Analisi Pilgrim Bank - Tesina

Luca Bajardi e Francesca Collini

31/07/2020

Introduzione

Utilizziamo il dataset fornito dalla "Harvad Business School" relativo alla redditività di alcuni clienti nella Pilgrim Bank nel corso degli anni 1999 e 2000.

Inizialmente, cerchiamo di predire il profitto di un cliente in base alle sue caratteristiche anagrafiche, questa analisi potrebbe essere utile per la banca nel caso in cui debba decidere ad esempio se concedere un prestito oppure no. Vorrei quindi una previsione molto accurata in modo da capire se quel determinato cliente sarà in grado di restituire il denaro e scegliere quelle azioni finanziarie che siano il meno rischiose possibile. Nella previsione è importante tenere conto del fatto che potrebbero mancare alcune informazioni relative ai clienti, in questo caso è necessario scegliere il modo opportuno per cercare di non buttare via quei dati, ma utilizzarli comunque per la nostra previsione.

Un altro tipo di analisi che vogliamo effettuare è quella nel tempo. Infatti, per ciascun cliente che è presente nella base dati nel 1999, abbiamo un'informazione sull'anno successivo: sappiamo se il cliente decide di lasciare la banca oppure di rimanerci e in quest'ultimo caso conosciamo anche il suo profitto in quell'anno. Quindi l'obiettivo della nostra analisi è quello di riuscire a prevedere se un cliente rimarrà fedele alla banca o meno. Per farlo, utilizziamo diversi metodi di classificazione tra questi l'albero di decisione e il knn. Inoltre poiché la variabile target alla quale siamo interessati è binaria, possiamo utilizzare anche la regressione logistica e l'svm.

Infine ci concentriamo sull'eventuale passaggio di un cliente dal servizio offline a quello online. Se riuscissimo a capire che, per una determinata tipologia di clienti, il passaggio al canale online, permette di guadagnare maggiormente, la banca potrebbe ad esempio portare avanti delle campagne pubblicitarie per fare in modo che i clienti scelgano l'opzione più profittevole. Per ottenere questo tipo di informazione, applichiamo un algoritmo di clustering (il k-means) considerando solamente quei clienti che nel 1999 utilizzavano il servizio di banca offline e che nel 2000 sono rimasti clienti della banca. All'interno di ogni cluster poi, calcoliamo la media, separatamente per coloro che hanno scelto di passare al digitale e per quelli che sono rimasti con il servizio standard. In questo modo riusciamo a definire delle tipologie di clienti e capire se per la banca risulta più o meno vantaggioso far passare il cliente alla banca online.

Esplorazione dei dati

Carichiamo i dati leggendo il file csv e settiamo il seme del generatore pseudo-casuale così da avere i risultati sempre uguali.

```
rm(list =ls())
set.seed(1)
Pilgrim = read.csv(file = "PilgrimABC.csv", header=T)
attach(Pilgrim)
```

Osserviamo che il nostro dataset contiene 31634 osservazioni e 11 variabili (anche se le ultime due non sono interessanti ai fini della nostra analisi). I dati sono relativi a due anni, 1999 e 2000, per ciascuno di essi, abbiamo a disposizione un'informazione relativa al profitto (o alla perdita) di un determinato cliente in quell'anno e un'informazione che ci dice se quel cliente in quell'anno ha utilizzato l'opzione di online banking

oppure no: la variabile Online quindi è una variabile binaria. Abbiamo poi a dispozione altre informazioni relative all'anagrafica del cliente:

- AGE: variabile categorica che indica a che fascia di età appartiene il cliente (1 = less than 15 years; 2 = 15-24 years; 3 = 25-34 years; 4 = 35-44 years; 5 = 45-54 years; 6 = 55-64 years; 7 = 65 years and older.)
- INCOME: variabile categorica che indica il range del reddito di ciascun cliente (1 = less than $\$15,000$; 2 = $\$15,000 - \$19,999$; 3 = $\$20,000 - \$29,999$; 4 = $\$30,000 - \$39,999$; 5 = $\$40,000 - \$49,999$; 6 = $\$50,000 - \$74,999$; 7 = $\$75,000 - \$99,999$; 8 = $\$100,000 - \$124,999$; 9 = $\$125,000$ and more.)
- TENURE: variabile quantitativa continua che rappresenta l'età di servizio.
- DISTRICT: variabile categorica che indica una delle tre regioni geografiche in cui si trova il cliente.

```
dim(Pilgrim)
```

```
## [1] 31634    11
```

```
names(Pilgrim[, 1:9])
```

```
## [1] "ID"         "X9Profit"   "X9Online"   "X9Age"      "X9Inc"
## [6] "X9Tenure"   "X9District" "X0Profit"   "X0Online"
```

Rinominiamo le colonne per evitare problemi con i nomi:

```
Profit=Pilgrim$X9Profit
Online=Pilgrim$X9Online
Age=Pilgrim$X9Age
Income=Pilgrim$X9Inc
Tenure=Pilgrim$X9Tenure
District=Pilgrim$X9District
```

Stampiamo le prime righe del dataset per avere un'idea più chiara dei dati che abbiamo a disposizione.

```
head(Pilgrim)
```

```
##   ID X9Profit X9Online X9Age X9Inc X9Tenure X9District X0Profit X0Online
## 1  1       21        0    NA    NA     6.33       1200       NA       NA
## 2  2       -6        0     6     3    29.50       1200      -32        0
## 3  3      -49        1     5     5    26.41       1100      -22        1
## 4  4       -4        0    NA    NA     2.25       1200       NA       NA
## 5  5      -61        0     2     9     9.91       1200       -4        0
## 6  6      -38        0    NA     3     2.33       1300       14        0
##   X9Billpay X0Billpay
## 1         0        NA
## 2         0         0
## 3         0         0
## 4         0        NA
## 5         0         0
## 6         0         0
```

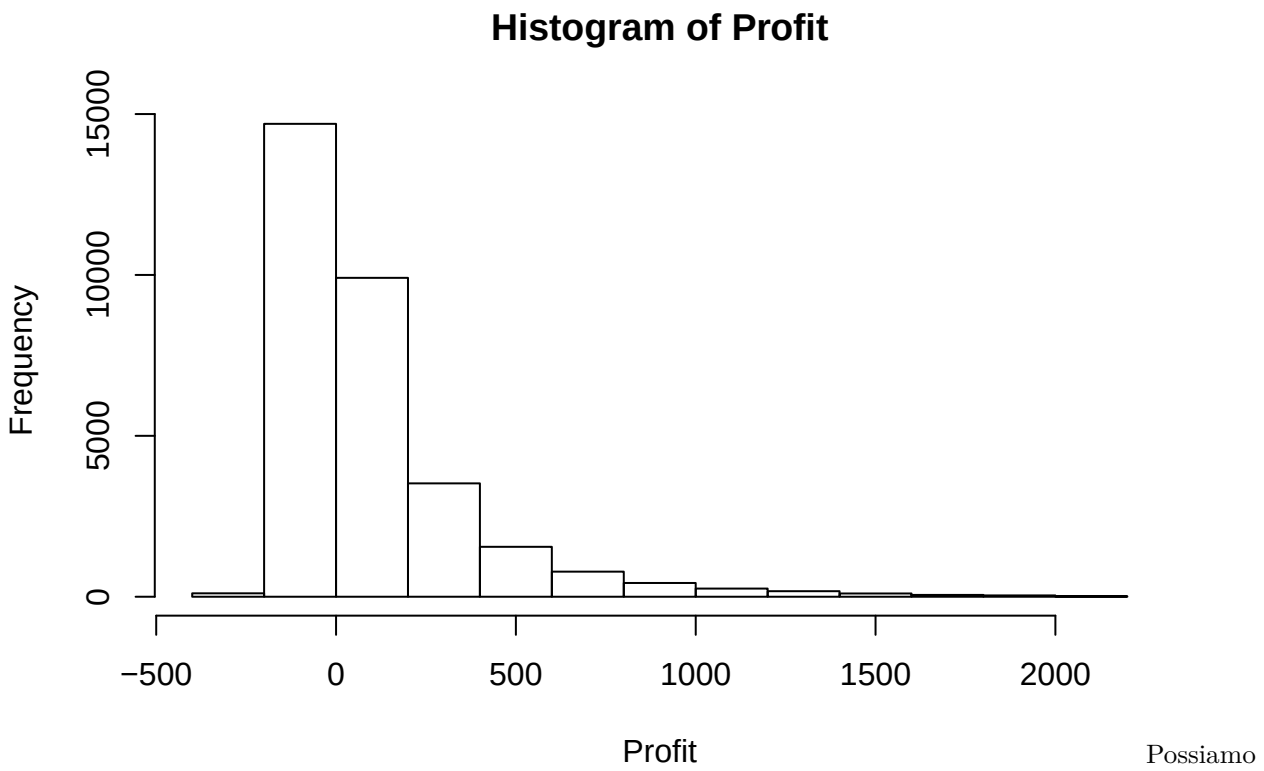
Dal summary del profitto possiamo facilmente osservare che la distribuzione è molto asimmetrica perchè media e mediana sono molto diverse tra loro. Inoltre notiamo che tutto il primo quartile è negativo, quindi c'è un numero abbastanza rilevante di persone che fa perdere soldi alla banca.

```
summary(Profit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -221.0   -34.0     9.0   111.5   164.0  2071.0
```

Anche dall'istogramma sul profitto nel 1999, possiamo notare che c'è un discreto numero di clienti che mi fa perdere e che l'istogramma è molto scodato a destra quindi ci sono pochissimi clienti che mi fanno guadagnare molto.

```
hist(Profit)
```



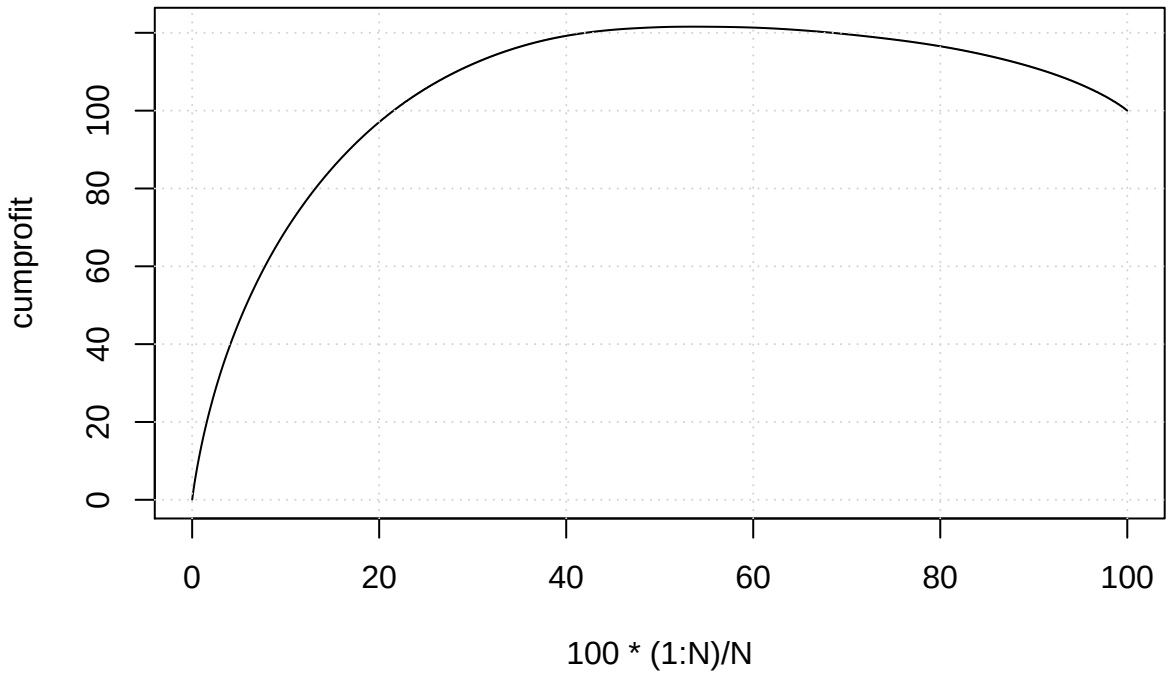
Possiamo notare che ci sono solo 16832 clienti che generano profitto sulle 31634 osservazioni presenti nel dataset.

```
N=length(Profit)
Nprofitable = sum(Profit>0)
cat('profitable = ', Nprofitable, ' out of ', N, '\n')
```

```
## profitable =  16832  out of  31634
```

Tramite la curva di Pareto possiamo confermare che poche persone fanno guadagnare tanto, mentre la maggior parte delle persone fa guadagnare poco o addirittura fa perdere guadagno.

```
cumprofit=cumsum(sort(Profit,decreasing=TRUE))*100/sum(Profit)
plot(100*(1:N)/N,cumprofit,type='l')
grid()
```



Online vs Offline

Dall'analisi dei profitti medi possiamo notare che il profitto generato da chi utilizza i servizi online è maggiore rispetto a chi li utilizza offline.

```
cat('average profit ', mean(Profit), '\n')
```

```
## average profit  111.5027
```

```
ProfitOnline = Profit[Online==1]
cat('average profit ON', mean(ProfitOnline), '\n')
```

```
## average profit ON 116.6668
```

```
ProfitOffline = Profit[Online==0]
cat('average profit OFF', mean(ProfitOffline), '\n')
```

```
## average profit OFF 110.7862
```

Quello che vorremmo capire è se questa differenza è statisticamente significativa, quindi se davvero coloro che utilizzano i servizi online mi permettono di guadagnare di più. Per questo motivo, possiamo eseguire un `t.test` sulle due popolazioni, i clienti che utilizzano il servizio online e coloro che non lo usano, per andare a vedere quanto vale il p-value. L'ipotesi nulla in questo caso è che le medie siano uguali e che quindi la differenza tra le medie delle due popolazioni non sia significativa. Analizziamo inoltre l'intervallo di confidenza per vedere se i dati sono sufficienti o no. Infatti se intervallo fosse troppo grande vorrebbe dire che avrei bisogno di più clienti per formulare una teoria generale.

```
cat('Conf int profit ', t.test(Profit)$conf.int, '\n')
```

```
## Conf int profit  108.496 114.5094
```

```
cat('p-value difference ',t.test(ProfitOnline,ProfitOffline)$p.value, '\n')
```

```
## p-value difference  0.2254368
```

Risulta esserci una differenza di circa 6 dollari tra le due popolazioni, quindi non risulta più di tanto significtiva dal lato business, ma anche il p-value è maggiore di 0.05 e quindi possiamo concludere che questa differenza

non è significativa nemmeno dal punto di vista statistico. In conclusione non possiamo rifiutare l'ipotesi nulla sulla differenza tra le medie.

Possiamo ottenere lo stesso risultato impostando il modello di regressione classico, utilizzando la variabile `Online` per prevedere la variabile `Profit`:

```
mod = lm(Profit ~ Online)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -337.67 -144.79 -101.79   52.21 1960.21
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  110.786      1.637  67.678   <2e-16 ***
## Online         5.881      4.690   1.254     0.21
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 272.8 on 31632 degrees of freedom
## Multiple R-squared:  4.97e-05,   Adjusted R-squared:  1.809e-05
## F-statistic: 1.572 on 1 and 31632 DF,  p-value: 0.2099
```

Dal summary del modello, possiamo osservare che il valore dell'intercetta è la media del profitto tra coloro che operano offline, mentre il valore aggiunto di quelli che operano online è di quasi 6 dollari come la differenza tra la media di quelli che operano online e la media di quelli che operano offline. Anche in questo caso però, il p-value è superiore a 0.05 come in precedenza. Questo significa che la variabile `Online` non è significativa. Tuttavia, questo potrebbe essere dovuto al fatto che stiamo mettendo insieme clienti molto diversi tra loro e quindi considerando un unico modello non riusciamo a distinguere bene i singoli effetti. Per questo introduciamo l'età nel modello di regressione.

Age

La variabile `Age` è riconosciuta numerica anche se in realtà è categorica. Nell'analisi dobbiamo trasformarla in `factor` perché altrimenti vi è troppa influenza dell'ordinamento delle variabili categoriali.

```
Age1 = as.factor(Age)
summary(Age1)
```

```
##    1    2    3    4    5    6    7 NA's
##  710 3650 5390 5376 3236 2290 2693 8289
```

Possiamo notare che ci sono 8289 osservazioni in cui non è presente l'età, infatti nel `summary` sottostante possiamo notare che nonostante ci siano 32 mila osservazioni ci sono solo 23 mila gradi di libertà in quanto quelle con i missing values sono eliminate automaticamente.

```
mod = lm(Profit ~ Online+Age1)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + Age1)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -404.52 -162.90  -84.62   68.80 1952.10
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.802     10.485  -0.172    0.864
## Online        27.246      5.519   4.937 8.01e-07 ***
## Age12         54.425     11.401   4.774 1.82e-06 ***
## Age13        112.699     11.098  10.155  < 2e-16 ***
## Age14        133.820     11.103  12.053  < 2e-16 ***
## Age15        144.986     11.531  12.574  < 2e-16 ***
## Age16        160.844     11.965  13.443  < 2e-16 ***
## Age17        193.072     11.757  16.422  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 277.9 on 23337 degrees of freedom
##   (8289 observations deleted due to missingness)
## Multiple R-squared:  0.02494,    Adjusted R-squared:  0.02464
## F-statistic: 85.26 on 7 and 23337 DF,  p-value: < 2.2e-16
```

Ora sulle fasce d'età vi è una significatività sia statistica che di business, infatti le diverse età potrebbero implicare diversi redditi, in linea generale un giovane tende ad usare di più l'online banking ma ha un reddito minore e quindi fa guadagnare di meno.

Posso notare che a prescindere dall'utilizzo dell'online o dell'offline i giovani sono meno profittevoli degli anziani:

```
lapply(split(Profit,as.factor(Age)),mean)
```

```
## $`1`
## [1] 3.45493
##
## $`2`
## [1] 58.48959
##
## $`3`
## [1] 115.1122
##
## $`4`
## [1] 135.6618
##
## $`5`
## [1] 145.7596
##
## $`6`
## [1] 160.41
##
## $`7`
## [1] 192.2614
```

Inoltre il 20% dei giovani usa l'online, invece questa percentuale scende per le fasce di età successive fino ad arrivare a poco più del 3%.

```r
lapply(split(Online,as.factor(Age)),mean)
```

```
## $`1`
## [1] 0.1929577
##
## $`2`
## [1] 0.2153425
##
## $`3`
## [1] 0.154731
##
## $`4`
## [1] 0.1337426
##
## $`5`
## [1] 0.09456119
##
## $`6`
## [1] 0.05021834
##
## $`7`
## [1] 0.03639064
```

Ci potrebbe essere un bias dovuto ai dati mancanti. Infatti ci sono molte osservazioni in cui non abbiamo l'informazione sull'età. Non sappiamo perchè questi dati siano mancanti, anche se è difficile pensare che il motivo sia semplicemente collegato a delle dimenticanze o sviste in fase di racconta dei dati. Probabilmente avrebbe più senso pensare ai conti cointestati oppure al fatto che il conto venga intestato ad una società.

Proviamo ora a formulare un modello di regressione introducendo una variabile binaria che indica se abbiamo o meno a disposizione l'informazione sull'età:

```r
sum(is.na(Age))
```

```
## [1] 8289
```

```r
AgeGiven = ifelse(is.na(Age),0,1) # 0 dove c'è NA, 1 se c'è l'età
mod = lm(Profit ~ AgeGiven)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ AgeGiven)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -346.19 -150.19  -90.96   50.81 1961.04
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   72.962      2.986   24.43   <2e-16 ***
## AgeGiven      52.224      3.476   15.02   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 271.9 on 31632 degrees of freedom
## Multiple R-squared:  0.007085,   Adjusted R-squared:  0.007054
```

```
## F-statistic: 225.7 on 1 and 31632 DF,  p-value: < 2.2e-16
```

C'è una differenza statisticamente significativa sul profitto tra le osservazioni in cui è presente l'informazione sull'età e dove non c'è. La variabile `AgeGiven` infatti risulta significativa, la distribuzione tra le due popolazioni di conseguenza non risulta omogenea. Quindi, eliminando i dati dove manca l'età, stiamo distorcendo l'analisi, perchè queste medie sono "sporcate". Infatti c'è una profittabilità media più alta tra chi mi ha dato l'età rispetto a chi non me l'ha data.

Rimpiazziamo i valori mancanti con il valore nullo

Possiamo provare diversi metodi per cercare di recuperare quelle osservazioni dove l'età non è presente, in modo da includerle comunque nell'analisi. Se avessimo delle righe complete ed alcune con un solo campo mancante, potremmo costruire un modello di regressione in modo da prevedere quel valore. Una possibile alternativa, più immediata, è quella di sostituire i valori mancanti con i valori nulli. Questa strada sembra abbastanza convincente, quando l'età è una variabile categorica e possiamo quindi scegliere un valore arbitrario. Quando invece la variabile è numerica non ha molto senso, perchè equivale a dire che coloro che non hanno fornito l'età, risultano neonati.

```
AgeZero = ifelse(is.na(Age),0,Age)
table(AgeZero)
```

```
## AgeZero
##    0    1    2    3    4    5    6    7
## 8289  710 3650 5390 5376 3236 2290 2693
```

Andiamo a formulare un nuovo modello di regressione, utilizzando questa nuova variabile in cui abbiamo sostituito l'età nulla a coloro per i quali questa infromazione non è disponibile.

```
mod = lm(Profit ~ Online+AgeZero)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + AgeZero)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -393.91 -147.07  -82.03   49.97 1976.97
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  57.0311     2.6014  21.923  < 2e-16 ***
## Online       13.7925     4.6487   2.967  0.00301 **
## AgeZero      17.6803     0.6697  26.402  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 269.9 on 31631 degrees of freedom
## Multiple R-squared:  0.02161,    Adjusted R-squared:  0.02155
## F-statistic: 349.3 on 2 and 31631 DF,  p-value: < 2.2e-16
```

Anche se questa variabile risulta significativa, i valori dei coefficienti sono dovuti al fatto che, in modo arbitrario, abbiamo scelto di sostituire i valori mancanti con il valore nullo.

Rimpiazziamo i valori mancanti con la media

Una possibile alternativa è quella di sostituire i valori mancanti con la media dell'età calcolata sui valori presenti.

```r
mm = mean(Age, na.rm=TRUE)
AgeAverage = ifelse(is.na(Age),mm,Age)
table(AgeAverage)
```

```
## AgeAverage
##                1                2                3                4
##              710             3650             5390             5376
## 4.04604840436924                5                6                7
##             8289             3236             2290             2693
```

```r
mod = lm(Profit ~ Online+AgeAverage)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + AgeAverage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -398.01 -144.99  -91.28   55.00 1981.04
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.911      4.776   1.028    0.304
## Online        22.005      4.699   4.683 2.84e-06 ***
## AgeAverage    25.682      1.090  23.572  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 270.5 on 31631 degrees of freedom
## Multiple R-squared:  0.01731,    Adjusted R-squared:  0.01725
## F-statistic: 278.6 on 2 and 31631 DF,  p-value: < 2.2e-16
```

Anche in questo caso, la variabile risulta significativa, ma ovviamente osserviamo che i valori dei coefficienti cambiano, ed in particolare cambia il contributo della variabile `Online`, quindi a seconda di come tappo il buco, di come scelgo di inserire i valori mancanti nel modello, otteniamo un risultato diverso. Questo ci suggerisce che probabilmente, non è questa la strada giusta per procedere.

Per tentare di evitare questo problema potremmo considerare i modelli ottenuti tenendo conto sia della variabile `Age` opportunamente modificata, sia del fatto che la variabile `Age` venga fornita oppure no. Questo ci permette di considerare una variabile categorica che indica la presenza o meno dell'informazione sull'età, quindi non importa più di tanto come vado a rimepire il buco dell'informazione mancante.

```r
mod = lm(Profit ~ Online+AgeZero+AgeGiven)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + AgeZero + AgeGiven)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -409.34 -144.14  -82.69   52.07 1967.12
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   70.926      3.000  23.643  < 2e-16 ***
## Online        19.649      4.685   4.194 2.75e-05 ***
## AgeZero       25.603      1.086  23.582  < 2e-16 ***
## AgeGiven     -51.849      5.598  -9.263  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 269.5 on 31630 degrees of freedom
## Multiple R-squared:  0.02426,    Adjusted R-squared:  0.02417
## F-statistic: 262.1 on 3 and 31630 DF,  p-value: < 2.2e-16
```

```
mod = lm(Profit ~ Online+AgeAverage+AgeGiven)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + AgeAverage + AgeGiven)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -409.34 -144.14  -82.69   52.07 1967.12
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -32.663      5.377  -6.074 1.26e-09 ***
## Online        19.649      4.685   4.194 2.75e-05 ***
## AgeAverage    25.603      1.086  23.582  < 2e-16 ***
## AgeGiven      51.740      3.448  15.006  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 269.5 on 31630 degrees of freedom
## Multiple R-squared:  0.02426,    Adjusted R-squared:  0.02417
## F-statistic: 262.1 on 3 and 31630 DF,  p-value: < 2.2e-16
```

Osserviamo che in entrambi i casi, tutte le variabili risultano significative perchè il p-value corrispondente è sufficientemente piccolo. Inoltre il valore della variabile `Online` è lo stesso del caso precedente anche se abbiamo scelto di tappare i buchi fissando dei valori arbitrari. Tuttavia il valore dell'R^2 risulta essere molto molto piccolo, quindi questo modello povero arriva a spiegare circa il 2.5% della variabilità.

Income

Per cercare un modello più valido, possiamo considerare un'altra variabile, ad esempio il reddito di ciascun cliente. Anche in questo caso osserviamo che ci sono alcuni valori NA e possiamo scegliere come includerli. Decidiamo per esempio di sostituire i valori mancnanti con quelli nulli.

```
IncomeZero = ifelse(is.na(Income),0,Income)
IncomeGiven = ifelse(is.na(Income),0,1)
mod = lm(Profit ~ Online+AgeAverage+AgeGiven+IncomeZero+IncomeGiven)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + AgeAverage + AgeGiven + IncomeZero +
##     IncomeGiven)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -459.03 -144.61  -74.61   50.17 1963.39
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -38.191      5.336  -7.158 8.38e-13 ***
## Online        11.867      4.650   2.552   0.0107 *
## AgeAverage    26.891      1.078  24.941  < 2e-16 ***
## AgeGiven      14.490      8.272   1.752   0.0799 .
## IncomeZero    18.771      0.748  25.094  < 2e-16 ***
## IncomeGiven  -63.553      9.047  -7.024 2.20e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 266.8 on 31628 degrees of freedom
## Multiple R-squared:  0.04365,    Adjusted R-squared:  0.0435
## F-statistic: 288.7 on 5 and 31628 DF,  p-value: < 2.2e-16
```

Possiamo notare, che il valore della variabile Online scende un po' (passando da circa 19$ ad 11$ e che l'R^2 è quasi raddoppiato, ma rimane comunque piccolo.

Altri modelli di regressione

Notiamo che non ci sono valori mancanti nella variabile `District`, tuttavia viene considerata numerica, quindi vorremmo trasformarla in categorica, in modo che ogni valore venga associato ad uno dei tre distretti. Quello che possiamo fare è introdurre due variabili binarie che rappresentano i distretti. Notiamo inoltre che non ci sono valori mancanti neanche nella variabile `Tenure`.

```
any(is.na(District))
```

```
## [1] FALSE
```

```
table(District)
```

```
## District
##  1100  1200  1300
##  3142 24342  4150
```

```
District1100 = ifelse(District==1100,1,0)
District1200 = ifelse(District==1200,1,0)
any(is.na(Tenure))
```

```
## [1] FALSE
```

Andando ad effettuare la regressione con tutte queste variabili, notiamo che l'R^2 (e anche l'R^2_{adj}) sta aumentando anche se sempre di poco, mentre il contributo della variabile Online adesso vale circa 13$.

```
mod = lm(Profit ~ Online+AgeAverage+AgeGiven+IncomeZero+IncomeGiven+Tenure
         +District1100+District1200)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + AgeAverage + AgeGiven + IncomeZero +
##     IncomeGiven + Tenure + District1100 + District1200)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -487.17 -141.21  -65.88   48.87 1993.27
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -44.2382     6.5180  -6.787 1.16e-11 ***
## Online        13.8233     4.6091   2.999  0.00271 **
## AgeAverage    16.6701     1.1482  14.519  < 2e-16 ***
## AgeGiven       4.3913     8.2017   0.535  0.59237
## IncomeZero    16.8530     0.7554  22.310  < 2e-16 ***
## IncomeGiven  -57.1191     8.9956  -6.350 2.19e-10 ***
## Tenure         4.7464     0.1918  24.742  < 2e-16 ***
## District1100  -7.9955     6.2582  -1.278  0.20140
## District1200  13.1986     4.4734   2.950  0.00318 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 264.2 on 31625 degrees of freedom
## Multiple R-squared:  0.06234,    Adjusted R-squared:  0.0621
## F-statistic: 262.8 on 8 and 31625 DF,  p-value: < 2.2e-16
```

Age come variabile categorica

```
AgeCat = ifelse(is.na(Age)==TRUE,0,Age)
Age1=as.factor(AgeCat)
levels(Age1)
```

```
## [1] "0" "1" "2" "3" "4" "5" "6" "7"
```

```
mod = lm(Profit ~ Online+Age1+IncomeZero+IncomeGiven+Tenure+District1100+District1200)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + Age1 + IncomeZero + IncomeGiven +
##     Tenure + District1100 + District1200)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -498.67 -141.61  -65.93   48.75 1993.55
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   22.8351     5.0941   4.483 7.40e-06 ***
## Online        13.8916     4.6093   3.014 0.002582 **
## Age11        -63.1264    12.3213  -5.123 3.02e-07 ***
## Age12        -34.6976     9.0630  -3.829 0.000129 ***
## Age13          1.5890     8.8141   0.180 0.856937
## Age14          4.4084     8.9231   0.494 0.621280
## Age15          7.3930     9.4162   0.785 0.432382
## Age16         26.5698     9.8813   2.689 0.007173 **
## Age17         64.8400     9.6484   6.720 1.84e-11 ***
## IncomeZero    16.7425     0.7762  21.570  < 2e-16 ***
## IncomeGiven  -57.4804     9.0048  -6.383 1.76e-10 ***
## Tenure         4.7925     0.1920  24.965  < 2e-16 ***
```

```
## District1100  -7.9082     6.2555  -1.264 0.206171
## District1200  13.2550     4.4722   2.964 0.003040 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 264.1 on 31620 degrees of freedom
## Multiple R-squared:  0.06342,    Adjusted R-squared:  0.06304
## F-statistic: 164.7 on 13 and 31620 DF,  p-value: < 2.2e-16
```

Anche in questo caso l'R^2 continua a rimanere molto piccolo e molti livelli di `Age` non risultano neppure significativi.

Income come variabile categorica

```
IncomeCat = ifelse(is.na(Income)==TRUE,0,Income)
Income1=as.factor(IncomeCat)
levels(Income1)
```

```
##  [1] "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
```

```
mod = lm(Profit ~ Online+Age1+Income1+Tenure+District1100+District1200)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + Age1 + Income1 + Tenure + District1100 +
##     District1200)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -533.29 -138.64  -65.41   49.37 1994.05
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   24.5467     5.0875   4.825 1.41e-06 ***
## Online        13.4983     4.5997   2.935 0.003342 **
## Age11        -63.5675    12.2956  -5.170 2.36e-07 ***
## Age12        -33.0700     9.0467  -3.655 0.000257 ***
## Age13          4.0186     8.8025   0.457 0.648010
## Age14          6.6551     8.9113   0.747 0.455185
## Age15          9.9292     9.4047   1.056 0.291081
## Age16         28.7596     9.8759   2.912 0.003593 **
## Age17         63.4023     9.6342   6.581 4.75e-11 ***
## Income11     -12.3684     9.7080  -1.274 0.202657
## Income12     -11.0678    12.2163  -0.906 0.364952
## Income13      -2.4894     9.4772  -0.263 0.792803
## Income14      -3.1228     9.7751  -0.319 0.749376
## Income15       2.6210     9.7503   0.269 0.788079
## Income16      26.8388     8.8454   3.034 0.002414 **
## Income17      46.8962     9.3559   5.012 5.40e-07 ***
## Income18      64.2256    10.2531   6.264 3.80e-10 ***
## Income19     132.7718     9.5773  13.863  < 2e-16 ***
## Tenure         4.7917     0.1916  25.009  < 2e-16 ***
## District1100  -8.6630     6.2490  -1.386 0.165665
## District1200  11.0437     4.4736   2.469 0.013569 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 263.5 on 31613 degrees of freedom
## Multiple R-squared:  0.06777,    Adjusted R-squared:  0.06718
## F-statistic: 114.9 on 20 and 31613 DF,  p-value: < 2.2e-16
```

Anche in questo caso l'R^2 continua a rimanere molto piccolo e molti livelli di `Age` e di `Income` non risultano neppure significativi. Un ulteriore miglioramento a questi modelli, potrebbe essere fatto andando ad aggiungere le interazioni tra le variabili. Tuttavia anche se l'R^2 migliora leggeremente, comunque non otteniamo dei risultati sufficientemente buoni per la previsione, come possiamo osservare nei due modelli seguenti:

```
mod = lm(Profit ~ Online+Age1+Income1+Tenure+District1100+District1200+Income1:Age1)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + Age1 + Income1 + Tenure + District1100 +
##     District1200 + Income1:Age1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -540.65 -138.83  -64.06   48.86 1992.37
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     26.3529     5.1203   5.147 2.67e-07 ***
## Online          13.1986     4.6004   2.869  0.00412 **
## Age11          -70.2085    28.5554  -2.459  0.01395 *
## Age12          -22.7249    22.9500  -0.990  0.32209
## Age13          -20.7780    23.5585  -0.882  0.37780
## Age14          -52.0245    32.5462  -1.598  0.10995
## Age15          -26.2834    43.3843  -0.606  0.54463
## Age16           -7.0280    45.9301  -0.153  0.87839
## Age17           20.0038    37.3765   0.535  0.59252
## Income11       -17.9786    23.5622  -0.763  0.44545
## Income12        -0.2011    48.1588  -0.004  0.99667
## Income13       -24.7235    27.4643  -0.900  0.36802
## Income14       -49.6047    38.5237  -1.288  0.19788
## Income15       -22.1993    41.7337  -0.532  0.59478
## Income16        33.7687    25.9885   1.299  0.19383
## Income17        10.5973    33.5687   0.316  0.75224
## Income18        10.5450    41.7433   0.253  0.80057
## Income19        43.4069    62.1257   0.699  0.48475
## Tenure           4.7471     0.1918  24.744  < 2e-16 ***
## District1100    -8.9367     6.2512  -1.430  0.15284
## District1200    10.9538     4.4747   2.448  0.01437 *
## Age11:Income11  25.6809    43.1608   0.595  0.55184
## Age12:Income11  -8.6396    35.3990  -0.244  0.80718
## Age13:Income11   3.4246    36.2742   0.094  0.92478
## Age14:Income11  68.9695    43.2982   1.593  0.11119
## Age15:Income11  26.1505    52.8877   0.494  0.62099
## Age16:Income11  23.6922    54.8945   0.432  0.66604
## Age17:Income11  75.8855    45.8312   1.656  0.09778 .
## Age11:Income12  20.2853    69.3945   0.292  0.77005
```

```
## Age12:Income12 -12.9816    58.3965  -0.222  0.82408
## Age13:Income12  -1.7301    58.7608  -0.029  0.97651
## Age14:Income12  41.1089    63.7963   0.644  0.51934
## Age15:Income12  27.2786    72.1030   0.378  0.70519
## Age16:Income12  35.0006    70.8021   0.494  0.62107
## Age17:Income12  22.8008    63.4736   0.359  0.71943
## Age11:Income13  23.1785    46.5075   0.498  0.61822
## Age12:Income13   9.4634    37.5281   0.252  0.80091
## Age13:Income13  23.3195    38.0466   0.613  0.53993
## Age14:Income13  96.6927    44.6963   2.163  0.03052 *
## Age15:Income13  33.8900    53.7738   0.630  0.52855
## Age16:Income13  53.9803    55.4002   0.974  0.32988
## Age17:Income13  97.9617    47.9505   2.043  0.04106 *
## Age11:Income14  85.8800    61.3514   1.400  0.16158
## Age12:Income14  49.6616    46.9681   1.057  0.29036
## Age13:Income14  54.2563    46.5845   1.165  0.24415
## Age14:Income14  87.2232    51.8943   1.681  0.09281 .
## Age15:Income14  89.1528    59.8676   1.489  0.13645
## Age16:Income14  84.2506    61.6151   1.367  0.17152
## Age17:Income14 111.7993    55.5682   2.012  0.04424 *
## Age11:Income15  34.9132    59.2477   0.589  0.55568
## Age12:Income15  42.9600    49.9523   0.860  0.38978
## Age13:Income15  43.2659    49.1501   0.880  0.37871
## Age14:Income15  73.8506    53.9840   1.368  0.17132
## Age15:Income15  53.2150    61.6343   0.863  0.38792
## Age16:Income15  79.2875    64.0659   1.238  0.21588
## Age17:Income15  51.8705    58.7821   0.882  0.37756
## Age11:Income16 -14.2291    45.8257  -0.311  0.75618
## Age12:Income16 -23.0508    35.8160  -0.644  0.51985
## Age13:Income16  21.1635    35.6935   0.593  0.55324
## Age14:Income16  41.9374    42.1412   0.995  0.31966
## Age15:Income16  31.6642    51.3542   0.617  0.53751
## Age16:Income16  25.1818    54.1175   0.465  0.64171
## Age17:Income16  53.1012    46.9862   1.130  0.25842
## Age11:Income17  21.4190    56.3050   0.380  0.70364
## Age12:Income17  16.9834    42.1693   0.403  0.68714
## Age13:Income17  73.0823    41.9624   1.742  0.08159 .
## Age14:Income17  95.8947    47.5433   2.017  0.04370 *
## Age15:Income17  93.5148    56.1783   1.665  0.09600 .
## Age16:Income17  50.0305    59.4907   0.841  0.40037
## Age17:Income17  32.9881    53.1518   0.621  0.53484
## Age11:Income18  51.4308    70.2721   0.732  0.46425
## Age12:Income18  30.6564    50.4162   0.608  0.54315
## Age13:Income18  80.7796    49.4822   1.632  0.10258
## Age14:Income18 114.3998    54.2478   2.109  0.03497 *
## Age15:Income18 111.5045    62.2722   1.791  0.07337 .
## Age16:Income18  87.6081    66.0962   1.325  0.18503
## Age17:Income18  66.8771    60.4453   1.106  0.26856
## Age11:Income19  -5.5391    98.0605  -0.056  0.95495
## Age12:Income19  62.9328    67.6952   0.930  0.35256
## Age13:Income19 133.3694    67.0406   1.989  0.04667 *
## Age14:Income19 166.2334    70.6076   2.354  0.01856 *
## Age15:Income19 107.3262    76.6466   1.400  0.16144
## Age16:Income19 142.1290    79.2137   1.794  0.07278 .
```

```
## Age17:Income19  25.8359    74.8013   0.345  0.72980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 263.3 on 31550 degrees of freedom
## Multiple R-squared:  0.07147,    Adjusted R-squared:  0.06902
## F-statistic: 29.26 on 83 and 31550 DF,  p-value: < 2.2e-16
```

```
mod = lm(Profit ~ Online+Age1+Income1+Tenure+District1100+District1200+Age1:Online)
summary(mod)
```

```
##
## Call:
## lm(formula = Profit ~ Online + Age1 + Income1 + Tenure + District1100 +
##     District1200 + Age1:Online)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -533.76 -138.52  -65.42   49.40 1992.56
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   25.9686     5.1607   5.032 4.88e-07 ***
## Online        -0.5564     9.5038  -0.059 0.953318
## Age11        -58.3584    13.2368  -4.409 1.04e-05 ***
## Age12        -33.1107     9.3462  -3.543 0.000397 ***
## Age13          1.0602     8.9812   0.118 0.906031
## Age14          4.2486     9.0715   0.468 0.639537
## Age15          4.8174     9.5711   0.503 0.614738
## Age16         26.9998    10.0063   2.698 0.006974 **
## Age17         62.5795     9.7272   6.433 1.27e-10 ***
## Income11     -12.3110     9.7078  -1.268 0.204749
## Income12     -11.0145    12.2169  -0.902 0.367290
## Income13      -2.2279     9.4778  -0.235 0.814158
## Income14      -2.9549     9.7759  -0.302 0.762454
## Income15       3.0427     9.7519   0.312 0.755033
## Income16      27.0293     8.8459   3.056 0.002248 **
## Income17      46.9508     9.3562   5.018 5.25e-07 ***
## Income18      64.1878    10.2537   6.260 3.90e-10 ***
## Income19     132.6767     9.5790  13.851  < 2e-16 ***
## Tenure         4.7888     0.1918  24.973  < 2e-16 ***
## District1100  -8.6826     6.2487  -1.389 0.164691
## District1200  11.1116     4.4738   2.484 0.013007 *
## Online:Age11 -21.2242    26.8073  -0.792 0.428523
## Online:Age12   6.8667    14.2488   0.482 0.629872
## Online:Age13  23.0077    13.7444   1.674 0.094147 .
## Online:Age14  20.4003    14.2079   1.436 0.151058
## Online:Age15  51.6058    18.4645   2.795 0.005195 **
## Online:Age16  17.7843    26.9519   0.660 0.509352
## Online:Age17  -6.1621    28.7507  -0.214 0.830293
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 263.5 on 31606 degrees of freedom
## Multiple R-squared:  0.06812,    Adjusted R-squared:  0.06732
```

```
## F-statistic: 85.57 on 27 and 31606 DF,  p-value: < 2.2e-16
```

In conclusione, i modelli trovati sono poco utili nella previsione del profitto perchè non riescono bene a spiegare la varibilità presente.

Profitto nel tempo

Possiamo provare a migliorare la situazione, utilizzando la storia della banca, e quindi anche le informazioni passate dei clienti.

```
#Rinominiamo le variabili
Profit9=X9Profit
Online9=X9Online
Profit0=X0Profit
Online0=X0Online
Income=X9Inc
Age=X9Age
```

Modello base

Proviamo a prevedere il profitto di ciascun cliente nel 2000, andando ad utilizzare l'informazione binaria riguardo all'uso dell'online banking. L'R^2 è troppo basso, quindi vorremmo migliorare questo modello.

```
mod1 = lm(Profit0 ~ Online9)
summary(mod1)
```

```
##
## Call:
## lm(formula = Profit0 ~ Online9)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5784.9  -172.9  -120.9    62.1 26944.1
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  141.900      2.565  55.319  < 2e-16 ***
## Online9       23.490      7.267   3.232  0.00123 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 389.9 on 26394 degrees of freedom
##   (5238 observations deleted due to missingness)
## Multiple R-squared:  0.0003957,  Adjusted R-squared:  0.0003578
## F-statistic: 10.45 on 1 and 26394 DF,  p-value: 0.001229
```

Per provare a far aumentare il valore dell'R^2, possiamo utilizzare non solo l'informazione della variabile `Online`, ma anche le caratteristiche anagrafiche a nostra disposizione relative al singolo cliente, quelle che abbiamo utilizzato nei modelli precedenti.

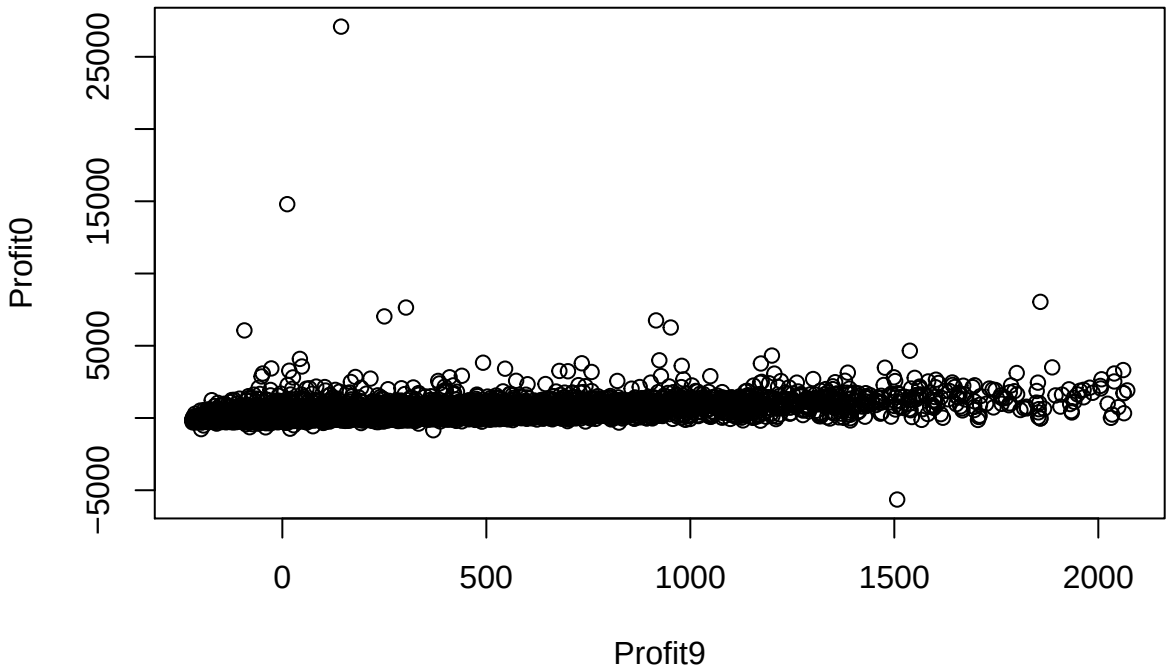
```
mod2 = lm(Profit0 ~ Online9+AgeZero+AgeGiven+IncomeZero+IncomeGiven+Tenure
          +District1100+District1200)
summary(mod2)
```

```
##
## Call:
```

```
## lm(formula = Profit0 ~ Online9 + AgeZero + AgeGiven + IncomeZero +
##     IncomeGiven + Tenure + District1100 + District1200)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5933.2  -168.7   -85.0    56.8 26797.3
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   58.7253     8.8328   6.649 3.02e-11 ***
## Online9       28.5864     7.2548   3.940 8.16e-05 ***
## AgeZero       13.4625     1.7582   7.657 1.97e-14 ***
## AgeGiven     -57.8432    14.3949  -4.018 5.88e-05 ***
## IncomeZero    21.5758     1.1483  18.789  < 2e-16 ***
## IncomeGiven  -85.7359    14.0981  -6.081 1.21e-09 ***
## Tenure         4.7547     0.3007  15.809  < 2e-16 ***
## District1100 -13.5127    10.0338  -1.347    0.178
## District1200  10.9915     7.1485   1.538    0.124
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 383.3 on 26387 degrees of freedom
##   (5238 observations deleted due to missingness)
## Multiple R-squared:  0.03419,    Adjusted R-squared:  0.0339
## F-statistic: 116.8 on 8 and 26387 DF,  p-value: < 2.2e-16
```

Questo migliora abbastanza le prestazioni, perchè così riusciamo a spiegare molta più della variabilità presente. Inoltre il modello potrebbe essere ulteriormente migliorato, infatti abbiamo a disposizione un'altra informazione sui singoli clienti: il profitto nel 1999.

```
plot(Profit9, Profit0)
```



Inserendo questa informazione nel modello è come se dicessimo che il profitto nell'anno successivo dipende dal profitto dell'anno in corso e dalle caratteristiche del singolo cliente. Quindi in linea di principio, coloro che

l'anno precdente mi hanno fatto guadagnare, continueranno a farmi guadagnare anche nell'anno successivo.

```
mod3 = lm(Profit0 ~ Profit9+Online9+AgeZero+AgeGiven+IncomeZero+IncomeGiven+Tenure
          +District1100+District1200)
summary(mod3)
```

```
##
## Call:
## lm(formula = Profit0 ~ Profit9 + Online9 + AgeZero + AgeGiven +
##     IncomeZero + IncomeGiven + Tenure + District1100 + District1200)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6948.0   -72.6   -33.2    28.6 26901.1
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   35.070248   7.185594   4.881 1.06e-06 ***
## Profit9        0.825353   0.007099 116.264  < 2e-16 ***
## Online9       15.063749   5.900660   2.553 0.010689 *
## AgeZero       -0.783422   1.434981  -0.546 0.585108
## AgeGiven      -2.298837  11.715494  -0.196 0.844438
## IncomeZero     7.123508   0.942052   7.562 4.11e-14 ***
## IncomeGiven  -32.355443  11.473583  -2.820 0.004806 **
## Tenure         0.922225   0.246777   3.737 0.000187 ***
## District1100  -8.012553   8.159471  -0.982 0.326112
## District1200  -1.517990   5.814085  -0.261 0.794026
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 311.7 on 26386 degrees of freedom
##   (5238 observations deleted due to missingness)
## Multiple R-squared:  0.3614, Adjusted R-squared:  0.3611
## F-statistic:  1659 on 9 and 26386 DF,  p-value: < 2.2e-16
```

Adesso l'R^2 risulta notevolmente migliorato: riusciamo a spiegare oltre il 36% della variabilità.

Possiamo anche supporre che alcuni dei dati anagrafici non siano rilevanti, ma magari non li abbiamo utilizzati nel modo giusto all'interno del modello. Infatti provando a tenere la variabile del profitto precedente come regressore, ma eliminando le variabili `Age` e `Income`, otteniamo un modello che spiega all'incirca la stessa variabilità.

```
mod4 = lm(Profit0 ~ Profit9+Online9+Tenure+District1100+District1200)
summary(mod4)
```

```
##
## Call:
## lm(formula = Profit0 ~ Profit9 + Online9 + Tenure + District1100 +
##     District1200)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6965.3   -70.3   -35.7    27.6 26908.6
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   31.023428   5.932798   5.229 1.72e-07 ***
## Profit9        0.831875   0.007017 118.547  < 2e-16 ***
## Online9       18.774205   5.841623   3.214  0.00131 **
## Tenure         0.919726   0.228984   4.017 5.92e-05 ***
## District1100 -11.205954   8.153961  -1.374  0.16936
## District1200   3.889211   5.776169   0.673  0.50075
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 312 on 26390 degrees of freedom
##   (5238 observations deleted due to missingness)
## Multiple R-squared:  0.3599, Adjusted R-squared:  0.3598
## F-statistic:  2968 on 5 and 26390 DF,  p-value: < 2.2e-16
```

Quello che possiamo osservare è come le due variabili legate al profitto del singolo cliente nei diversi anni, siano correlate tra loro. Osserviamo infatti che c'è una correlazione di circa il 60%. Sono presenti inoltre solamente 8 osservazioni in cui il profitto nel 2000 risulta essere maggiore di 5000 e una osservazione con profitto nel 2000 minore di -5000, quindi possiamo pensare che quelle osservazioni siano degli outliers.

```
cor(Profit9,Profit0,use="complete.obs")
```

```
## [1] 0.5993369
```

```
sum(Profit0>5000,na.rm=TRUE)
```

```
## [1] 8
```

```
sum(Profit0<(-5000),na.rm=TRUE)
```

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

Quello che possiamo fare è andare ad osservare il valore dell'R^2 nel caso in cui questi valori vengano eliminati, per capire se il modello risulta migliore.

```
c=which(abs(Profit0)>5000)
detach(Pilgrim)
data=Pilgrim[-c,]
attach(data)
Profit9=X9Profit
Online9=X9Online
Age=X9Age
Income=X9Inc
Tenure=X9Tenure
District=X9District
Profit0=X0Profit
Online0=X0Online
District1100 = ifelse(District==1100,1,0)
District1200 = ifelse(District==1200,1,0)
AgeGiven = ifelse(is.na(Age),0,1)
AgeZero = ifelse(is.na(Age),0,Age)
IncomeZero = ifelse(is.na(Income),0,Income)
IncomeGiven = ifelse(is.na(Income),0,1)
mod3 = lm(Profit0 ~ Profit9+Online9+AgeZero+AgeGiven+IncomeZero+IncomeGiven+Tenure
          +District1100+District1200)
summary(mod3)
```

```
##
## Call:
```

```
## lm(formula = Profit0 ~ Profit9 + Online9 + AgeZero + AgeGiven +
##     IncomeZero + IncomeGiven + Tenure + District1100 + District1200)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1711.1   -69.9   -31.1    30.8  4010.9
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   36.234524   5.178529   6.997 2.68e-12 ***
## Profit9        0.821781   0.005123 160.402  < 2e-16 ***
## Online9       18.161148   4.252329   4.271 1.95e-05 ***
## AgeZero       -1.760224   1.034234  -1.702 0.088775 .
## AgeGiven       2.183430   8.442798   0.259 0.795935
## IncomeZero     6.252322   0.678954   9.209  < 2e-16 ***
## IncomeGiven  -27.438630   8.268377  -3.319 0.000906 ***
## Tenure         0.917593   0.177904   5.158 2.52e-07 ***
## District1100 -13.050532   5.881041  -2.219 0.026489 *
## District1200  -6.504494   4.190384  -1.552 0.120616
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 224.6 on 26377 degrees of freedom
##   (5238 observations deleted due to missingness)
## Multiple R-squared:  0.5174, Adjusted R-squared:  0.5173
## F-statistic:  3143 on 9 and 26377 DF,  p-value: < 2.2e-16
```

```
mod4 = lm(Profit0 ~ Profit9+Online9+Tenure+District1100+District1200)
summary(mod4)
```

```
##
## Call:
## lm(formula = Profit0 ~ Profit9 + Online9 + Tenure + District1100 +
##     District1200)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1712.1   -67.9   -33.5    30.1  4009.2
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   33.956128   4.278295   7.937 2.16e-15 ***
## Profit9        0.827164   0.005068 163.228  < 2e-16 ***
## Online9       22.032087   4.212378   5.230 1.71e-07 ***
## Tenure         0.852151   0.165189   5.159 2.51e-07 ***
## District1100 -16.081030   5.880673  -2.735  0.00625 **
## District1200  -1.713170   4.165725  -0.411  0.68089
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 225 on 26381 degrees of freedom
##   (5238 observations deleted due to missingness)
## Multiple R-squared:  0.5158, Adjusted R-squared:  0.5157
## F-statistic:  5620 on 5 and 26381 DF,  p-value: < 2.2e-16
```

Quindi andando a ripetere il codice precedente, quando però eliminiamo gli outliers, miglioriamo l'R^2 che negli ultimi due modelli raggiunge oltre il 50%. Per questo motivo nelle successive analisi considereremo il dataset senza outliers.

Inoltre, come nel caso precedente, nel momento in cui andiamo ad introdurre delle possibili interazioni tra le variabili, come nei successivi modelli, logicamente l'R^2 migliora, ma non significativamente. Quindi piuttosto conviene spostare la nostra attenzione su un altro tipo di analisi.

```
mod5 = lm(Profit0 ~ Profit9+Online9+Tenure+District1100+District1200+Profit9:Online9)
summary(mod5)
```

```
##
## Call:
## lm(formula = Profit0 ~ Profit9 + Online9 + Tenure + District1100 +
##     District1200 + Profit9:Online9)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1719.9   -68.2   -33.3    30.0  4009.5
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      33.518741   4.283390   7.825 5.25e-15 ***
## Profit9           0.831216   0.005441 152.757  < 2e-16 ***
## Online9          25.735425   4.585527   5.612 2.02e-08 ***
## Tenure            0.844632   0.165220   5.112 3.21e-07 ***
## District1100    -16.032445   5.880368  -2.726  0.00641 **
## District1200     -1.668952   4.165530  -0.401  0.68868
## Profit9:Online9  -0.029766   0.014568  -2.043  0.04104 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 225 on 26380 degrees of freedom
##   (5238 observations deleted due to missingness)
## Multiple R-squared:  0.5158, Adjusted R-squared:  0.5157
## F-statistic:  4684 on 6 and 26380 DF,  p-value: < 2.2e-16
```

```
mod6 = lm(Profit0 ~ Profit9+Online9+AgeZero+AgeGiven+IncomeZero+IncomeGiven+Tenure+District1100+Distric
summary(mod6)
```

```
##
## Call:
## lm(formula = Profit0 ~ Profit9 + Online9 + AgeZero + AgeGiven +
##     IncomeZero + IncomeGiven + Tenure + District1100 + District1200 +
##     IncomeZero:AgeZero)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1710.7   -70.0   -31.2    30.9  4011.4
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         35.694127   5.183451   6.886 5.86e-12 ***
## Profit9              0.821679   0.005123 160.389  < 2e-16 ***
## Online9             18.001066   4.252555   4.233 2.31e-05 ***
## AgeZero              1.906425   1.901991   1.002   0.3162
```

```
## AgeGiven            -4.309090   8.902726  -0.484   0.6284
## IncomeZero           9.362177   1.514559   6.181 6.44e-10 ***
## IncomeGiven        -35.609936   9.000554  -3.956 7.63e-05 ***
## Tenure               0.923588   0.177909   5.191 2.10e-07 ***
## District1100       -13.109145   5.880620  -2.229   0.0258 *
## District1200        -6.530159   4.190059  -1.558   0.1191
## AgeZero:IncomeZero  -0.766873   0.333859  -2.297   0.0216 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 224.6 on 26376 degrees of freedom
##   (5238 observations deleted due to missingness)
## Multiple R-squared:  0.5175, Adjusted R-squared:  0.5174
## F-statistic:  2829 on 10 and 26376 DF,  p-value: < 2.2e-16
```

Retain

Quello che possiamo fare, è considerare una variabile aggiuntiva `Retain` che ci dice se il cliente rimane nella banca anche nell'anno successivo. A questo punto vorremmo capire come prevedere al meglio se un cliente rimane con la banca nell'anno successivo oppure no, questo potrebbe dipendere da tutte le caratteristiche del cliente.

Si considera che il cliente rimane nella banca se è presente il valore di `Profit0`.

```
Retain = ifelse(is.na(Profit0),0,1)
mod1 = lm(Retain ~ Online9)
summary(mod1)
```

```
##
## Call:
## lm(formula = Retain ~ Online9)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8534  0.1466  0.1683  0.1683  0.1683
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.831731   0.002230 372.904  < 2e-16 ***
## Online9     0.021668   0.006389   3.391 0.000696 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3717 on 31623 degrees of freedom
## Multiple R-squared:  0.0003636,  Adjusted R-squared:  0.000332
## F-statistic:  11.5 on 1 and 31623 DF,  p-value: 0.0006963
```

```
mod2 = lm(Retain ~ Online9+AgeZero+AgeGiven+IncomeZero+IncomeGiven+Tenure
          +District1100+District1200)
summary(mod2)
```

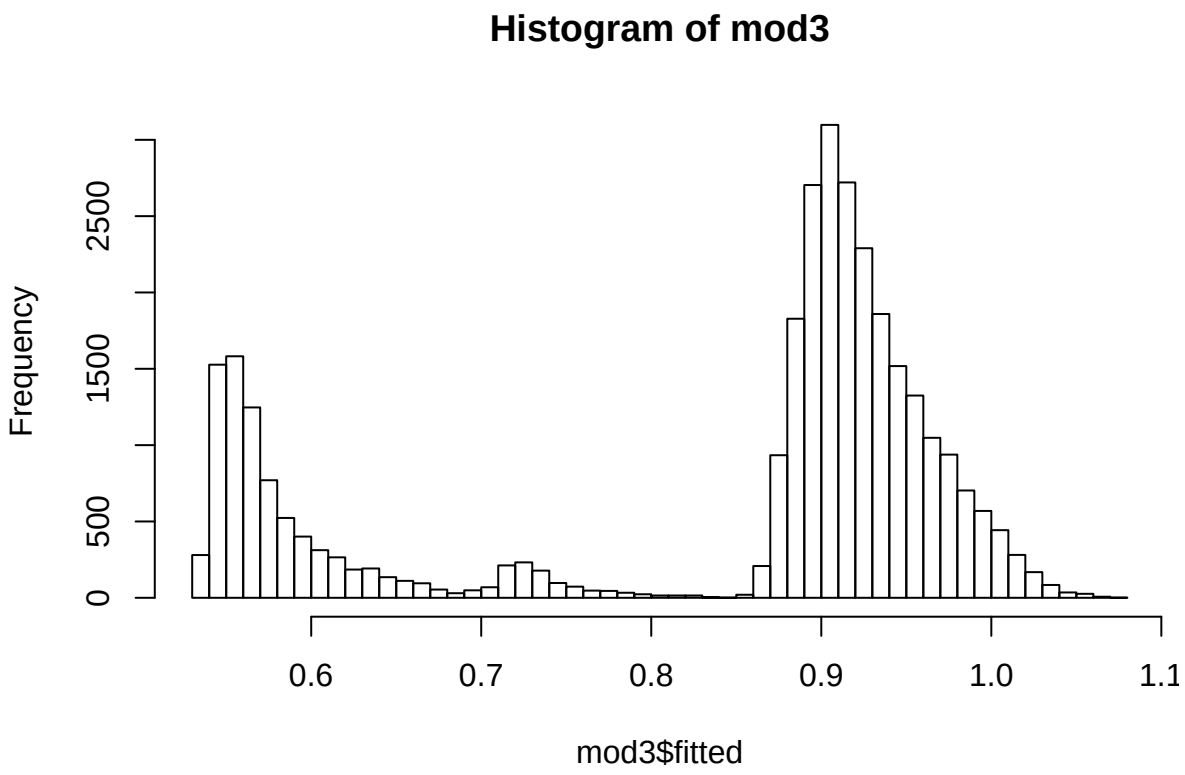
```
##
## Call:
## lm(formula = Retain ~ Online9 + AgeZero + AgeGiven + IncomeZero +
##     IncomeGiven + Tenure + District1100 + District1200)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04745  0.02557  0.07969  0.10979  0.46685
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5356312  0.0065091  82.289  < 2e-16 ***
## Online9       0.0107948  0.0058871   1.834 0.066719 .
## AgeZero       0.0014012  0.0014667   0.955 0.339418
## AgeGiven      0.1664643  0.0117311  14.190  < 2e-16 ***
## IncomeZero    0.0035132  0.0009650   3.641 0.000272 ***
## IncomeGiven   0.1501841  0.0114900  13.071  < 2e-16 ***
## Tenure        0.0039240  0.0002451  16.009  < 2e-16 ***
## District1100 -0.0031135  0.0079946  -0.389 0.696949
## District1200  0.0074261  0.0057144   1.300 0.193766
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3375 on 31616 degrees of freedom
## Multiple R-squared:  0.176,  Adjusted R-squared:  0.1758
## F-statistic: 844.3 on 8 and 31616 DF,  p-value: < 2.2e-16
```

```
mod3 = lm(Retain ~ Profit9+Online9+AgeZero+AgeGiven+IncomeZero+IncomeGiven+Tenure
          +District1100+District1200)
summary(mod3)
```

```
##
## Call:
## lm(formula = Retain ~ Profit9 + Online9 + AgeZero + AgeGiven +
##     IncomeZero + IncomeGiven + Tenure + District1100 + District1200)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.05798  0.02531  0.07933  0.11024  0.46756
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.352e-01  6.511e-03  82.203  < 2e-16 ***
## Profit9       1.895e-05  7.191e-06   2.635  0.00842 **
## Online9       1.053e-02  5.887e-03   1.789  0.07369 .
## AgeZero       1.085e-03  1.471e-03   0.738  0.46073
## AgeGiven      1.677e-01  1.174e-02  14.283  < 2e-16 ***
## IncomeZero    3.195e-03  9.724e-04   3.285  0.00102 **
## IncomeGiven   1.513e-01  1.150e-02  13.158  < 2e-16 ***
## Tenure        3.834e-03  2.474e-04  15.496  < 2e-16 ***
## District1100 -2.966e-03  7.994e-03  -0.371  0.71061
## District1200  7.172e-03  5.715e-03   1.255  0.20947
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3375 on 31615 degrees of freedom
## Multiple R-squared:  0.1762, Adjusted R-squared:  0.176
## F-statistic: 751.4 on 9 and 31615 DF,  p-value: < 2.2e-16
```

```r
hist(mod3$fitted,nclass=50, main ="Histogram of mod3")
```



Andare a considerare il valore dell'R^2 per valtutare la bontà del modello non è molto opportuno perchè stiamo prevedendo se il cliente rimane oppure no, quindi si tratta di una variabile binaria. Dall'istogramma possiamo facilmente dedurre che ci sono due gruppi abbastanza distinti, quindi quello che possiamo fare è fissare una certa soglia al di sopra della quale possiamo concludere che il cliente è sicuro, mentre al di sotto si trovano quei clienti che probabilmente il prossimo anno cambieranno la banca. Infatti ci sono due mode molto alte e sufficientemente lontane, mentre tra i valori di 0.7 e 0.8, troviamo una "zona grigia", dove la frequenza è molto bassa e i clienti sono abbastanza incerti. Inoltre saremmo interessati ad interpretare i valori dell'istogramma come delle probabilità, anche se ci sono dei valori più grandi di 1. Per questa ragione, è preferibile usare i metodi di classificazione che sono adatti a valori di risposta qualitativi.

Analisi di Retain con Regressione logistica

Proviamo a prevedere il comportamento futuro di un cliente, utilizzando la regressione logistica.

```r
glm.out = glm(Retain ~ Profit9+Online9+AgeZero+AgeGiven+IncomeZero+
  IncomeGiven+Tenure+District1100+District1200,family=binomial(logit))
summary(glm.out)
```

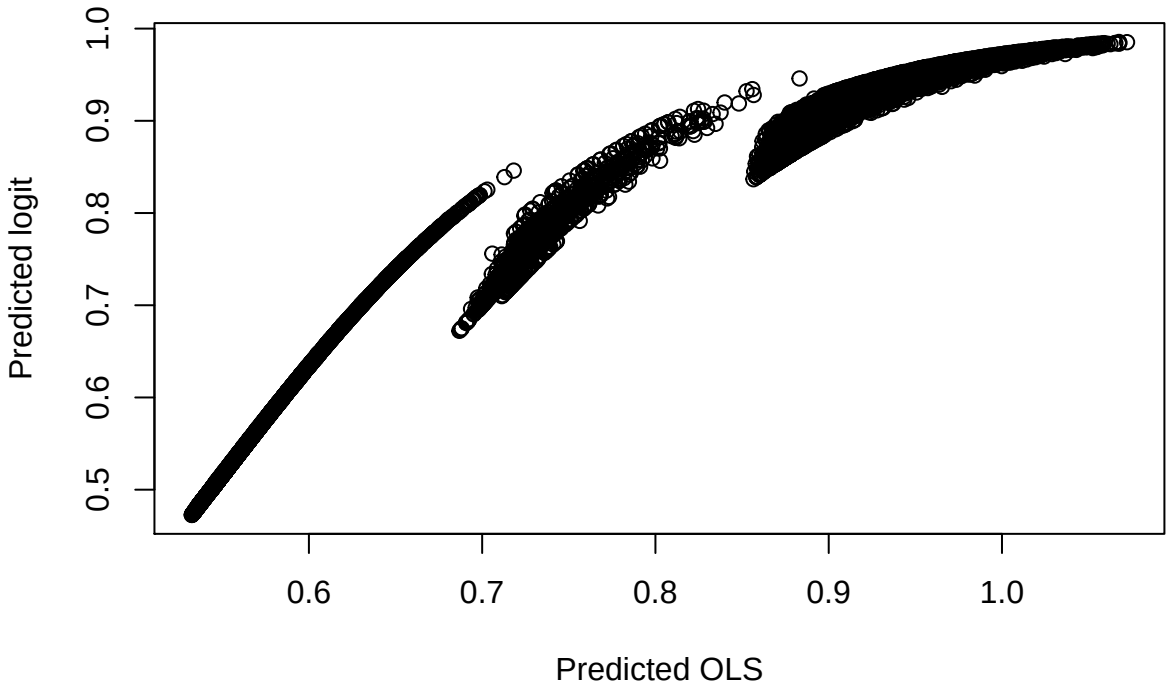
```
##
## Call:
## glm(formula = Retain ~ Profit9 + Online9 + AgeZero + AgeGiven +
##     IncomeZero + IncomeGiven + Tenure + District1100 + District1200,
##     family = binomial(logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8377   0.2917   0.3884   0.4667   1.2243
##
## Coefficients:
```

```
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -7.981e-02  5.145e-02  -1.551   0.1209
## Profit9       1.832e-04  7.182e-05   2.550   0.0108 *
## Online9       1.058e-01  5.333e-02   1.984   0.0472 *
## AgeZero       6.841e-02  1.596e-02   4.287 1.81e-05 ***
## AgeGiven      8.372e-01  9.307e-02   8.996  < 2e-16 ***
## IncomeZero    5.320e-02  1.058e-02   5.030 4.89e-07 ***
## IncomeGiven   7.752e-01  8.999e-02   8.615  < 2e-16 ***
## Tenure        3.748e-02  2.413e-03  15.530  < 2e-16 ***
## District1100 -3.193e-02  6.798e-02  -0.470   0.6386
## District1200  6.678e-02  4.915e-02   1.359   0.1742
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28392  on 31624  degrees of freedom
## Residual deviance: 23297  on 31615  degrees of freedom
## AIC: 23317
##
## Number of Fisher Scoring iterations: 5
```

```
hist(glm.out$fitted.values,nclass=50, main="Histogram of Logistic Regression")
```

Questo metodo risulta molto veloce,in quanto si ferma dopo solo 5 iterazioni. Inoltre le diagnostiche del modello sono un po' diverse, notiamo che non abbiamo più a disposizione l'R^2, ma non cambia il senso generale, come osserviamo dall'istogramma.

```
plot(mod3$fitted, glm.out$fitted, xlab="Predicted OLS", ylab="Predicted logit")
```



I dati utilizzati per costruire questo modello non sono bilanciati e quindi questo non è ottimale. Per fare una previsione più accurata possiamo procedere con il bilanciamento del dataset.

Bilanciamento dataset

Andando ad osservare la distribuzione della variabile Retain, notiamo che il dataset risulta molto sbilanciato, perchè solamente 5238 clienti decidono di lasciare la banca nell'anno successivo.

```
Retain = as.factor(Retain)
summary(Retain)
```

```
##     0     1
##  5238 26387
```

Per questo motivo è utile andare a bilanciare i dati mediante un oversampling:

```
library(ROSE)
dim_data_balanced = max(sum(Retain==1),sum(Retain==0))*2
data_balanced_over <- ovun.sample(as.factor(Retain) ~ Profit9+Online9+AgeZero+AgeGiven
                                  +IncomeZero+IncomeGiven+Tenure+District1100
                                  +District1200, method = "over",
                                  N = dim_data_balanced)$data
```

Per analizzare meglio i vari algoritmi di classificazione dividiamo in due parti il dataset bilanciato, una parte per il training e una parte per la validazione del modello. In questo modo riusciamo anche a testare il modello su un insieme esterno rispetto a quello con cui abbiamo formulato il modello. Scegliamo arbitrariamente di utilizzare il 70% dei dati per il training set e la restante parte per il test set.

```
set.seed(1)
train = sample(1:nrow(data_balanced_over), nrow(data_balanced_over)*0.7)
df.train = data_balanced_over[train,]
df.test  = data_balanced_over[-train,]
target.train = df.train$Retain
target.test  = df.test$Retain
```

Modelli di classificazione con dataset bilanciato

Per capire qual è il modello migliore per prevedere il comportamento futuro di un cliente, quindi per capire se il cliente rimarrà anche il prossimo anno oppure verrà perso, possiamo analizzare la confusion matrix e le misure ad essa collegate. Nel caso specifico, la variabile target è binaria (vogliamo prevedere se il cliente rimane nella banca anche il prossimo anno oppure no), quindi sono possibili solamente due tipi di errori: coloro che in realtà rimangono fedeli, ma utilizzando il modello, prevediamo che cambieranno, e questi possiamo definirli come "falsi negativi" (FN), o viceversa coloro che vengono predetti come clienti fedeli e poi l'anno successivo non saranno più clienti della nostra banca e questi sono i "falsi positivi" (FP). L'accuratezza di un classificatore si definisce come il rapporto tra le osservazioni predette correttamente e tutte le osservazioni, vorrei che questo valore fosse il più possibile vicino ad 1:

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP}$$

Una politica più scupolosa da parte della banca porterebbe a far pesare di più l'errore sui falsi positivi. Infatti nel caso in cui si trattasse di un cliente che mi fa guadagnare molto e prevedo che rimanga con me anche l'anno prossimo, anche se nella realtà non sarà così, non faccio nessuna azione. Invece vorrei riuscire a prevedere che probabilmente quel cliente è incerto, in modo da attuare delle strategie di marketing per convicerlo a rivalutare la sua decisione. In questo caso piuttosto che guardare al valore dell'accuratezza siamo più interessati ad altre misure:

- La PRECISIONE che rappresenta la percentuale di osservazioni predette positive corrette:
$$Precision = \frac{TP}{TP + FP}$$
- La RECALL che rappresenta invece la percentuale di osservazioni realmente positive che vengono classificate correttamente:
$$Precision = \frac{TP}{TP + FN}$$
- La SPECIFICITA' che rappresenta la percentuale di osservazioni negative che vengono effettivamente classificate come negative:
$$Specificity = \frac{TN}{TN + FP}$$

Decision Tree

Un possibile modello di classificazione da utilizzare, è quello dell'albero di decisione. Questo è un grafo a forma di albero appunto, dove ogni nodo interno rappresenta un test su uno degli attributi, ed ogni ramo rappresenta il risultato di questo test. Quindi il cammino dalla radice fino ad una foglia, rappresenta una regola di classificazione, infatti ogni foglia è associata ad un'etichetta di classe. Il nodo alla radice rappresenta l'intera popolazione, l'algoritmo consiste nello scegliere il miglior attributo su cui fare lo split in modo che le osservazioni con lo stesso valore di quell'attributo vadano a formare dei sottoinsiemi distinti dagli altri. Questi step vengono ripetuti finchè non troviamo un nodo foglia in ogni ramo dell'albero. Nel nostro caso, impostiamo dei parametri diversi da quelli di default, in modo che l'albero non sia nè troppo specifico, nè troppo generico. Infine validiamo il modello sull'insieme di test.

```
library(tree)
setup<-tree.control(nrow(df.train),mincut = 2, minsize = 6, mindev = 0.001)

tree.Pilgrim = tree(Retain ~ Profit9+Online9+AgeZero+AgeGiven+IncomeZero+IncomeGiven
                    +Tenure+District1100+District1200, data=df.train, control = setup)
tree.pred = predict(tree.Pilgrim, df.test, type="class", probability=TRUE)
library(caret)
cf = confusionMatrix(data = tree.pred, reference = target.test, mode = "prec_recall")
cf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    0
##          1 6606 2873
##          0 1273 5081
##
##                Accuracy : 0.7381
##                  95% CI : (0.7312, 0.745)
##     No Information Rate : 0.5024
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4768
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##               Precision : 0.6969
##                  Recall : 0.8384
##                      F1 : 0.7611
##              Prevalence : 0.4976
##          Detection Rate : 0.4172
##    Detection Prevalence : 0.5987
##       Balanced Accuracy : 0.7386
##
##        'Positive' Class : 1
##
```

Logistic Regression

In alternativa possiamo utilizzare la regressione logistica, infatti la variabile da prevedere è binaria. In questo caso, invece di una funzione lineare, si utilizza la funzione logistica in modo che l'output sia compreso tra 0 e 1:

$$f(x) = \frac{e^x}{1 + e^x}$$

Utilizzando questa funzione, siamo in grado di collegare il logaritmo della probabilità della classe di default, in particolare il logaritmo dell'odds, con l'espressione lineare del modello di regressione:

$$\log\left(\frac{p}{1-p}\right) = \beta^T \mathbf{X} + \epsilon$$

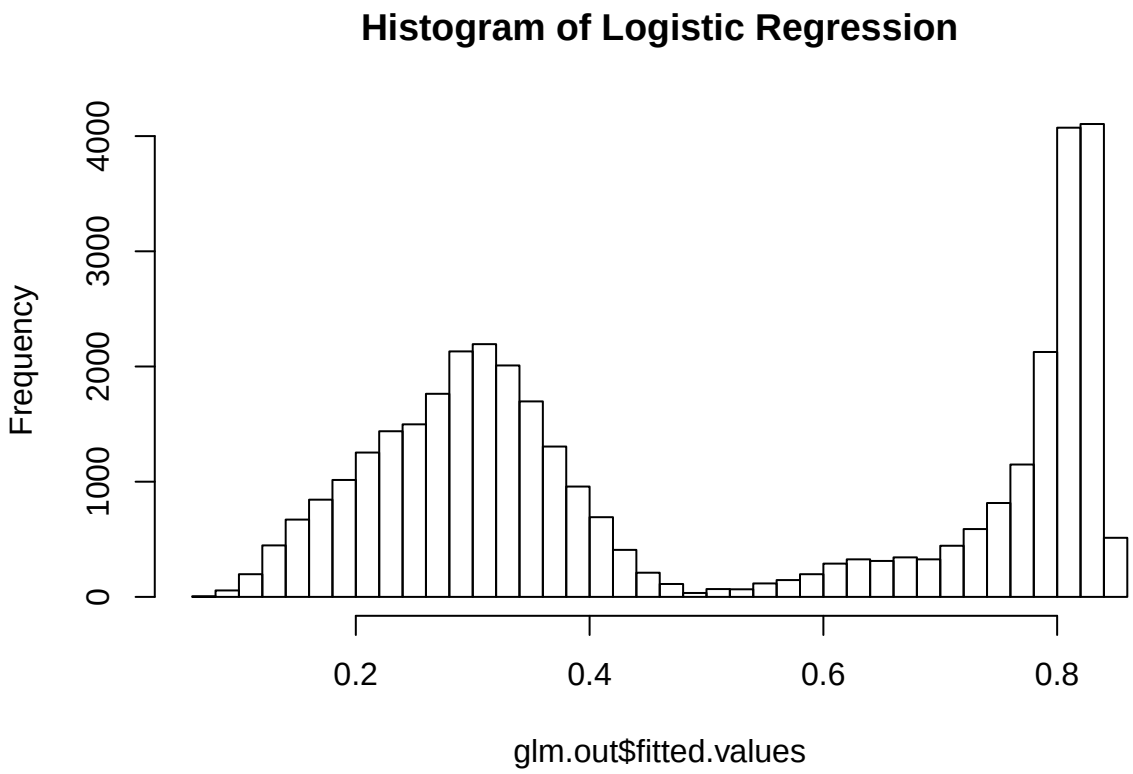
Dall'istogramma notiamo che la distribuzione dei valori predetti, rimane molto simile a quella ottenuta con il modello lineare: sempre due picchi abbastanza distanti ed una zona intermedia, dove le frequenze sono piuttosto basse.

```
glm.out = glm(Retain ~ Profit9+Online9+AgeZero+AgeGiven+IncomeZero+
  IncomeGiven+Tenure+District1100+District1200, family=binomial(logit), data=df.train)
summary(glm.out)
```

```
##
## Call:
## glm(formula = Retain ~ Profit9 + Online9 + AgeZero + AgeGiven +
##     IncomeZero + IncomeGiven + Tenure + District1100 + District1200,
##     family = binomial(logit), data = df.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.9623  -0.8565  -0.4466   0.7157   2.1846
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.646e+00  3.890e-02  42.325  < 2e-16 ***
## Profit9      -1.709e-04  4.948e-05  -3.454 0.000552 ***
## Online9      -1.456e-02  3.751e-02  -0.388 0.697850
## AgeZero      -6.681e-02  9.847e-03  -6.785 1.16e-11 ***
## AgeGiven     -8.429e-01  7.006e-02 -12.031  < 2e-16 ***
## IncomeZero   -5.580e-02  6.642e-03  -8.401  < 2e-16 ***
## IncomeGiven  -7.540e-01  6.807e-02 -11.077  < 2e-16 ***
## Tenure       -3.657e-02  1.651e-03 -22.151  < 2e-16 ***
## District1100  1.236e-01  4.957e-02   2.493 0.012663 *
## District1200 -5.624e-02  3.576e-02  -1.573 0.115817
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51211  on 36940  degrees of freedom
## Residual deviance: 41028  on 36931  degrees of freedom
## AIC: 41048
##
## Number of Fisher Scoring iterations: 4
```

```
hist(glm.out$fitted.values,nclass=50, main="Histogram of Logistic Regression")
```




```
pred=predict(glm.out, df.test, type="response")
logit.pred = rep(0, dim(df.test)[1])
logit.pred[pred > .5] = 1
```

```
logit.pred=factor(logit.pred,labels=c("1","0"))
cf = confusionMatrix(data = logit.pred, reference = target.test, mode = "prec_recall",
                     positive="1")
cf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    0
##          1 6347 2666
##          0 1532 5288
##
##                Accuracy : 0.7349
##                  95% CI : (0.7279, 0.7417)
##     No Information Rate : 0.5024
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4701
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##               Precision : 0.7042
##                  Recall : 0.8056
##                      F1 : 0.7515
##              Prevalence : 0.4976
##          Detection Rate : 0.4009
##    Detection Prevalence : 0.5693
##       Balanced Accuracy : 0.7352
##
##        'Positive' Class : 1
##
```

```
specificity=specificity(logit.pred,target.test, positive="1")
cat('Specificity:',specificity, '\n')
```

```
## Specificity: 0.6648227
```

KNN

Un ulteriore modello che possiamo utilizzare è il KNN. Questo algoritmo necessita di un parametro K da fissare e che rappresenta il numero di vicini da considerare nella fase successiva. Dopodichè, per ogni osservazione si considerano le prime K osservazioni più vicine, tra queste si valuta qual è la classe più comune ed infine questa viene assegnata all'osservazione che stiamo etichettando. Dopo aver provato diversi valori di K, in modo da scegliere il miglior valore per cercare di massimizzare l'accuratezza, formuliamo il modello.

```
set.seed(1)
train.control=trainControl(method="repeatedcv", number=3, repeats=1)
fit=train(as.factor(Retain) ~ Profit9+Online9+AgeZero+AgeGiven+IncomeZero+IncomeGiven
          +Tenure+District1100+District1200,
          method="knn",
          tuneGrid=expand.grid(k=c(4,8,12,16)),
          trControl=train.control,
          metric="Accuracy",
          data=df.train)
fit
```

```
## k-Nearest Neighbors
##
## 36941 samples
##     9 predictor
##     2 classes: '1', '0'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold, repeated 1 times)
## Summary of sample sizes: 24627, 24628, 24627
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    4  0.7439701  0.4881729
##    8  0.7311117  0.4623654
##   12  0.7295145  0.4591158
##   16  0.7294602  0.4589681
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 4.
```

```
library(class)
knn_pred <- knn(df.train,df.test, cl=target.train,k=4)
cf = confusionMatrix(data = knn_pred, reference = target.test, mode = "prec_recall")
cf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    0
##          1 5880  513
##          0 1999 7441
##
##                Accuracy : 0.8413
##                  95% CI : (0.8356, 0.847)
##     No Information Rate : 0.5024
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6824
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##               Precision : 0.9198
##                  Recall : 0.7463
##                      F1 : 0.8240
##              Prevalence : 0.4976
##          Detection Rate : 0.3714
##    Detection Prevalence : 0.4038
##       Balanced Accuracy : 0.8409
##
##        'Positive' Class : 1
##
```

```
specificity=specificity(knn_pred,target.test, positive="1")
cat('Specificity:',specificity, '\n')
```

```
## Specificity: 0.9355041
```

SVM

Utilizzando infine il metodo del Support Vector Machines, l'obiettivo è quello di trovare un iperpiano in N-1 dimensioni (dove N è il numero di variabili che utilizziamo per prevedere) che riesca a separare le diverse osservazioni a seconda della variabile target binaria di appartenenza. I punti più vicini all'iperpiano prendono il nome di vettori di supporto e quello che vogliamo è fare in modo che la loro distanza dall'iperpiano, che viene definita "margine", sia la più grande possibile in modo da ridurre l'errore di misclassificazione per le nuove osservazioni. I due iperparametri da fissare sono il costo, che rappresenta il trade-off tra l'errore di classificazione e un confine di decisione regolare, e "gamma" che invece stabilisce qual è il peso massimo che una singola osservazione nel training può raggiungere. Dopo aver provato diversi valori, scegliamo la coppia che minimizza l'errore. Per poter scegliere il valore ottimale dei parametri, estraiamo un sample dall'insieme di training, utilizzando il metodo dell'under sampling, in modo da accellare il processo.

```
library(e1071)
dim_data_balanced_under = min(sum(Retain==1),sum(Retain==0))*2
data_balanced_under <- ovun.sample(as.factor(Retain) ~ Profit9+Online9+AgeZero+AgeGiven
                                   +IncomeZero+IncomeGiven+Tenure+District1100
                                   +District1200, method = "under",N = dim_data_balanced_under)$data
set.seed(1)
train = sample(1:nrow(data_balanced_under), nrow(data_balanced_under)*0.7)
df.train=data_balanced_under[train,]
df.test = data_balanced_under[-train,]
target.train = df.train$Retain
target.under.test = df.test$Retain
g=c(-5, -3, 1)
c=c(-3,1)
svm_par=tune.svm(as.factor(Retain) ~ Profit9+Online9+AgeZero+AgeGiven+IncomeZero
                 +IncomeGiven+Tenure+District1100+District1200,
                 data=df.train, kernel="linear", gamma=10^g, cost=10^c,
                 scale=TRUE)
summary(svm_par)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  gamma cost
##  1e-05   10
##
## - best performance: 0.261832
##
## - Detailed performance results:
##   gamma  cost     error dispersion
## 1 1e-05 1e-03 0.2634692 0.01351011
## 2 1e-03 1e-03 0.2634692 0.01351011
## 3 1e+01 1e-03 0.2634692 0.01351011
## 4 1e-05 1e+01 0.2618320 0.01216201
## 5 1e-03 1e+01 0.2618320 0.01216201
## 6 1e+01 1e+01 0.2618320 0.01216201
```

A questo punto possiamo utilizzare i valori dei due parametri trovati precedentemente, utilizzando tutto l'insieme di training e valutare la nostra previsione sull'insieme test.

```
svm_model=svm(as.factor(Retain) ~ Profit9+Online9+AgeZero+AgeGiven+IncomeZero+IncomeGiven
              +Tenure+District1100+District1200, data = df.train, kernel = "linear",
              cost = 10, gamma=1e-05, scale = TRUE)
svm_pred=predict(svm_model, df.test)
cf = confusionMatrix(data = svm_pred, reference = target.under.test, mode = "prec_recall")
cf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    0
##          1 1293  547
##          0  269 1034
##
##                Accuracy : 0.7404
##                  95% CI : (0.7247, 0.7556)
##     No Information Rate : 0.503
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4813
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##               Precision : 0.7027
##                  Recall : 0.8278
##                      F1 : 0.7601
##              Prevalence : 0.4970
##          Detection Rate : 0.4114
##    Detection Prevalence : 0.5854
##       Balanced Accuracy : 0.7409
##
##        'Positive' Class : 1
##
```

```
specificity=specificity(svm_pred,target.under.test, positive="1")
cat('Specificity:',specificity, '\n')
```

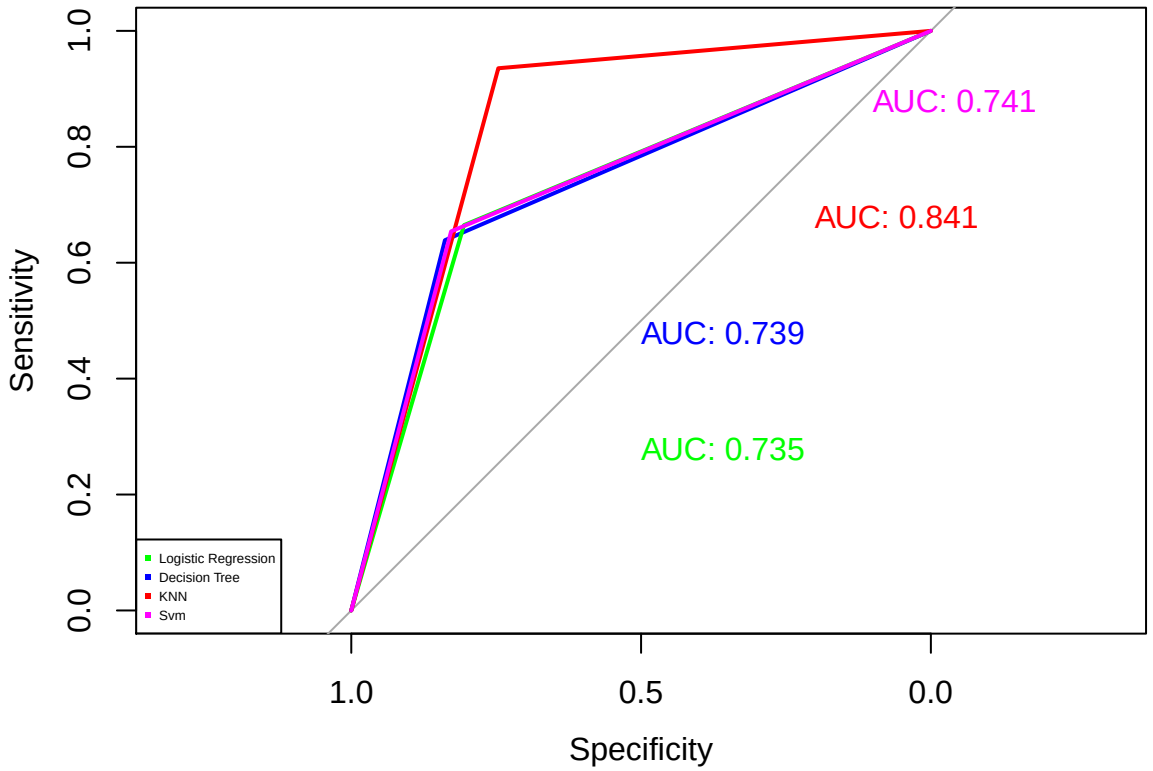
```
## Specificity: 0.6540164
```

Confronto tra modelli di classificazione

Un utile modo per confrontare la performance di un modello per un classificatore binario, sono la curva ROC e l'area sotto di essa (AUC). La ROC plotta 1 meno la specificità e la recall (o sensitività), quindi il modello migliore è quello che ha un AUC più vicino ad 1.

```
library(pROC)
treeROC = roc(target.test,factor(tree.pred, ordered = TRUE), plot=TRUE, print.auc=TRUE,
              col="blue")
logit_roc=plot.roc(target.test, factor(logit.pred, ordered = TRUE), add=TRUE,
                   col="green", print.auc=TRUE, print.auc.y=0.3)
knn_roc = plot.roc(target.test,factor(knn_pred, ordered = TRUE), add=TRUE,
                   col="red", print.auc=TRUE, print.auc.y=0.7, print.auc.x=0.2)
svm_roc = plot.roc(target.under.test,factor(svm_pred, ordered = TRUE), add=TRUE,
                   col="magenta", print.auc=TRUE, print.auc.y=0.9, print.auc.x=0.1)
```

```
legend("bottomleft", legend = c("Logistic Regression", "Decision Tree", "KNN", "Svm"),
       col=c("green", "blue", "red", "magenta"), pch=15, cex=0.4)
```



Possiamo quindi concludere che il modello che ci dà la migliore previsione relativa al comportamento futuro dei clienti è il KNN, con una precisione e una specificità entrambe oltre il 90%.

Clustering

Ora, dal dataset senza outliers, consideriamo solo coloro che nel 1999 erano offline (`Online9=0`) e che sono rimasti clienti nel 2000 (`Retain=1`). Inoltre creiamo delle variabili binarie che indicano la fascia d'età e di reddito alla quale ciascun cliente appartiene. Infine discretizziamo la variabile `Tenure`, che rappresenta gli anni di servizio, utilizzando altre variabili categoriche. In questo modo, siamo in grado di avere tutte le informazioni relative ad un cliente, escluso il profitto, attraverso delle variabili binarie.

```
detach(data)
newdata = data[Online9==0 & Retain==1,]
attach(newdata)
Profit9=X9Profit

Profit0=X0Profit
Online0=X0Online

District=X9District
District1100 = ifelse(District==1100,1,0)
District1200 = ifelse(District==1200,1,0)

Age=X9Age
Age1 = ifelse(Age==1&!is.na(Age),1,0)
Age2 = ifelse(Age==2&!is.na(Age),1,0)
```

```
Age3 = ifelse(Age==3&!is.na(Age),1,0)
Age4 = ifelse(Age==4&!is.na(Age),1,0)
Age5 = ifelse(Age==5&!is.na(Age),1,0)
Age6 = ifelse(Age==6&!is.na(Age),1,0)
Age7 = ifelse(Age==7&!is.na(Age),1,0)

Income=X9Inc
Income1 = ifelse(Income==1&!is.na(Income),1,0)
Income2 = ifelse(Income==2&!is.na(Income),1,0)
Income3 = ifelse(Income==3&!is.na(Income),1,0)
Income4 = ifelse(Income==4&!is.na(Income),1,0)
Income5 = ifelse(Income==5&!is.na(Income),1,0)
Income6 = ifelse(Income==6&!is.na(Income),1,0)
Income7 = ifelse(Income==7&!is.na(Income),1,0)
Income8 = ifelse(Income==8&!is.na(Income),1,0)
Income9 = ifelse(Income==9&!is.na(Income),1,0)

Tenure=X9Tenure
Tenure1 = ifelse(Tenure<10,1,0)
Tenure2 = ifelse(Tenure>=10&Tenure<20,1,0)
Tenure3 = ifelse(Tenure>=20&Tenure<30,1,0)
Tenure4 = ifelse(Tenure>=30&Tenure<40,1,0)
```

Possiamo osservare che nel caso in cui in uno dei valori di `Age` mancasse l'informazione, ovvero compare NA, tutte le variabili `AgeX` assumono il valore 0, quindi questo indica un'altra classe. Stessa cosa accade per la variabile `Income` e quando il valore della variabile `Tenure` è maggiore o uguale a 40.

A questo punto, il nostro obiettivo è quello di applicare un algoritmo di clustering. Quindi vorremmo cercare di raggruppare i clienti che hanno delle caratteristiche simili, in modo da applicare delle politiche di business specifiche del gruppo, per capire quale sia la strategia migliore da applicare con ciascuna tipologia di cliente.

```
set.seed(1)
num_tot_cluster = 25
df = data.frame(Age1, Age2, Age3, Age4, Age5, Age6, Age7, Income1, Income2, Income3, Income4, Income5,
k2 <- kmeans(df, centers = num_tot_cluster, nstart = 25)
```

Il metodo di clustering più conosciuto ed utilizzato per la sua immediatezza è il k-means. Per utilizzarlo, bisogna definire in anticipo qual è k, ovvero il numero finale di cluster che vogliamo formare. L'approccio è basato su un problema di ottimizzazione che consiste nel minimizzare la somma degli errori al quadrato data una specifica partizione. L'algoritmo consiste nel:

- Partire da una partizione arbitraria dove si scelgono k semi, uno per ogni cluster e ad ogni osservazione viene assegnato il cluster del seme più vicino.
- Selezionare un'osservazione ed eventualmente riassegnarla ad un altro cluster, se questa operazione migliora la qualità generale del cluster. La distanza va sempre calcolata dall'osservazione che stiamo considerando al centroide del cluster, che può essere aggiornato quando un nuovo elemento viene aggiunto al cluster.
- Ripetere finchè nessun miglioramento è possibile.

Nel caso specifico, consideriamo le variabili binarie relative all'età, al reddito, agli anni di servizio e al distretto di residenza, per formare 25 diversi cluster. All'interno di ciascun cluster andiamo a calcolare due diverse medie, quella per il clienti che sono rimasti Offline anche nel 2000 e coloro che invece in quest'anno sono passati al servizio della banca Online. In quei cluster in cui la differenza è positiva, vuol dire che in media guadagno di più quando il cliente decide di passare al canale online, quindi sarebbe utile portare avanti delle campagne pubblicitarie o applicare sconti per invogliare i clienti a cambiare. Anche se in realtà osserviamo

che questa differenza in media non assume valori troppo grandi, la massima differenza è pari a 16 dollari, quindi bisogna tenere conto di questo aspetto quando si sceglierà se e quale strategia adottare.

```
deltaProfit = Profit0-Profit9
deltaOnOff = rep(0,num_tot_cluster)
for(num_cluster in 1:num_tot_cluster) {
  deltaProfitOnline  = mean(deltaProfit[Online0[k2$cluster==num_cluster]==1])
  deltaProfitOffline = mean(deltaProfit[Online0[k2$cluster==num_cluster]==0])
  if(is.na(deltaProfitOnline)){deltaProfitOnline=0}
  if(is.na(deltaProfitOffline)){deltaProfitOffline=0}
  deltaOnOff[num_cluster] = deltaProfitOnline-deltaProfitOffline
}
sort(deltaOnOff, decreasing = TRUE)
```

```
##  [1]  16.1980207   9.3386800   6.5645477   5.8080562   5.7848845
##  [6]   5.7350611   4.0722603   3.7592511   2.3208486  -0.5822352
## [11]  -1.0123178  -1.0832612  -1.4748410  -2.0509612  -2.3209335
## [16]  -2.3222040  -2.4323903  -2.4871949  -2.5681359  -4.2206428
## [21]  -4.6739807  -4.7189654  -7.9957390 -10.4063965 -11.1045145
```