**TAREA MINERÍA**

**Uxía Taboada Nieto**

**45147622W**

1. **Introducción al objetivo del problema y las variables implicadas**

El objetivo de esta tarea será seleccionar las variables objetivo “ActEconom\_Cuali” y “ActEconom” para establecer los modelos de regresión lineal y logísitca y obtener el mejor modelo posible en cada caso implementando técnicas de depuración y optimización.

1. **Importación del conjunto de datos y asignación correcta de los tipos de variables.**

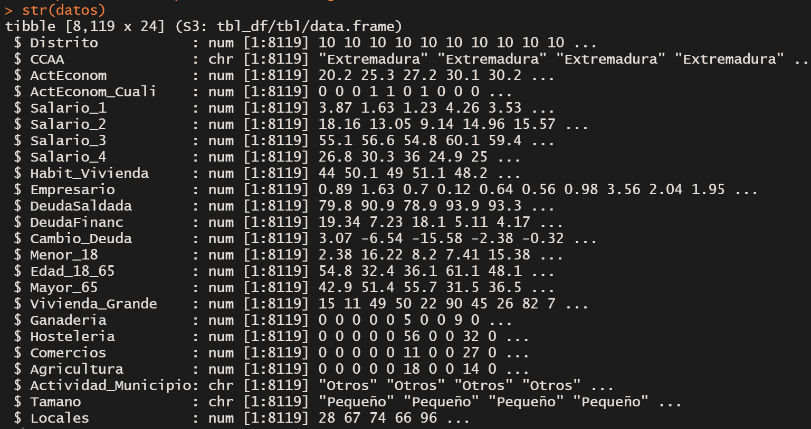
*datos <- read\_excel("C:\\Users\\uxiat\\Desktop\\MasterBD\\Mineria\\Documentación minería de datos y modelización predictiva - Rosa Espinola-20221111\\Tarea\\DatosImpuestos\_Tarea.xlsx")*

Descartamos las posibles variables objetivo excepto las que hemos seleccionado.

*datos <- datos[,-(5:9)]*

Visualizamos el análisis descriptivo del tipo de variables por si ha habido errores a la hora de la asignación.

*str(datos)*



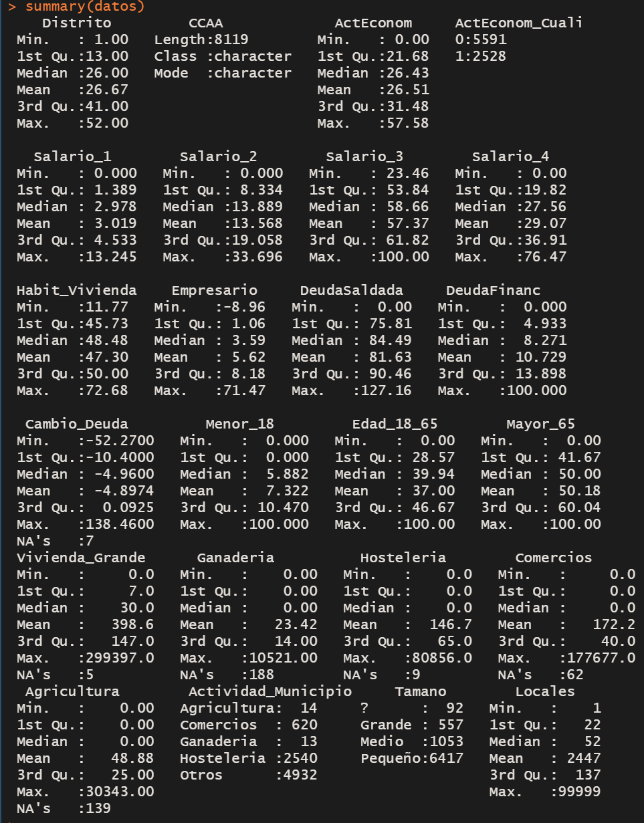
Podemos observar que las variables *ActEconom\_Cuali*, *Actividad\_Municipio* y *Tamano*, deben ser convetrtidas a factor ya que son variables categóricas.

*datos[,c(4,22,23)] <- lapply(datos[,c(4,22,23)], factor)*

*str(datos)*

1. **Análisis descriptivo del conjunto de datos.**

*summary(datos)*



Datos erróneos:

Empresario presenta valores negativos no permitidos.

DeudaSaldada tiene valores fuera de rango superiores a 100.

Tamano tiene una categoría no definda (“?”).

Locales presenta un valor atípico “99999” que seguramente se trata de un error.

Cambio\_Deuda, Vivienda\_Grande, Ganadería, Hosteleria, Comercios, y Agricultura presentan valores desaparecidos (NA’s).

Vemos el reparto de categorías de las variables cualitativas.

*freq(datos$Actividad\_Municipio)*

*freq(datos$Tamano)*

*freq(datos$CCAA)*

Contamos el numero de valores diferentes para las variables numéricas para ver si se puede categorizar alguna. En este caso a simple vista parece que todas tienen un numero muy elevado de valores distintos y no se requiere ninguna modificación.

sapply(Filter(is.numeric, datos),function(x) length(unique(x)))

*freq(datos$ActEconom\_Cuali)*

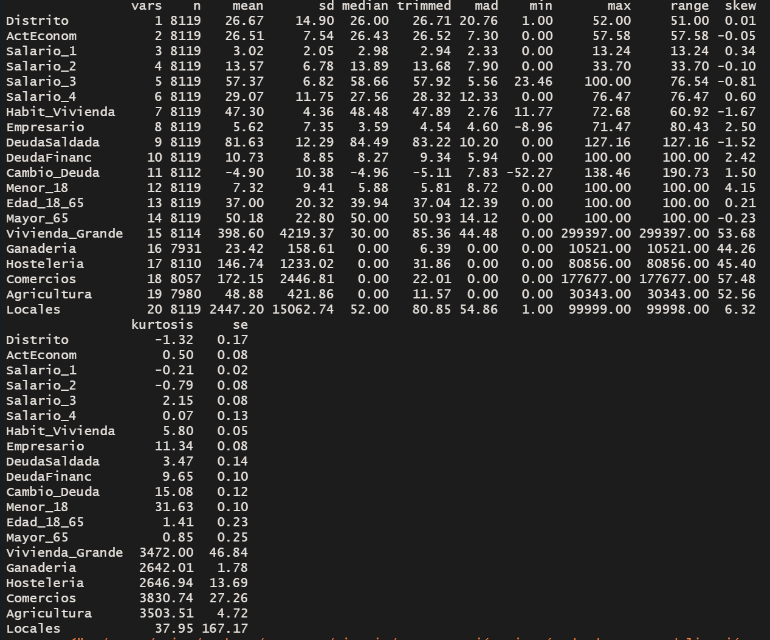
*freq(datos$DeudaFinanc)*

*freq(datos$Habit\_Vivienda)*

*freq(datos$Vivienda\_Grande)*

Para ver el resto destadisticos usamos:

*describe(Filter(is.numeric, datos))*



Las variables Vivienda\_Grande, Ganadería, Hosteleria, Comercios y agricultura presentan asimetría y curtosis muy elevadas, lo cual no es bueno para nuestros datos, ya que muchos valores de las variables tendrán muy poca representación y esto hará que los modelos no se ajusten bien.

*freq(datos$Vivienda\_Grande)*

*freq(datos$Ganaderia)*

*freq(datos$Hosteleria)*

*freq(datos$Comercios)*

*freq(datos$Agricultura)*

1. ***Corrección de los errores detectados***

*Corrección de los datos no declarados (“missings”) de las variables cualitativas. Los pasamos a no declarados (“NA”)*

datos$Tamano<-recode.na(datos$Tamano,"?")

summary(datos)

Corrección de los “missings” no declarados en las variables cuantitativas (-1, 99999)

*datos$DeudaSaldada<-replace(datos$DeudaSaldada,*

*which((datos$DeudaSaldada < 0)|(datos$DeudaSaldada>100)), NA)*

*datos$Empresario<-replace(datos$Empresario,*

*which((datos$Empresario < 0)|(datos$Empresario>100)), NA)*

*summary(datos)*

Corregir los errores en las variables categóricas. En este caso, vamos a crear un único campo en la variable Actividad\_Municipio que será Campo, que contenga la información de las categorías Agricultura y Ganadería. Esto lo hacemos porque como hemos visto antes, estas categorías están muy poco representadas.

*datos$Actividad\_Municipio<-recode(datos$Actividad\_Municipio, "c('Agricultura','Ganaderia')='Campo'")*

*freq(datos$Actividad\_Municipio)*

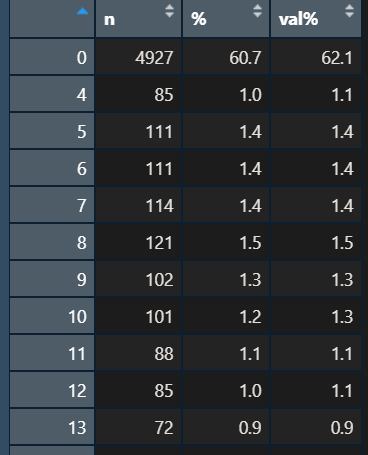
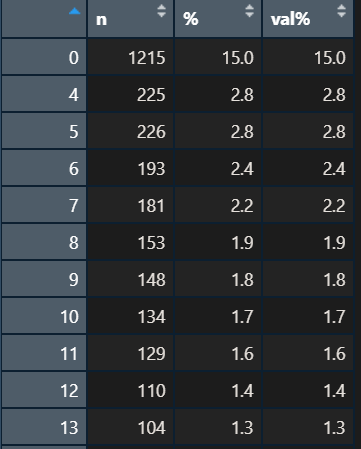
Creamos también la nueva categoría, Ceuta y Melilla como union de las CCAA Ceuta y Melilla ya que no tienen representación. Para poder hacerlo, primero debemos convertir la variable CCAA a factor.

*datos$CCAA <- factor(datos$CCAA)*

*datos$CCAA<-recode(datos$CCAA, "c('Ceuta','Melilla')='Ceuta y Melilla'")*

*freq(datos$CCAA)*

Corrección de las variables cualitativas con categorías poco representadas. Las variables Ganadería, Hosteleria, Comercios, Agricultura y Vivienda\_Grande, presentan una asimetría y curtosis elvadisimas, al calcular sus frecuencias nos damos cuenta que el numero de municipios que tienen estas variables el valor 0 presentan un porcentaje de significación considerablemente elevado respecto al resto de valores. Esta misma casuística afecta a las 5 variables en cuestión.

De forma que resulta imposible pasarlas a factor ya que el numero de categorías seria elevadísimo, crear intervalos de categorías no me parecía una solución viable por la ausencia de un patrón claro, y dejar estos datos con tanta asimetría y curtosis en el modelo daría lugar a errores significativos. Por ello, decidí convertirlas a variables categóricas binarias. Es decir con solamente la información de si existen municipios con o sin las variables mencionadas.

Haciendo esto perdemos información, pero que a mi parecer no resulta muy reveladora, ya que haciendo uso de este método el porcentaje de representación de la categoría 0 frente a la 1 se homogeniza, al contrario que tal como están las variables originales como se puede ver en la imagen.

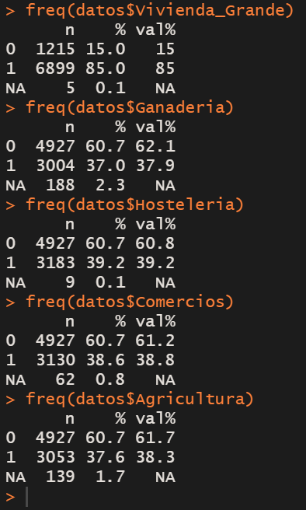
*datos$Ganaderia <- factor(replace(datos$Ganaderia, which(datos$Ganaderia>0),1))*

*datos$Hosteleria<- factor(replace(datos$Hosteleria, which(datos$Hosteleria>0),1))*

*datos$Comercios <- factor(replace(datos$Comercios, which(datos$Comercios>0),1))*

*datos$Agricultura <- factor(replace(datos$Agricultura, which(datos$Agricultura>0),1))*

*datos$Vivienda\_Grande <- factor(replace(datos$Vivienda\_Grande, which(datos$Vivienda\_Grande>0),1))*

**

Ahora vemos que los porcentajes de significación son mas próximos entre las categorías y por lo tanto toda la información que aportan estas variables esta bien representada.

Para la gestión de valores atípicos y “missings” necesitamos separar del conjunto de datos las variables objetivo.

varObjCont<-datos$ActEconom

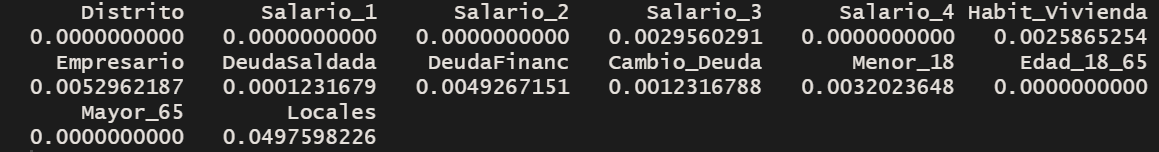
varObjBin<-datos$ActEconom\_Cuali

Creamos el nuevo conjunto de datos input sin las variables objetivo que se corresponden con las columnas 3 y 4.

input<-as.data.frame(datos[,-(3:4)])

Corrección de valores atípicos. El tratamiento para este tipo de variables será pasarlos a valor perdido. Para ello, primero, hacemos un recuento del porcentaje de valores atípicos contenido en cada variable con la función:

*sapply(Filter(is.numeric, input),function(x) atipicosAmissing(x)[[2]])/nrow(input)*

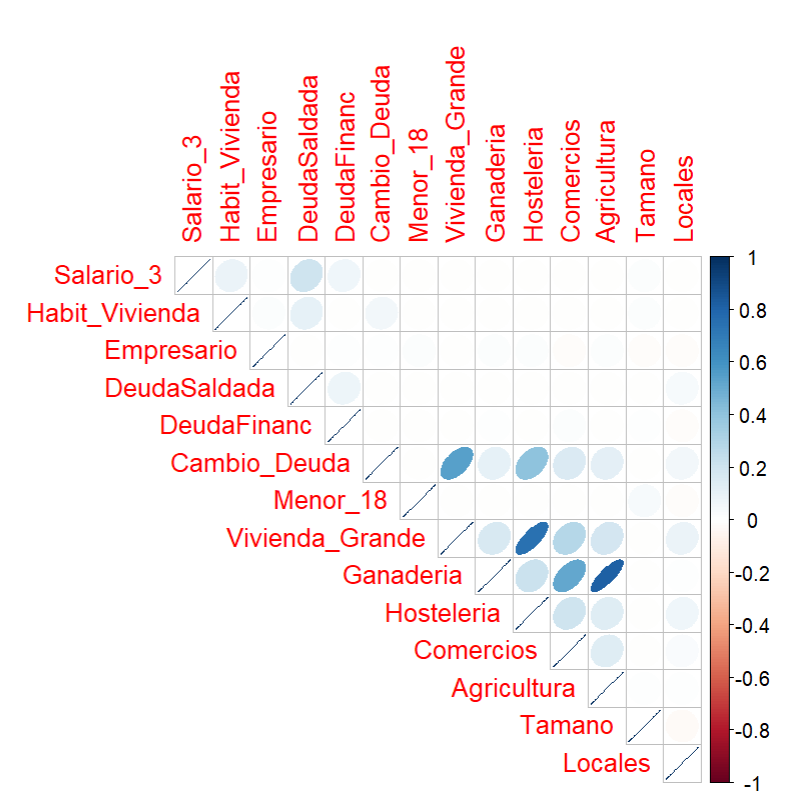


Y en segundo lugar convertimos los convierto a “missings”

*input[,as.vector(which(sapply(input, class)=="numeric"))]<-sapply(Filter(is.numeric, input),function(x) atipicosAmissing(x)[[1]])*

Buscamos si existe algún patrón en los valores perdidos que pueda darnos alguna información adicional. Para buscar esta relación utilizamos el siguiente grafico.

*corrplot(cor(is.na(input[colnames(input)[colSums(is.na(input))>0]])),method = "ellipse",type = "upper")*



Hacemos un estudio de la proporción de “missings” por variable y observación.

Primero hacemos el estudio por filas, es decir por observaciones.

*input$prop\_missings<-apply(is.na(input),1,mean)*

Nos ha creado la nueva variable prop\_missings en el dataset input. Vemos el summary de esta variable que tiene por filas la proporción de valores perdidos por observacion.

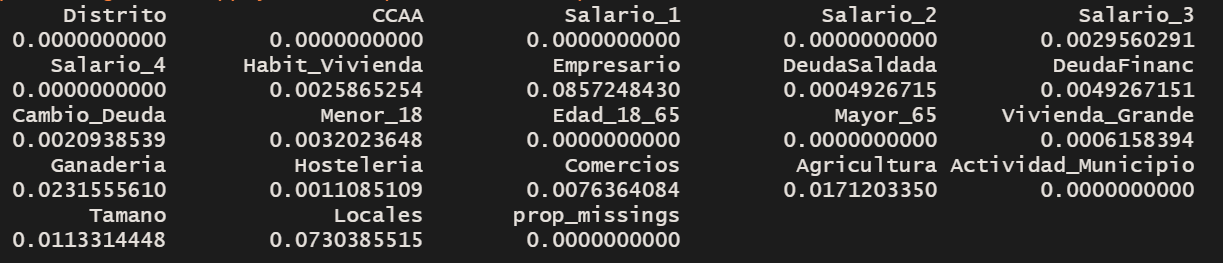
*summary(input$prop\_missings)*

**

Esta grafica nos muestra que al menos hay una variable con el 30% de las variables perdidas, este valor es inferior al 50% por lo que no tenemos un porcentaje suficientemente elevado de valores perdidos y por lo tanto no podemos eliminar esta variable.

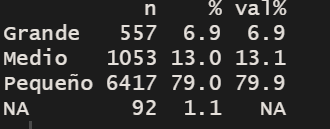
Ahora procedemos a hacer el mismo estudio pero por columnas, es decir, por variables.

*(prop\_missingsVars<-apply(is.na(input),2,mean))*

**

Como se observa, todas las variables presentan menos de un 50% de valores perdidos y por lo tanto no podemos eliminar ninguna variable.

Correcciones en las variables categóricas. El método es imputar las variables categóricas si el porcentaje de NA es suficientemente elevado. Volviendo al análisis de la grafica de las variables cateoricas anterior, nos centramos en la variable Tamano que no habíamos modificado.



No procedemos a recategorizar la variable Tamano porque el porcentaje de “NA” es del 1.1% y por tanto es mejor no inputarla. Para la actividad municipio no haría falta porque no tiene valores perdidos.

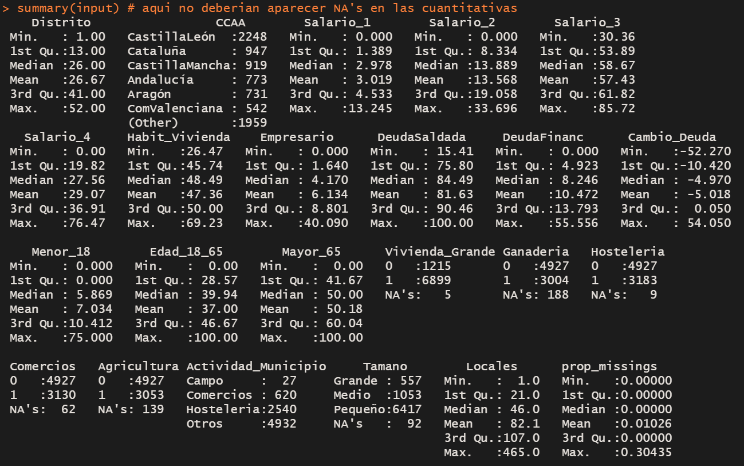
Inputaciones de las variables cuantitativas.

Seleccionamos el tipo de inputación aleatorio.

summary(datos)

input[,as.vector(which(sapply(input, class)=="numeric"))]<-sapply(Filter(is.numeric, input),function(x) ImputacionCuant(x,"aleatorio")) #ejecutar 2 veces

summary(input)



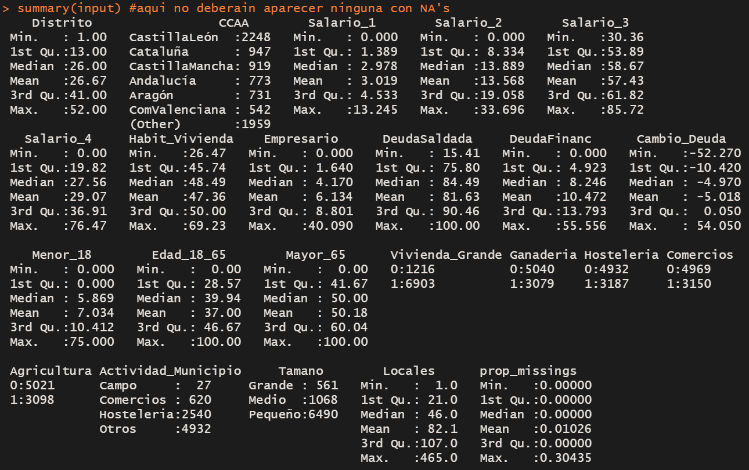
Inputaciones para las variables cualitativas.

*summary(datos)*

*input[,as.vector(which(sapply(input, class)=="factor"))]<-sapply(Filter(is.factor, input),function(x) ImputacionCuali(x,"aleatorio"))*

A veces se cambia el tipo de factor a character al imputar, asi que hay que indicarle que es factor.

*input[,as.vector(which(sapply(input, class)=="character"))] <- lapply(input[,as.vector(which(sapply(input, class)=="character"))] , factor)*

**

Llegados a este punto, ya no tenemos valores “NA” en ninguna de las variables y nuestros datos están limpios y listos para aplicarles los modelos de regresión. Guardamos nuestra base de datos depurada para hacer el siguiente paso que es contruir el modelo. Una vez finalizado este proceso, se puede considerar que los datos estan depurados. Los guardamos

saveRDS(cbind(varObjBin,varObjCont,input),"datosTareaDep")

Se va a llamar datosVinoDep

*setwd("C:\\Users\\uxiat\\Desktop\\MasterBD\\Mineria\\Documentacion minera­a de datos y modelizacion predictiva - Rosa Espinola-20221111\\codigos\_buenos")*

1. **Modelo de regresión lineal**

Para este modelo partimos de los datos depurados.

*datos<-readRDS("C:\\Users\\uxiat\\Desktop\\MasterBD\\Mineria\\Documentación minería de datos y modelización predictiva - Rosa Espinola-20221111\\Tarea\\datosTareaDep")*

*varObjCont<-datos$varObjCont*

*varObjBin<-datos$varObjBin*

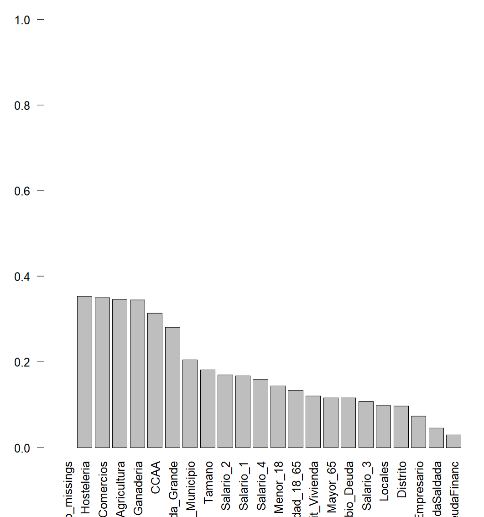
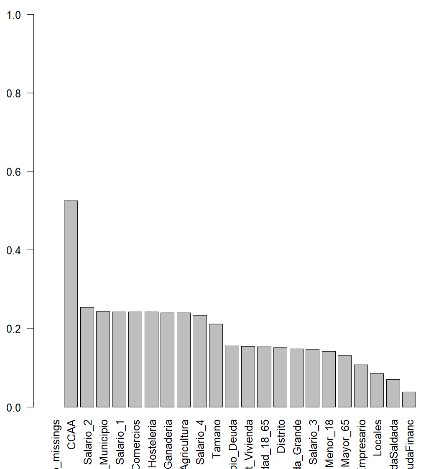
*input<-datos[,-(1:2)]*

Obtenemos la importancia de las variables. Falla si hay alguna variable cuantitativa con menos de 6 valores diferentes.

Los gráficos de Kramer nos dicen la correlacion entre las variables y las variables objetivo.

*graficoVcramer(input,varObjBin)*

*graficoVcramer(input,varObjCont)*



Vemos grafiicamente el efecto de dos variables cualitativas sobre la binaria.

mosaico\_targetbinaria(input$Actividad\_Municipio,varObjBin,"Actividad\_municipio")

mosaico\_targetbinaria(input$Tamano,varObjBin,"Tamano")

barras\_targetbinaria(input$Actividad\_Municipio,varObjBin,"Actividad\_municipio")

barras\_targetbinaria(input$Tamano,varObjBin,"Tamano")

Vemos graficamente el efecto de dos variables cuantitativas sobre la binaria

summary(input)

boxplot\_targetbinaria(input$Edad\_18\_65,varObjBin,"Edad\_18\_65")

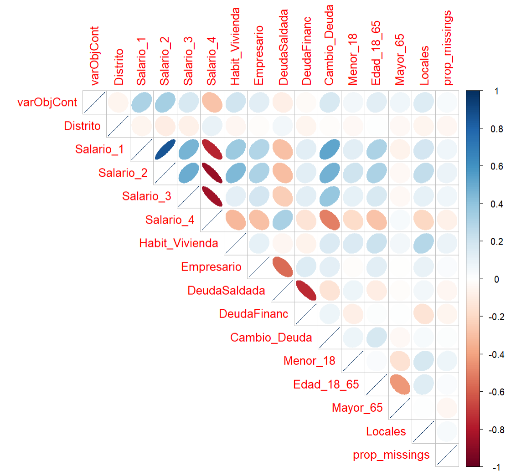
boxplot\_targetbinaria(input$Locales,varObjBin,"Local")

hist\_targetbinaria(input$Edad\_18\_65,varObjBin,"Edad\_18\_65")

hist\_targetbinaria(input$Acidez,varObjBin,"Acidez")

graficoCorrelacion(varObjCont,input)

corrplot(cor(cbind(varObjCont,Filter(is.numeric, input)), use="pairwise", method="pearson"), method = "ellipse",type = "upper")



Buscamos las mejores transformaciones para las variables numéricas respecto a los dos tipos de variables.

input\_cont<-cbind(input,Transf\_Auto(Filter(is.numeric, input),varObjCont))

input\_bin<-cbind(input,Transf\_Auto(Filter(is.numeric, input),varObjBin))

saveRDS(data.frame(input\_bin,varObjBin),"todo\_bin")

saveRDS(data.frame(input\_cont,varObjCont),"todo\_cont")

Comenzamos con la regression lineal y obtenemos la partición.

todo<-data.frame(input,varObjCont)

set.seed(123456)

trainIndex <- createDataPartition(todo$varObjCont, p=0.8, list=FALSE)

data\_train <- todo[trainIndex,]

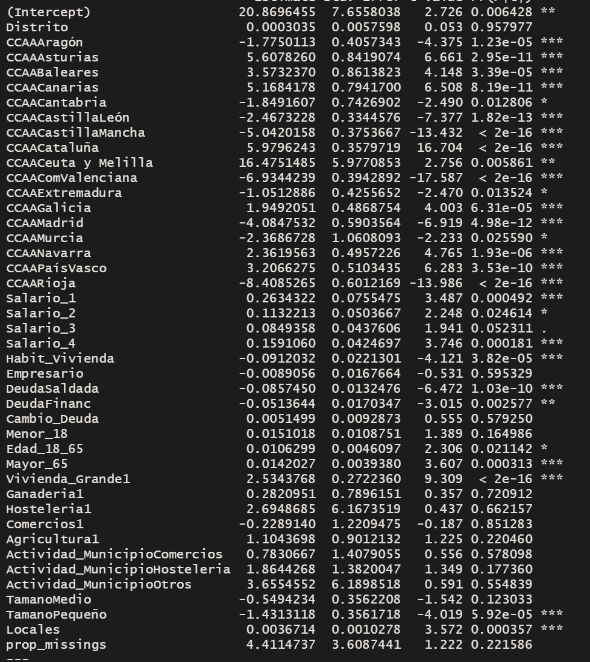
data\_test <- todo[-trainIndex,]

La idea del modelo será partir de este modelo con todas las varibales e ir creando nuevos modelos basándonos en los valores del r cuadrado, del numero de variables y de si todas las variables del modelo son o no significativas.

modelo1<-lm(varObjCont~.,data=data\_train)

summary(modelo1)

Interpretación de una variable continua y otra categórica.



Variable Continua:

Para cada distrito con vivienda grande, el porcentaje de impago aumenta en 2,55 unidades.

Variable Categórica:

Recordemos que el summary anterior, nos muestra todas las categorías de las variables categóricas -1. La que falta es la que falta por defecto , en este caso municipios grandes. Por tanto, por cada municipio de tamaño pequeño, el porcentaje de deuda disminuye en 1.41 unidades con respecto a los municipios grandes Variable Continua:

Para cada distrito con vivienda grande, el porcentaje de impago aumenta en 2,55 unidades.

Variable Categórica:

Recordemos que el summary anterior, nos muestra todas las categorías de las variables categóricas -1. La que falta es la que falta por defecto , en este caso municipios grandes. Por tanto, por cada municipio de tamaño pequeño, el porcentaje de deuda disminuye en 0.549 unidades con respecto a los municipios grandes. Esto implica que hay mas tasa de impago en los grandes municipios.

Esto implica que hay mas tasa de impago en los grandes municipios.

Rsq(modelo1,"varObjCont",data\_train)

Rsq(modelo1,"varObjCont",data\_test)

modelEffectSizes(modelo1)

barplot(sort(modelEffectSizes(modelo1)$Effects[-1,4],decreasing =T),las=2,main="Importancia de las variables (R2)")

En base a los resultados anteriores, contruimos un modelo con menos variables pero que estas sean más relevantes.

graficoVcramer(todo,varObjCont)

modelo2<-lm(varObjCont~CCAA+DeudaSaldada+Locales+Vivienda\_Grande+Tamano+Actividad\_Municipio+Mayor\_65+Salario\_4+Salario\_1+DeudaFinanc,data=data\_train)

modelEffectSizes(modelo2)

summary(modelo2)

Rsq(modelo2,"varObjCont",data\_train)

Rsq(modelo2,"varObjCont",data\_test

modelEffectSizes(modelo2)

barplot(sort(modelEffectSizes(modelo2)$Effects[-1,4],decreasing =T),las=2,main="Importancia de las variables (R2)")

Modelo solo con las 3 variables mas correlacionadas del modelo 1:

modelox<-lm(varObjCont~CCAA+Vivienda\_Grande+DeudaSaldada,data=data\_train)

summary(modelox)

Rsq(modelox,"varObjCont",data\_train)

Rsq(modelox,"varObjCont",data\_test)

Modelo contruido eliminando variables menos relevantes del modelo2.

modelo3<-lm(varObjCont~CCAA+DeudaSaldada+Locales+Vivienda\_Grande+Tamano+Actividad\_Municipio+Habit\_Vivienda,data=data\_train)

summary(modelo3)

Rsq(modelo3,"varObjCont",data\_train)

Rsq(modelo3,"varObjCont",data\_test)

modelEffectSizes(modelo3)

barplot(sort(modelEffectSizes(modelo3)$Effects[-1,4],decreasing =T),las=2,main="Importancia de las variables (R2)")

modelo4<-lm(varObjCont~CCAA+DeudaSaldada+Locales+Vivienda\_Grande+Tamano,data=data\_train)

summary(modelo4)

Rsq(modelo4,"varObjCont",data\_train)

Rsq(modelo4,"varObjCont",data\_test)

modelo5<-lm(varObjCont~CCAA+Vivienda\_Grande+Actividad\_Municipio+DeudaSaldada,data=data\_train)

summary(modelo5)

Rsq(modelo5,"varObjCont",data\_train)

Rsq(modelo5,"varObjCont",data\_test)

Para validar la calidad de los modelos hacemos las validaciones cruzadas

modelo1VC <- train(formula(modelo1),

data = todo,method = "lm",

trControl = trainControl(method="repeatedcv", number=5, repeats=20, returnResamp="all"))

modelo2VC <- train(formula(modelo2),

data = todo,method = "lm",

trControl = trainControl(method="repeatedcv", number=5, repeats=20, returnResamp="all"))

modelo3VC <- train(formula(modelo3),

data = todo,method = "lm",

trControl = trainControl(method="repeatedcv", number=5, repeats=20, returnResamp="all"))

modelo4VC <- train(formula(modelo4),

data = todo,method = "lm",

trControl = trainControl(method="repeatedcv", number=5, repeats=20, returnResamp="all"))

modelo5VC <- train(formula(modelo5),

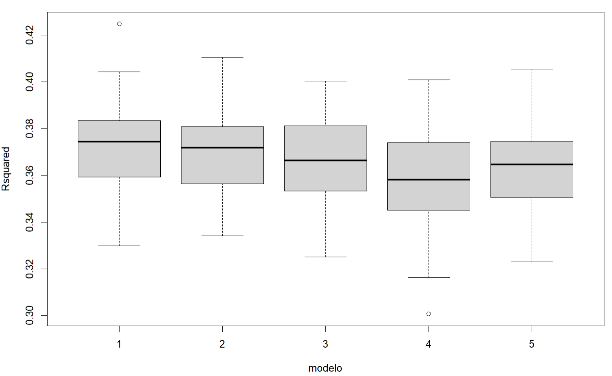
data = todo,method = "lm",

trControl = trainControl(method="repeatedcv", number=5, repeats=20, returnResamp="all"))

De toda la info solo nos quedamos con resample que guarda el error de todos los cruces de todos los tests de cada modelo.

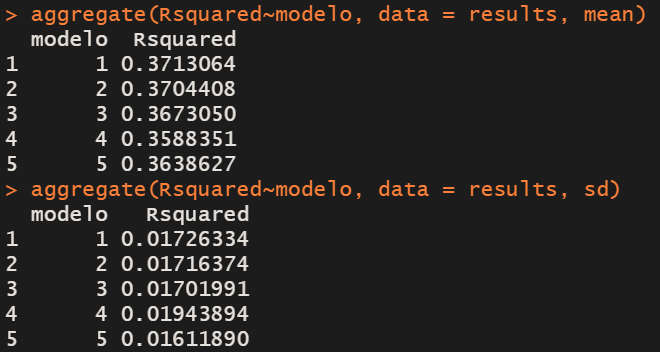
results<-data.frame(rbind(modelo1VC$resample,modelo2VC$resample,modelo3VC$resample,modelo4VC$resample,modelo5VC$resample),modelo=c(rep(1,100),rep(2,100),rep(3,100),rep(4,100),rep(5,100)))

boxplot(Rsquared~modelo,data=results)



aggregate(Rsquared~modelo, data = results, mean)

aggregate(Rsquared~modelo, data = results, sd)



Por ultimo visualizamos el numero de parámetros de cada modelo.

length(coef(modelo1));length(coef(modelo2));length(coef(modelo3));length(coef(modelo4));length(coef(modelo5))

Modelo1 : 43 variables

Modelo2: 30 variables

Modelo 3: 27 variables

Modelo4: 23 variables

Modelo 5: 23 variables

Elección del modelo.

Si bien es cierto que el modelo 1 tiene mayor r2 y teniendo en cuenta aue la variabilidad de todos los modelos es muy similar, nos fijamos en el numero de parametros para tomar la decision final.

El modelo 4 ya tiene un 4 r2 considerablemente mas bajo que el 4 y con el mismo numero de factores por lo tanto lo descartamos.

Para este caso el modelo 5 parece el mejor ya que tiene 4 factores menos que el modelo 3 (segundo mejor) y un r2 muy similar. Además el 5 tiene presenta menor variabilidad.

Vemos los coeficientes del modelo ganador

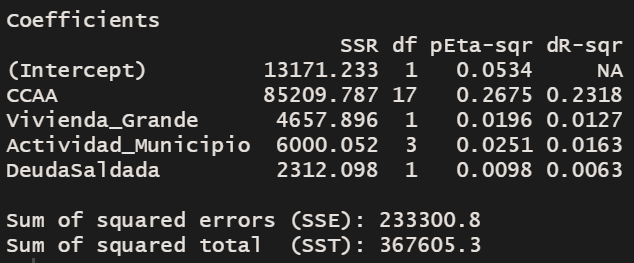
coef(modelo5)

Evaluamos la estabilidad del modelo a partir de las diferencias en train y test:

Rsq(modelo5,"varObjCont",data\_train)

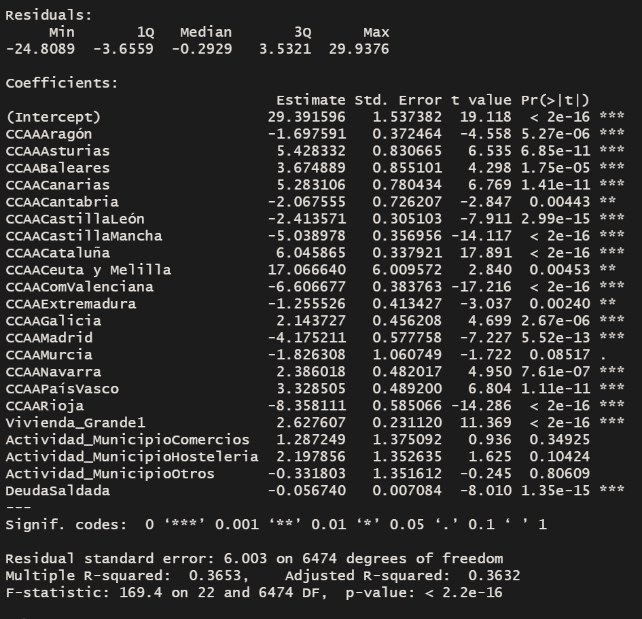
Rsq(modelo5,"varObjCont",data\_test)

modelEffectSizes(modelo5)



barplot(sort(modelEffectSizes(modelo5)$Effects[-1,4],decreasing =T),las=2,main="Importancia de las variables (R2)")

summary(modelo5)



1. **Modelo de regression logísitica**

todo<-readRDS("todo\_bin")

Veo el reparto original (sin transformaciones ni interacciones). Comprobamos que la variable objetivo tome valor 1 para el evento y 0 para el no evento.

freq(todo$varObjBin)

Hacemos la separación entre las variables originales de entrenamiento y test.

set.seed(123456)

trainIndex <- createDataPartition(todo$varObjBin, p=0.8, list=FALSE)

Primero eliminamos las variables transformadas por el momento.

data\_train <- todo[trainIndex,c(1:23,39)]

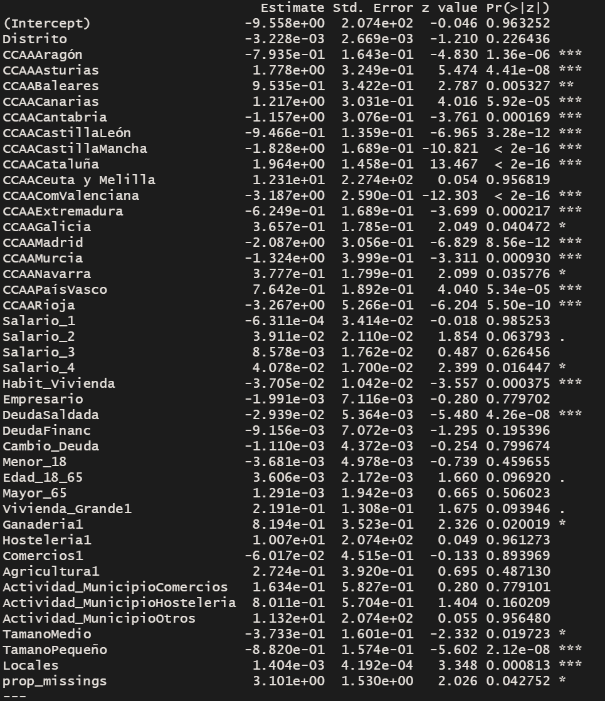
Nos quedamos con las variables sin transformar y con la objetivo.

data\_test <- todo[-trainIndex,c(1:23,39)]

Primeramente probamos un modelo sin las transformadas, hacemos la regresión logisitica para los datos de entrenamiento.

modeloInicial<-glm(varObjBin~.,data=data\_train,family=binomial)

summary(modeloInicial)



Hay variables que no son significativas y por tanto no van a influir en la variable respuesta.

Con el summary no nos sale el valor de r2 por tanto no podemos estimar la calidad del modelo con estos datos

Procedemos a interpretar un parametro asociado a la variable continua y otro a la variable categórica.

Variable Continua:

Para cada distrito con vivienda grande, el porcentaje de impago aumenta en 0.21 unidades.

Variable Categórica:

Recordemos que el summary anterior, nos muestra todas las categorías de las variables categóricas -1. La que falta es la que falta por defecto , en este caso municipios grandes. Por tanto, por cada municipio de tamaño pequeño, el porcentaje de deuda disminuye en 0.882 unidades con respecto a los municipios grandes y por cada municipio de tamaño medio disminuye en 0.377 unidas. Esto implica que hay mas tasa de impago en los grandes municipios.

pseudoR2(modeloInicial,data\_train,"varObjBin")

pseudoR2(modeloInicial,data\_test,"varObjBin")

Es normal que el pseudor2 baje un poco en los datos de prueba (justo lo contrario que en la lineal) porque estamos midiendo el error con datos que no se han usado para estimar el valor.

Vemos el numero de parámetros de este modelo. Tiene 43, son demasiados.

modeloInicial$rank

fijandonos en la significacion de las variables,el modelo con las variables más significativas queda.

modelo2<-glm(varObjBin~CCAA+DeudaSaldada+Locales+Vivienda\_Grande+Tamano+Salario\_2+Actividad\_Municipio+Habit\_Vivienda,data=data\_train,family=binomial)

summary(modelo2)

pseudoR2(modelo2,data\_train,"varObjBin")#es un poquito peor que el anterior, pero el n. de parametros es casi la mitad

pseudoR2(modelo2,data\_test,"varObjBin")

modelo2$rank

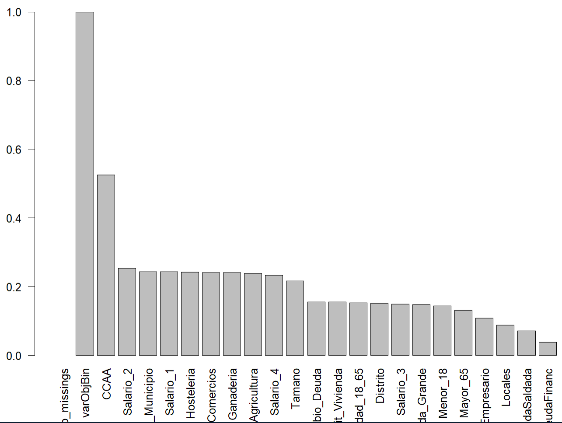
Es un poquito peor que el anterior, pero hemos conseguido reducir el numero de parametros es casi la mitad.

impVariablesLog(modelo2,"varObjBin")

Quizas sobren las 2 ultimas salario\_2 y vivienda\_grande que son las menos significativas. Por lo tanto construimos otro modelo.

Miramos el gráfico V de Cramer para ver las variables mas importantes.

graficoVcramer(todo[,c(1:23,39)],todo$varObjBin)



modelo3<-glm(varObjBin~CCAA+DeudaSaldada+Locales+Tamano+Actividad\_Municipio+Habit\_Vivienda,data=data\_train,family=binomial)

summary(modelo3)

pseudoR2(modelo3,data\_train,"varObjBin)

pseudoR2(modelo3,data\_test,"varObjBin")

modelo3$rank

Es mejor que el 2 porque tiene menos parametros y no disminuye pseudor2

Eliminamos las variables que no son significativas del modelo anterior. que perece ser la variable “actividad municipio”

modelo3\_bis<-glm(varObjBin~CCAA+DeudaSaldada+Locales+Tamano+Habit\_Vivienda,data=data\_train,family=binomial)

summary(modelo3\_bis)

pseudoR2(modelo3\_bis,data\_train,"varObjBin")#baja un poco r2 pero al menos todos los parametros son significativos

pseudoR2(modelo3,data\_test,"varObjBin")

modelo3$rank

Pruebamos alguna iteracción sobre el modelo 3\_bis que parece el mejor.

modelo4<-glm(varObjBin~CCAA+DeudaSaldada+Locales+Tamano+Habit\_Vivienda+Tamano:Habit\_Vivienda,data=data\_train,family=binomial)

summary(modelo4)

pseudoR2(modelo4,data\_train,"varObjBin")

pseudoR2(modelo4,data\_test,"varObjBin")

modelo4$rank

No es muy viable xq la pseudor2 es parecida al 4 pero aumentan hasta 25 los factores.

modelo5<-glm(varObjBin~CCAA+DeudaSaldada+Locales+Tamano+Habit\_Vivienda+CCAA:Locales,data=data\_train,family=binomial)

pseudoR2(modelo5,data\_train,"varObjBin")

pseudoR2(modelo5,data\_test,"varObjBin")

modelo5$rank #si es cierto que aumenta un poco pseudor2 pero se van a 40 los facores

summary(modelo5)

Si es cierto que aumenta un poco pseudor2 pero pasamos a tener 40 facores por no mencionar que hay factores muy poco significativos.

modelo6<-glm(varObjBin~CCAA+DeudaSaldada+Locales+Tamano+Habit\_Vivienda+Tamano:DeudaSaldada,data=data\_train,family=binomial)

No es significativa esta interacción.

summary(modelo6)

pseudoR2(modelo6,data\_train,"varObjBin")

pseudoR2(modelo6,data\_test,"varObjBin")

modelo6$rank

Aumenta pseudor2 un poco pero hay 25 factores (aunque la mayoria significativos)

Validacion cruzada repetida para elegir entre todos.

auxVarObj<-todo$varObjBin

Formateamos la variable objetivo para que funcione el código.

todo$varObjBin<-make.names(todo$varObjBin)

total<-c()

modelos<-sapply(list(modelo1,modelo2,modelo3,modelo3,modelo4,modelo5),formula)

for (i in 1:length(modelos)){

set.seed(1712)

vcr<-train(as.formula(modelos[[i]]), data = todo,

method = "glm", family="binomial",metric = "ROC",

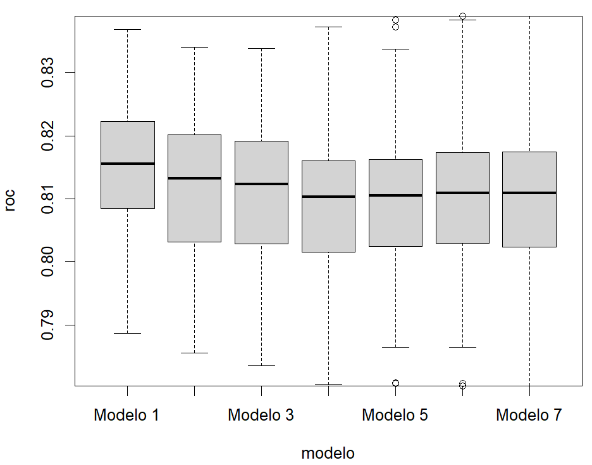
trControl = trainControl(method="repeatedcv", number=5, repeats=20,

summaryFunction=twoClassSummary,

classProbs=TRUE,returnResamp="all"))

total<-rbind(total,data.frame(roc=vcr$resample[,1],modelo=rep(paste("Modelo",i),

nrow(vcr$resample))))}



boxplot(roc~modelo,data=total,main="Area bajo la curva ROC") #el 3 es peor, los otros parecidos

aggregate(roc~modelo, data = total, mean) # media

aggregate(roc~modelo, data = total, sd)

En vista solo de estos valores, todos los modelos presentan la misma area bajo la curva roc. Podriamos descartar 4,5 y 6 porque es ligeramente menor (pero veamos primero el numero de parametros ). Y todos presentan una variabilidad muy similar tambien (desviacion tipica). Por tanato para concluir con que modelo es mejor, debemos ver el numero de parámetros. Recupero la variable objetivo en su formato

todo$varObjBin<-auxVarObj

Miramos el numero de parámetros.

modelo1$rank

modelo2$rank

modelo3$rank

modelo4$rank

modelo5$rank

modelo6$rank

Modelos 1 y 5 descartados xq tienen muchismos factores.

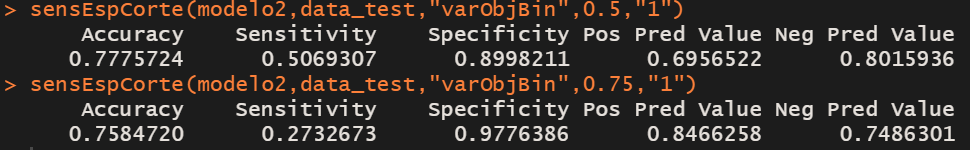
El modelo 4 tiene una roc 0.03 veces menor que el siguiente modelo con mejor roc y parámetros que es el 2. El 2 tiene tiene un factor mas que el 4 pero mejor roc

Entre 4 y 6 estan empatados en factores pero ambos un roc menor que el 2,Asi que voy a escoger el modelo 2 como ganador. Buscamos el mejor punto de corte

Comprobamos las medidas de error. Para disntintos puntos de corte para ver cual es mejor. Los puntos de corte son necesarios para crear la matriz de confusión.Probamos dos

*sensEspCorte(modelo2,data\_test,"varObjBin",0.5,"1")*

*sensEspCorte(modelo2,data\_test,"varObjBin",0.75,"1")*



En este caso de ejemplo seria mejor el 0.75. Generamos una rejilla de puntos de corte

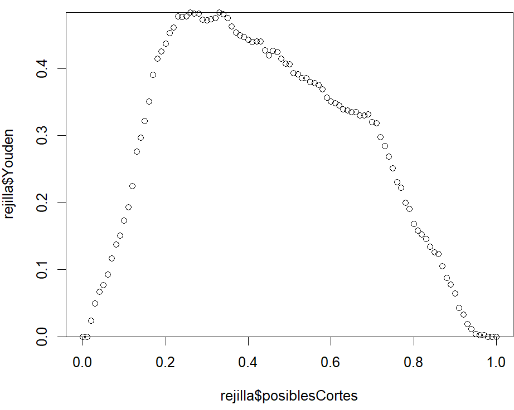
*posiblesCortes<-seq(0,1,0.01)*

*rejilla<-data.frame(t(rbind(posiblesCortes,sapply(posiblesCortes,function(x) sensEspCorte(modelo2,data\_test,"varObjBin",x,"1")))))*

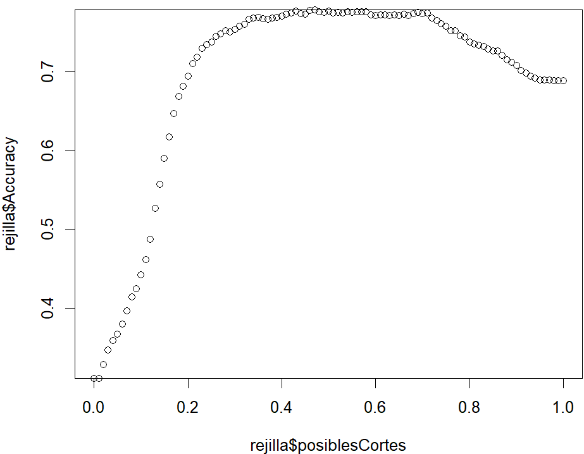
*rejilla$Youden<-rejilla$Sensitivity+rejilla$Specificity-1*

Representamos el indice de Youden para cada punto de corte

plot(rejilla$posiblesCortes,rejilla$Youden)



plot(rejilla$posiblesCortes,rejilla$Accuracy)



plot(rejilla$posiblesCortes,rejilla$Accuracy)

Cogeremos punto de corte que maximice la accuracy. Lo vemos numéricamente

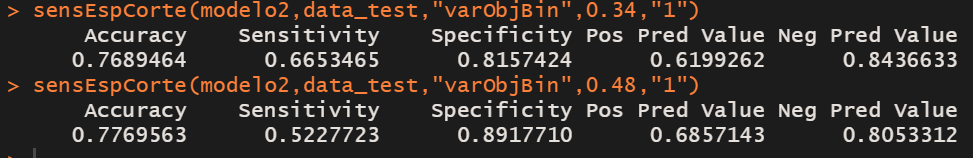
rejilla$posiblesCortes[which.max(rejilla$Youden)] # 0.34

rejilla$posiblesCortes[which.max(rejilla$Accuracy)] # 0.48

Son 2 puntos bastante diferentes por lo que vamos a evaluar las medidas en ambos puntos de corte y ver cual nos interesa mas. El resultado es 0.34 para youden y 0.48 para Accuracy.Los comparamos

sensEspCorte(modelo2,data\_test,"varObjBin",0.34,"1")

sensEspCorte(modelo2,data\_test,"varObjBin",0.48,"1")



Viendo ambos, elegiriamos el punto 0.48 porque nos maximiza la Accuracy.Tambien nos interesa mas que acierte en los unos que en los ceros (specificity).Aunque acierta mas "unos" (sensitivity) el 0.34.Por ello, en vista de que la accuracy es muy similar, y que en el 0.48 acierta mas unos. Me quedo con el 0.34.

Vemos las variables mas importantes del modelo ganador.

impVariablesLog(modelo2,"varObjBin")

Vemos los coeficientes del modelo ganador.

coef(modelo2)

Evaluamos la estabilidad del modelo a partir de las diferencias en train y test:

pseudoR2(modelo2,data\_train,"varObjBin")

pseudoR2(modelo2,data\_test,"varObjBin")

Baja pseudor2 en test que es bueni. Es poca la diferencia, por lo que el modelo se puede considerar robusto.

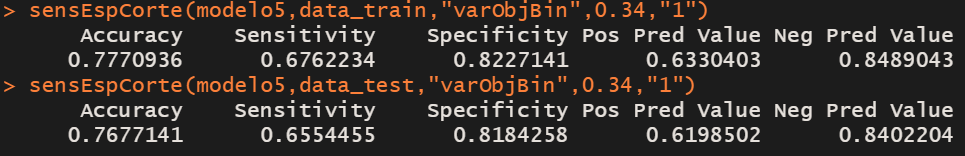
roc(data\_train$varObjBin, predict(modelo5,data\_train,type = "response"))

roc(data\_test$varObjBin, predict(modelo5,data\_test,type = "response"))

Area bajo la curva roc no exageradamente buena pero es bastante próxima. Tambien es poca la diferencia en el area bajo la curva roc y para el punto de corte por tanto parece un modelo robusto. Por ultimo comprobamos que los datos de entramiento y de los tests no hayan cambiado mucho.

sensEspCorte(modelo2,data\_train,"varObjBin",0.34,"1")

sensEspCorte(modelo2,data\_test,"varObjBin",0.34,"1")



Summary(modelo2)

