Insegnamento di Analisi dei dati (Data mining)
Prova d'esame dell' 11 luglio 2017 - parte pratica

Daniele Cugnigni

2023-02-20

Testo d'esame

Alcuni ricercatori sono interessati alla classificazione di una voce ottenuta da un file audio, come maschile o femminile a seconda di alcune proprietà acustiche. A tale scopo, nel file voce.csv, è disponibile un campione di registrazioni della durata di 2-5 minuti, tratti da 62 file audio provenienti da diversi capitoli di audiolibri. Di questi capitoli, 32 sono letti da lettori di sesso maschile e i restanti 30 da lettrici. Per ciascun file è stato creato uno spettrogramma (rappresentazione grafica dell'intensità di un suono in funzione del tempo e della frequenza) dal quale sono stati estratti i campioni che effettivamente rappresentano un suono, tralasciando il rumore bianco. Si sono quindi isolati 19090 spezzoni dei quali 10313 provenienti da lettori di sesso maschile e 8777 da lettrici. Per ciascun spezzone sono disponibili le seguenti variabili:

- meanfreq: frequenza media (in kHz)
- sd: deviazione standard della frequenza
- median: mediana della frequenza (in kHz)
- Q25: primo quartile (in kHz)
- Q75: terzo quartile (in kHz)
- IQR: scarto interquartile (in kHz)
- skew: misura di asimmetria della distribuzione
- kurt: misura di curtosi della distribuzione
- sp.ent: entropia spettrale
- sfm: piattezza spettrale
- mode: moda della frequenza (in kHz)
- centroid: centroide della frequenza
- peakf: picco di frequenza
- meanfun: media della frequenza fondamentale
- minfun: frequenza fondamentale minima
- maxfun: frequenza fondamentale massima
- meandom: media della frequenza dominante

- mindom: frequenza dominante minima
- maxdom: frequenza dominante massima
- dfrange: range della frequenza dominante
- modindx: indice di modulazione

E' inoltre disponibile la variabile qualitativa "genere" che identifica il gruppo di appartenenza.

```
dati <- read.csv("voce.csv", stringsAsFactors = TRUE)

str(dati)
```

```
## 'data.frame':    19090 obs. of  23 variables:
##  $ X          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ sound.files: Factor w/ 62 levels "1914_07_brooke_128kb.wav",..: 13 13 13 13 13 13 13 13 13 13 ...
##  $ meanfreq   : num  0.222 0.199 0.161 0.166 0.182 ...
##  $ sd         : num  0.0597 0.0746 0.0425 0.0501 0.0437 ...
##  $ median     : num  0.236 0.157 0.169 0.167 0.191 ...
##  $ Q25        : num  0.236 0.148 0.165 0.167 0.177 ...
##  $ Q75        : num  0.246 0.296 0.174 0.178 0.191 ...
##  $ IQR        : num  0.01071 0.14776 0.00968 0.01111 0.01364 ...
##  $ skew       : num  2.89 3.03 4.74 2.97 2.55 ...
##  $ kurt       : num  9.97 11.26 26.09 10.32 7.93 ...
##  $ sp.ent     : num  0.613 0.664 0.557 0.659 0.638 ...
##  $ sfm        : num  0.233 0.195 0.182 0.339 0.314 ...
##  $ mode       : num  0.246 0.152 0.169 0.167 0.177 ...
##  $ centroid   : num  0.222 0.199 0.161 0.166 0.182 ...
##  $ meanfun    : num  0.23 0.214 0.187 0.195 0.191 ...
##  $ minfun     : num  0.184 0.174 0.174 0.175 0.173 ...
##  $ maxfun     : num  0.283 0.298 0.258 0.296 0.233 ...
##  $ meandom    : num  0.172 0.163 0.172 0.172 0.172 ...
##  $ mindom     : num  0.1723 0.0861 0.1723 0.1723 0.1723 ...
##  $ maxdom     : num  0.172 0.172 0.172 0.172 0.172 ...
##  $ dfrange    : num  0 0.0861 0 0 0 ...
##  $ modindx    : num  0 0.222 0 0 0 ...
##  $ genere     : Factor w/ 2 levels "female","male": 2 2 2 2 2 2 2 2 2 2 ...
```

```
dim(dati)
```

```
## [1] 19090    23
```

```
summary(dati)
```

```
##        X                                          sound.files
##  Min.   :    1   adventuresofgrandfatherfrog_05_burgess.wav:  775
##  1st Qu.: 4773   adventuresofgrandfatherfrog_13_burgess.wav:  711
##  Median : 9546   adventuresofgrandfatherfrog_16_burgess.wav:  673
##  Mean   : 9546   adventuresofgrandfatherfrog_06_burgess.wav:  647
##  3rd Qu.:14318   columbianorator_09_bingham_128kb.wav      :  561
##  Max.   :19090   ansiedelungmeeresgrunde_04_kraft_64kb.wav :  534
##                  (Other)                                   :15189
```

```
##     meanfreq              sd              median            Q25
##  Min.   :0.008392   Min.   :0.01092   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.181692   1st Qu.:0.03452   1st Qu.:0.1732   1st Qu.:0.1385
##  Median :0.195567   Median :0.05267   Median :0.1979   Median :0.1650
##  Mean   :0.197600   Mean   :0.04994   Mean   :0.1998   Mean   :0.1666
##  3rd Qu.:0.212311   3rd Qu.:0.06555   3rd Qu.:0.2250   3rd Qu.:0.1929
##  Max.   :0.289073   Max.   :0.10499   Max.   :0.3000   Max.   :0.2857
##
##       Q75                IQR               skew              kurt
##  Min.   :0.006667   Min.   :0.002439   Min.   :-0.4391   Min.   : 1.333
##  1st Qu.:0.201266   1st Qu.:0.015789   1st Qu.: 1.9547   1st Qu.: 5.670
##  Median :0.227273   Median :0.033333   Median : 2.5646   Median : 8.725
##  Mean   :0.229414   Mean   :0.062851   Mean   : 2.6701   Mean   :10.449
##  3rd Qu.:0.257143   3rd Qu.:0.120000   3rd Qu.: 3.2982   3rd Qu.:13.388
##  Max.   :0.300000   Max.   :0.232941   Max.   : 7.6599   Max.   :66.179
##
##      sp.ent            sfm                mode           centroid
##  Min.   :0.3086   Min.   :0.007939   Min.   :0.0000   Min.   :0.008392
##  1st Qu.:0.5893   1st Qu.:0.128004   1st Qu.:0.1650   1st Qu.:0.181692
##  Median :0.6649   Median :0.185318   Median :0.1950   Median :0.195567
##  Mean   :0.6594   Mean   :0.198670   Mean   :0.1971   Mean   :0.197600
##  3rd Qu.:0.7295   3rd Qu.:0.254886   3rd Qu.:0.2278   3rd Qu.:0.212311
##  Max.   :0.9466   Max.   :0.846956   Max.   :0.3000   Max.   :0.289073
##
##     meanfun            minfun            maxfun           meandom
##  Min.   :0.08694   Min.   :0.08647   Min.   :0.08785   Min.   :0.0000
##  1st Qu.:0.19697   1st Qu.:0.17294   1st Qu.:0.26250   1st Qu.:0.1458
##  Median :0.21399   Median :0.17431   Median :0.28636   Median :0.1723
##  Mean   :0.20640   Mean   :0.16626   Mean   :0.26878   Mean   :0.1595
##  3rd Qu.:0.22721   3rd Qu.:0.17854   3rd Qu.:0.29400   3rd Qu.:0.1723
##  Max.   :0.28248   Max.   :0.27055   Max.   :0.29797   Max.   :0.2584
##
##      mindom            maxdom          dfrange           modindx
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08613   1st Qu.:0.1723   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.12920   Median :0.1723   Median :0.04307   Median :0.03571
##  Mean   :0.11813   Mean   :0.1762   Mean   :0.05806   Mean   :0.18125
##  3rd Qu.:0.17227   3rd Qu.:0.1723   3rd Qu.:0.08613   3rd Qu.:0.31818
##  Max.   :0.25840   Max.   :0.2584   Max.   :0.25840   Max.   :1.00000
##
##     genere
##  female: 8777
##  male  :10313
##
##
##
##
##
```

```
all.equal(dati$X, 1:NROW(dati))
```

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

```
length(table(dati$sound.files))
```

```
## [1] 62
```

Pulizia del dataset

Il file "voce.csv" è composto da 19090 unità statistiche (gli spezzoni dei 62 file audio) sulle quali sono state rilevate complessivamente 23 variabili, con la variabile *genere* che rappresenta la variabile risposta.
Prima di procedere all'analisi del dataset, è opportuno effettuare delle operazioni di pulizia. In primo luogo si nota come la variabile X non è nient'altro che l'indicatore di riga, pertanto viene eliminata. Inoltre si nota come la variabile *sound.files* faccia riferimento a quale dei 62 file audio (o capitoli) appartiene il singolo spezzone, ovvero è una variabile indicatrice dell'unità statistica originale (il singolo file audio), pertanto viene eliminata.
A questo punto, tenendo conto anche del fatto di avere a disposizione un numero relativamente piccolo di variabili esplicative, è opportuno effettuare delle considerazioni meramente statistiche. Prima di tutto si nota che sono state rilevate le variabili *Q25*, *Q75* e *IQR*, e la relazione che lega queste tre variabili è lineare ed è data da $IQR = Q75 - Q25$, pertanto l'informazione contenuta nello scarto interquartile può essere ricavata dalla differenza tra terzo quartile e primo quartile. Alla luce di questa considerazione, viene deciso di eliminare la variabile *IQR*. Con un ragionamento analogo, emerge che sono state rilevate le variabili riguardanti la frequenza dominante massima, *maxdom*, la frequenza dominante minima, *mindom*, e il range della frequenza dominante, *dfrange*, e viene deciso di eliminare quest'ultima. Inoltre vi è la presenza sia della variabile *meanfreq* che della variabile *centroid*: la correlazione tra queste due variabili è pari ad 1, in particolare i valori della variabile *meanfreq* e della variabile *centroid* sono identici per ogni unità statistica, pertanto l'informazione portata dalle due variabili è la medesima e si decide di eliminare la variabile *centroid*.

```
cor(dati$Q75 - dati$Q25, dati$IQR)
```

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

```
cor(dati$maxdom - dati$mindom, dati$dfrange)
```

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

```
cor(dati$meanfreq, dati$centroid)
```

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

```
all.equal(dati$meanfreq, dati$centroid)
```

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

```
dati$X <- NULL
dati$sound.files <- NULL
dati$IQR <- NULL
dati$dfrange <- NULL
dati$centroid <- NULL
dim(dati)
```

```
## [1] 19090    18
```

Inoltre, nel dataset non sono presenti valori mancanti.

```
#Controllo della presenza di NA
na_get <- function(data){
  na_vars <- sapply(data, function(col) sum(is.na(col)))
  na_vars <- sort(na_vars[na_vars > 0])
  na_vars <- data.frame(
    variabile <- names(na_vars),
    freq_assoluta <- as.numeric(na_vars),
    freq_relativa <- round(as.numeric(na_vars)/nrow(data), 4)
  )
  na_vars
}
na_tab <- na_get(dati)
na_tab
```

```
## [1] variabile....names.na_vars.
## [2] freq_assoluta....as.numeric.na_vars.
## [3] freq_relativa....round.as.numeric.na_vars..nrow.data...4.
## <0 righe> (o 0-length row.names)
```

In seguito a queste operazioni, il dataset è composto da 19090 unità statistiche e 18 variabili. A questo punto, prima di procedere con la modellazione dei dati:

- si verifica l'assenza di variabili esplicative degeneri (e quindi inutili per l'analisi);
- per tenere in considerazione il compromesso tra varianza e distorsione, si procede con la divisione del dataset in insieme di stima (80%) e insieme di verifica (20%), ottenendo un insieme di stima con 15272 osservazioni ed un insieme di verifica con 3818 osservazioni;
- si verifica che nell'insieme di stima le classi della variabile risposta siano bilanciate. In particolare, il 46% degli spezzoni sono di voce femminile ed il 54% di voce maschile, pertanto, poichè le classi della variabile risposta risultano essere bilanciate, non si ha la necessità di svolgere ulteriori operazioni e si può procedere all'analisi esplorativa nell'insieme di stima.

```
#Divisione variabili quantitative e variabili qualitative
tipo_var <- sapply(dati, class)
table(tipo_var)
```

```
## tipo_var
##  factor numeric
##       1      17
```

```
var_qualitative <- names(dati)[tipo_var == "factor"]
var_quantitative <- setdiff(names(dati), var_qualitative)
var_qualitative
```

```
## [1] "genere"
```

```
var_quantitative
```

```
##  [1] "meanfreq" "sd"       "median"   "Q25"      "Q75"      "skew"
##  [7] "kurt"     "sp.ent"   "sfm"      "mode"     "meanfun"  "minfun"
## [13] "maxfun"   "meandom"  "mindom"   "maxdom"   "modindx"
```

```
#Rimozione delle variabili quantitative degeneri
ids.deg <- which(apply(dati, 2, var) == 0)
ids.deg
```

```
## named integer(0)
```

```
#Rimozione delle variabili qualitative degeneri
for(col in var_qualitative) cat(col,":", nlevels(dati[,col]), "livelli \n")
```

```
## genere : 2 livelli
```

```
#Rimozione/trasformazione in fattori di variabili quantitative che assumono poche modalità
const <- apply(dati[,var_quantitative], 2, function(x) length(unique(x)) < 4)
summary(dati[,var_quantitative][,const])
```

```
## < table of extent 0 x 0 >
```

```
#Salvo l'indice della risposta
ids.leak <- which(names(dati) %in% c("genere"))
ids.leak
```

```
## [1] 18
```

```
tipo_var <- sapply(dati[, -ids.leak], class)
table(tipo_var)
```

```
## tipo_var
## numeric
##      17
```

```
var_qualitative <- names(dati)[-ids.leak][tipo_var == "factor"]
for(col in var_qualitative) cat(col,":", nlevels(dati[,col]), "livelli \n")
var_quantitative <- setdiff(names(dati)[-ids.leak], var_qualitative)

var_qualitative
```

```
## character(0)
```

```
var_quantitative
```

```
##  [1] "meanfreq" "sd"       "median"   "Q25"      "Q75"      "skew"
##  [7] "kurt"     "sp.ent"   "sfm"      "mode"     "meanfun"  "minfun"
## [13] "maxfun"   "meandom"  "mindom"   "maxdom"   "modindx"
```

```r
#Divisione in insieme di stima e insieme di verifica

n <- dim(dati)[1]
p <- dim(dati)[2]
set.seed(12)
ind <- sample(1:n, round((4/5)*n), replace = T)
stima <- dati[ind, ]
ver <- dati[-ind, ]
rm(dati)

dim(stima)
```

```
## [1] 15272    18
```

```r
dim(ver)
```

```
## [1] 8515   18
```

```r
prop.table(table(stima$genere)) #classi bilanciate nell'insieme di stima
```

```
##
##    female      male
## 0.4585516 0.5414484
```

Analisi esplorativa

Tenendo in considerazione che la variabile risposta è una variabile categoriale con due modalità e le variabili esplicative risultano essere tutte quantitative, un'analisi esplorativa (abbastanza) completa ed adeguata si avrebbe con la discretizzazione di ciascuna delle variabili esplicative e l'analisi della distribuzione della variabile dipendente al variare delle singole variabili indipendenti. Poichè l'obiettivo primario non è quello di effettuare l'analisi esplorativa ma di adattare i modelli, si valuta la distribuzione della risposta solamente per alcune variabili esogene.

```r
par(mfrow = c(2,3))
nomi <- c("meanfreq", "Q25", "Q75", "median", "sfm", "sp.ent")
for(i in 1:length(nomi)) {
  classi <- cut(stima[,nomi[i]], breaks = round(summary(stima[,nomi[i]])[-4],2),
                include.lowest = T)
  nuovo <- data.frame(genere = stima$genere, esplicativa = classi)
  colnames(nuovo) <- c("genere", nomi[i])
  condizionata <- prop.table(table(nuovo),2)
  barplot(condizionata,beside = T, xlab = nomi[i], ylab = "genere",
          ylim = c(0,1.05), col = c(3,4),legend.text = c("female", "male"),
          cex.axis = 1.1, cex.names = 1.09, cex.lab = 1.4)
}
```

```r
rm(nuovo)
```

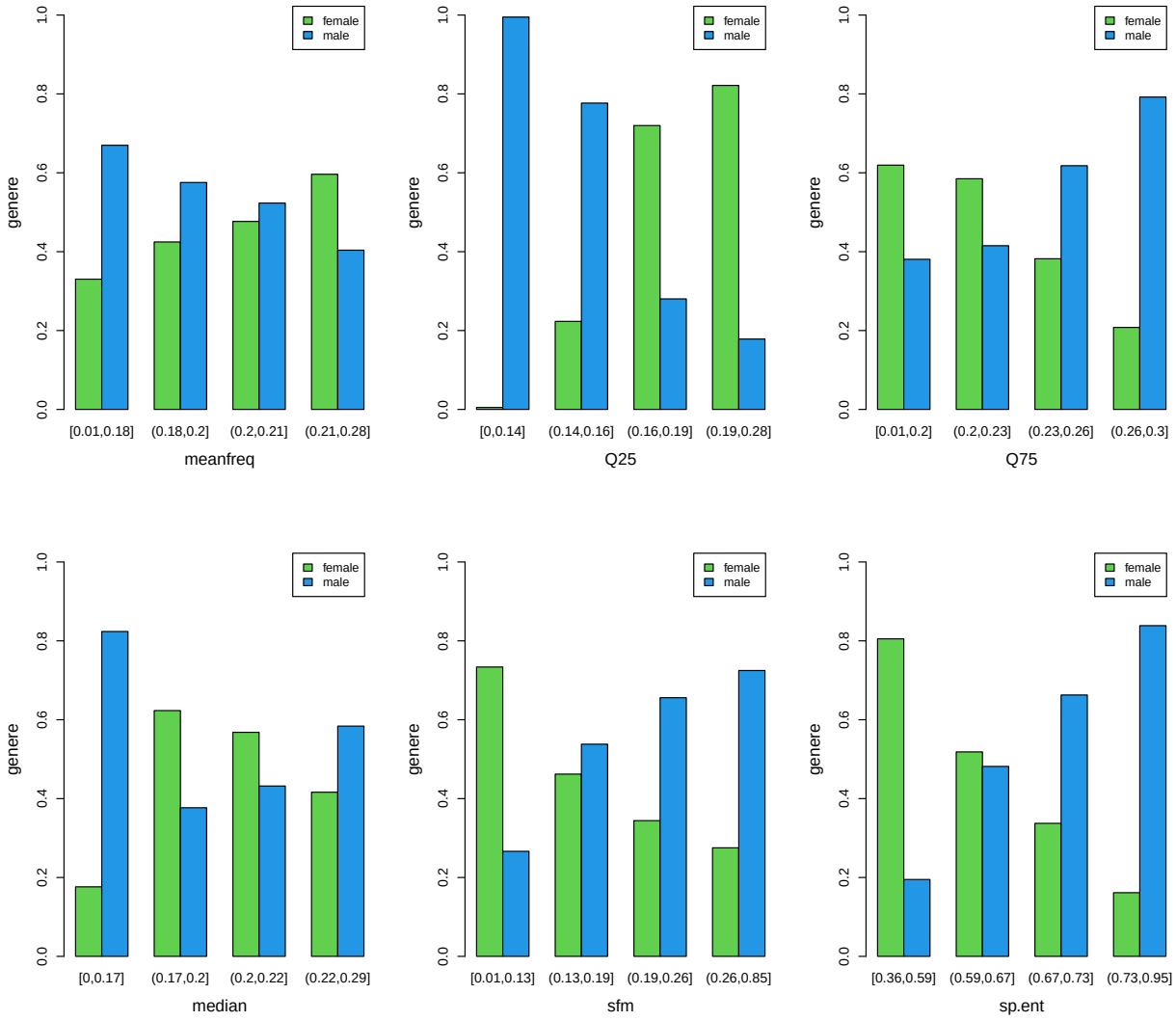

Figure 1: Barplot della variabile risposta rispetto ad alcune variabili esplicative

I barplot in Figura 1 danno indicazione di un possibile effetto molto significativo del primo quartile della frequenza (*Q25*): per valori piccoli del primo quartile la percentuale di spezzoni con voce femminile risulta essere inferiore rispetto alla percentuale di spezzoni con voce maschile, mentre per valori grandi del primo quartile vi è una presenza maggiore di spezzoni con voce femminile. Per quanto riguarda le altre variabili, sembrerebbe esserci un effetto significativo del terzo quartile (*Q75*), dell'entropia spettrale (*sp.ent*) e della piattezza spettrale (*sfm*) in quanto, per tutte e tre, le donne presentano un andamento decrescente all'aumentare del valore della variabile esplicativa mentre gli uomini presentano un andamento crescente. Infine, sembrerebbe non esserci un effetto significativo della frequenza media (*meanfreq*) mentre si nota come per valori piccoli e grandi della mediana (*median*) si ha un numero maggiore di spezzoni con voce femminile, mentre per valori intermedi si hanno più spezzoni con voce maschile.

Conclusa l'analisi esplorativa nell'insieme di stima, si può procedere alla modellazione dei dati.

Modellazione dei dati

Poichè l'interesse dei ricercatori è rivolto alla classificazione della voce, senza focalizzare l'attenzione sul classificare correttamente le voci maschili o le voci femminili, e poichè le classi della variabile risposta nell'insieme di stima risultano essere bilanciate, si utilizzerà una soglia pari a 0.5 e come metrica per il confronto tra i modelli il tasso di errata classificazione.

```
#Formula del modello completo
nomi <- names(stima)
form <- as.formula(paste("genere ~ ", paste(nomi[-ids.leak],collapse ="+")))

#Funzione che calcola matrice di confusione e gli errori di classificazione
tabella.sommario <- function(previsti, osservati){
  n <-  table(previsti,osservati)
  err.tot <- 1-sum(diag(n))/sum(n)
  print(n)
  cat("errore totale: ", format(err.tot),"\n")
  invisible(n)
}

#Errori
tab <- list()
```

Modello logistico

Il primo modello che si adatta è il modello di regressione logistica su tutte le variabili esplicative (senza interazione) con funzione di legame la funzione *logit*.

```
mlog1 <- glm(form, data = stima, family = binomial)
summary(mlog1)
```

```
##
## Call:
## glm(formula = form, family = binomial, data = stima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.4837  -0.6173   0.1386   0.4033   2.9260
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   7.88975    0.61050  12.923  < 2e-16 ***
## meanfreq    -24.24700    4.16778  -5.818 5.97e-09 ***
## sd           60.25396    3.24774  18.553  < 2e-16 ***
## median       18.26871    1.80912  10.098  < 2e-16 ***
## Q25         -29.74227    1.68030 -17.701  < 2e-16 ***
## Q75           3.92532    1.80802   2.171 0.029927 *
## skew         -0.86651    0.12724  -6.810 9.76e-12 ***
## kurt          0.06668    0.01673   3.986 6.73e-05 ***
## sp.ent        1.45503    0.49185   2.958 0.003094 **
## sfm          -6.40843    0.44753 -14.320  < 2e-16 ***
## mode         -7.86226    1.14514  -6.866 6.61e-12 ***
## meanfun      11.61912    1.99655   5.820 5.90e-09 ***
## minfun       -8.05619    1.56154  -5.159 2.48e-07 ***
## maxfun       -3.89409    1.07993  -3.606 0.000311 ***
## meandom      -6.72214    2.13648  -3.146 0.001653 **
## mindom       -0.89741    0.58728  -1.528 0.126491
## maxdom       -3.45990    1.81400  -1.907 0.056477 .
## modindx       0.29532    0.16598   1.779 0.075207 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21066  on 15271  degrees of freedom
## Residual deviance: 11259  on 15254  degrees of freedom
## AIC: 11295
##
## Number of Fisher Scoring iterations: 6
```

```
mlog1.pred <- predict(mlog1, newdata = ver, type = "response")
mlog1.tab <- tabella.sommario(mlog1.pred > 0.5, ver$genere)
```

```
##         osservati
## previsti female male
##    FALSE   3471  810
##    TRUE     446 3788
## errore totale:  0.1475044
```

```
tab <- c(tab, list(Logistico = mlog1.tab))
```

Ad un livello di significatività del 5%, le uniche variabili che risultano avere un effetto statisticamente nullo sulla variabile risposta risultano essere la frequenza dominante minima, la frequenza dominante massima e l'indice di modulazione.
Il tasso di errata classificazione nell'insieme di verifica è pari al **14.75%**.

Modello logistico stepwise

Poichè il modello logistico adattato in precedenza ha messo in luce l'effetto non significativo di alcune variabili esplicative, si ritiene ragionevole adattare un modello di regressione logistica stepwise basato sulla minimizzazione dell'AIC, con ricerca in entrambe le direzioni e a partire dal modello con la sola intercetta.

```r
mlog1 <- glm(genere ~ 1, weights = NULL, data = stima, family = binomial)
mlog2 <- step(mlog1, scope = form, direction = "both", trace = F)
summary(mlog2)
```

```
##
## Call:
## glm(formula = genere ~ Q25 + sd + meanfreq + skew + sfm + meandom +
##     median + mode + kurt + sp.ent + minfun + meanfun + maxfun +
##     Q75 + mindom, family = binomial, data = stima, weights = NULL)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.4605  -0.6167   0.1379   0.4068   2.9205
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   7.68750    0.58589  13.121  < 2e-16 ***
## Q25         -30.32663    1.64537 -18.431  < 2e-16 ***
## sd           60.85650    3.22633  18.862  < 2e-16 ***
## meanfreq    -23.65968    4.13880  -5.717 1.09e-08 ***
## skew         -0.87879    0.12707  -6.916 4.64e-12 ***
## sfm          -6.40268    0.44714 -14.319  < 2e-16 ***
## meandom      -9.14857    1.73951  -5.259 1.45e-07 ***
## median       17.99049    1.80309   9.978  < 2e-16 ***
## mode         -7.75189    1.14361  -6.778 1.21e-11 ***
## kurt          0.06755    0.01674   4.036 5.44e-05 ***
## sp.ent        1.33749    0.48657   2.749 0.005981 **
## minfun       -7.55728    1.46791  -5.148 2.63e-07 ***
## meanfun      11.94019    1.98796   6.006 1.90e-09 ***
## maxfun       -3.99015    1.07965  -3.696 0.000219 ***
## Q75           3.86840    1.80398   2.144 0.032003 *
## mindom       -0.81766    0.55206  -1.481 0.138578
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21066  on 15271  degrees of freedom
## Residual deviance: 11264  on 15256  degrees of freedom
## AIC: 11296
##
## Number of Fisher Scoring iterations: 6
```

```r
logist.step.var <- names(mlog2$model)[-1]
length(logist.step.var)
```

```
## [1] 15
```

```r
mlog2.pred <- predict(mlog2, newdata = ver, type = "response")
mlog2.tab <- tabella.sommario(mlog2.pred > 0.5, ver$genere)
```

```
##         osservati
```

```
## previsti female male
##    FALSE   3461  813
##    TRUE     456 3785
## errore totale:  0.1490311
```

```
tab <- c(tab, list(Logistico.stepwise = mlog2.tab))
names(tab)[2] <- "Logistico stepwise"
```

Nel modello finale sono incluse 15 delle 17 variabili esplicative, ovvero tutte ad eccezione della frequenza dominante massima e dell'indice di modulazione.
Il tasso di errata classificazione nell'insieme di verifica è pari al **14.90%**.

```
#Divisione training e validation set

set.seed(1234)
ind <- sample(1:nrow(stima), round((3/4)*nrow(stima)))
stima.rid <- stima[ind,]
conv <- stima[-ind,]
rm(ind)
```

Albero di classificazione

Si prosegue la fase di modellazione con l'adattamento di un albero di classificazione, con l'entropia come funzione da minimizzare. Poichè questo modello prevede la selezione del numero di foglie ottimale, si divide l'insieme di stima in due sottoinsiemi: un insieme di stima ridotto in cui far crescere l'albero e un insieme di convalida in cui effettuare la fase di potatura. Nella fase di crescita dell'albero viene impostata una numerosità minima di osservazioni per foglia pari a 2 e una diminuzione dell'entropia per consentire uno split pari almeno a 0.000005, in modo da far diventare l'albero il più profondo possibile. Nella fase di potatura viene valutata la devianza nell'insieme di convalida al variare del numero di foglie dell'albero. Il grafico in Figura 2 mostra come il minimo si ottenga con un albero con 14 foglie.

```
library(tree)
set.seed(1)
mtree.or <- tree(genere ~., weights = NULL, data = stima.rid,
                 control = tree.control(nobs =nrow(stima.rid), minsize = 2,
                                        mindev = 0.000005))
prune.mtree <- prune.tree(mtree.or, newdata = conv)
plot(prune.mtree)
J.opt <- prune.mtree$size[which.min(prune.mtree$dev)]
abline(v = J.opt, col = 2, lty = "dashed")
```

Uno dei pregi di questo modello è la facile interpretabilità nel caso in cui l'albero sia poco profondo. A tal riguardo, il grafico in Figura 3 mostra gli split dell'albero selezionato, mettendo in luce che il primo quartile e la deviazione standard della frequenza sono le variabili che entrano in gioco nelle prime suddivisioni dell'albero.

```
mtree <- prune.tree(mtree.or, best = J.opt)
plot(mtree, type = "uniform")
text(mtree, pretty = 4, cex = 0.7)
```

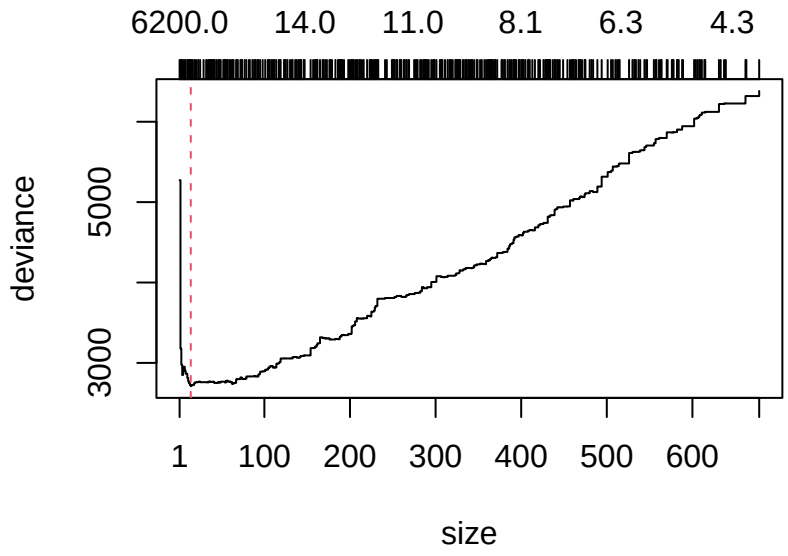

Figure 2: Errore nell'insieme di convalida in funzione del numero di foglie

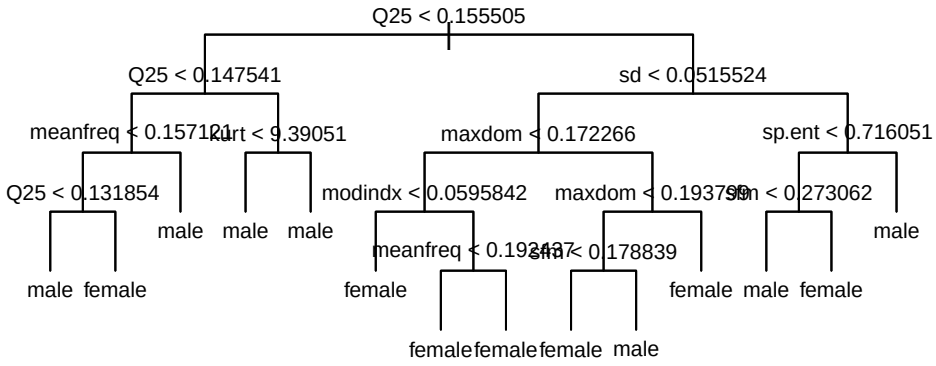

Figure 3: Albero di classificazione selezionato

```
mtree.pred.prob <- predict(mtree, newdata = ver, type = "vector")[,2]
tree.tab <- tabella.sommario(mtree.pred.prob > 0.5, ver$genere)
```

```
##         osservati
## previsti female male
##    FALSE   3462  772
##    TRUE     455 3826
## errore totale:  0.1440986
```

```
tab <- c(tab, list(Albero = tree.tab))
```

Nell'insieme di verifica il tasso di errata classificazione è pari al **14.41%**.

Modello additivo

Il modello successivamente adattato è il modello additivo generalizzato (GAM). Sono utilizzate le splines di lisciamento con al massimo 3 gradi di libertà equivalenti come lisciatori per le variabili quantitative e una procedura di tipo passo a passo ibrida basata sulla minimizzazione dell'AIC implementata nell'insieme di stima.

```
library(gam)
gam1 = gam(genere ~ 1, weights = NULL, family = binomial, data = stima)
scope = gam.scope(stima[,-ids.leak], arg = c("df = 2", "df = 3"))
gam.step = step.Gam(gam1, scope = scope, trace = F)
summary(gam.step)
```

```
##
## Call: gam(formula = genere ~ s(sd, df = 3) + s(median, df = 3) + s(Q25,
##     df = 3) + s(Q75, df = 3) + s(skew, df = 3) + s(kurt, df = 3) +
##     s(sp.ent, df = 3) + s(sfm, df = 3) + s(meandom, df = 3) +
##     s(mindom, df = 3) + modindx, family = binomial, data = stima,
##     weights = NULL, trace = FALSE)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -4.90647 -0.59247  0.06604  0.30693  2.47512
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 21066.42 on 15271 degrees of freedom
## Residual Deviance: 10206.88 on 15240 degrees of freedom
## AIC: 10270.88
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                       Df Sum Sq Mean Sq  F value    Pr(>F)
## s(sd, df = 3)          1   1994 1994.42 101.5002 < 2.2e-16 ***
## s(median, df = 3)      1    192  191.98   9.7702  0.001777 **
## s(Q25, df = 3)         1    751  750.93  38.2162 6.494e-10 ***
## s(Q75, df = 3)         1     17   17.30   0.8803  0.348136
## s(skew, df = 3)        1     41   40.74   2.0735  0.149901
```

```
## s(kurt, df = 3)        1    146  146.26   7.4436  0.006373 **
## s(sp.ent, df = 3)      1      3    2.65   0.1347  0.713575
## s(sfm, df = 3)         1    123  122.69   6.2438  0.012473 *
## s(meandom, df = 3)     1     66   66.16   3.3670  0.066536 .
## s(mindom, df = 3)      1      1    0.86   0.0436  0.834552
## modindx                1      5    5.11   0.2599  0.610173
## Residuals          15240 299457   19.65
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                     Npar Df Npar Chisq    P(Chi)
## (Intercept)
## s(sd, df = 3)            2      67.77 1.887e-15 ***
## s(median, df = 3)        2      51.97 5.181e-12 ***
## s(Q25, df = 3)           2     318.79 < 2.2e-16 ***
## s(Q75, df = 3)           2     166.42 < 2.2e-16 ***
## s(skew, df = 3)          2      20.56 3.432e-05 ***
## s(kurt, df = 3)          2      28.08 7.994e-07 ***
## s(sp.ent, df = 3)        2      60.42 7.605e-14 ***
## s(sfm, df = 3)           2      83.67 < 2.2e-16 ***
## s(meandom, df = 3)       2      60.77 6.373e-14 ***
## s(mindom, df = 3)        2     101.92 < 2.2e-16 ***
## modindx
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#plot(gam.step, ask = TRUE, se = TRUE)
gam.step.pred = predict(gam.step, newdata = ver, type = "response")
gam.step.tab = tabella.sommario(gam.step.pred > 0.5, ver$genere)
```

```
##         osservati
## previsti female male
##    FALSE   3565  814
##    TRUE     352 3784
## errore totale:  0.1369348
```

```
tab = c(tab, list(gam.step = gam.step.tab))
names(tab)[4] <- "Gam stepwise"
```

Il modello finale include la deviazione standard, la mediana, il primo e il terzo quartile della frequenza, la curtosi e l'asimmetria della distribuzione, la piattezza e l'entropia spettrale, la media e il minimo della frequenza dominante. Nel dettaglio, gli effetti delle variabili appena menzionato sono tutti stimati tramite splines di lisciamento con 3 gradi di libertà equivalenti.
Nell'insieme di verifica si ottiene un tasso di errata classificazione pari al **13.69%**.

Random forest

Si procede con l'adattamento del *random forest*. Il parametro di regolazione del modello è il numero di covariate da considerare ad ogni suddivisione dell'albero. A tal riguardo, l'insieme di stima viene diviso in un insieme di stima ridotto e uno di convalida e viene adattato il *random forest* con 250 alberi in corrispondenza di ognuno dei possibili valori del numero di covariate considerate. Il numero di covariate selezionato è il

valore corrispondente al modello con tasso di errata classificazione minore nell'insieme di convalida. La Figura 4 mostra che, con tale procedura, si sceglie un numero di colonne da campionare in ogni albero pari a 8.

```
library(randomForest)
mtries <-  c(1, 2, seq(5, 17, 3))
err <- rep(NA, length(mtries))
set.seed(123)
for(i in 1:length(mtries)){
  rf <- randomForest(x = stima.rid[, -ids.leak], y = stima.rid$genere,
                     xtest = conv[, -ids.leak], ytest = conv$genere,
                     ntree = 250, mtry = mtries[i],
                     nodesize = 5, weights = NULL)
  err[i] <- rf$test$err.rate[250,1]
  cat(i, "")
}
```

```
## 1 2 3 4 5 6 7
```

```
plot(mtries, err, type = "l", xlab = "Numero di covariate campionate",
     ylab = "Tasso di errata classificazione", main = "")
mtry.opt <- mtries[which.min(err)]
abline(v = mtry.opt, col = 2, lty = "dashed")
```

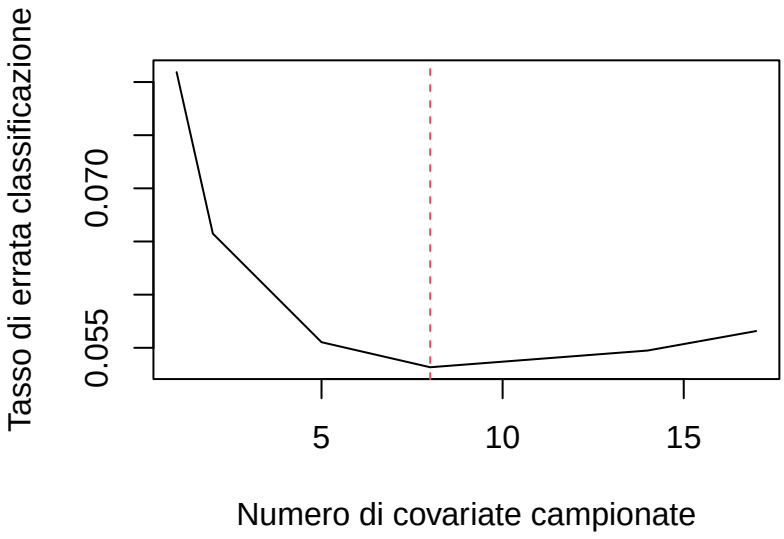

Figure 4: Errore nell'insieme di convalida in funzione del numero di covariate campionate

Successivamente il modello selezionato è adattato sull'intero insieme di stima e permette di ottenere un tasso di errata classificazione nell'insieme di verifica pari al **9.71%**.
Questo modello permette di ottenere una misura di importanza delle variabili esplicative, senza però avere indicazione sulla direzione dell'effetto di esse sulla risposta. In queso caso, la Figura 5 mette in luce che le

variabili più importanti in termini di diminuzione dell'errore di previsione risultano essere il primo quartile, la piattezza spettrale, la media e il massimo della frequenza fondamentale e la frequenza media.

```
set.seed(2222)
rf <- randomForest(x = stima[, -ids.leak], y = stima$genere, ntree = 250,
                   mtry = mtry.opt, importance = TRUE, weights = NULL)
rf.pred.prob <- predict(rf, newdata = ver, type = "prob")[,2]
rf.tab <- tabella.sommario(rf.pred.prob > 0.5, ver$genere)
```

```
##         osservati
## previsti female male
##    FALSE   3647  556
##    TRUE     270 4042
## errore totale:  0.09700528
```

```
tab <- c(tab, list(randomforest = rf.tab))
names(tab)[5] <- "Random Forest"
varImpPlot(rf, type = "1", main = "")
```

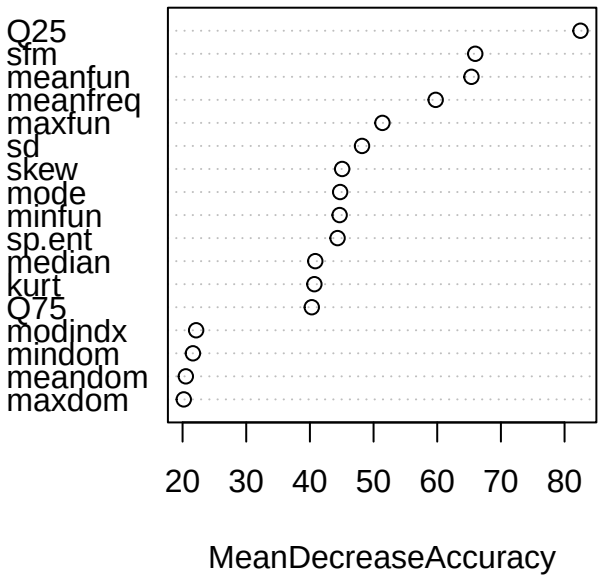

Figure 5: Importanza delle variabili nel random forest

Bagging

Si adatta un *bagging* con alberi di classificazione. Viene calcolato l'errore OOB per diversi valori del numero di campioni boostrap (e quindi di alberi) utilizzato dal modello, scegliendo il valore per cui l'errore OOB è

minore. In questo caso è pari a 150, come si evince dalla Figura 6, in cui si riporta il grafico dell'errore OOB in funzione del numero di campioni bootstrap.
Il modello selezionato ottiene sull'insieme di verifica un tasso di errata classificazione pari al **10.05%**.

```
library(ipred)
nbag <- seq(10, 190, by = 10)
err <- rep(NA, length(nbag))
set.seed(5678)
for(i in 1:length(nbag)){
  bag <- bagging(stima$genere ~., data = stima,
                 nbagg = nbag[i], coob = TRUE)
  err[i] <- bag$err
  cat(i, "")
}
```

```
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

```
plot(nbag, err, xlab = "Numero di campioni bootstrap", ylab = "Errore OOB", type = "l",
     main = "")
nbag.opt <- nbag[which.min(err)]
abline(v = nbag.opt, col = 2, lty = "dashed")
```

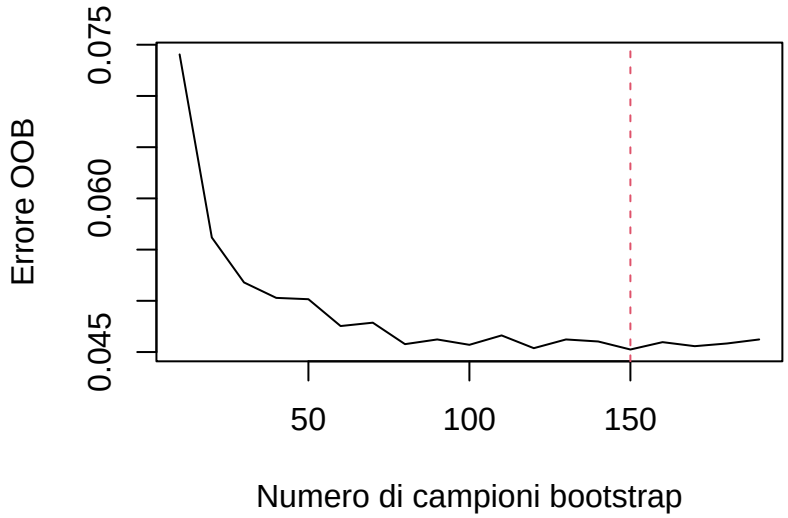

Figure 6: Errore OOB (Out-Of-Bag) nell'insieme di stima in funzione del numero di campioni bootstrap

```
set.seed(567)
bag <- bagging(stima$genere ~., data = stima,
               nbagg = nbag.opt, coob = TRUE)
bag.pred.prob <- predict(bag, newdata = ver, type = "prob")[,2]
bag.tab <- tabella.sommario(bag.pred.prob > 0.5, ver$genere)
```

```
##         osservati
## previsti female male
##    FALSE   3640  579
##    TRUE     277 4019
## errore totale:  0.1005285
```

```
tab <- c(tab, list(Bagging = bag.tab))
```

Boosting

Si adatta un *boosting* con alberi di classificazione. Per individuare il numero di alberi necessari a stabilizzare l'errore di previsione, si divide l'insieme di stima in un insieme di stima ridotto e uno di convalida. La Figura 7 mostra l'errore di previsione nell'insieme di convalida in funzione del numero di iterazioni dell'algoritmo, facendo notare che l'errore si stabilizza dopo 130 iterazioni.

```
library(ada)
set.seed(99)
boost <- ada(stima.rid$genere~., data = stima.rid,
             test.x = conv[, -ids.leak], test.y = conv$genere, iter = 200)
plot(boost, test = TRUE)
```

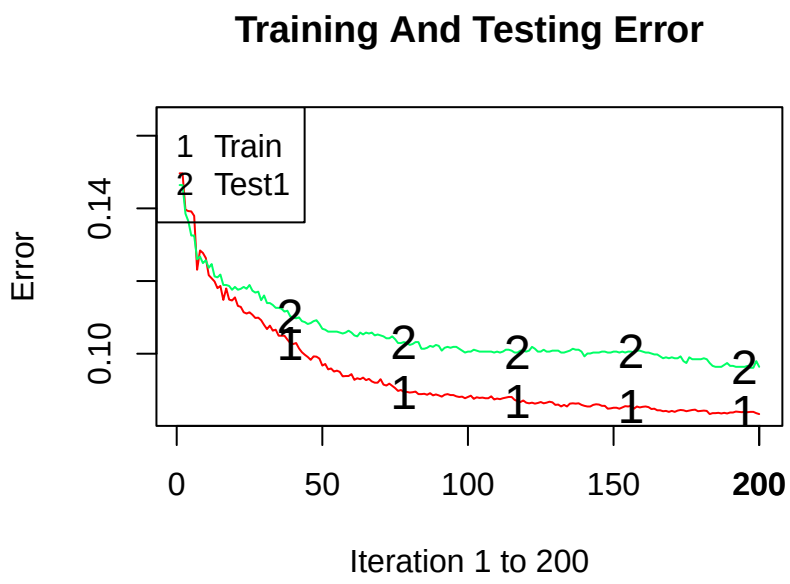

Figure 7: Errore di previsione nell'insieme di convalida in funzione del numero di iterazioni

Il modello selezionato è riadattato sull'intero insieme di stima e ottiene un tasso di errata classificazione nell'insieme di verifica pari al **10.51%**.
Anche questo modello ha il pregio di portare informazione sull'importanza delle variabili esplicative. La Figura 8 permette di far notare che le variabili maggiormente presenti negli stumps risultano essere la mediana, la moda, il primo e il terzo quartile, la media e il massimo della frequenza dominante.

```
set.seed(111)
boost <- ada(stima$genere ~., data = stima, iter = 130)
boost.pred.prob <- predict(boost, newdata = ver, type = "prob")[,2]
boost.tab <- tabella.sommario(boost.pred.prob > 0.5, ver$genere)
```

```
##         osservati
## previsti female male
##    FALSE   3653  631
##    TRUE     264 3967
## errore totale:  0.1051086
```

```
tab <- c(tab, list(Boosting = boost.tab))
varplot(boost)
```

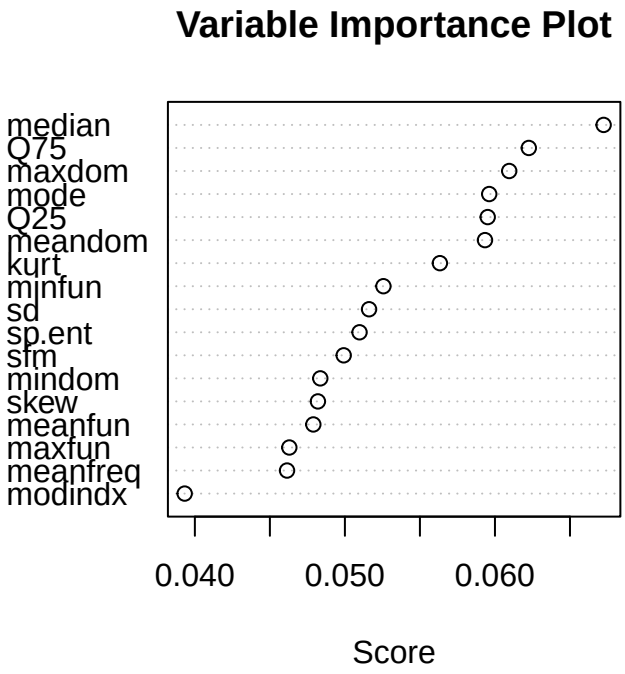

Figure 8: Importanza delle variabili nel boosting

Risultati

Nella Tabella 1 si riportano i risultati ottenuti coi diversi modelli adattati in termini di accuratezza (ovvero il complemento ad 1 del tasso di errata classificazione).

```
metriche.class = function(lista){
  n.mod = length(lista)
  nomi = names(lista)
  nomi.num = rep(NA, n.mod)
```

```
  for(i in 1:n.mod) nomi.num[i] = nomi[i]
  mat = matrix(NA, n.mod, 5)
  rownames(mat) = nomi.num
  colnames(mat) = c("Accuratezza", "Sensibilita'", "Specificita'",
                    "Precisione", "F1 Score")
  for(i in 1:n.mod){
    mat[i,1] = acc = sum(diag(lista[[i]]))/sum(lista[[i]])
    mat[i,2] = sens = lista[[i]][2,2]/sum(lista[[i]][,2])
    mat[i,3] = spec = lista[[i]][1,1]/sum(lista[[i]][,1])
    mat[i,4] = prec = lista[[i]][2,2]/sum(lista[[i]][2,])
    mat[i,5] = f1 = 2/((1/sens) + (1/prec))
  }
  return(mat)
}

knitr::kable(sort(metriche.class(tab)[,1], decreasing = T),
             caption = "Tasso di accuratezza dei modelli adattati",
             col.names = "Accuratezza", align = "c",
             digits = 4 ,format = "simple")
```

Table 1: Tasso di accuratezza dei modelli adattati

	Accuratezza
Random Forest	0.9030
Bagging	0.8995
Boosting	0.8949
Gam stepwise	0.8631
Albero	0.8559
Logistico	0.8525
Logistico stepwise	0.8510

Si nota come il modello che permette di riconoscere in maniera migliore la voce dei file audio risulta essere il *random forest*, in quanto ha un'accuratezza del 90.29%, seguito dal *bagging* e dal *boosting*, i quali hanno un tasso di accuratezza praticamente identico a quello del modello migliore e pari rispettivamente all'89.95% e all'89.49%. Una capacità predittiva peggiore si ha con il modello additivo generalizzato (86.31%), l'albero di classificazione (85.59%), il modello logistico (85.25%) e il modello logistico stepwise (85.10%).
Focalizzando l'attenzione sul *random forest*, come già è stato detto in precedenza, questo modello permette di avere una misura di importanza delle variabili esplicative, senza però avere indicazione sulla direzione dell'effetto di queste variabili sulla risposta. In questo caso, le variabili maggiormente importanti risultano essere il primo quartile, la piattezza spettrale, la media e il massimo della frequenza fondamentale e la frequenza media.