Progetto DM

Importazione dati

Il dataset considerato contiene dati relativi a 4238 soggetti.
L'outcome binario è TenYearCHD e indica la presenza o meno di malattie cardiache dopo 10 anni dalla rilevazione dei dati.
Le variabili indipendenti indicano alcune caratteristiche degli individui (sesso, età, educazione,…) e sono sia continue che fattori (binari o su più livelli).

Si procede con delle analisi preliminari e con il fit di alcuni modelli sul dataset di training.

STEP1: FIT MODELLI

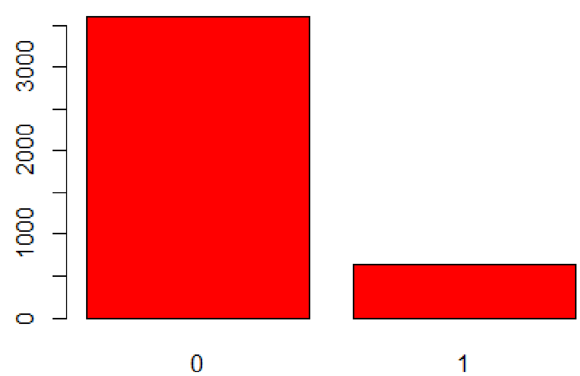
Controllo e sistemazione variabili

```
##     variable q_zeros p_zeros q_na p_na q_inf p_inf    type unique
## 1    glucose       0    0.00  388 9.16     0     0 numeric    143
## 2  education       0    0.00  105 2.48     0     0  factor      4
## 3     BPMeds    4061   95.82   53 1.25     0     0  factor      2
## 4    totChol       0    0.00   50 1.18     0     0 numeric    248
## 5 cigsPerDay    2144   50.59   29 0.68     0     0 numeric     33
## 6        BMI       0    0.00   19 0.45     0     0 numeric   1363
```

Sono state modificate le variabili non importate correttamente.
Non è stata eliminata alcuna variabile perchè nessuna presenta più del 20/30% di valori mancanti.

Distribuzione del target

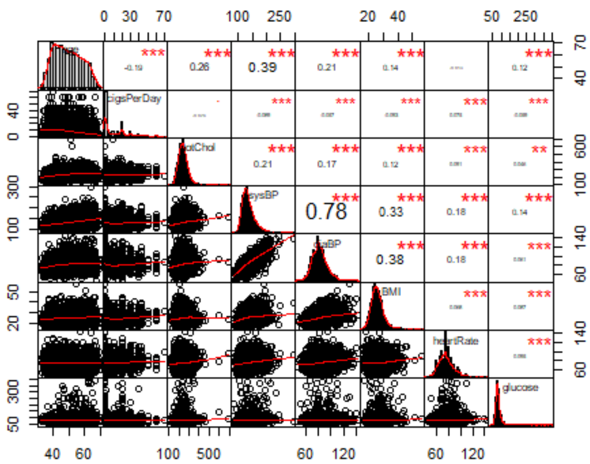
```
##         0         1
## 0.8480415 0.1519585
```



Solo il 15% circa dei soggetti (644) ha TenYearCHD pari a 1 (evento).
Si considera quindi il dataset non bilanciato.

Grafico correlazioni e conteggio valori mancanti

L'unica correlazione bivariata discretamente alta (0.78) è quella tra sysBP e diaBP.



Solo alcune variabili presentano valori mancanti.

```
##            male             age       education   currentSmoker      cigsPerDay
##               0               0             105               0              29
##          BPMeds prevalentStroke    prevalentHyp        diabetes         totChol
##              53               0               0               0              50
##           sysBP           diaBP             BMI       heartRate         glucose
##               0               0              19               1             388
##      TenYearCHD
##               0
```

Imputazione

Sono stati imputati i valori mancanti delle covariate numeriche sfruttando la funzione mice, mentre quelli relativi ai fattori con il comando impute.

Collinearità

```
## [1] "sysBP"
```

X1 <fctr>	Row <fctr>	Column <fctr>	Chi.Square <dbl>	df <int>	p.value <dbl>	n <int>	u1 <dbl>	u2 <dbl>	nMinu1u2 <dbl>	Chi.Square.norm <dbl>
1	male	education	88.440	3	0.000	4238	1	3	4238	0.0208683259772862624015221655326968
2	male	currentSmoker	164.673	1	0.000	4238	1	1	4238	0.0388561831377291777567251074287924
3	male	BPMeds	10.650	1	0.001	4238	1	1	4238	0.0025130798654163940229755436206460
4	male	prevalentStroke	0.009	1	0.926	4238	1	1	4238	0.0000020554153715847714378132243818
5	male	prevalentHyp	0.098	1	0.755	4238	1	1	4238	0.0000230227720708079624340955393613
6	male	diabetes	0.855	1	0.355	4238	1	1	4238	0.0002017027493116030280751599379485
7	education	currentSmoker	18.764	3	0.000	4238	3	1	4238	0.0044274676270551127296726257043247
8	education	BPMeds	0.726	3	0.867	4238	3	1	4238	0.0001713630754807319407032745184338
9	education	prevalentStroke	4.778	3	0.189	4238	3	1	4238	0.0011274117722798228120179020450564
10	education	prevalentHyp	33.565	3	0.000	4238	3	1	4238	0.0079199166273846606511899182123670
11	education	diabetes	9.980	3	0.019	4238	3	1	4238	0.0023548975509290158425468320047003
12	currentSmoker	BPMeds	9.345	1	0.002	4238	1	1	4238	0.0022050786779875615646606679121078
13	currentSmoker	prevalentStroke	3.790	1	0.052	4238	1	1	4238	0.0008943849888956820507954215138113
14	currentSmoker	prevalentHyp	44.743	1	0.000	4238	1	1	4238	0.0105575110801111610864255752062491
15	currentSmoker	diabetes	7.765	1	0.005	4238	1	1	4238	0.0018322171827976484626693265411745
16	BPMeds	prevalentStroke	47.141	1	0.000	4238	1	1	4238	0.0111233013893734748039898718730001
17	BPMeds	prevalentHyp	280.316	1	0.000	4238	1	1	4238	0.0661434808067117713248350696630951
18	BPMeds	diabetes	9.351	1	0.002	4238	1	1	4238	0.0022063675856389729570961044657906
19	prevalentStroke	prevalentHyp	21.666	1	0.000	4238	1	1	4238	0.0051122548115705114188389579510385
20	prevalentStroke	diabetes	0.000	1	1.000	4238	1	1	4238	0.0000000000000000000000000000515073
21	prevalentHyp	diabetes	24.606	1	0.000	4238	1	1	4238	0.0058060837940963998249355171310526

Per quanto riguarda le covariate numeriche, quella che causa collinearità è sysBP (come già osservato), mentre le coppie di fattori non restituiscono alcun Chi-quadro normalizzato maggiore di 0.9.

SysBP non viene eliminata poichè la correlazione non è eccessivamente elevata, il pre-processing è diverso per ogni modello e verrà fatto in seguito.

Zero variance / near zero variance

```
##                 freqRatio percentUnique zeroVar   nzv
## y                5.580745    0.04719207   FALSE FALSE
## age              1.049451    0.92024540   FALSE FALSE
## cigsPerDay       2.929444    0.77866918   FALSE FALSE
## totChol          1.197183    5.85181689   FALSE FALSE
## sysBP            1.049020    5.52147239   FALSE FALSE
## diaBP            1.723684    3.44502124   FALSE FALSE
## BMI              1.000000   32.16139689   FALSE FALSE
## heartRate        1.462338    1.72251062   FALSE FALSE
## glucose          1.152174    3.37423313   FALSE FALSE
## male             1.329852    0.04719207   FALSE FALSE
## education        1.456504    0.09438414   FALSE FALSE
## currentSmoker    1.023878    0.04719207   FALSE FALSE
## BPMeds          33.177419    0.04719207   FALSE  TRUE
## prevalentStroke 168.520000    0.04719207   FALSE  TRUE
## prevalentHyp     2.220365    0.04719207   FALSE FALSE
## diabetes        37.880734    0.04719207   FALSE  TRUE
```

Nessuna variabile ha varianza pari a 0, mentre BPMeds, prevalentStroke e diabetes soffrono di near zero variance.
Anche in questo caso, il relativo pre-processing verrà fatto in seguito.

Divisione training dataset e testing dataset

Il dataset di partenza viene suddiviso in training, test e score (campionamento stratificato).

In seguito, vengono fittati diversi modelli sul dataset di training.
Ognuno viene tunato in modo da massimizzare la specificity, in quanto l'evento di interesse è il verificarsi della malattia, quindi si vogliono massimizzare i TP e minimizzare i FN, ma R considera come evento la classe del target più frequente (sani).
Inoltre, ogni volta viene applicato un diverso pre-processing in base al modello che si considera.

Glm senza model selection

```
ctrl <- trainControl(method="cv", number=10, classProbs = TRUE,
summaryFunction=twoClassSummary)
glm <- train(r~., data=train_imputed, method="glm", metric="Spec",
          trControl=ctrl, tuneLength=5, trace=TRUE, na.action=na.pass,
preProcess=c("corr", "nzv", "BoxCox"))

##   ROC        Sens      Spec
##   0.7211134  0.995605  0.05165165

##             Reference
## Prediction    r0   r1
##           r0 84.4 14.4
##           r1  0.4  0.8
##
##  Accuracy (average) : 0.8522
```

Nonostante l'accuracy sia abbastanza elevata (0.85 circa), la percentuale di FN è piuttosto alta (specificity=0.05 circa).

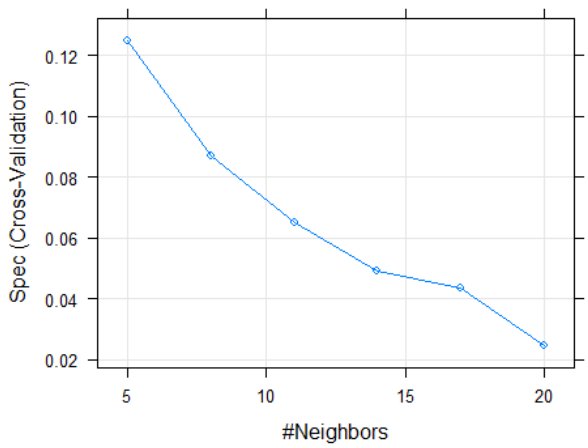
Knn con model selection (da modello logistico)

```
glm2 <- glm(y~., data=train_imputed, family=binomial(link="logit"))

## Model:
## y ~ age + cigsPerDay + totChol + sysBP + diaBP + BMI + heartRate +
##     glucose + male + education + currentSmoker + BPMeds + prevalentStroke +
##     prevalentHyp + diabetes
##                 Df Deviance    AIC    LRT            Pr(>Chi)
## <none>              1803.3 1839.3
## age              1   1861.7 1895.7 58.459 0.00000000000002076 ***
## cigsPerDay       1   1806.8 1840.8  3.531           0.0602245 .
## totChol          1   1806.7 1840.7  3.398           0.0652937 .
## sysBP            1   1815.6 1849.6 12.297           0.0004538 ***
## diaBP            1   1803.7 1837.7  0.444           0.5049995
## BMI              1   1803.3 1837.3  0.086           0.7691767
## heartRate        1   1803.3 1837.3  0.084           0.7721974
## glucose          1   1807.9 1841.9  4.642           0.0312040 *
## male             1   1816.2 1850.2 12.895           0.0003294 ***
## education        3   1806.7 1836.7  3.469           0.3248273
## currentSmoker    1   1804.0 1838.0  0.687           0.4070969
## BPMeds           1   1804.2 1838.2  0.935           0.3334884
## prevalentStroke  1   1807.7 1841.7  4.455           0.0348047 *
## prevalentHyp     1   1804.2 1838.2  0.938           0.3328447
## diabetes         1   1803.8 1837.8  0.490           0.4840745
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

ctrl <- trainControl(method="cv", number=10, classProbs = TRUE,
summaryFunction=twoClassSummary)
grid <- expand.grid(k=seq(5,20,3))
knn <-
train(r~age+cigsPerDay+totChol+sysBP+glucose+male+prevalentStroke,data=train_imputed,
method="knn", metric="Spec",
          trControl=ctrl, tuneLength=5, na.action=na.pass,tuneGrid=grid,
preProcess=c("scale", "corr", "nzv"))

##   k   ROC        Sens       Spec
##    5  0.6465292  0.9604615  0.12492492
##    8  0.6614651  0.9746222  0.08701201
##   11  0.6772318  0.9868173  0.06516517
##   14  0.6789196  0.9897441  0.04894895
##   17  0.6800711  0.9951100  0.04354354
##   20  0.6783720  0.9941320  0.02447447
##
## Spec was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```



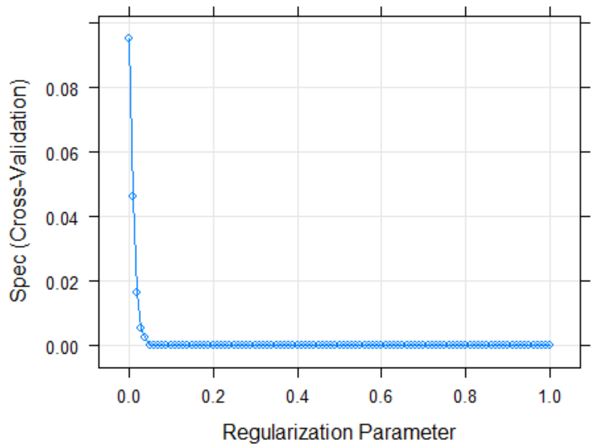

```
##           Reference
## Prediction   r0   r1
##         r0 81.5 13.3
##         r1  3.4  1.9
##
##  Accuracy (average) : 0.8336
```

K=5 massimizza la specificity (0.12 circa), mentre l'accuracy è pari a 0.83 circa.

Lasso

```
ctrl <- trainControl(method="cv", number=10, classProbs = TRUE,
summaryFunction=twoClassSummary)
grid <- expand.grid(.alpha=1, .lambda=seq(0, 1, by = 0.01))
lasso <- train(r~., data=train_imputed, method="glmnet", metric="Spec", trControl=ctrl,
tuneLength=5, na.action=na.pass,
            tuneGrid=grid)

##   lambda  ROC        Sens       Spec
##   0.00    0.7265227  0.9907269  0.095270270
##   0.01    0.7267783  0.9975586  0.046246246
##   0.02    0.7214943  0.9995122  0.016291291
##   0.03    0.7142528  1.0000000  0.005405405
##   0.04    0.7133991  1.0000000  0.002702703
##   0.05    0.7128220  1.0000000  0.000000000
##   0.06    0.7112470  1.0000000  0.000000000
##   0.07    0.7071104  1.0000000  0.000000000
##   0.08    0.6939221  1.0000000  0.000000000
##   0.09    0.5246598  1.0000000  0.000000000
##   0.10    0.5000000  1.0000000  0.000000000
##   0.11    0.5000000  1.0000000  0.000000000
##   0.12    0.5000000  1.0000000  0.000000000
## Tuning parameter 'alpha' was held constant at a value of 1
## Spec was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 1 and lambda = 0.
```



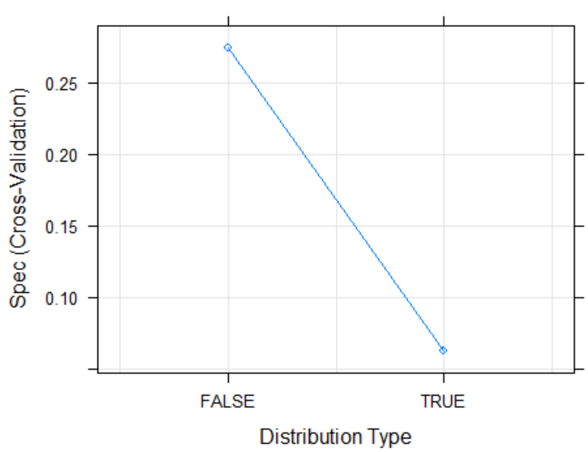

```
##           Reference
## Prediction    r0   r1
##         r0 84.0 13.7
##         r1  0.8  1.4
##
##  Accuracy (average) : 0.8547
```

Lambda=0 massimizza la specificity (0.09 circa), mentre l'accuracy è pari a 0.85 circa.

Naive Bayes

```
ctrl <- trainControl(method="cv", number=10, classProbs = TRUE,
summaryFunction=twoClassSummary)
naivebayes <- train(r~.,data=train_imputed, method="naive_bayes", metric="Spec",
            trControl=ctrl, tuneLength=5, na.action=na.pass, preProcess=c("corr", "nzv"))

##   usekernel  ROC        Sens       Spec
##   FALSE      0.7150953  0.9033668  0.27462462
##    TRUE      0.7076367  0.9863343  0.06238739
##
## Tuning parameter 'laplace' was held constant at a value of 0
## Tuning parameter 'adjust' was held constant at a value of 1
## Spec was used to select the optimal model using the largest value.
## The final values used for the model were laplace = 0, usekernel = FALSE and adjust = 1.
```




```
##           Reference
## Prediction    r0   r1
```

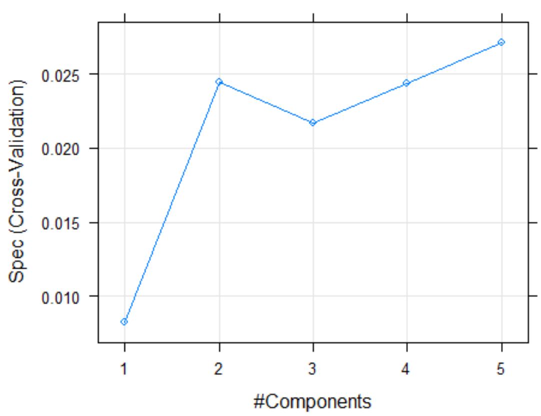
```
##         r0 76.6 11.0
##         r1  8.2  4.2
##
##  Accuracy (average) : 0.8079
```

Usekernel=FALSE massimizza la specificity (0.27 circa), mentre l'accuracy è pari a 0.81 circa.

Pls

```
Control <- trainControl(method="cv",number=10, classProbs = TRUE,
summaryFunction=twoClassSummary)
pls <- train(r~.,data=train_imputed, method="pls", metric="Spec",trControl=Control,
tuneLength=5)

##   ncomp  ROC        Sens       Spec
##   1      0.6643055  0.9995122  0.008183183
##   2      0.6889822  0.9980488  0.024474474
##   3      0.7126952  0.9995122  0.021696697
##   4      0.7186795  0.9980488  0.024399399
##   5      0.7218648  0.9980488  0.027177177
##
## Spec was used to select the optimal model using the largest value.
## The final value used for the model was ncomp = 5.
```




```
##           Reference
## Prediction   r0   r1
##         r0 84.6 14.8
##         r1  0.2  0.4
##
##  Accuracy (average) : 0.8506
```

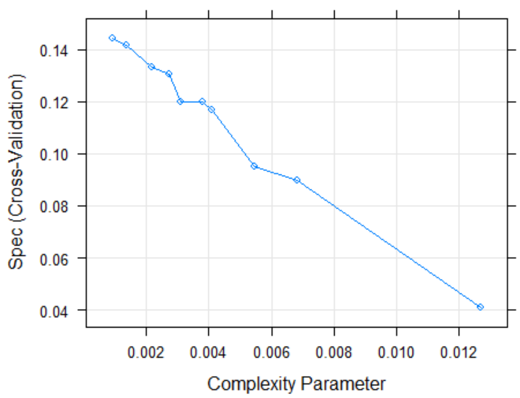
Ncomp=5 massimizza la specificity (0.03 circa), mentre l'accuracy è pari a 0.85 circa.

Tree

```
cvCtrl <- trainControl(method="cv", number=10, search="grid", classProbs = TRUE,
summaryFunction=twoClassSummary, savePredictions=TRUE)
tree <- train(r~.,data=train_imputed, method="rpart", metric="Spec",tuneLength=10,
trControl=cvCtrl, na.action=na.pass)

##   cp            ROC        Sens       Spec
##   0.0009082652  0.6295340  0.9253228  0.14444444
##   0.0013623978  0.6476412  0.9336155  0.14166667
##   0.0021798365  0.6729780  0.9409374  0.13333333
```

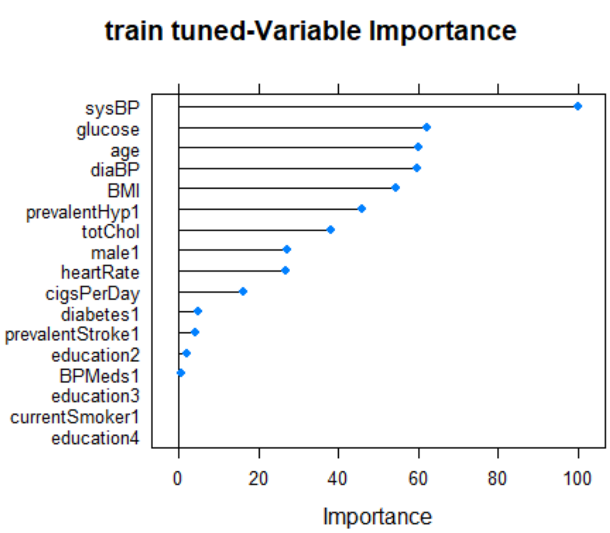
```
##    0.0027247956  0.6710081  0.9424008  0.13055556
##    0.0031140522  0.6468250  0.9560665  0.11966967
##    0.0038147139  0.6450320  0.9565543  0.11966967
##    0.0040871935  0.6498376  0.9580177  0.11696697
##    0.0054495913  0.6341489  0.9760808  0.09512012
##    0.0068119891  0.6347224  0.9770564  0.08971471
##    0.0127157130  0.5857075  0.9882807  0.04076577
##
## Spec was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.0009082652.
```




```
##            Reference
## Prediction    r0   r1
##         r0 78.5 13.0
##         r1  6.3  2.2
##
##  Accuracy (average) : 0.8067
```

Cp=0.0009082652 massimizza la specificity (0.14 circa), mentre l'accuracy è pari a 0.81 circa.

Importanza variabili albero



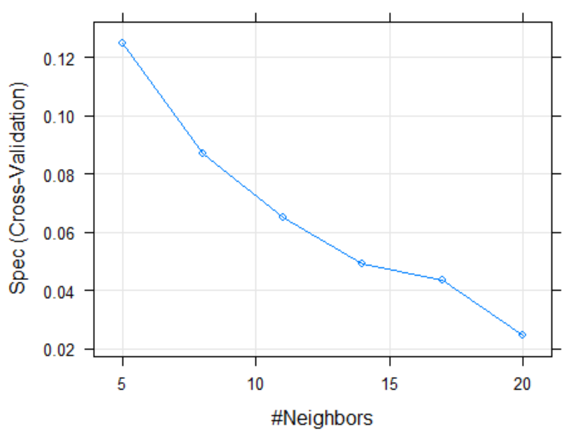

```
##                    Overall
## sysBP             100.0000
## glucose            62.2785
## age                60.1564
## diaBP              59.7399
## BMI                54.5404
## prevalentHyp1      45.9992
## totChol            38.2383
## male1              27.2910
## heartRate          26.6505
## cigsPerDay         16.0563
## diabetes1           4.6751
## prevalentStroke1    4.1606
## education2          2.1420
## BPMeds1             0.6515
## education3          0.0000
## education4          0.0000
## currentSmoker1      0.0000
```

Si considerano come importanti le variabili la cui importanza relativa è superiore al 20% di quella più importante (sysBP).
Si escludono quindi cigsPerDay, diabetes, prevalentStroke, education, BPMeds e currentSmoker.

Knn con model selection (da albero)

```
ctrl <- trainControl(method="cv", number=10, classProbs = TRUE,
summaryFunction=twoClassSummary)
grid <- expand.grid(k=seq(5,20,3))
knn2 <- train(r~., data=train_tree, method="knn", metric="Spec", trControl=ctrl,
tuneLength=5, na.action=na.pass,
              tuneGrid=grid, preProcess=c("scale", "corr", "nzv"))

##    k   ROC        Sens       Spec
##    5  0.6026974  0.9585175  0.06509009
##    8  0.6202987  0.9755978  0.06246246
##   11  0.6451585  0.9897513  0.03521021
##   14  0.6593986  0.9902367  0.03528529
##   17  0.6632814  0.9936561  0.03265766
##   20  0.6723769  0.9941416  0.01358859
##
## Spec was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```




```
##           Reference
## Prediction   r0   r1
##         r0 81.5 13.3
##         r1  3.4  1.9
##
##  Accuracy (average) : 0.8336
```

K=5 massimizza la specificity (0.06 circa), mentre l'accuracy è pari a 0.83 circa.

Glm con model selection (da albero)

```
ctrl <- trainControl(method="cv", number=10, classProbs = TRUE,
summaryFunction=twoClassSummary)
glm2 <- train(r~., data=train_tree, method="glm", metric="Spec",
          trControl=ctrl, tuneLength=5, trace=TRUE, na.action=na.pass,
preProcess=c("corr", "nzv", "BoxCox"))

##    ROC        Sens       Spec
##    0.7187519  0.9970732  0.03543544
```

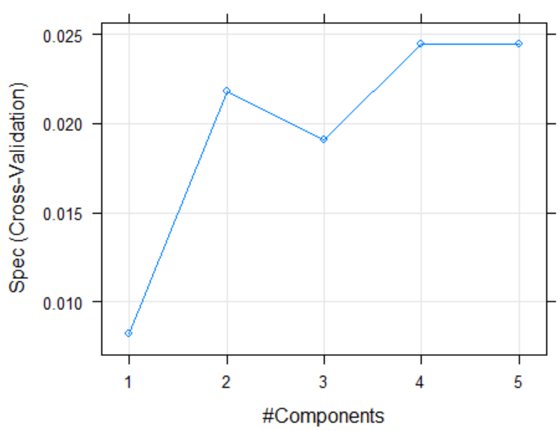
```
##            Reference
## Prediction    r0   r1
##         r0 84.6 14.7
##         r1  0.2  0.5
##
##  Accuracy (average) : 0.851
```

Anche in questo caso, nonostante l'accuracy sia abbastanza elevata (0.85 circa), la percentuale di FN è piuttosto alta (specificity=0.03 circa).

PLS con model selection (da albero)

```
Control <- trainControl(method="cv",number=10, classProbs = TRUE,
summaryFunction=twoClassSummary)
pls2 <- train(r~., data=train_tree, method="pls", metric="Spec",trControl=Control,
tuneLength=5)

##   ncomp  ROC        Sens       Spec
##   1      0.6633362  0.9995122  0.008183183
##   2      0.6864394  0.9980488  0.021771772
##   3      0.7043362  0.9995122  0.019069069
##   4      0.7097068  0.9980488  0.024474474
##   5      0.7119393  0.9980488  0.024474474
##
## Spec was used to select the optimal model using the largest value.
## The final value used for the model was ncomp = 4.
```



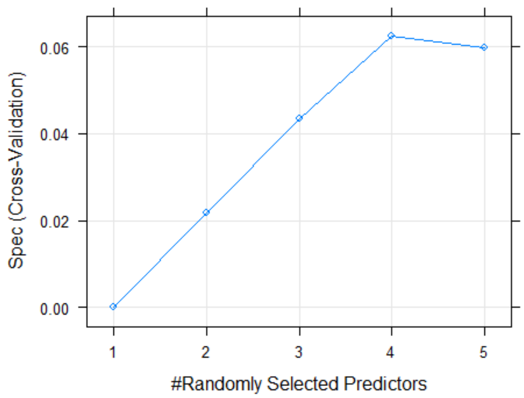

```
##            Reference
## Prediction    r0   r1
##         r0 84.6 14.8
##         r1  0.2  0.4
##
##  Accuracy (average) : 0.8502
```

Ncomp=4 massimizza la specificity (0.02 circa), mentre l'accuracy è pari a 0.85 circa.

Random forest

```
control <- trainControl(method="cv", number=10, search="grid",
summaryFunction=twoClassSummary, classProbs = TRUE)
tunegrid <- expand.grid(.mtry=c(1:5))
rf <- train(r~.,data=train_imputed, method="rf", metric="Spec",
            tuneGrid=tunegrid, ntree=250, trControl=control, na.action=na.pass)
```

```
##    mtry  ROC        Sens       Spec
##    1     0.6839798  1.0000000  0.00000000
##    2     0.6938821  0.9946270  0.02169670
##    3     0.6899173  0.9873051  0.04339339
##    4     0.6799142  0.9858417  0.06253754
##    5     0.6815047  0.9809613  0.05983483
##
## Spec was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```



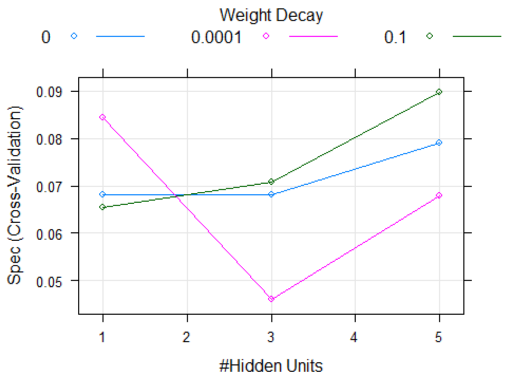

```
##           Reference
## Prediction   r0   r1
##         r0 83.6 14.2
##         r1  1.2  1.0
##
##  Accuracy (average) : 0.8456
```

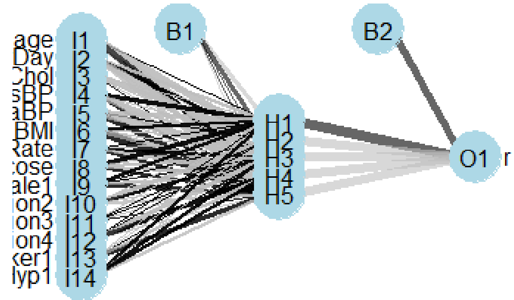
Mtry=4 massimizza la specificity (0.06 circa), mentre l'accuracy è pari a 0.85 circa.

Rete neurale

```
ctrl <- trainControl(method="cv", number=10, search="grid",
summaryFunction=twoClassSummary, classProbs=T)
rete <- train(r~., data=train_imputed, method="nnet",
preProcess=c("range","corr","nzv"),metric="Spec", trControl=ctrl,
              trace=TRUE, maxit=300,na.action=na.pass)

##    size  decay   ROC        Sens       Spec
##    1     0.0000  0.6996444  0.9926758  0.06816817
##    1     0.0001  0.7275263  0.9921879  0.08438438
##    1     0.1000  0.7276079  0.9960928  0.06539039
##    3     0.0000  0.6764977  0.9838905  0.06816817
##    3     0.0001  0.6860053  0.9853491  0.04602102
##    3     0.1000  0.7025519  0.9892563  0.07079580
##    5     0.0000  0.6511845  0.9824199  0.07905405
##    5     0.0001  0.6542794  0.9717073  0.06786787
##    5     0.1000  0.6870414  0.9858393  0.08978979
##
## Spec was used to select the optimal model using the largest value.
## The final values used for the model were size = 5 and decay = 0.1.
```





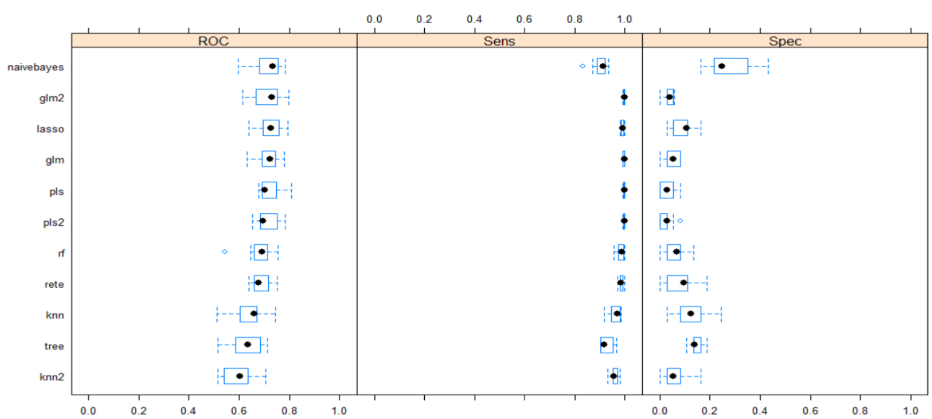

```
##            Reference
## Prediction    r0    r1
##           r0 83.6 13.8
##           r1  1.2   1.4
##
##  Accuracy (average) : 0.8498
```

Size=5 e decay=0.1 massimizza la specificity (0.09 circa), mentre l'accuracy è pari a 0.85 circa.

Si procede con il confronto dei modelli fittati utilizzando il dataset di validation.

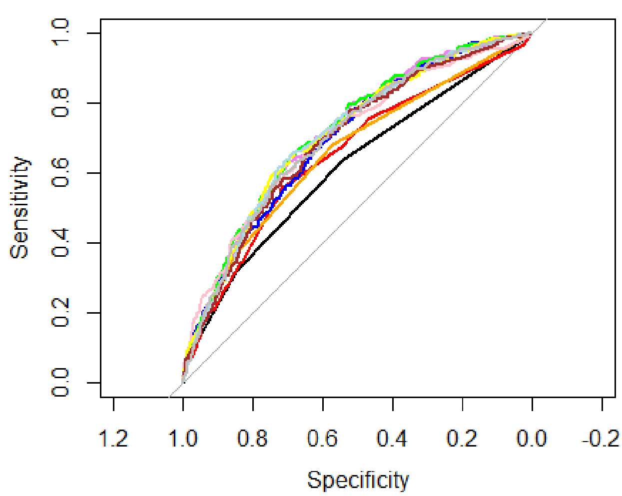
STEP2: ASSESSMENT

Comparazione risultati crossvalidati (no vero assessment)



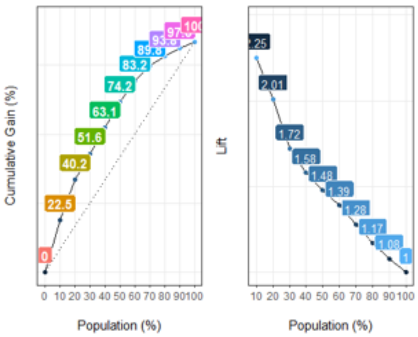
Il modello naivebayes è quello con specificity più elevata, ma è anche il più instabile (maggiore variabilità).

Curve ROC

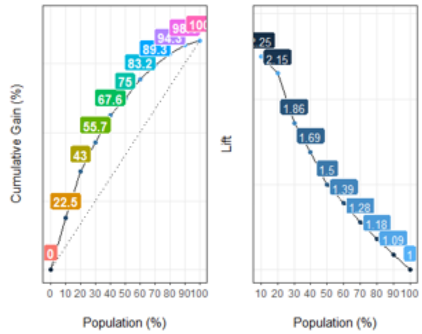


Poichè le curve ROC si sovrappongono, procediamo analizzando le curve lift dei modelli che sembrano avere ROC migliore.

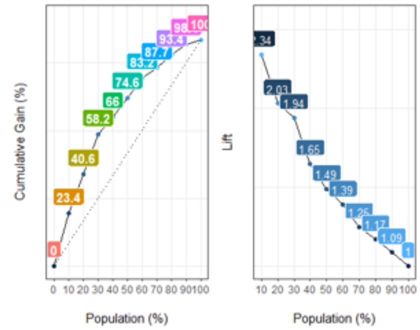
Lift



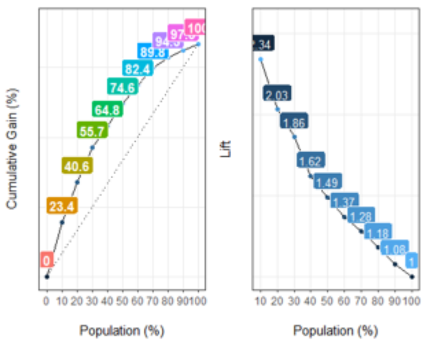
Naive bayes



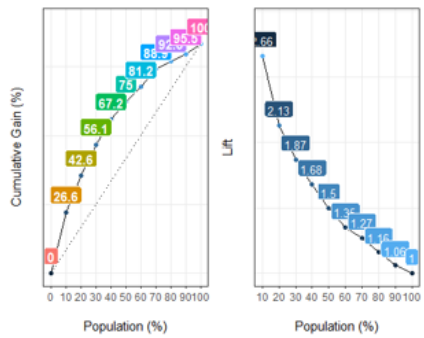
Lasso



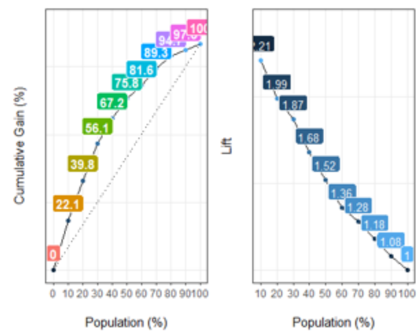
Pls



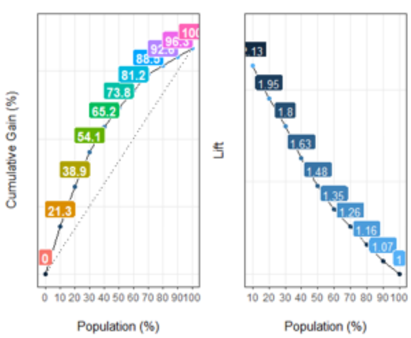
Rete neurale



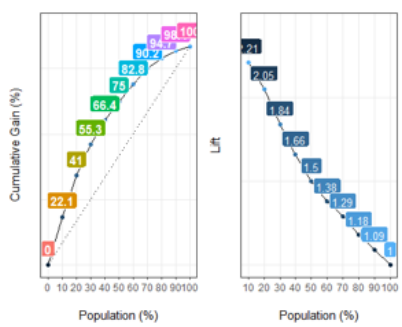
Random Forest



Pls 2



Glm 2



Glm 1

Tenendo conto del 20% della popolazione e osservando le curve lift, si sceglie come modello vincente il lasso (lift=2.15 e cumulative gain=43%).

In seguito, si fissa la soglia che massimizza la misura di interesse (specificity), si ricavano i valori previsti del target e si valuta la metrica classificativa scelta sulla matrice di confusione considerando i dati di validation.

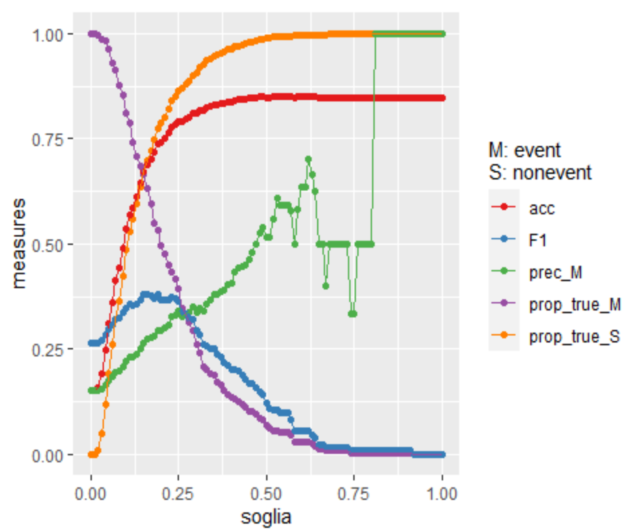
STEP3: SCELTA SOGLIA

Misure rispetto alle soglie

```
##    soglia prop_true_M prop_true_S true_M true_S fn_M
## 1    0.00   1.0000000 0.000000000    244      0    0
## 2    0.01   1.0000000 0.000000000    244      0    0
## 3    0.02   0.9959016 0.008791209    243     12    1
## 4    0.03   0.9877049 0.049816850    241     68    3
## 5    0.04   0.9836066 0.117948718    240    161    4
## 6    0.05   0.9631148 0.192673993    235    263    9
## 7    0.06   0.9303279 0.260073260    227    355   17
## 8    0.07   0.9139344 0.322344322    223    440   21
## 9    0.08   0.8770492 0.364102564    214    497   30
## 10   0.09   0.8524590 0.423443223    208    578   36
```

Come già detto, la metrica di interesse è la specificity (prop_true_S, dove S=sani).

Grafico con sensitivity, specificity e altre misure



La soglia che consente di avere una specificity soddisfacente e al tempo stesso una discreta sensibility è 0.2.

Predetti sul validation e matrice di confusione

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   r0   r1
##         r0 1073  123
##         r1  292  121
##
##                Accuracy : 0.7421
##                  95% CI : (0.72, 0.7633)
##     No Information Rate : 0.8484
##     P-Value [Acc > NIR] : 1
##
##
##             Sensitivity : 0.4959
```

```
##             Specificity : 0.7861
##
##        'Positive' Class : r1
```

Con la soglia considerata, si ottiene una specificity pari all'80% circa, una sensitivity pari al 50% circa e un'accuracy pari al 74% circa.

Poiché il modello è abbastanza soddisfacente, si procede con la classificazione di nuovi soggetti.

STEP4: SCORE NUOVE OSSERVAZIONI

Score dataset e previsione

```
##          r0         r1
## 1 0.8061843 0.19381566
## 2 0.9133995 0.08660045
## 3 0.8581781 0.14182194
## 4 0.7765300 0.22346997
## 5 0.8004077 0.19959228
## 6 0.7844270 0.21557298
```

```
##     age cigsPerDay totChol sysBP diaBP   BMI heartRate glucose male education
## 7    63          0     205 138.0    71 33.11        60      85    0         1
## 60   40         20     205 158.0   102 25.45        75      87    0         4
## 100  56         15     269 121.0    75 22.36        50      66    0         1
## 147  59          1     259 141.0    86 25.97        70      86    0         1
## 209  67          0     249 128.0    68 25.81        70      87    0         2
## 216  45         43     191 139.5    75 22.30        77      71    1         1
##     currentSmoker BPMeds prevalentStroke prevalentHyp diabetes   prob.r0
## 7               0      0               0            0        0 0.8061843
## 60              1      0               0            0        0 0.9133995
## 100             1      0               0            0        0 0.8581781
## 147             1      0               0            1        0 0.7765300
## 209             0      0               0            0        0 0.8004077
## 216             1      0               0            1        0 0.7844270
##        prob.r1 pred_y
## 7   0.19381566      S
## 60  0.08660045      S
## 100 0.14182194      S
## 147 0.22346997      M
## 209 0.19959228      S
## 216 0.21557298      M
```