
#Boxplots:
par(mfrow = c(2,2))

boxplot(Credit[training, 'Balance'] ~ Credit[training, 'Education'],
        ylab='', xlab='', main = 'Boxplot de educación contra Balance')
boxplot(Credit[training, 'Balance'] ~ Credit[training, 'Married'],
        ylab='', xlab='', main = 'Boxplot de Matrimonio contra Balance')
boxplot(Credit[training, 'Balance'] ~ Credit[training, 'Ethnicity'],
        ylab='', xlab='', main = 'Boxplot de etnicidad contra Balance')
boxplot(Credit[training, 'Balance'] ~ Credit[training, 'Student'],
        ylab='', xlab='', main = 'Boxplot de Estudiante contra Balance')
```





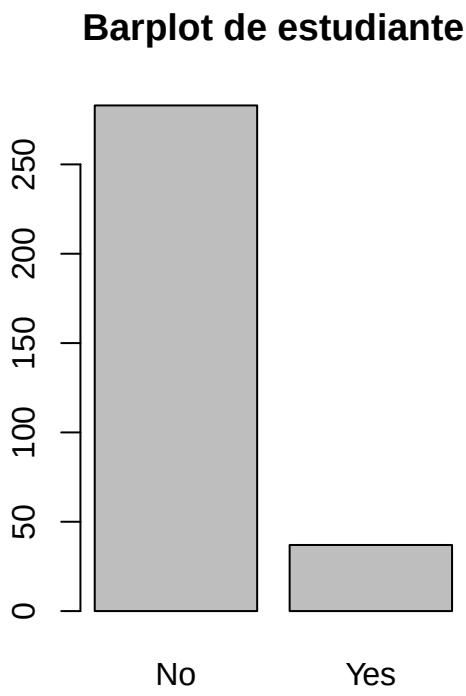


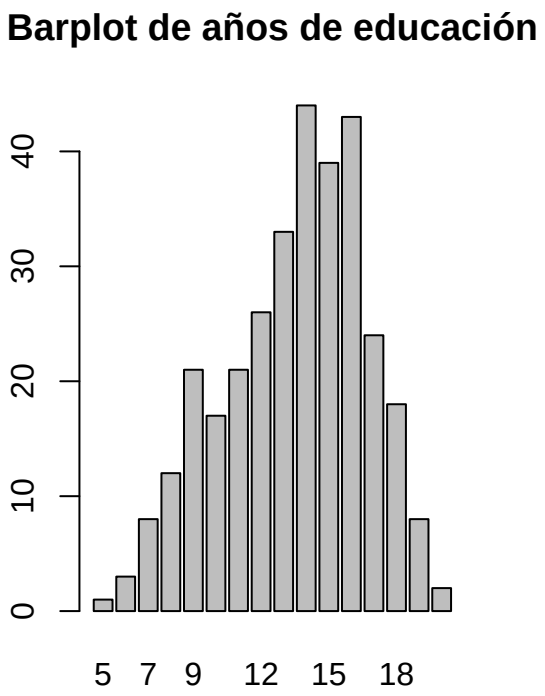


En los diagramas de cajas y bigotes se puede observar que las variables categóricas en las que a priori se relacionen con el valor del balance son la variable Estudiante y la variable años de educación.

Se construye un par de gráficas de barras para evaluar el balanceo en estas variables:

```
par(mfrow = c(1,2))
barplot(table(Credit[training, 'Student']), main = 'Barplot de estudiante')
barplot(table(Credit[training, 'Education']), main = 'Barplot de años de educación')
```





Se observa desbalanceo en la variable estudiante, para este caso este desbalanceo no es tratado ya que se asume que la naturaleza de los datos permite justificar la inclusión de esta variable cómo característica en los futuros modelos.

Finalmente se seleccionan las siguientes variables para modelar el Balance en la base de datos:

- Puntaje crediticio
- Limite crediticio
- Años de estudio
- Condición de estudiante (Estudiante o no estudiante)

Construcción de análisis de varianza de modelos anidados:

Se plantean 4 modelos gam con los cuales se busca modelar el Balance crediticio en un análisis de varianza de modelos anidados, se incluyen poco a poco variables buscando que al disminuir los grados de libertad mediante el modelo anterior incluyendo nuevas características se obtenga una mejora significativa en el ajuste de los datos:

```
#Modelamiento:
#Balance en base a las covariables usando loess en una covariable
mod_gam1 <- gam(Balance ~ lo(Limit),
                data = Credit,
                subset = training)
#Balance en base a las covariables usando splines y loess en las covariables
mod_gam2 <- gam(Balance ~ lo(Limit) + s(Rating),
                data = Credit,
```

```
                subset = training)
#Balance en base a las covariables usando loess, splines y anadiendo una variable categorica
mod_gam3 <- gam(Balance ~ lo(Limit) + s(Rating) + Student,
                data = Credit,
                subset = training)
#Balance en base a las covariables usando loess y splines
mod_gam4 <- gam(Balance ~ lo(Limit) + s(Rating) + Student + Education,
                data = Credit,
                subset = training)

pander::pander(anova(mod_gam1, mod_gam2,mod_gam3, mod_gam4, test = 'F'))
```

Table 2: Analysis of Deviance Table

Resid. Df	Resid. Dev	Df	Deviance	F	Pr(>F)
314.6	15833119	NA	NA	NA	NA
310.6	15564546	4.001	268573	1.907	0.1091
309.6	10863212	1	4701334	133.6	6.513e-26
308.6	10861739	1	1473	0.04186	0.838

De la tabla anterior se concluye que el tercer modelo es posiblemente el candidat a elegir ya que es el que según su estadístico F presenta significancia respecto a los anteriores modelos.

```
errores1 <- predict(mod_gam1, Credit[test, c('Limit', 'Rating', 'Student', 'Education')])
errores2 <- predict(mod_gam2, Credit[test, c('Limit', 'Rating', 'Student', 'Education')])
errores3 <- predict(mod_gam3, Credit[test, c('Limit', 'Rating', 'Student', 'Education')])
errores4 <- predict(mod_gam4, Credit[test, c('Limit', 'Rating', 'Student', 'Education')])

errores1 <- mean((Credit[test, 'Balance'] - errores1)^2)
errores2 <- mean((Credit[test, 'Balance'] - errores2)^2)
errores3 <- mean((Credit[test, 'Balance'] - errores3)^2)
errores4 <- mean((Credit[test, 'Balance'] - errores4)^2)
```

Se construye una tabla anova del modelo elegido:

```
pander::pander(summary(mod_gam3)$anova)
```

Table 3: Anova for Nonparametric Effects

	Npar Df	Npar F	Pr(F)
(Intercept)	NA	NA	NA
lo(Limit)	3.4	11.37	9.805e-08
s(Rating)	3	2.636	0.04985
Student	NA	NA	NA

De la tabla anterior se observa que en el modelo a plantear el efecto de las variables númericas con suavizadores es significativo.

Finalmente comparamos los errores en el conjunto de datos de prueba:

```
kable(data.frame(errores1, errores2, errores3, errores4),
      col.names = c('Modelo1', 'Modelo2', 'Modelo3', 'Modelo4'))
```

Modelo1	Modelo2	Modelo3	Modelo4
39871.36	38563.04	29894.41	30116.88

Según la validación cruzada el modelo que arroja el menor error en el conjunto de prueba es el modelo 3 en dónde se aplica loess al limite crediticio y splines al ratio crediticio en el conjunto de datos de prueba, esto posiblemente se deba a que al suavizar el conjunto de variables se mejore el ajuste sacrificando interpretabilidad y que cómo se observa de manera gráfica hayan diferencias en el balance crediticio respecto al estatus como estudiante de un individuo.