Otsustuspuud

praktikum

Tänast praktikumi alustame loengus käsitletud näiteanalüüsi kordamisega – proovime prognoosida tudengi sugu. Selleks kasutame TÜ (arstiteaduskonna) tudengite andmestikku:

```
load(url("http://www-1.ms.ut.ee/mart/kvalitatiivsed/tudengid.RData"))
tudengid[1:3,]
attach(tudengid)
set.seed(1)
```

Otsustuspuude koostamiseks vajame lisamoodulit rpart:

```
library(rpart)
```

ning samuti võiksime tunnuse sugu (1-naine; 2-mees) muuta paremini loetavaks

```
table(sugu)
suguF=factor(sugu, levels=c(1,2), labels=c("naine", "mees"))
table(suguF)
table(sugu, suguF)
```

Nüüd on saabunud sobiv hetk koostada esimese otsustuspuu:

```
m1=rpart(suguF~factor(olu)+pikkus)
plot(m1)
text(m1)
```

Või proovi ka saadud puud sellisel viisil välja joonistada:

```
plot(m1)
text(m1, use.n=T)
```

Sõltuvalt kasutatavast arvutist või ekraanist võid parema tulemuse saada käsuga

```
plot(m1, margin=0.1)
text(m1, use.n=T)
```

Mida lisaparameeter margin= teeb?

Loodud puid tuleb vahel pügada. Uurime ka meie, kas puul on vesiharusid ehk midagi liigset:

```
printcp(m1)
```

Näeme, et xerror – suhteline viga uue vaatluse prognoosimisel – tuleb väikseim karistusliikme (complexity parameeter) 0.013514 korral. Seega pügame saadud puud kasutades valitud kasitusliiget:

```
m2=prune(m1, cp=0.0136)
plot(m2)
text(m2)
```

Näide 2

Prognoosi tudengi tervislikku seisundit:

```
m3=rpart(factor(tervis)~., data=tudengid)
printcp(m3)
```

Millise mudelini jõuad?

Näide 3

Prognoosime seda, kas tudeng on alakaaluline (KMI<18,5); ülekaaluline (KMI>25) või normaalkaalus:

```
KMI = kaal/(pikkus/100)**2
kaaluklass=factor(
      cut(KMI, c(0, 18.5, 25, Inf)),
      labels=c("alakaaluline", "norm", "ülekaaluline"))

table(kaaluklass)
m2=rpart(kaaluklass~., data=tudengid)
```

Kas saadud mudel on loogiline?

Ülesanne

Prognoosi seda, palju tudeng õlut joob (tunnus olu). Millist otsustuspuuni lõpuks jõuad? Kui hästi saadud puu võiks prognoosida?

Proovimiseks

```
set.seed(1)
tudengid2=na.omit(data.frame(pereF=factor(perekonnaseis), pikkus, kaal,
                             viin, vanus, factor(tervis), factor(olu), sugu))
n=nrow(tudengid2)
abi=sample(n, 0.7*n)
treening=tudengid2[abi,]
test=tudengid2[-abi,]

mudel1=rpart(pereF~. , data=treening)

library(randomForest)
mudel2=randomForest(pereF~. , data=treening, ntree=1000, mtry=5)

prog1=predict(mudel1, newdata=test, type="class")
prog2=predict(mudel2, newdata=test, type="class")

table(test$pereF, prog1)
table(test$pereF, prog2)

mean(prog1==test$pereF)
mean(prog2==test$pereF)
```