Laboratorio 7: Classificazione

Un esempio di classificazione logistica

L.Egidi, N. Torelli

Contents

Valutazione del merito di credito

Illustrazione dei dati e obiettivo dell'analisi

Nell'emettere un credito al cliente, le banche verificano la "solvibilità" o il "merito creditizio" del cliente, ovvero la sua capacità di rimborsare il credito. Si tratta quindi di classificare i clineti in due categorie: quelli "solvibili" e quelli che si ritiene non ripagheranno il credito. Si tratta quindi di un problema di classificazione binaria. Affronteremo ora questo semplice problema utilizzando un modello di regerssione logistica. Il modello ci consentirà di stimare la probabilità che un credito venga rimborsato in fuznione di alcune caratteristiche di chi richeide il prestito.

Prenderemo in considerazione un dataset di $n = 1000$ crediti emessi da una banca tedesca. Ad ogni cliente è associata una risposta binaria definita come:

$$y_i = 0 \text{ il cliente ha rimborsato il prestito}$$
$$y_i = 1 \text{ il cliente NON ha rimborsato il prestito}$$

Le altre variabili da utilizzare per la classificazione sono:

Variable name	Description
acc	nessun conto corrente, conto corrente ma con frequenti sconfinamenti, buon conto corrente

Variable name	Description
duration	durata per il rimborso del prestito
amount	ammontare in K-euro del prestito
moral	Comportamento in precedenti prestiti. 1 = good
intuse	uso del prestito. 1 = fini privati, 0 = lavoro

Le informazioni disponibili sono registrate come file di testo in `credit1.txt`, e possono essere lette dal comando

```
Credit <- read.table("credit1.txt", header=TRUE)
```

I dati sono in un data frame con 1000 righe e 7 colonne. Diamo uno sguardo ai primi 5 record:

```
Credit[1:5,]
```

```
##   y  acc duration    amount moral intuse
## 1 0   no       24 1.5118900     1      1
## 2 0  bad       12 0.7280797     1      1
## 3 0 good       18 1.0486600     1      1
## 4 0 good       12 2.3902900     1      1
## 5 0 good       24 1.5226270     1      0
```

Per rendere queste variabili direttamente disponibile nel seguito

```
attach(Credit)
```

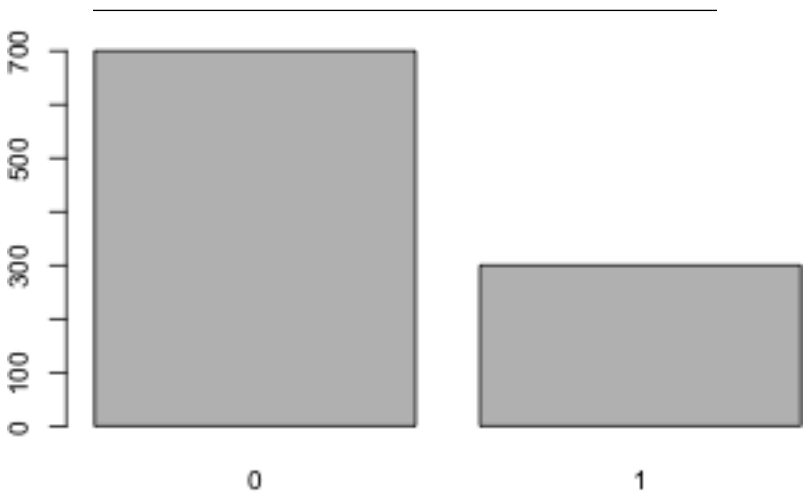
Stimiamo un modello di regressione logistica

Per prima cosa vediamo la distribuzione della variabile risposta. Quanti clienti non hanno rimborsato il prestito?

```
table(y)
```

```
## y
##   0   1
## 700 300
```

```
barplot(table(y))
```



Possiamo ora provare a stimare un primo modello di regressione logistica (utilizziamo tutti i dati per ora)

```
mod1=glm(y~ acc+duration+amount+moral+intuse,family=binomial(link=logit))
summary(mod1)
```

```
##
## Call:
## glm(formula = y ~ acc + duration + amount + moral + intuse, family = binomial(link = logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8876  -0.8440  -0.4628   0.9629   2.3620
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.284402   0.302579  -0.940 0.347255
## accgood     -1.337748   0.201127  -6.651 2.91e-11 ***
## accno        0.617659   0.174728   3.535 0.000408 ***
## duration     0.033233   0.007746   4.290 1.78e-05 ***
## amount       0.045875   0.064092   0.716 0.474134
## moral       -0.986066   0.250891  -3.930 8.49e-05 ***
## intuse      -0.425536   0.158272  -2.689 0.007174 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.7  on 999  degrees of freedom
## Residual deviance: 1029.8  on 993  degrees of freedom
## AIC: 1043.8
##
## Number of Fisher Scoring iterations: 4
```

Osservando p-values possiamo vedere che tutte le variabili sono significativamente diverse da zero tranne `importo`.

I segni dei coefficienti stimati sono quelli attesi. Ricordiamo che stiamo valutando l'effetto delle covariate sulla probabilità di non essere meritevoli di credito.

Il non avere alcun conto corrente sembra aumentare la probabilità di insolvenza anche rispetto a quelli con un "cattivo" conto corrente. Mentre avere un buon conto corrente diminuisce notevolmente la probabilità di essere insolvente. Possiamo valutare l'effetto sulle probabilità calcolando l'odds ratio

```
exp(mod1$coefficients[3])
```

```
##    accno
## 1.854582
```

A parità di altre condizioni, per chi ha un cattivo conto gli odds di insolvenza sono sono quasi il doppio ripseto a chi ha un buon conto.

Il coefficiente associato alla `durata` è significativamente diverso da 0 e ha un segno positivo. Ciò significa che più lungo è il termine per rimborsare il debito e maggiore è la probabilità di insolvenza.

Un modello più complesso

Il modello può essere complicato cercando di vedere se c'è un effetto non lineare delle due covariate quantitative.

Introduciamo anche i quadrati delle variabili quantitative

```
amountsq=amount^2
durationsq=duration^2
```

Proviamo il nuovo modello

```
mod2=glm(y~acc+duration+durationsq+amount+amountsq+moral+intuse,family=binomial(link=logit))
summary(mod2)
```

```
##
## Call:
## glm(formula = y ~ acc + duration + durationsq + amount + amountsq +
##     moral + intuse, family = binomial(link = logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4157  -0.8124  -0.4719   0.9585   2.6326
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.4877826  0.3896495  -1.252 0.210625
## accgood     -1.3374022  0.2024364  -6.607 3.93e-11 ***
## accno        0.6178347  0.1761733   3.507 0.000453 ***
## duration     0.0921909  0.0252941   3.645 0.000268 ***
## durationsq  -0.0009094  0.0004133  -2.200 0.027781 *
## amount      -0.5165866  0.1926907  -2.681 0.007342 **
## amountsq     0.0878646  0.0285982   3.072 0.002124 **
## moral       -0.9953315  0.2551618  -3.901 9.59e-05 ***
## intuse      -0.4039789  0.1601534  -2.522 0.011654 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.7  on 999  degrees of freedom
## Residual deviance: 1017.4  on 991  degrees of freedom
## AIC: 1035.4
##
## Number of Fisher Scoring iterations: 4
```

Sembra che il secondo modello sia migliore (la devianza residua è ora 1017,4, prima era 1029,8).

Stimare la probabilità di insolvenza

Se assumiamo che le informazioni sulle covariate sono disponibili nel momento in cui la banca deve decidere se dare il prestito a un nuovo cliente, allora tale modello può essere la base per costruire un algoritmo di classificazione logistica.

In primo luogo dovremo stimare la probabilità che un creditore sia insolvente. Si consideri un cliente con le seguenti caratteristiche: Non ha conto corrente, il termine per il rimborso del debito è di 36 mesi, l'importo è di 10000 euro, il comportamento di pagamento precedente era pessimo e i soldi sono destinati ad essere utilizzati per affari. Quanto è rischioso concedergli un prestito? La probabilità prevista di insolvenza è:

```
mio=data.frame(acc="no",duration=36,durationsq=36^2,amount=10,amountsq=100,moral=0,intuse=0)
predict(mod2,newdata=mio, type="response")
```

```
##         1
## 0.9972431
```

In questo caso il rischio di insolvenza è molto elevato.

Costruire l'algoritmo di classificazione

Suddividiamo casualmente i dati disponibili in training set (70% per la creazione di un modello predittivo) e test set (30% per la valutazione del modello). L'obiettivo è poi di misurare la *precisione predittiva* del modello in base al test set.

```
library(dplyr)
# observations' length
n <- length(Credit$y)
amountsq=amount^2
durationsq=duration^2
Credit<-cbind(Credit, amountsq, durationsq)
# training set
set.seed(1234)
train <- sample_n(Credit, 0.7*n)
# test set
test <- setdiff(Credit, train)
```

E costruiamo il modello usando il solo training set

```
# Fit the model
full.model <- glm(y ~., data = train,
                  family = binomial)
```

Ora possiamo stimare le probabilità di insolvenza per il test set e poi definire una regola predittiva fissando una soglia. Usiamo la regola per cui se $\hat{p} > 0.5$ classifichiamo il creditore come insolvente.

```
# Make predictions
probabilities <- predict(full.model,
                         newdata = test,
                         type = "response")
previsto <- ifelse(probabilities > 0.5, "insolvente", "solvente")
previsto<-relevel(factor(previsto), "solvente")
osservato<-factor(test$y)
levels(osservato)=c("solvente","insolvente")
```

Possiamo ora calcolare la matrice di confusione

```
# Matrice di confusione
conf<-table(previsto,osservato)
conf
```

```
##             osservato
## previsto     solvente insolvente
##   solvente        183         47
##   insolvente       26         44
```

```
# Accuratezza
acc<-sum(diag(conf))/length(test$y)
acc
```

```
## [1] 0.7566667
```

```
# Sensibilità o Recall (True positive rate)
Spec<-conf[2,2]/sum(conf[,2])
Spec
```

```
## [1] 0.4835165
```

```
# Specificità (True negative rate)
recall<-conf[1,1]/sum(conf[,1])
recall
```

```
## [1] 0.8755981
```

```
# False positive rate (1-specificità)
1-Spec
```

```
## [1] 0.5164835
```

```
# Precisione
precision<-conf[1,1]/sum(conf[1,])
precision
```

```
## [1] 0.7956522
```

```
#F1
F1=2*(precision*recall)/(recall+precision)
F1
```

```
## [1] 0.833713
```

Regressione logistica con GAM

```
library(mgcv)
model.gam <- gam(y ~ acc+s(duration)+s(amount)+moral+intuse,family=binomial, data = train)
summary(model.gam)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ acc + s(duration) + s(amount) + moral + intuse
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.7929     0.3819   2.076 0.037873 *
## accgood      -1.1987     0.2475  -4.844 1.27e-06 ***
## accno         0.9224     0.2202   4.188 2.81e-05 ***
## moral        -1.2752     0.3498  -3.645 0.000267 ***
## intuse       -0.4642     0.1972  -2.354 0.018570 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df Chi.sq  p-value
## s(duration) 8.786  8.955  38.81 7.67e-06 ***
## s(amount)   2.696  3.387  15.02  0.00295 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.222   Deviance explained = 20.5%
## UBRE = 0.016185  Scale est. = 1         n = 700
```

```
prob.gam <- predict(model.gam,
                        newdata = test,
                        type = "response")
previsto.gam <- ifelse(prob.gam > 0.5, "insolvente", "solvente")
previsto.gam<-relevel(factor(previsto.gam), "solvente")
# osservato<-factor(test$y)
# levels(osservato)=c("solvente","insolvente")

conf<-table(previsto.gam,osservato)
conf
```

```
##             osservato
## previsto.gam solvente insolvente
##   solvente        178         53
##   insolvente       31         38
```

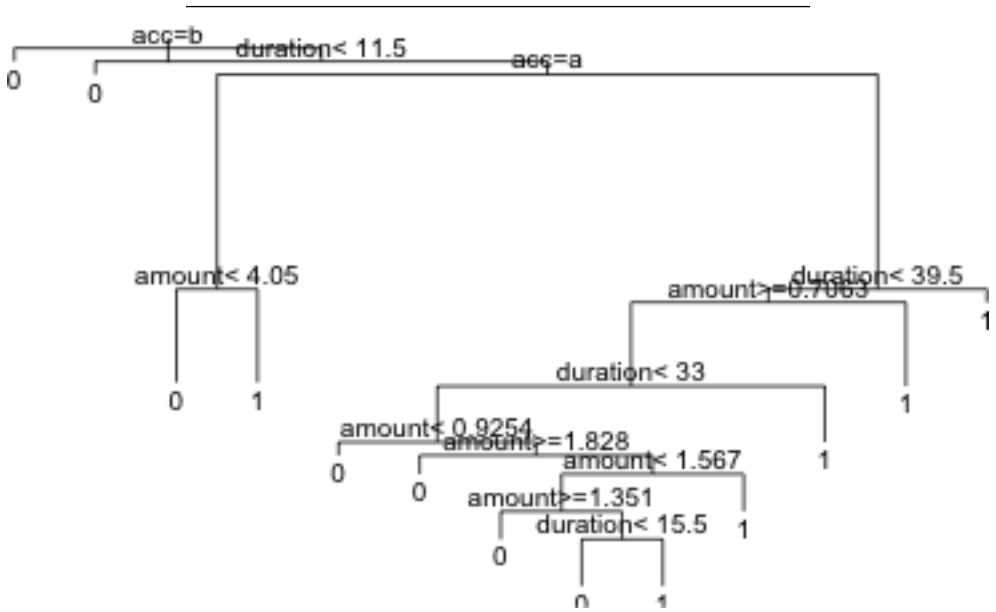
Albero di Classificazione

```
library(rpart)
model.tree <- rpart(y~acc+duration+amount+moral+intuse, data = train, method='class')
model.tree
```

```
## n= 700
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##    1) root 700 209 0 (0.7014286 0.2985714)
##      2) acc=good 278  33 0 (0.8812950 0.1187050) *
##      3) acc=bad,no 422 176 0 (0.5829384 0.4170616)
##        6) duration< 11.5 69  14 0 (0.7971014 0.2028986) *
##        7) duration>=11.5 353 162 0 (0.5410765 0.4589235)
##         14) acc=bad 184  66 0 (0.6413043 0.3586957)
##           28) amount< 4.050454 164  51 0 (0.6890244 0.3109756) *
##           29) amount>=4.050454 20   5 1 (0.2500000 0.7500000) *
##         15) acc=no 169  73 1 (0.4319527 0.5680473)
##           30) duration< 39.5 153  72 1 (0.4705882 0.5294118)
##             60) amount>=0.7063497 109  50 0 (0.5412844 0.4587156)
##              120) duration< 33 93  39 0 (0.5806452 0.4193548)
##                240) amount< 0.9254383 14   3 0 (0.7857143 0.2142857) *
##                241) amount>=0.9254383 79  36 0 (0.5443038 0.4556962)
##                  482) amount>=1.828124 29  10 0 (0.6551724 0.3448276) *
##                  483) amount< 1.828124 50  24 1 (0.4800000 0.5200000)
##                    966) amount< 1.566854 38  17 0 (0.5526316 0.4473684)
##                     1932) amount>=1.350833 9   1 0 (0.8888889 0.1111111) *
##                     1933) amount< 1.350833 29  13 1 (0.4482759 0.5517241)
##                       3866) duration< 15.5 7   1 0 (0.8571429 0.1428571) *
##                       3867) duration>=15.5 22   7 1 (0.3181818 0.6818182) *
```

```
##                     967) amount>=1.566854 12   3 1 (0.2500000 0.7500000) *
##                121) duration>=33 16   5 1 (0.3125000 0.6875000) *
##               61) amount< 0.7063497 44  13 1 (0.2954545 0.7045455) *
##             31) duration>=39.5 16   1 1 (0.0625000 0.9375000) *
```

```
plot(model.tree); text(model.tree)
```




```
prob.tree <- predict(model.tree, newdata = test, type="class")
previsto.tree<-factor(prob.tree, labels=c("solvente","insolvente"))
# length(prob.tree)
# table(prob.tree)
# previsto.tree <- ifelse(prob.tree > 0.5, "insolvente", "solvente")
# previsto.tree<-relevel(factor(previsto.tree), "solvente")
# osservato<-factor(test$y)
# levels(osservato)=c("solvente","insolvente")

 conf<-table(previsto.tree,osservato)
 conf
```

```
##              osservato
## previsto.tree solvente insolvente
##    solvente        187         64
##    insolvente       22         27
```

Esercizi

Dati sul diabete degli indiani Pima

Descrizione dei dati

Il dataset `PimaIndiansDiabetes2` contiene 768 osservazioni su 9 variabili (si trova nel pacchetto "mlbench"). L'obiettivo è quello di classificare i pazienti come "diabetici" o "non diabetici" sulla base di alcune variabili cliniche elencate di seguito:

Nome della variabile	Descrizione
pregnant	Number of times pregnant
glucose	Plasma glucose concentration (glucose tolerance test)
pressure	Diastolic blood pressure (mm Hg)
triceps	Triceps skin fold thickness (mm)
insulin	2-Hour serum insulin (mu U/ml)
mass	Body mass index (weight in kg/(height in m)^2)
pedigree	Diabetes pedigree function
age	Age (years)
diabetes	Class variable (test for diabetes)

Esercizio 1.

1. Usare la regressione logistica per costruire un classificatore che preveda il diabete
2. Costruire e disegnare la curva ROC
3. Calcolare l'indicatore AUC

Esercizio 2.

Per i dati sul diabete degli indiani Pima: 1. Usare regressione logistica semiparametrica 2. Usare un albero di classificazione 3. Confrontare l'accuratezza e costruire la curva ROC in entrambi i classificatori

Esercizio 3.

1. Considerarer i dati sugli iris (sono presenti in R, basta renderli disponibili con `data(iris)`)
2. Costruire un albero di classificazione per le 3 specie di iris e ottenere lka matrice di confusione