UNIVERSIDAD NACIONAL DE COLOMBIA

IAA-Modulo 1

Autor:

Daniela Pico

Juan Sebastian Falcón

Jhonatan Smith

Profesor:

Juan Carlos Salazar

2021-02

1) Considere el estadistico Leverage

$$h_{ii} = \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_{j=1}^{n}(x_j - \overline{x})^2}$$

Demuestre que

$$\frac{1}{n} \leq h_{ii} \leq 1$$

Por el lado izquierdo

Sabemos que la diferencia de cuadrados $(x_i - \overline{x})^2$ siempre va ser ≥ 0 por tanto la sumatoria de las diferencias de cuadrados $\sum_{j=1}^{n}(x_j - \overline{x})^2$ tambien sera un numero ≥ 0. Por lo tanto $\frac{(x_i-\overline{x})^2}{\sum_{j=1}^{n}(x_j-\overline{x})^2} \geq 0$. Luego, tomando el caso de que $x_i = \overline{x}$, tendriamos que

$$h_{ii} = \frac{1}{n} + \frac{0}{\sum_{j=1}^{n}(x_j - \overline{x})^2} = \frac{1}{n}$$

Se demuestra que

$$\frac{1}{n} \leq h_{ii}$$

Por el lado derecho

Tenemos la matriz Hat $H = X(X^TX)^{-1}X^T$, cuyas diagonales son los valores h_{ii}

Se tiene la matriz **H** como

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & \cdots & \cdots & h_{nn} \end{bmatrix}$$

Se sabe que la matriz Hat tiene las siguientes propiedades Es una matriz simetrica. $H = H^T$ Es una matriz idenpotente. $H = H^2$

A partir de las anteriores propiedades se soluciona H^2

$$H^2 = \begin{bmatrix} h_{11}^2 + \sum_{i \neq j}^{n} h_{ij}^2 & \cdots & \cdots & \cdots \\ \vdots & h_{22}^2 + \sum_{i \neq j}^{n} h_{ij}^2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \cdots & \cdots & h_{nn}^2 + \sum_{i \neq j}^{n} h_{ij}^2 \end{bmatrix}$$

Observando la diagonal principal de la matriz H^2 tendriamos que

$$h_{ii} = h_{ii}^2 + \sum_{i \neq j}^{n} h_{ij}^2$$

De lo anterior tendrimos que h_{ii} es igual a su cuadrado mas una sumatoria, esto solo seria posible si $h_{ii} \geq h_{ii}^2$, y esto a su vez implica que $h_{ii} \leq 1$.

Se demuestra que

$$h_{ii} \leq 1$$

Finalmente

$$\frac{1}{n} \leq h_{ii} \leq 1$$

2) Considere el conjunto de datos anexo el cual tiene 17 variables. Asuma que el supervisor es la variable Loan

a) Cree un conjunto de datos de entrenamiento del 75% y el restante 25 % tratelo como datos de test o de prueba

```
bank <- read.csv(file.choose(), sep = ",")
```

Se procede a realizar la division de la base de datos.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
set.seed(1039705595) # Se selecciona una semilla para la extraccion de la muestra aleatoria
datab <- sort(sample(nrow(bank), nrow(bank)*.75))
# Datos de entrenamiento
train<-bank[datab,]
# Datos de prueba
test<-bank[-datab,]
ytrain<-bank[datab,8]
ytest<-bank[-datab,8]
newdata1<- train %>% dplyr::select(-8)
newdata2<- test %>% dplyr::select(-8)
```

b) Se ajusta Naive Bayes con los datos de entrenamiento. En R...

```
library(naivebayes)
```

```
## Warning: package 'naivebayes' was built under R version 4.0.5
```

```
## naivebayes 0.9.7 loaded
```

```
modelo_NB <- naive_bayes(loan ~ ., data = train)
```

c) Para implementar Knn se debe utilizar variables indicadoras para las categoricas y normalizar las continuas. Tenga presente que de las 17 variables de la base de datos 9 (8 sin contar la supervisora) son categoricas.

Primero, se implementan las variables indicadoras (Dummies) y luego se normalizan las continuas.

```
## Warning: package 'ISLR' was built under R version 4.0.5
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'bank':
##
##     housing
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Warning: package 'Amelia' was built under R version 4.0.5
```

```
## Loading required package: Rcpp
```

```
## Warning: package 'Rcpp' was built under R version 4.0.5
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.0, built: 2021-05-26)
## ## Copyright (C) 2005-2021 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
## Warning: package 'pscl' was built under R version 4.0.5
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
## Warning: package 'caret' was built under R version 4.0.5
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Loading required package: lattice
```

```
## Warning: package 'pROC' was built under R version 4.0.5
```

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

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

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

```
## Warning: package 'ROCR' was built under R version 4.0.5
```

```
## Warning: package 'wesanderson' was built under R version 4.0.5
```

```
## Warning: package 'e1071' was built under R version 4.0.5
```

```
## Warning: package 'mlbench' was built under R version 4.0.5
```

```
## Warning: package 'InformationValue' was built under R version 4.0.5
```

```
##
## Attaching package: 'InformationValue'
```

```
## The following objects are masked from 'package:caret':
##
##     confusionMatrix, precision, sensitivity, specificity
```

Se construye un df con todas las variable para luego sacar de alli los datos de prueba y entrenamiento dada las indicadoras y las varuables normalizadas.

Dado el siguiente codigo, se normalizan las variables continuas, pues se requieren bajo la misma escala para proceder.

```
normalize <- function(x) {
norm <- ((x - min(x))/(max(x) - min(x)))
return (norm)
}
train2<-(newdata[newdatasort,])
train2n<-normalize(train2[-50])
test2<-(newdata[-newdatasort,])
test2n<-normalize(test2[-50])
# Datos de entrenamiento para Knn
ytrain2<-newdata[newdatasort,50]
# Datos de prueba para Knn
ytest2<-newdata[-newdatasort,50]
```

Se simulan knn con k de 1 a 10

```
for (i in 1:10) {
# Se modela con los datos de entrenamiento con un k=i
mod2train<- knn(train = train2n, test = train2n, cl = ytrain2, k=i, prob=TRUE)
# Se modela con los datos de prueba con un k=i
mod2test<- knn(train = train2n, test = test2n, cl = ytrain2, k=i,prob=TRUE)
tk<-table(mod2train,ytrain2)
t1n<-table(mod2test,ytest2)
# Training error
Training_error_KNN<-(tk[1,2]+tk[2,1])/(sum(tk))
print(table(Training_error_KNN,i))
}
```

```
##                   i
## Training_error_KNN 1
##                  0 1
##                    i
## Training_error_KNN  2
##   0.113009198423127 1
##                    i
## Training_error_KNN  3
##   0.105841595986143 1
##                    i
## Training_error_KNN  4
##   0.122088161509975 1
##                    i
## Training_error_KNN  5
##   0.121132481185044 1
##                    i
## Training_error_KNN  6
##   0.124955202484769 1
##                    i
## Training_error_KNN  7
##   0.126388722972166 1
##                    i
## Training_error_KNN  8
##   0.127344403297097 1
##                    i
## Training_error_KNN  9
##   0.128300083622028 1
##                    i
## Training_error_KNN  10
##   0.129016843865727  1
```

Bajo estos criterios se escoge k = 3 puesto que tiene un error de entrenamiento bajo y un k mas pequeño puede estar sobre ajustando el modelo (como posiblemente sucede en k=1)

Se ajusta el modelo con el K seleccionado.

```
modelo_knn<-knn(train = train2n, test = train2n, cl = ytrain2, k=3, prob=TRUE)
```

d) La Regresion logistica será tal que:

```
modelo_logistico <- glm(as.factor(train2$loan) ~ . ,data=train2n, family = "binomial")
```

e) Se implementa LDA con R:

```
# ytrain: variable loan con 1,0
newdata1<- train %>% dplyr::select(-8)
modelo_LDA=lda(ytrain~., data = newdata1)
```

f) Se calcula la matriz de confusion, el error de training MSE y se grafica.

Todos los siguientes calculos a presentar seran dados con los datos de entrenamiento.

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
##         ytrain
## glm.pred   no  yes
##        0 7270 1075
##        1   22    4
```

```
# Modelo Naive Bayes
tabla_NB_E<-table(train_NB,ytrain)
tabla_NB_E
```

```
##         ytrain
## train_NB   no  yes
##      no  6753  892
##      yes  539  187
```

```
# Knn
tabla_knn_E<-table(modelo_knn,ytrain)
tabla_knn_E
```

```
##           ytrain
## modelo_knn   no  yes
##        no  6906  981
##        yes  386   98
```

```
# Modelo logistico
tabla_lo_E<-table(glm.pred,ytrain)
tabla_lo_E
```

```
##         ytrain
## glm.pred   no  yes
##        0 7270 1075
##        1   22    4
```

```
##          ytrain
## train_lda   no  yes
##         0 7248   40
##         1   44 1039
```

Las tablas anteriores son las respectivas matrices de confusion asociada a cada modelo. Note que los elementos de la diagonal principal son los aciertos de prediccion y cualquier elemento fuera de ella representa un error a la hora de predecir.

LDA es el modelo que mejor acierta a la hora de predecir, luego, seguiria Naive Bayes, Logistico y Knn en ultimo.

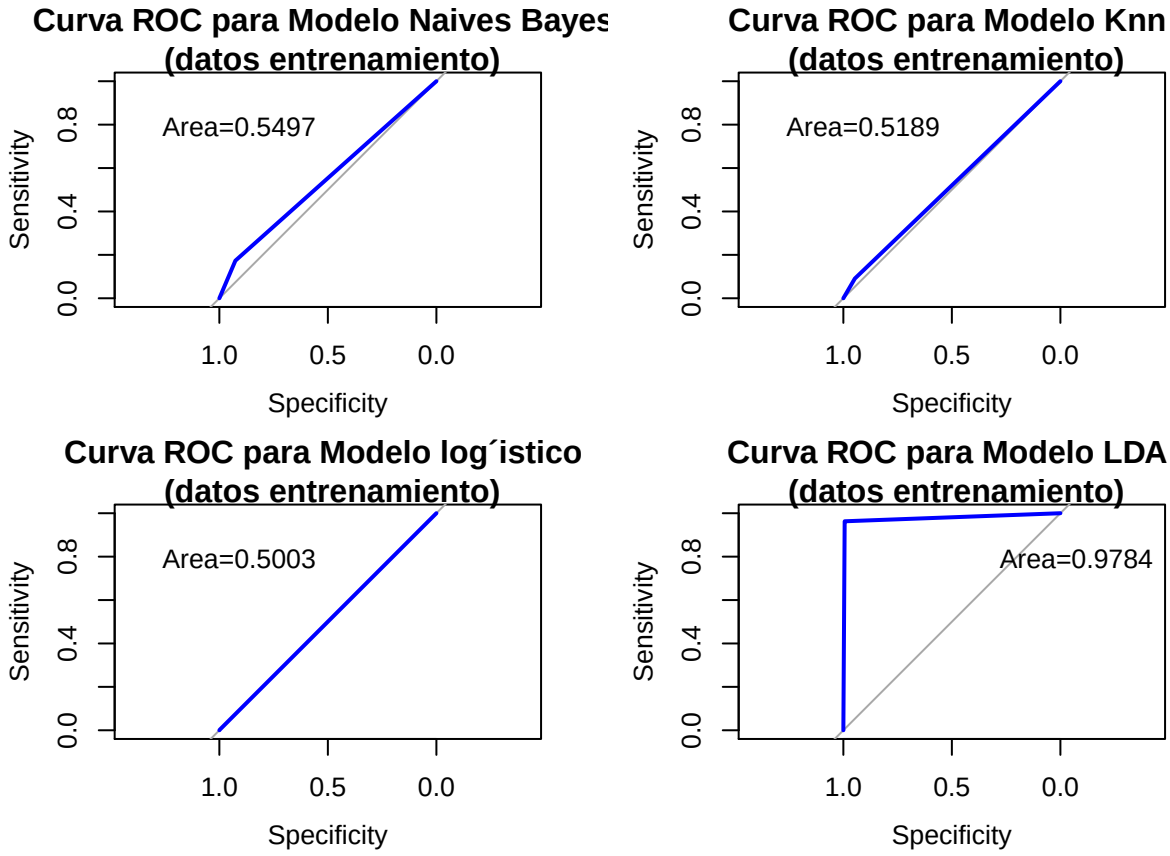
Se calcula el trainning MSE para cada modelo:

```
##   Naive Bayes    Knn Logistico     LDA
## 1     0.17095 0.1633   0.13105 0.01003
```

Y se observa que el modelo con menor MSE es LDA.

Graficas ROC:

```
## Setting levels: control = no, case = yes
## Setting direction: controls < cases
## Setting levels: control = no, case = yes
## Setting direction: controls < cases
## Setting levels: control = no, case = yes
## Setting direction: controls < cases
## Setting levels: control = no, case = yes
## Setting direction: controls < cases
```



El modelo que, mejor resultados ofrece a la hora de predecir segun la corva ROC es el modelo LDA. Basta ver la curva y su valor AUC de casi 100%.

g) La matriz de consfusion para los 4 modelos de prueba es:

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

Matriz de Naive Bayes

```
tabla_NB_T
```

```
##        ytest
## test_NB   no  yes
##     no  2211  308
```

```
##     yes  199   73
```

Matriz de Knn

```
##         ytest
## test_knn   no  yes
##      no  2218  352
##      yes  192   29
```

Matris de Regresion Logistica

```
##          ytest
## test_logs   no  yes
##         1 2410  381
```

En este resultado, el valor asignado a la segunda fila (1) es cero. Implicando que la matriz es de la 2x1, 2 columnas y una fila

Matriz LDA

```
##         ytest
## test_LDA   no  yes
##        0 2389    9
##        1   21  372
```

Calculo del MSE para cada modelo:

Para Naiev Bayes

```
MSE_NB_T = (tabla_NB_T[1,2]+tabla_NB_T[2,1])/sum(tabla_NB_T[,])
MSE_NB_T
```

```
## [1] 0.1816553
```

Para Knn

```
MSE_Knn_t = (tabla_knn_T[1,2]+tabla_knn_T[2,1])/sum(tabla_knn_T[,])
MSE_Knn_t
```

```
## [1] 0.1949122
```

Para Regresion Losgistica:

```
MSE_RL =  (tabla_lo_T[1,2])/sum(tabla_NB_T[,])
MSE_RL
```

```
## [1] 0.1365102
```

Para LDA:

```
MSE_LDA = (tabla_LDA_T[1,2]+tabla_LDA_T[2,1])/sum(tabla_LDA_T[,])
MSE_LDA
```

```
## [1] 0.01074884
```

El MSE mas pequeño es de LDA.

Finalmente, graficas ROC:

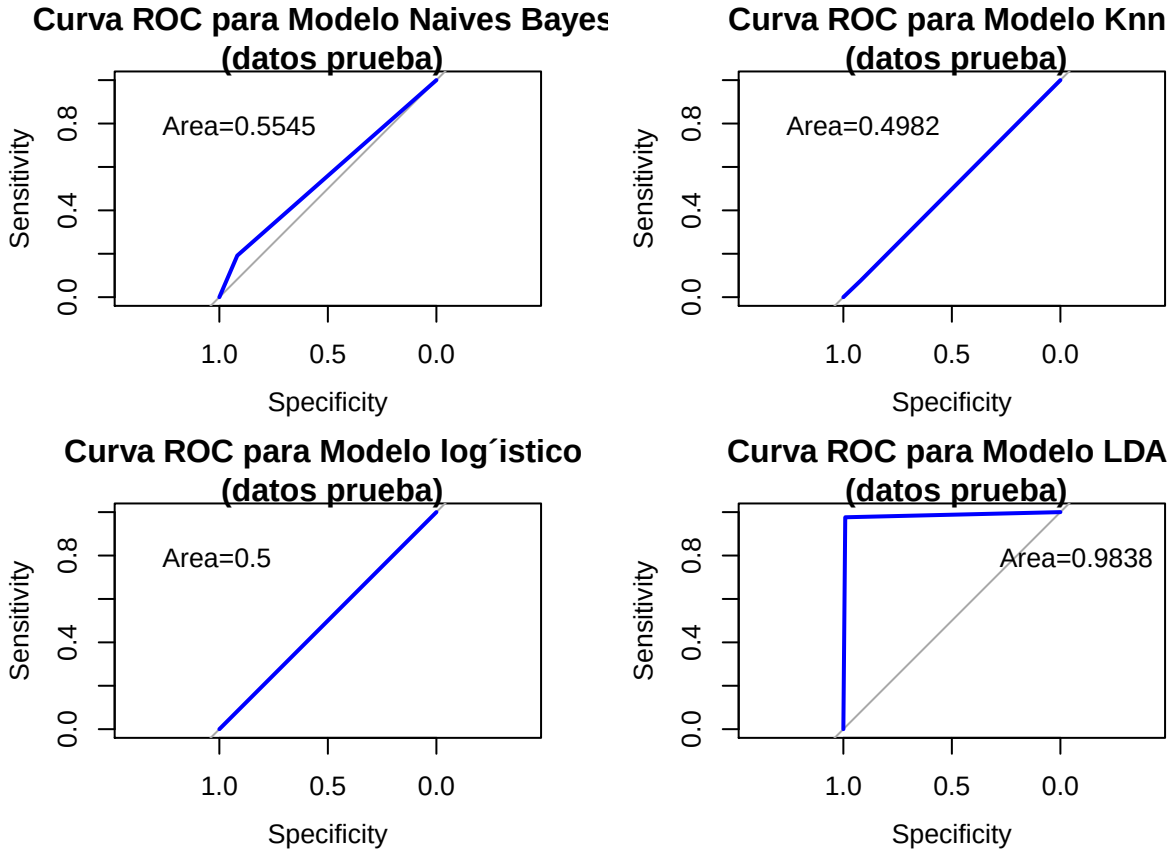
```
## Setting levels: control = no, case = yes
```

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

```
## Setting levels: control = no, case = yes
```

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

```
## Setting levels: control = no, case = yes
## Setting direction: controls < cases
## Setting levels: control = no, case = yes
## Setting direction: controls < cases
```



H) El modelo con mejor desempeño fue el del modelo de LDA dada las estimaciones de sus parametros, teniendo en cuenta que fue el que mejor MSE ha tenido a lo largo de todas las pruebas.

3) Considere el conjunto de datos anexo el cual contiene 12 variables incluyendo ID. Asuma que el supervisor es la variable income.

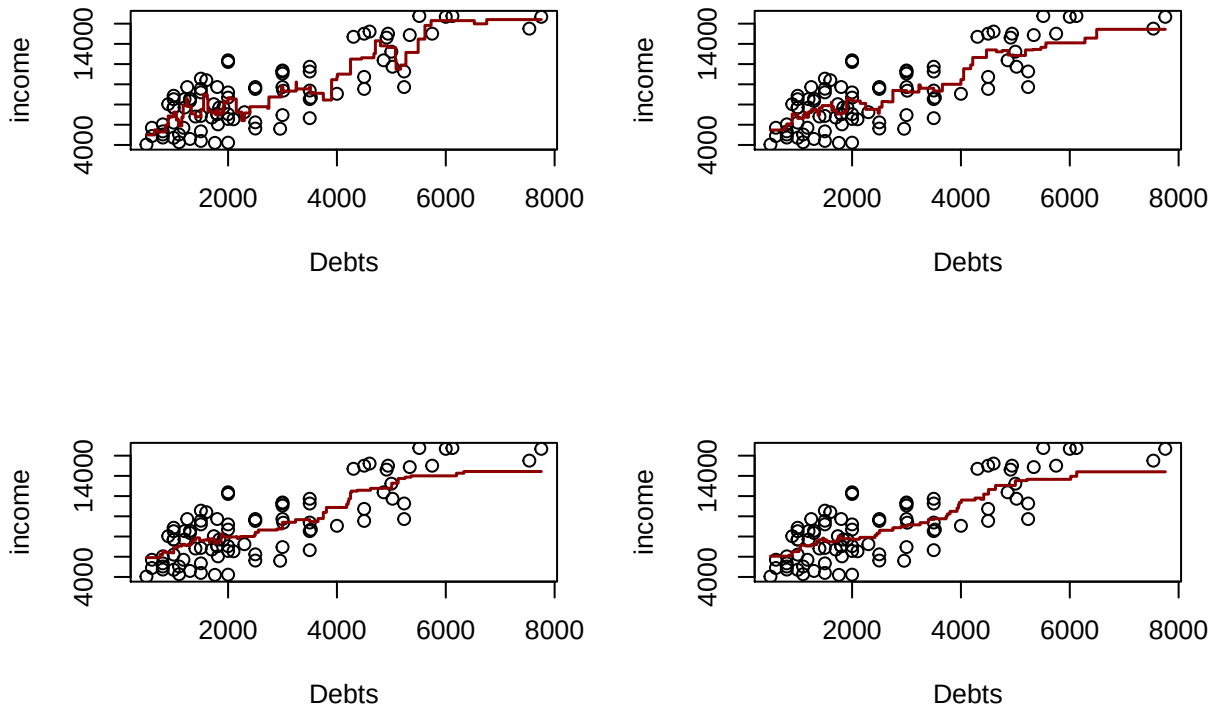
a) Cree un conjunto de datos de entrenamineto del 75% y el restante 25% tratelo como datos de test o de prueba.

```
##
## Attaching package: 'FNN'
## The following objects are masked from 'package:class':
##
##     knn, knn.cv
```

b) Con los datos de entrenamiento implemente KNN (con al menos tres valores para K), usando icome como supervisor y debts como predictor, grafique e interprete.

Del Punto anterior se obtuvo una base de entrenamiento de 86 observaciones mientras que la base de prueba fue de 28, se usa función knn.reg de la libreria FNN para realizar la regresión de k-vecinos más cercanos.

La variable respuesta income y su respectiva variable predictora debts. En este caso se uso K= 4,8,12 y 16 respectivamente

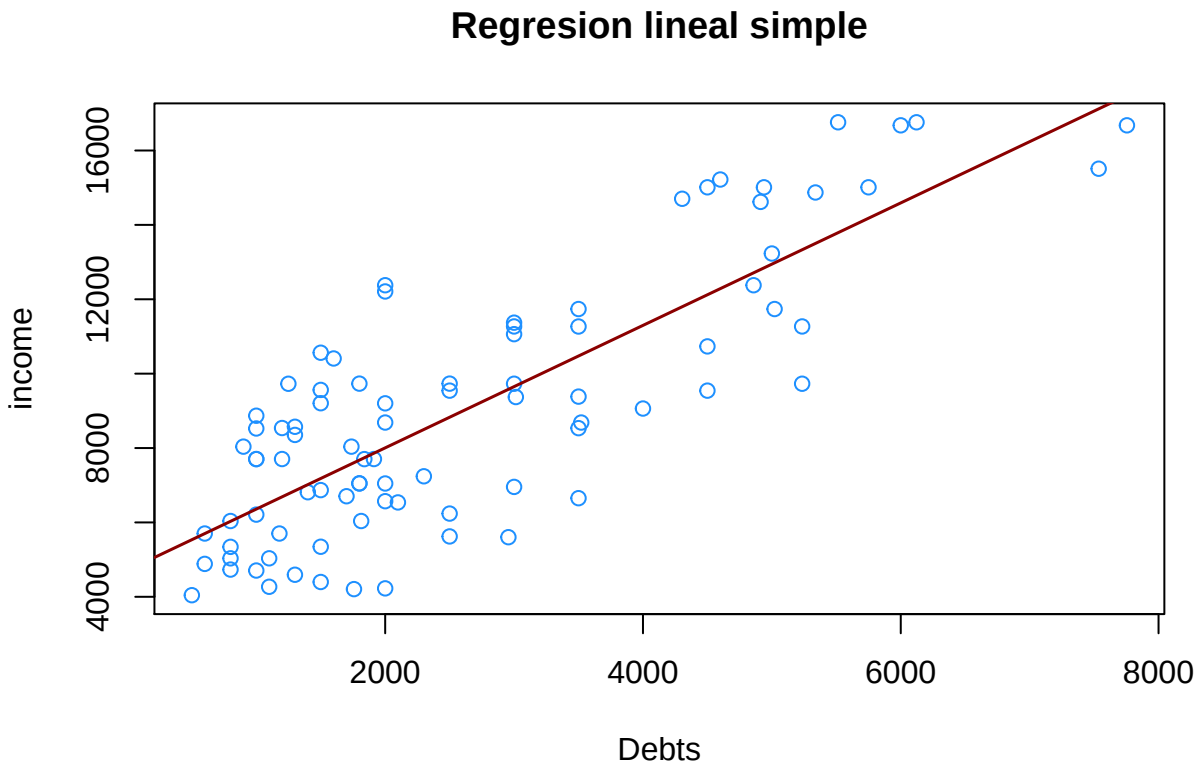


De las graficas anteriores se observa que cuando K=4 proporciona un ajuste flexible y con mayor variabilidad en comparación al resto de graficas y mientras mayor es el tamaño de k la linea roja tiende a ser mas suave, es decir que se logra una mejor tendencia cuando k se incrementa, por tanto se opta por el modelo de regresión con k=16, sin embargo se debe tener en cuenta que cuando k toma un valor muy elevado el suavizamiento puede incrementar el sesgo al enmascarar algo de la estructura de F(x).

c) Con los datos de entrenamiento, implemente regresion lineal simple usando income como el supervisor y dbts como predictor. Grafique e interprete.

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3976.0 -1490.2  -151.9  1694.4  4373.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4711.3692   403.4548   11.68   <2e-16 ***
## x              1.6466     0.1286   12.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2023 on 84 degrees of freedom
```

```
## Multiple R-squared:  0.661,  Adjusted R-squared:  0.657
## F-statistic: 163.8 on 1 and 84 DF,  p-value: < 2.2e-16
```



Se oberva:

1. La relación entre el número Debts y income se puede modelar por un MRLS ya que tiene una tendencia lineal.
2. Esta relación es del tipo creciente, esto es, a medida que aumenta el Debts el income aumenta.
3. La dispersion entre Debs y income es similar, sin embargo, se observa gran variabilidad.
4. El modelo lineal sigue la tendencia de los datos de entrenamiento, no pasa por toda lo nube de puntos, pero parece tener buen comportamiento de ajuste de los datos.

d) Use los repectivos ajustes de cada uno de los modelos anteriores y con el conjunto de prueba, calcule el TEST-MSE. ¿Qué observa?.

```
## Warning: package 'kableExtra' was built under R version 4.0.4
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

Tipo	MSE_test
Knn regresión k=4	4655926
Knn regresión k=8	4632614
Knn regresión k=12	4475809
Knn regresión k=16	4220362
Regresión simple	3996076

Según el MSE test el peor ajuste fue cuando se utilizo regresión con k=4, ya que como fue mencionado anteriormente este modelo genera problemas en la variabilidad, también se observa que los modelos que tienen mejor desempeño en el ajuste son el modelo de regresión con k=16 y el modelo de regresión lineal ya que tienen un valor de MSE test menor en comparación al resto de modelos, esto es de esperarse ya que cuando se desconoce a f(x) knn con valores de k grandes es casi igual que OLS cuando f(x) es lineal, por lo cual Knn resulta ser un buen competidor con la regresión lineal.

e) Usando todos los datos y regresión lineal multiple seleccione un modelo usando forward, backward y stepwise.

Backward Elimination Summary

Variable	AIC	RSS	Sum Sq	R-Sq	Adj. R-Sq
Full Model	-261.890	0.222	8.640	0.97493	0.95198
state_NY	-263.890	0.222	8.640	0.97493	0.95198
gender_Male	-265.890	0.345	8.518	0.96112	0.93027

```
ols_step_forward_aic(lm(income~., data=df))
```

Selection Summary

Variable	AIC	Sum Sq	RSS	R-Sq	Adj. R-Sq
age	-99.897	6.226	2.636	0.70251	0.69986
occupation_Accout	-257.978	8.215	0.647	0.92695	0.92563
occupation_Manager	-290.601	8.384	0.478	0.94608	0.94461
debts	-299.351	8.427	0.435	0.95094	0.94913
state_GA	-306.598	8.461	0.401	0.95476	0.95266
state_RI	-309.757	8.479	0.383	0.95676	0.95433
state_MT	-313.099	8.496	0.366	0.95874	0.95601
loan_type_Credit	-314.557	8.507	0.355	0.95997	0.95692
state_MD	-316.349	8.519	0.343	0.96128	0.95793
marital_status_Married	-317.500	8.528	0.334	0.96234	0.95868
`race_American Indian or Alaska Native`	-319.678	8.540	0.322	0.96369	0.95978
state_IA	-322.276	8.553	0.309	0.96513	0.96098
loan_type_Home	-324.255	8.564	0.298	0.96632	0.96195
loan_decision_type_Approved	-326.959	8.576	0.286	0.96769	0.96312
state_AK	-327.677	8.582	0.280	0.96845	0.96362
state_OR	-328.017	8.588	0.274	0.96909	0.96399

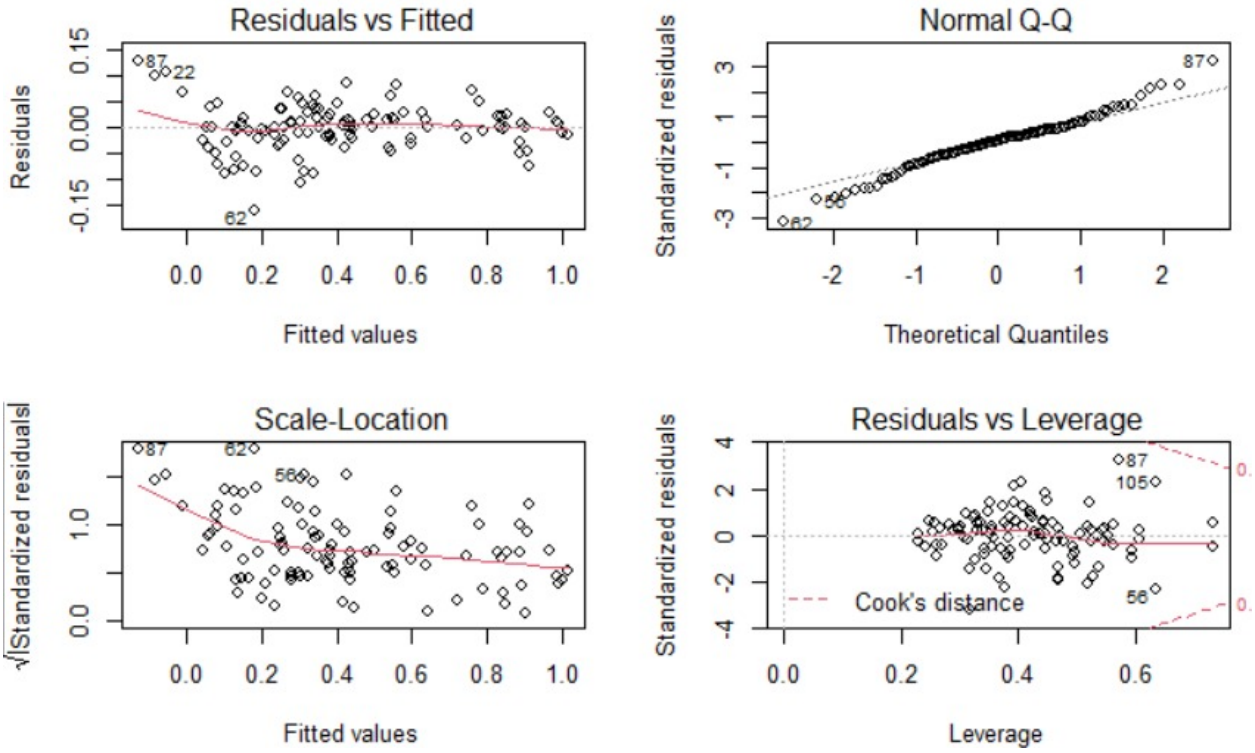
Stepwise Summary

Variable	Method	AIC	RSS	Sum Sq	R-Sq	Adj. R-Sq
age	addition	-99.897	2.636	6.226	0.70251	0.6998
occupation_Accout	addition	-257.978	0.647	8.215	0.92695	0.9256
occupation_Manager	addition	-290.601	0.478	8.384	0.94608	0.9446
debts	addition	-299.351	0.435	8.427	0.95094	0.9491
state_GA	addition	-306.598	0.401	8.461	0.95476	0.9526
state_RI	addition	-309.757	0.383	8.479	0.95676	0.9543
state_MT	addition	-313.099	0.366	8.496	0.95874	0.9560
loan_type_Credit	addition	-314.557	0.355	8.507	0.95997	0.9569
state_MD	addition	-316.349	0.343	8.519	0.96128	0.9579
marital_status_Married	addition	-317.500	0.334	8.528	0.96234	0.9586
`race_American Indian or Alaska Native`	addition	-319.678	0.322	8.540	0.96369	0.9597
state_IA	addition	-322.276	0.309	8.553	0.96513	0.9609
loan_type_Home	addition	-324.255	0.298	8.564	0.96632	0.9619
loan_decision_type_Approved	addition	-326.959	0.286	8.576	0.96769	0.9631
state_AK	addition	-327.677	0.280	8.582	0.96845	0.9636
state_OR	addition	-328.017	0.274	8.588	0.96909	0.9639

Dek metodo backward se observa una leve menoria en el AIC, además tiende a ser más pausible porque deja de estimar un parametro, mientras que con los metodos forward y stepwise se una la misma cantidad de parametros para hacer las estimaciones y predicciones.

e) Seleccione uno de los modelos del paso anterior y responda con argumentacion la pregunta: ¿ajusta bien dicho modelo?

Vamos a seleccionar el modelo deacuerdo al metodo "forward" que tiene un R2adj = 0.9587, a continuación se validaran los supuestos del modelo



- Se observa varianza aproximadamente constante ya que los errores oscilan alrededor del cero de forma aleatoria, es decir, no siguen ningun patrón.
- En el Q-Q plot parece haber algunos outliers que nos hacen alertar sobre el supuesto de normalidad, y las colas no parecen ajustarse correctamente a la recta, sin embargo el 95% de puntos se ajusta la recta, por tanto, se cumple el supuesto de normalidad en los errores.
- Se identifican igualmente los puntos con leverage que podrían afectar el ajuste.
- En general, se concluye que aunque hay problemas con puntos atipicos y leverage, no parecen ser significativos, por tanto, se puede decir que el modelo hace un buen ajuste.