8. loeng

Otsustuspuud

Diskriminantanalüüs (klassifitseerimine)

Soovime määrata, millisesse klassi (või alampopulatsiooni) uus objekt kuulub. Ülesande lahendamiseks võime kasutada valimit, kuhu kuuluvad objektid on juba eksperdi poolt õigetesse klassidesse paigaldatud.

Näiteks soovime määrata liblikate liiki nende tiivapikkuse ja värvi järgi. Kasutades varem püütud liblikaid (kelle liik on putukatundja poolt määratud) peab diskriminantanalüüs koostama määramiseeskirja mida saab kasutada ka uue liblika liigi määramiseks (ilma, et me eksperti uuesti tülitaksime).

Kui jagame objekte kahte klassi...

Ühte klassifitseerimiseks sobilikku meetodit me juba tunneme –

logistilist regressiooni

saab suurepäraselt kasutada jagamaks vaatluseid kahte gruppi.

Aga mis saab siis, kui tahame jaotada vaatluseid kolme, nelja või *k*-sse klassi?

Ning mis saab siis, kui soovime jõuda mõne lihtsama ja arusaadavama klassifitseerimiseeskirjani?

Logistilise regressiooni mudel:

$$P(Adenoma) = \frac{\exp(0.74 + 165.42I(Fluidlevel = Fluid) + 48(T2 = separation) + \ldots)}{1 + \exp(0.74 + 165.42I(Fluidlevel = Fluid) + 48(T2 = separation) + \ldots)}$$

Otsustuspuu (decision tree):

Tulemust sageli märksa mugavam kasutada;

Mõne tunnuse väärtuse mõõtmine võib osutuda mittevajalikuks (mõnel objektil) – see aga võib tähendada kokkuhoidu nii rahas kui ajas

Klassifitseerimistäpsus sageli peaaegu sama hea kui logistilisel regressioonil

Saab kasutada ka siis, kui jagame vaatluseid 3, 4, ... gruppi

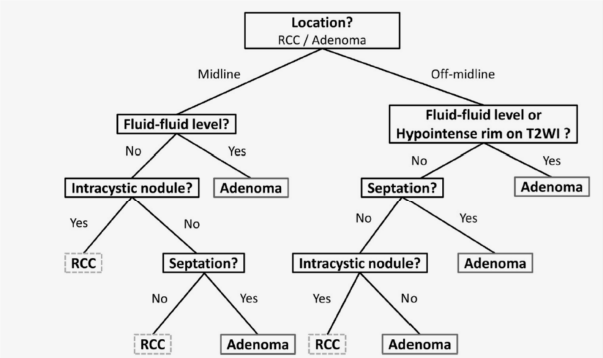


FIG 3. Diagnostic decision tree for the differentiation of cystic pituitary adenomas and RCCs using MR imaging.

Kuidas luuakse otsustuspuid?

Võimalus 1 – kasutades entroopiat

$$H(Y) = \mathrm{E}\left(\log_2\left(\frac{1}{P(Y)}\right)\right) = \sum_i^J p_i \log_2\left(\frac{1}{p_i}\right) = -\sum_i^J p_i \log_2\left(p_i\right)$$

y_i	A	B
P($Y=y_i$)	0,5	0,5
P($Y=y_i$)	0,1	0,9

$H(Y) = 1$

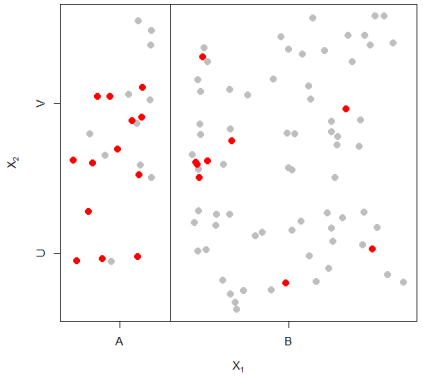
$H(Y) = 0{,}469$

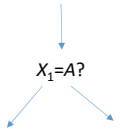
$$H(Y|X) = H(Y|X=1)P(X=1) + H(Y|X=2)P(X=2)$$

Kuidas luuakse otsustuspuid?

Võimalus 1 – kasutades entroopiat

$$H(Y) = \mathrm{E}\left(\log_2\left(\frac{1}{P(Y)}\right)\right) = \sum_i^J p_i \log_2\left(\frac{1}{p_i}\right) = -\sum_i^J p_i \log_2\left(p_i\right)$$



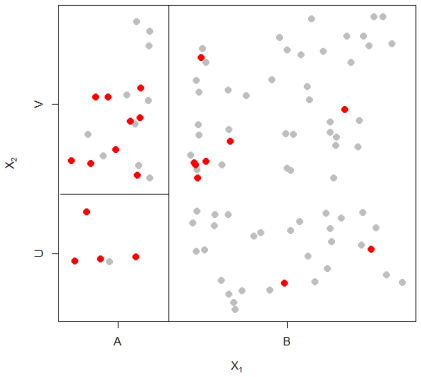


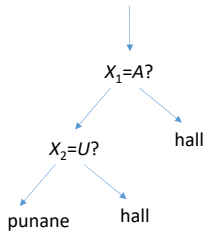
$H(Y|X_1) < H(Y|X_2)$

Kuidas luuakse otsustuspuid?

Võimalus 1 – kasutades entroopiat

$$H(Y) = \mathrm{E}\left(\log_2\left(\frac{1}{P(Y)}\right)\right) = \sum_i^J p_i \log_2\left(\frac{1}{p_i}\right) = -\sum_i^J p_i \log_2\left(p_i\right)$$





Alternatiiv

- Kasuta Gini indeksit

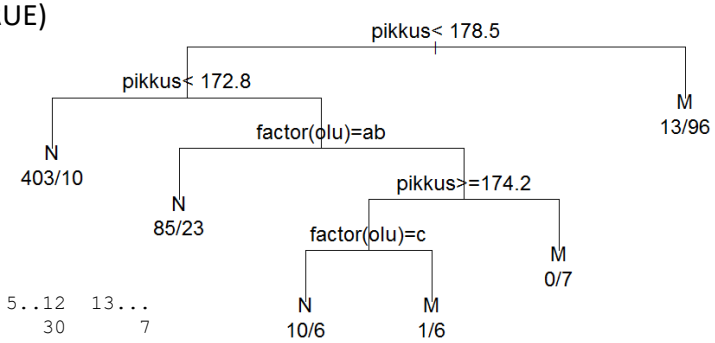
$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}), \qquad \hat{p}_{mk} = \frac{\#\{y = k | \mathbf{x} \in R_m\}}{|R_m|}.$$

- Pidevate uuritavate tunnuste korral jääkide ruutude summat

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2.$$

Otsusepuud R'is. Samm 1 – esialgne puu

```
library(rpart)
m1=rpart(factor(sugu)~pikkus+factor(olu))
plot(m1, margin=0.1, uniform=TRUE)
text(m1, use.n=TRUE)
```




```
> table(olu)
olu
     0   0..1   1..4  5..12  13...
   266    265     92     30      7
```

Otsusepuud R'is. Samm 2 – puu pügamine

(väldime ülesobitamist)

```
printcp(m1)

Classification tree:
rpart(formula = factor(sugu) ~ pikkus + factor(olu))

Variables actually used in tree construction:
[1] factor(olu) pikkus

Root node error: 148/660 = 0.22424

n= 660

        CP nsplit rel error  xerror     xstd
1 0.560811      0   1.00000 1.00000 0.072399
2 0.027027      1   0.43919 0.48649 0.054115
3 0.013514      3   0.38514 0.47973 0.053784
4 0.010000      5   0.35811 0.47973 0.053784
```

Karistusparameetri (CP) määramine.

Vali selline CP väärtus mille korral on *xerror* (ristvalideerimisel saadav viga) väikseim või otsusta sarnaste *xerror*-väärtuste korral lihtsama mudeli kasuks.

Otsusepuud R'is. Samm 2 – puu pügamine

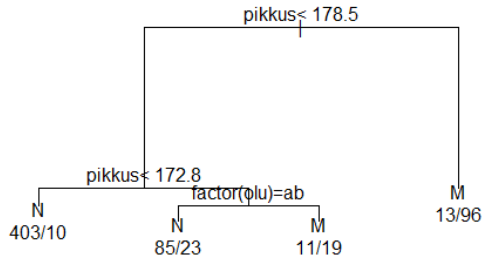
```
printcp(m1)

Classification tree:
rpart(formula = factor(sugu) ~ pikkus + factor(olu))

Variables actually used in tree construction:
[1] factor(olu) pikkus

Root node error: 148/660 = 0.22424

n= 660

        CP nsplit rel error  xerror     xstd
1 0.560811      0   1.00000 1.00000 0.072399
2 0.027027      1   0.43919 0.48649 0.054115
3 0.013514      3   0.38514 0.47973 0.053784
4 0.010000      5   0.35811 0.47973 0.053784
```

Karistusparameetri (CP) määramine.

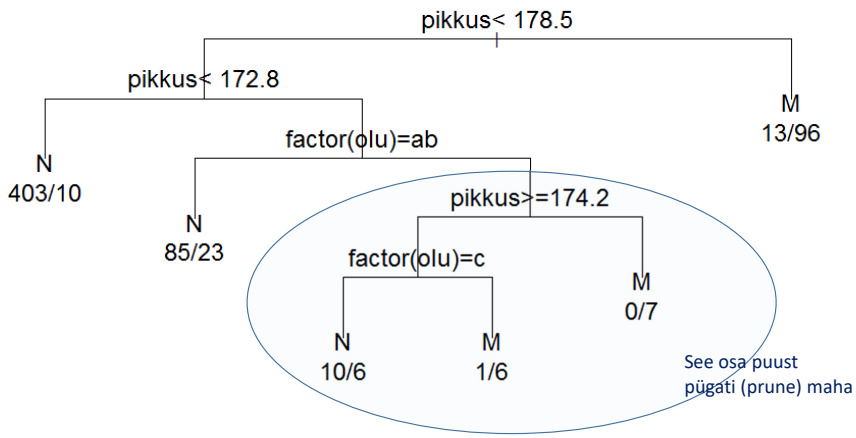
Vali selline CP väärtus mille korral on *xerror* (ristvalideerimisel saadav viga) väikseim või otsusta sarnaste *xerror*-väärtuste korral lihtsama mudeli kasuks.

Otsusepuud R'is. Samm 2 – puu pügamine

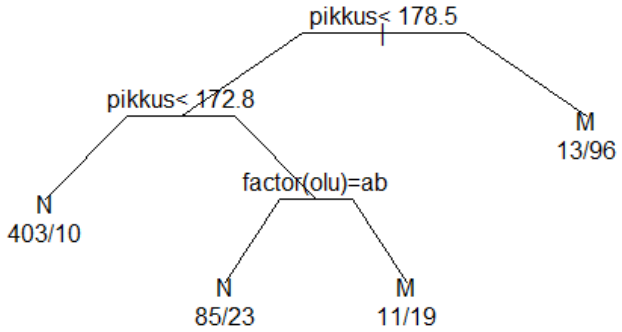
```
m2=prune(m1, cp=0.0136)
plot(m2, margin=0.2)
text(m2, use.n=TRUE)
```




```
        CP nsplit rel error  xerror     xstd
1 0.560811      0   1.00000 1.00000 0.072399
2 0.027027      1   0.43919 0.48649 0.054115
3 0.013514      3   0.38514 0.47973 0.053784
4 0.010000      5   0.35811 0.47973 0.053784
```




```
plot(m2, margin=0.2, branch=0.4, uniform=TRUE)
text(m2, use.n=TRUE)
```



Keerukamaid valikuid...

Kasuta hargnemiste loomisel entroopiat

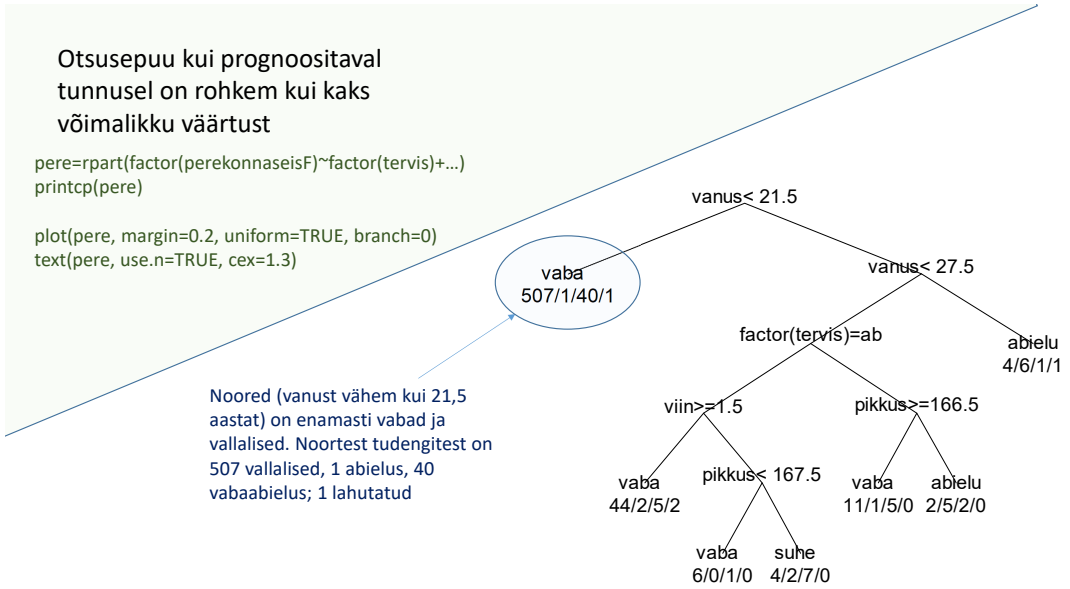
```
m1=rpart(factor(sugu)~pikkus+factor(olu), data=tudengid,
        parms=list(prior=c(0.5, 0.5), split="information"),
        loss=rbind(   c(0,   1),
                      c(0.5, 0)))
```

Algsetes andmetes oli mehi 3x vähem kui naisi. Kui tahame tulevikus prognoose leida olukorras, kus naisi ja mehi on võrdselt, siis peaksime seda näitama prior-parameetri abil (kui algsetes andmetes sattub mingisse alamgruppi 15 naist ja 10 meest, siis tasub sellesse gruppi sattuv tudeng prognoosida naiseks. Kui aga suurendame meeste arvu 3x, siis oleks sellesse alamgruppi kuuluv tudeng meheks prognoosida: 15 naist 30 mehe kohta…)

Kuidas karistame vigu?
Ridades on tegelikud väärtused, veergudes mudeli otsused.
Kui tunnuse sugu väärtused on järjekorras „naine“, „mees“, siis loeme naise meheks klassifitseerimist kaks korda tõsisemaks veaks kui ekslikult mehe naiseks pidamist.

Otsusepuu kui prognoositaval tunnusel on rohkem kui kaks võimalikku väärtust

```
pere=rpart(factor(perekonnaseisF)~factor(tervis)+…)
printcp(pere)

plot(pere, margin=0.2, uniform=TRUE, branch=0)
text(pere, use.n=TRUE, cex=1.3)
```

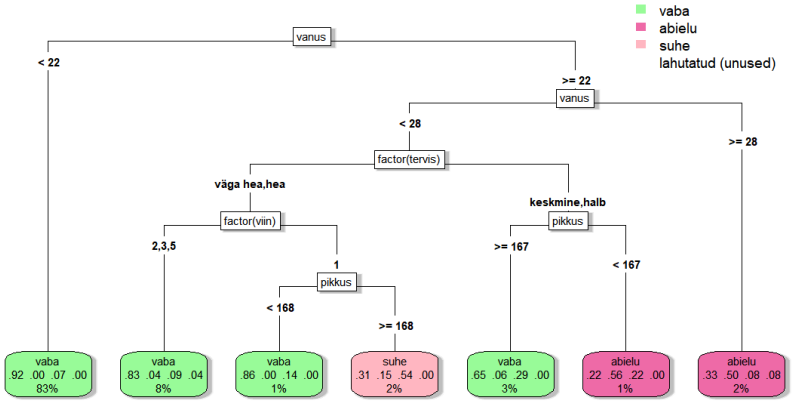


Noored (vanust vähem kui 21,5 aastat) on enamasti vabad ja vallalised. Noortest tudengitest on 507 vallalised, 1 abielus, 40 vabaabielus; 1 lahutatud

Prognoosimine otsustuspuu abil

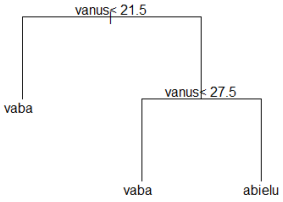
```
> predict(pere, newdata=data.frame(vanus=22, pikkus=161,
                  viin=1, tervis="väga hea"))

       vaba abielu      suhe lahutatud
1 0.8571429      0 0.1428571         0


> predict(pere, newdata=data.frame(vanus=22, pikkus=161,
                  viin=1, tervis="väga hea"), type="class")
   1
vaba
```

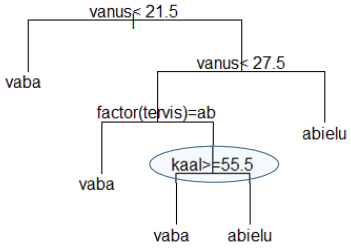
```
library(rpart.plot)
rpart.plot(pere, type=5)
```

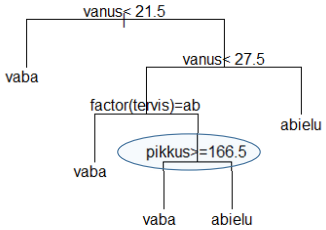


Palju puid = mets

Hinnatakse palju otsustuspuid, iga otsustuspuu paari juhuslikult valitud tunnuse abil. Hiljem kombineeritakse erinevate otsustuspuude prognoosid.







Palju puid = mets

kasutame tunnuseid kus on vähe puuduvaid väärtuseid

vajadusel muuda juba siin osad tunnused faktortunnusteks!

```
library(randomForest)
tudengid2=na.omit(data.frame(perekonnaseisF, pikkus, kaal, viin,
                  vanus, factor(tervis), factor(olu), sugu))

n=nrow(tudengid2)
abi=sample(n, 0.7*n)
treening=tudengid2[abi,]
test=tudengid2[-abi,]
```

jagame andmestiku treening ja testandmestikuks

```
mudel=randomForest(factor(perekonnaseisF)~. , data=treening,
                                                ntree=1000, mtry=4)
```

hindame metsa milles on 1000 puud.
Iga puu on tehtud kasutades 4 tunnust

```
prog=predict(mudel, newdata=test,type="class")
table(prog, test$perekonnaseis)
mean(prog==test$perekonnaseis)
```

Katsetame kui hästi „mets“ testandmete peal töötab

Palju puid = mets

Kolm tegelikult abielus olevat tudengit oleme ekslikult prognoosinud vallalisteks

```
> table(prog, test$perekonnaseis)

prog         vaba abielu suhe lahutatud
  vaba        178      3    2         0
  abielu        0      2    0         0
  suhe          1      0   10         0
  lahutatud     0      0    0         1

> mean(prog==test$perekonnaseis)
[1] 0.9695431
```

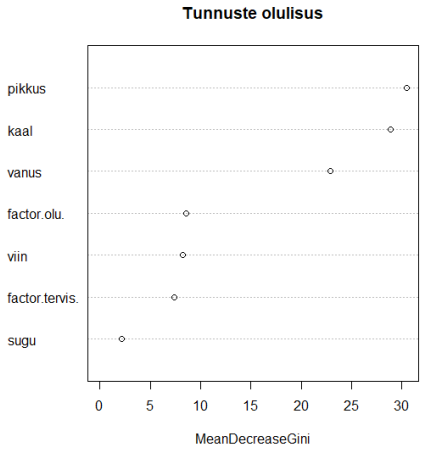
Suudame õigesti ära arvata 97% uute tudengite perekonnaseisu (uus tudeng - mudeli ehk metsa hindamiseks mittekasutatud tudeng)

Mida vajame selleks, et kasvaks hea mets?

Otsustuspuud vaadates näeme kohe, milliseid tunnuseid kasutatakse otsuse tegemiseks. Metsa puhul tuleb aga ise järgi uurida – milliseid tunnuseid siis ikkagi kasutati lõppotsuse tegemiseks ja kui tähtsat rolli nad mängisid…

```
> importance(mudel)
               MeanDecreaseGini
pikkus                30.451607
kaal                  28.909940
viin                   8.215171
vanus                 22.905772
factor.tervis.         7.362742
factor.olu.            8.606642
sugu                   2.166504
```

varImpPlot(mudel)



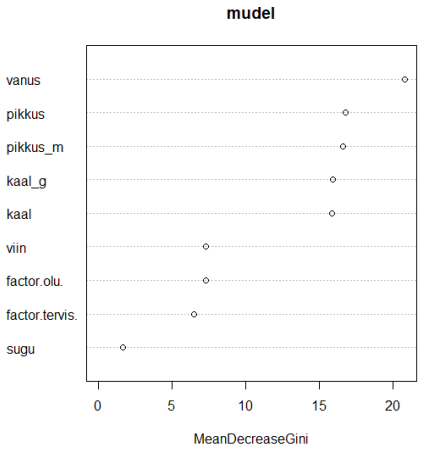
Mida vajame selleks, et kasvaks hea mets?

Samad andmed, aga nüüd on vanus tähtsam kui pikkus?

Tunnuste olulisuse tõlgendamisel tasub teadvustada, et tunnuse olulisus võib jaguneda tugevalt sõltuvate tunnuste vahel laiali.

```
> importance(mudel)
               MeanDecreaseGini
pikkus                16.773516
kaal                  15.871718
viin                   7.308804
vanus                 20.778859
factor.tervis.         6.473296
factor.olu.            7.269541
sugu                   1.652921
pikkus_m              16.625814
kaal_g                15.898106
```

varImpPlot(mudel)



Alternatiivid puudele ja metsale?

Hindame mitu logistilistilise regressioonanalüüsi mudelit?

```
mudel1=glm(1*(perekonnaseisF=="vaba")~vanus+ I(vanus**2)
mudel2=glm(1*(perekonnaseisF=="abielu")~vanus+ I(vanus**
mudel3=glm(1*(perekonnaseisF=="suhe")~vanus+ I(vanus**2)
mudel4=glm(1*(perekonnaseisF=="lahutatud")~vanus+ I(vanu
```

Probleem: prognooside tõenäosuste summa ei pruugi tulla 1

Vanus 25:

mudel 1: P(vaba|vanus=25)= 0,59
mudel 2: P(abielus|vanus=25)= 0,14
mudel 3: P(suhe|vanus=25)= 0,24
mudel 3: P(lahutatud|vanus=25)= 0,24

0,59+0,14+0,24+0,24 = **1,21**

probleem