Data Mining - Prova d'esame del 13.2.2020

Laboratorio: punteggi e soluzione

Punteggi

files	ACC	Percentuale	Punti
790430.txt	0.7681355	96.9%	8.7
802746.txt	0.7215701	19.3%	1.7
807498.txt	0.7379257	46.5%	4.2
807782.txt	0.7689051	98.2%	8.8
808162.txt	0.7562055	77%	6.9
808644.txt	0.7177218	12.9%	1.2
808693.txt	0.7719838	100%	9.0

Commenti

808162 | .txt non conforme, .html commentato |
808693 | Manca il file .html, solo .Rmd |

Soluzione

E' stato proposto un dataset reale, il dataset *wine* dal *repository* di dati di UCI Machine Learning. L'obiettivo è prevedere la qualità del vino, di cui ci sono 7 valori (numeri interi 3-9). Questo problema è stato trasformato in un problema di classificazione binaria per prevedere se un vino è "Good" (valori $6, 7, 8, 9$) o "Bad" (valori $3, 4, 5$).

Il modello di benchmark poteva essere facilmente migliorato selezionando il valore di K attraverso il metodo della convalida incrociata:

```
library(kknn)
( K = train.kknn(y ~ ., data=tr, kmax=100)$best.parameters$k )
```

```
[1] 46
```

```
fit = kknn(y ~ ., tr, te, , k = K)
yhat = fit$fitted.values
( ACC = mean( yhat == y.te ) )
```

```
[1] 0.7388878
```

oppure considerando un semplice modello logistico

```
fit = glm(y ~ ., tr, family=binomial())
phat = predict(fit,te, type="response")
yhat = ifelse(phat > 0.5, "Good","Bad")
( ACC = mean( yhat == y.te ) )
```

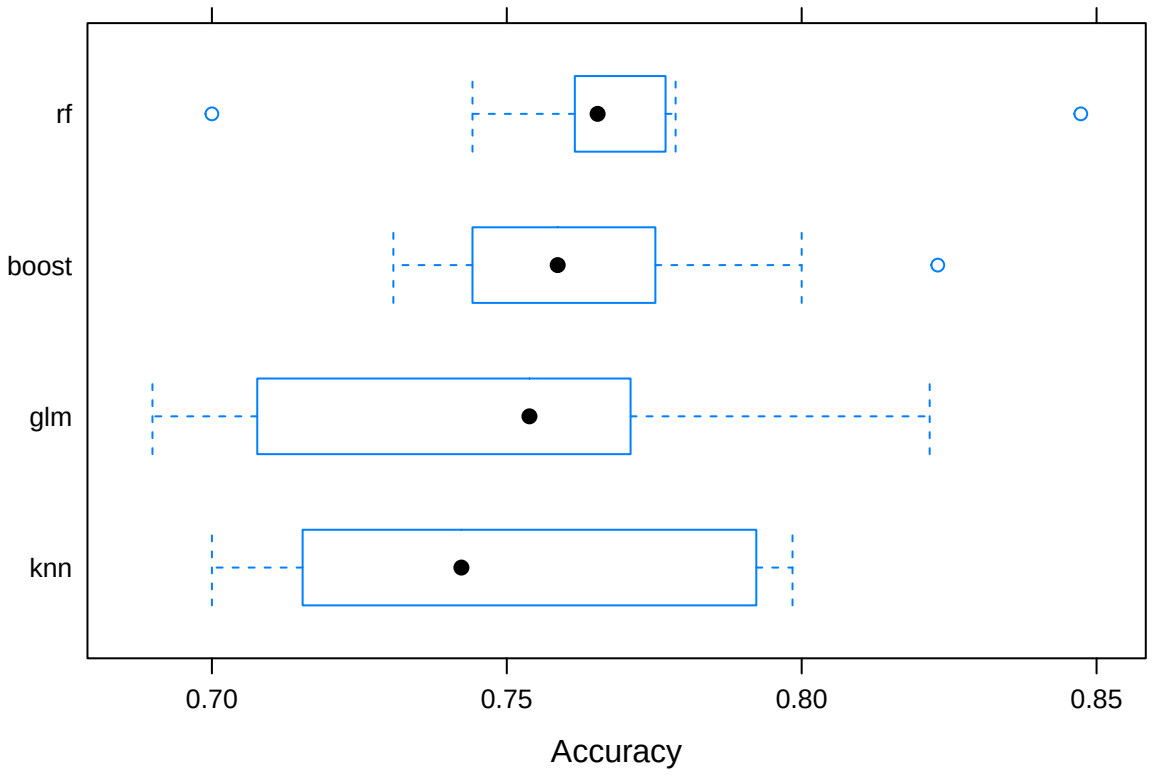
```
[1] 0.7402348
```

Un miglioramento significativo si poteva ottenere con il metodo Random Forest:

```
library(randomForest)
set.seed(123)
fit = randomForest(y ~ ., tr)
yhat <- predict(fit,newdata=te,type="response")
( ACC = mean( yhat == y.te ) )
```

```
[1] 0.7690976
```

Infine, adottando l'approccio "forza bruta" si potevano confrontare i diversi modelli con il metodo della convalida incrociata:




```
$knn
[1] 0.7192611

$glm
[1] 0.7402348

$rf
[1] 0.7694824

$boost
[1] 0.7565903
```