Informe sobre la práctica de Logistic Regression

German Credit

Jorge Candia

PASO 1: Observar los datos

Como siempre, lo primero a realizar antes de empezar a probar modelos es echar un vistazo rápido a los datos para familiarizarnos con ellos y, en caso de ser necesario, realizar las técnicas de preprocesado que creamos convenientes.

Empezando esto último, empleamos el comando `any(is.na(data))` con la intención de averiguar si existe alguna observación incompleta. En este caso, la salida del método da *FALSE,* pero en caso contrario se deberían de eliminar estas filas con `data <- na.omit(data)`

El siguiente método a utilizar será `summary(data)`, que primero se empleará para detectar variables innecesarias, y luego se usará con la intención de comprender orientativamente qué significa cada variable, entre qué rangos se mueven y la magnitud de su dispersión.

En este caso no hace falta eliminar ninguna variable, pero desde el panel de variables se ha observado que 17 de las 20 variables dependientes son variables discretas, por lo que se procede a declararlas como tal con el método `factor(x)`.

Por último, se hace un scatterplot con las variables numéricas. Se puede observar cierta relación de colinealidad entre `Duration.of.Credit..month.` y `Credit.Amount` El comando utilizado para el scatterplot fue:

```
pairs(~Creditability+Duration.of.Credit..month.+Credit.Amount+Age..years.,
      main='scatterplots', col=c('blue'), data=data)
```

PASO 2: Proposición de un modelo

Con la visión global de los datos que se tiene ya, se procede a buscar un modelo de regresión logística que aproxime con la mayor exactitud posible la variable Creditability a partir de las otras 20 que vienen dadas en el dataset. Para valorar y comparar la efectividad de los modelos, observaremos las diferencias entre la *Null deviance* y la *Residual deviance* y el valor del parámetro *AIC*. Cuanta mayor diferencia entre las *deviances*, mejor. Para mayor facilidad a la hora de comparar modelos, se priorizará aquel que consiga un menor *AIC*.

Después de una prueba exhaustiva de distintos modelos, nos quedamos con el siguiente:

```
log_modelC <-glm(Creditability~Account.Balance+
       Duration.of.Credit..month.*Payment.Status.of.Previous.Credit
       +Purpose+Value.Savings.Stocks+Guarantors, family="binomial", data=data)
```

Con el comando `summary(log_modelC)` observamos un *AIC* de 997.4, y con `with(log_modelC, null.deviance -deviance)`, una diferencia entre deviances de 280.5, lo mejor que se ha podido obtener sin complicar el modelo con demasiadas variables. Por ejemplo, si añadir o hacer interaccionar dos variables baja el AIC menos 3 ó 4 unidades, se prefiere no complicar el modelo.

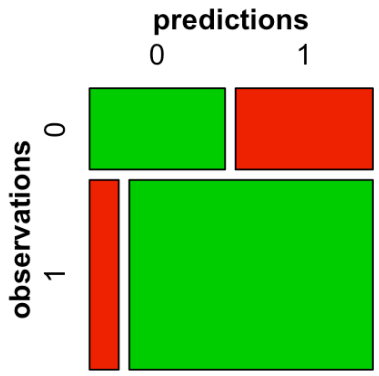
Ya con el modelo que creemos más adecuado, debemos comprobar que verdaderamente es útil y si es posible que sobre alguna variable. Para ello, se procede con los tests de *Likelihood* y de *Wald* respectivamente.

```
#Test of fit
#p value of degrees of freedom
with(log_modelC, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
#p value = 5.444555e-44 => The model fits significantly better than an empty model
```

```
#Wald tests
wald.test(b=coef(log_modelC), Sigma=vcov(log_modelC), Terms=2:4) #Account.Balance is significant
wald.test(b=coef(log_modelC), Sigma=vcov(log_modelC), Terms=10:18) #Purpose is significant
wald.test(b=coef(log_modelC), Sigma=vcov(log_modelC), Terms=21:22) #Guarantors is significant
```

Por último, plottearemos la matriz de confusión para una p umbral arbitraria de 0.5 para ver la distribución de predicciones que esperaríamos al introducir datos nuevos:

```
             predictions
observations   0   1
           0 149 151
           1  75 625
```



Como se puede observar, las predicciones de datos conocidos son bastante acertadas.

PASO 3: Capacidad de predicción del modelo

La distribución de esta última matriz es la que obtendríamos idealmente con datos nuevos si nuestro dataset de entrenamiento estuviese distribuido de igual manera que los nuevos inputs. Es decir, si nuestra muestra fuese perfectamente representativa de la realidad y nuestro modelo no sufriera de underfit u overfit.

Para comprobar que el modelo tiene una buena capacidad de predicción, se debe de probar con datos aún no explorados. Para esto, se hará una partición del dataset dado, guardando en la variable *trainingSet* el 80% de los datos, y en *testSet* el 20% restante. La idea es usar

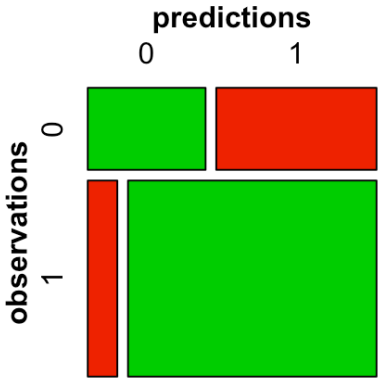
trainingSet para entrenar nuestro modelo y usar *testSet* para probar que el modelo se comporta de manera similar con datos nuevos que con los que se ha entrenado. Si esto es así, hemos terminado de construir nuestro modelo y se podrá utilizar con seguridad para hacer predicciones. En caso contrario, se deberán hacer los cambios pertinentes para reorientar el modelo, ya sea por el camino de modificar el set de variables o(/y) por el de limpiar el dataset, quitando outliers por ejemplo.

La distribución de los sets de training y de test debe de ser la misma, es decir, por precaución no se puede coger directamente el primer 80% de los datos para un grupo y el último 20% para el otro, sinó que se deberá hacer de manera aleatoria. Esto se hace de la siguiente forma:

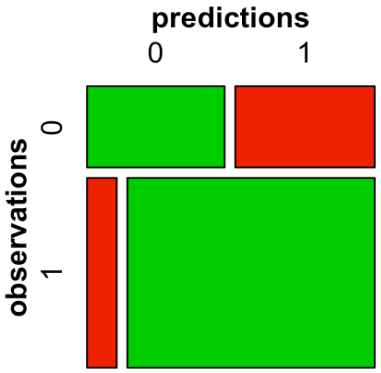
```
#Setteo un seed por si quiero repetir el estudio con exactamente las mismas muestras
set.seed(100)

#Creo un index con 800 números aleatorios que estén entre el 1 y el nº de muestras de
data (100) sin repetirse
index <- sample(nrow(data), 0.8*nrow(data), replace = F)
#Con el, hago la partición aleatoria de data para crear el trainSet y el testSet
trainSet <- data[index,]
testSet <- data[-index,]

#Entreno el modelo con trainSet, los datos de testSet no se contemplan en el modelo
train_modelC <- glm(Creditability~Account.Balance+
                    Duration.of.Credit..month.*Payment.Status.of.Previous.Credit
                  +Purpose+Value.Savings.Stocks+Guarantors, family="binomial",
data=trainSet)
summary(train_modelC) #Baja mucho el AIC y las deviances por tener muchas menos filas
summary(log_modelC) #Summary del modelo principal

#Con el modelo de entrenamiento, hago predicciones de los datos de test
testPredictions = predict(train_modelC, testSet, type="response")
testPredictions = as.numeric(testPredictions)

#Hago una matriz de confusión para ver si tiene la distribución esperada (uso la p de antes)
# p=0.5 => Si la predicción es mayor de 0.5, se cuenta el dato  como 1, si es menor o igual, 0
predictions<-ifelse(test=testPredictions>0.5,yes=1, no=0)
confusion_matrixTest<-table(testSet$Creditability,predictions,
                        dnn=c("observations", "predictions"))

confusion_matrixTest #Miro la tabla de predicciones
#La plotteo para ver los números gráficamente en forma de superficies
mosaic(confusion_matrixTest,shade=T,colorize=T,
       gp=gpar(fill=matrix(c("green3", "red2", "red2", "green3"),2,2)))

#La comparo con la primera
confusion_matrix
mosaic(confusion_matrix,shade=T,colorize=T,
       gp=gpar(fill=matrix(c("green3", "red2", "red2", "green3"),2,2)))
```

El resultado de esta nueva matriz es el siguiente, comparado con la primera matriz (la nueva arriba):

```
             predictions
observations   0   1
           0  25  34
           1  15 126
```




```
             predictions
observations   0   1
           0 149 151
           1  75 625
```



Como se puede apreciar, ambas matrices son muy parecidas. La sumas de las diagonales principales de las matrices entre el nº de predicciones son 0.755 para la matriz de test y 0.774 para matriz con todo el dataset para training. Estos números son interpretados como el porcentaje de acierto del modelo, por lo que se puede asegurar que este apenas sufre de overfitting/underfitting.

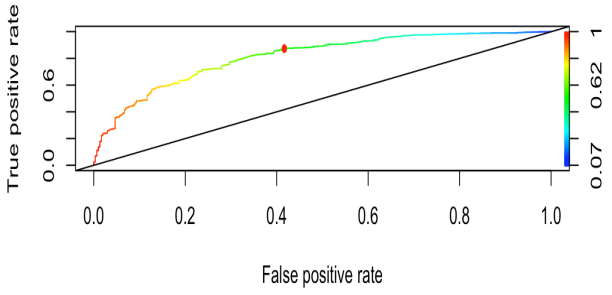
PASO 4: Curva ROC y Lift Charts

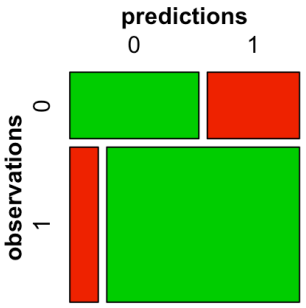
Hasta ahora hemos usado una p arbitraria de valor 0.5 para construir nuestra matriz de confusión pero, ¿cuál sería realmente el valor adecuado de la p umbral?. La curva ROC da la respuesta a esta pregunta. Esta curva representa todas las relaciones posibles en un modelo entre el True Positive Rate (eje de ordenadas) y el False Positive Rate (eje de abscisas), contemplando así todos los valores de p de 0 a 1.

Mediante el siguiente código, se puede plottear esta curva y obtener el punto donde la p es óptima, es decir, para la cual se maximizan los True Positives y los True Negatives (regiones coloreadas de verde en la matriz de confusión):

```
pred <- prediction(predictions= log_modelC$fitted.values, labels =
log_modelC$model$Creditability)
perf <- performance(pred,"tpr","fpr") #La curva ROC

# Punto de corte óptimo del clasificador (probabilidad óptima)
cost.perf <- performance(pred, measure ="cost")
opt.cut <- pred@cutoffs[[1]][which.min(cost.perf@y.values[[1]])] #La p óptima (0.543)

#coordenadas de la probabilidad óptima sobre la ROC
x<-perf@x.values[[1]][which.min(cost.perf@y.values[[1]])]
y<-perf@y.values[[1]][which.min(cost.perf@y.values[[1]])]

#La ROC pintada con el punto
plot(perf,colorize=TRUE,type="l")
abline(a=0,b=1)
points(x,y, pch=20, col="red")

# Área bajo la curva, para comparaciones entre modelos
AUC       <- performance(pred,measure="auc")
AUCaltura <- AUC@y.values[[1]]


#Confussion con la p umbral óptima de la ROC (0.543)
predictions<-ifelse(test=log_modelC$fitted.values>opt.cut,yes=1, no=0)
confusion_matrixOpt<-table(log_modelC$model$Creditability,predictions,
                           dnn=c("observations", "predictions"))

confusion_matrixOpt
#plotting the confussion matrix
mosaic(confusion_matrixOpt,shade=T,colorize=T,
       gp=gpar(fill=matrix(c("green3", "red2", "red2", "green3"),2,2)))
```

Los plots son los siguientes:



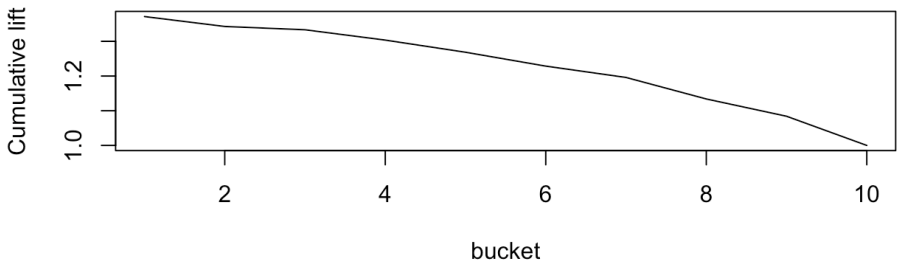



```
> confusion_matrixOpt
            predictions
observations   0   1
           0 175 125
           1  90 610
```

Con esta nueva probabilidad, el porcentaje de acierto sube del 0.774 respectivo a la p de 0.5 al 0.785 de la p de 0.543 . Un pequeño cambio pero que evidencia la efectividad de este cálculo. Además, la curva ROC sirve también por si queremos escoger una p con la que maximizar el True Positive Rate (a cambio de más falsos positivos) o bien sacrificar True Positives para minimizar los Falsos Positivos.

Otra manera de sacar información del modelo es con un Lift Chart y con un Cumulative Chart. Estas curvas representan de forma visual la efectividad del modelo de predicción calculada como la ratio entre los resultados obtenidos con y sin el modelo. Cuanto mayor sea el área entre las curvas y la línea base (que representa a los datos sin modelo), mejor es el modelo. El código para plottear el Lift Chart es el siguiente:

```
library(lift) #for lift curve
plotLift(log_modelC$fitted.values,log_modelC$model$Creditability, cumulative = TRUE,
         n.buckets = 10)
TopDecileLift(log_modelC$fitted.values,log_modelC$model$Creditability)
#1.371, correspondiente al primer decil
```



La interpretación del lift chart es la siguiente: Si se concede el crédito únicamente al 10% de las personas con mayor p, el número de personas que lo devolverá será igual a 1.371 (el valor correspondiente al decil/bucket 1) multiplicado por el número de personas que devolvería el crédito sin usar ningún modelo (repartiendo créditos a todas las personas del dataset). Si se lo concedemos al 20% con mayor p, lo devolverá el valor correspondiente al decil 2 por el número de personas que lo devolvería sin usar ningún modelo. Esto se extiende hasta llegar al décimo decil, que es el 100% del dataset. Como es lógico, en este valor coincide el número de personas que devuelven el crédito usando el modelo como sin usarlo.

El gráfico de Ganancias Acumuladas describe lo mismo pero en otra escala, comparando los deciles (interpretable como el porcentaje de personas contactadas) con el porcentaje de personas que devuelven el crédito respecto del 100% que lo devuelven. Por ejemplo, para el 10% de mayor p, devuelven el crédito el 13.71% de las personas que devolverían el crédito si se concediera a todo el mundo (sin modelo este porcentaje bajaría al 10%).

PASO 5: Otras características interesantes de cara al objetivo perseguido

De cara a conseguir un modelo mejor con una búsqueda aún más exhaustiva, podrían usarse comparaciones de AUCs. Como se ha mencionado anteriormente, por lo general cuanto mayor AUC mejor es el modelo.

Otro observación relevante sobre este estudio es que a la hora de comparar la matriz de confusión construida a partir de las predicciones del training set, esta se comparó con la matriz dada por el modelo completo (construida a partir de las predicciones de todo el dataset). Como realmente el modelo usado fue el de training, se podría haber comparado con la matriz de training en vez de con la completa, pero como ambas son casi iguales, no tiene demasiada importancia.

Por último, si el objetivo del modelo no fuera el de obtener el mayor número de predicciones correctas posibles sino por ejemplo priorizar el detectar el mayor número de morosos posible, se podría usar la ROC curve con este fin, obteniendo la p umbral correspondiente a obtener el ratio de Falsos Positivos (positivo significa que devuelve el crédito) que se crea conveniente, teniendo en cuenta que cuanto más bajo sea el FPR, menor será también el de Verdadero Positivo. Es decir, el número de personas a las que prestar un crédito será significativamente menor, pero con mayor probabilidad de que este sea devuelto.

La obtención del FPR conveniente no es objeto de este proyecto. Es un problema de optimización en el que se maximiza la función retorno de inversión teniendo en cuenta las variables de balance positivo al recibir el crédito de vuelta más el retorno negativo de que no se devuelva el crédito, siendo las variables el FPR y por ejemplo los intereses del crédito.

PASO 6: Conclusiones y recomendaciones del estudio

En esta práctica se ha hecho evidente lo positivo y útil que es el empleo de un modelo de regresión logística frente a no usarlo a la hora de hacer predicciones. Específicamente se ha visto muy claro el el Lift chart, además de lo representado en la matriz de confusión, ya que ajustada a la ROC se obtuvo un porcentaje de acierto del 78.5% sobre la posible morosidad del deudor.

Como recomendación, cabe recordar lo dicho anteriormente. Es conveniente hallar el FPR con el que obtener mayor beneficio y, en mi opinión, sería también conveniente en lo referente a los límites próximos a la p correspondiente a ese FPR, que la decisión no sea conceder el crédito o no, sinó desarrollar una forma de asignar distintos tipos de interés a las personas que se encuentren en los alrededores de estos márgenes.

Nota: Se muestra en gran medida el código de la práctica con la idea de poder estudiar para el examen desde este informe.