Esempio di classificazione logistica

Valutazione del merito creditizio

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
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
```

Prima di procedere all'analisi è buona prassi definire, anche al fine del corretto uso nei successivi modelli, le variabili categoriali come fattori (anche se come nel caso di `y`, `moral` e `intuse` sono lette come valori numerici interi).

```
Credit$y<-factor(Credit$y)
Credit$acc<-factor(Credit$acc)
Credit$moral<-factor(Credit$moral)
Credit$intuse<-factor(Credit$intuse)
```

Per i fattori espressi con valori numerici può risultare utile anche attribuire un attributo testuale a ogni livello per rendere poi più comprensibili i risultati

```
levels(Credit$moral)<-c("cattivo","buono")
levels(Credit$intuse)<-c("lavoro","privato")
```

Per rendere queste variabili direttamente disponibile nel seguito

```
attach(Credit)
```

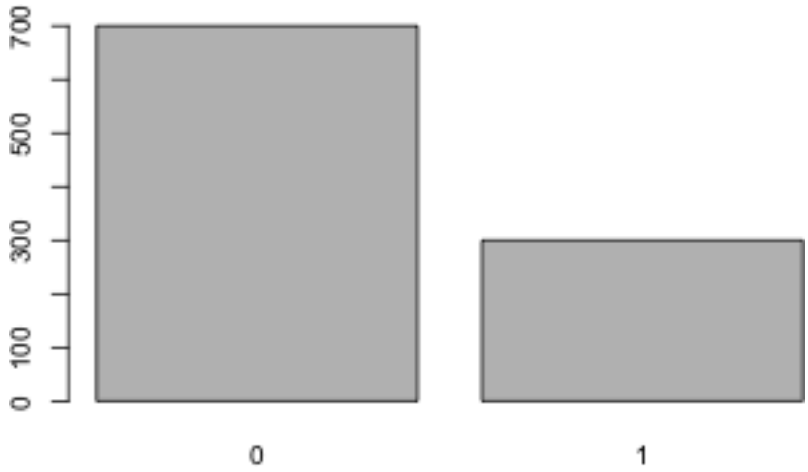
Stimiamo un modello di regressione logistica

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))
##
## 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
## moralbuono    -0.986066   0.250891  -3.930 8.49e-05 ***
## intuseprivato -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 i coefficienti stimati associati a tutte le variabili possono ritenersi significativamente diversi da zero tranne `amount`.

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 rispetto a chi ha un buon conto.

Il coefficiente associato a `duration` 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 quindi 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))
##
## 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 **
## moralbuono    -0.9953315  0.2551618  -3.901 9.59e-05 ***
## intuseprivato -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). Tale differenza è significativa in quanto il valore 1029,8-1017,4=12,4 è grande per essere plausibilmente tratto da una variabile χ^2_2 (che è l'approssimazione asintotica per la distribuzione della differenza tra le devianze nell'ipotesi che i due modelli siano equivalenti - ovvero che i coefficienti associati alle due nuove variabili siano entrambi uguali a 0).

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 risulta pari a:

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

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

In questo caso il rischio di insolvenza è molto elevato e quindi la banca non concederà il prestito.

Costruire l'algoritmo di classificazione

Al fine di costruire e poi valutare l'algoritmo di classificazione basato sul modello di regressione logistica precedentemente stimato, 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.

Carichiamo il pacchetto `dplyr`, estremamente utile per manipolare velocemente data frame. Useremo i comandi `slice_sample` e `setdiff` per creare training e test set.

```
library(dplyr)

n <- length(Credit$y) # dimensione del data frame (numero di righe)
amountsq=amount^2
durationsq=duration^2
Credit<-cbind(Credit, amountsq, durationsq) # Aggiungiamo al data frame le variabili al quadrato

# training set
set.seed(1234)
ntrain= round(0.7*n)
train <- slice_sample(Credit,n=ntrain)
# test set
test <- setdiff(Credit, train)
```

Ora 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 la stima della probabilità di insolvenza $\hat{p} > 0.5$ classifichiamo il creditore come insolvente.

```
# Make predictions
probabilities <- predict(full.model,
                         newdata = test,
                         type = "response")
previsto <- ifelse(probabilities > 0.5, "insolvente", "solvente")
```

```r
previsto<-relevel(factor(previsto), "solvente")
osservato<-factor(test$y)
levels(osservato)=c("solvente","insolvente")
```

Calcolare le misure di accuratezza

Possiamo ora calcolare la matrice di confusione

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

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

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

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

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

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

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

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

```r
# False positive rate (1-specificità)
1-spec
```

```
## [1] 0.1244019
```

```r
# Precisione
precision<-conf[2,2]/sum(conf[2,])
precision
```

```
## [1] 0.6285714
```

```r
#F1 (con beta=1)
F1=2*(precision*recall)/(recall+precision)
F1
```

```
## [1] 0.5465839
```