

8. loeng

Otsustuspuud

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?

Diskriminantanalüüs (klassifitseerimine)

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

Logistilise regressiooni mudel:

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

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

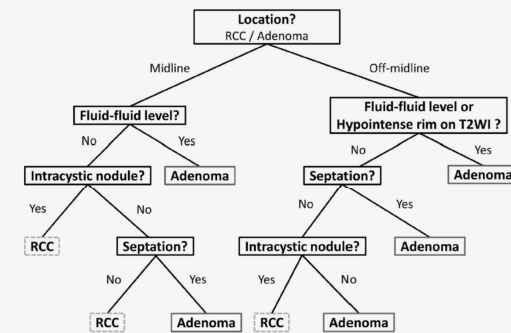


FIG 3. Diagnostic decision tree for the differentiation of cystic pituitary adenomas and RCCs using MR imaging.

Kuidas luuakse otsustuspuid?

Võimalus 1 – kasutades entroopiat

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

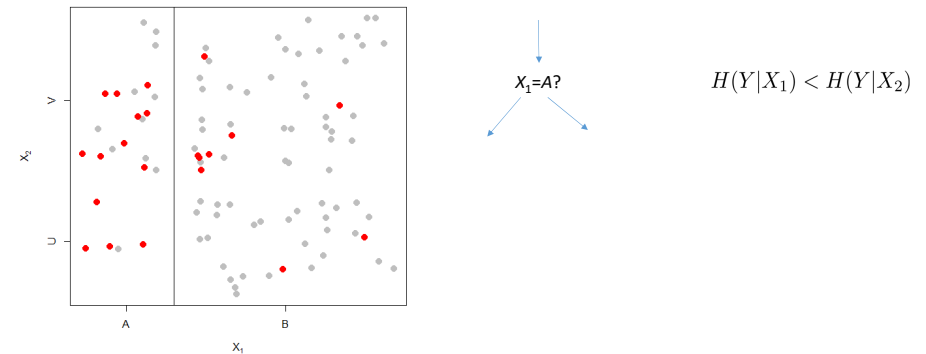
y_i	A	B	
$P(Y=y_i)$	0,5	0,5	$H(Y) = 1$
$P(Y=y_i)$	0,1	0,9	$H(Y) = 0,469$

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

Kuidas luuakse otsustuspuid?

Võimalus 1 – kasutades entroopiat

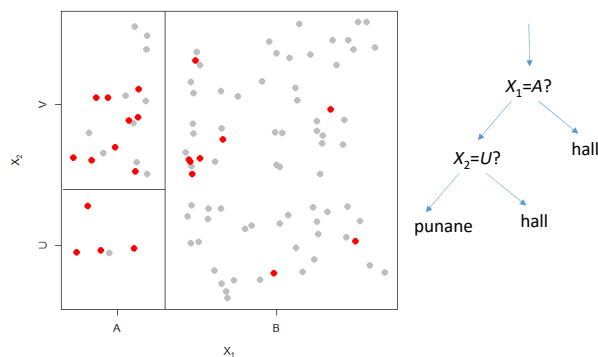
$$H(Y) = E \left(\log_2 \left(\frac{1}{P(Y)} \right) \right) = \sum_i^J p_i \log_2 \left(\frac{1}{p_i} \right) = - \sum_i^J p_i \log_2 (p_i)$$



Kuidas luuakse otsustuspuid?

Võimalus 1 – kasutades entroopiat

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



Alternatiiv

- Kasuta Gini indeksit

$$G = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

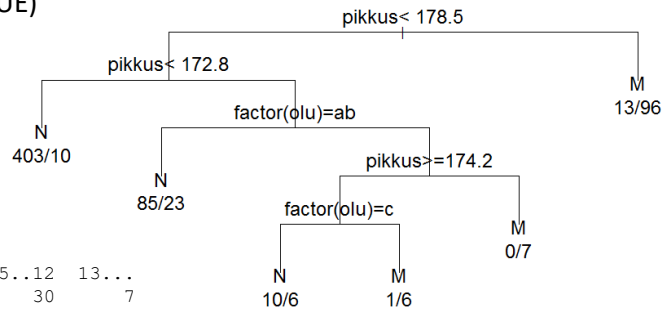
$$\hat{p}_{mk} = \frac{\#\{y = k | \mathbf{x} \in R_m\}}{|R_m|}.$$

- 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.

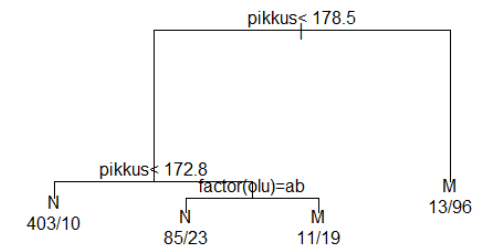
Vali selline CP väärtus mille korral on *xerror* (ristvalideerimisel saadav viga) väikseim või otsusta sarnaste *xerror*-väärtuste korral lihtsama mudeli kasuks.

Otsusepuud R'is. Samm 2 – puu pügamine

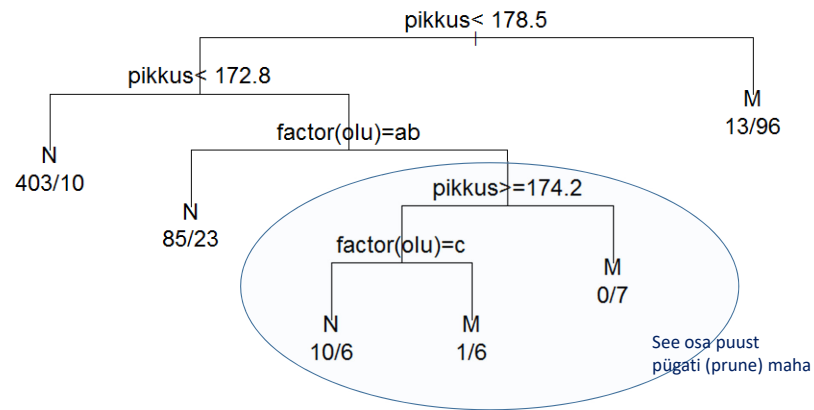
m2=prune(m1, cp=0.0136)

plot(m2, margin=0.2)

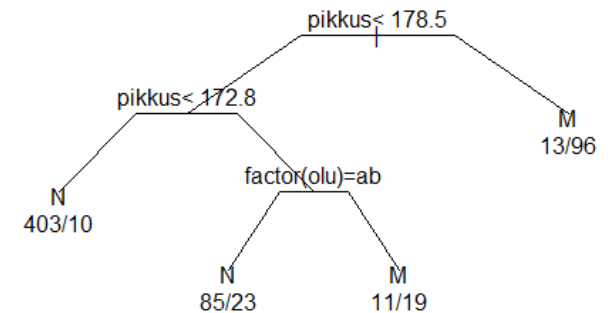
text(m2, use.n=TRUE)



	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)
```

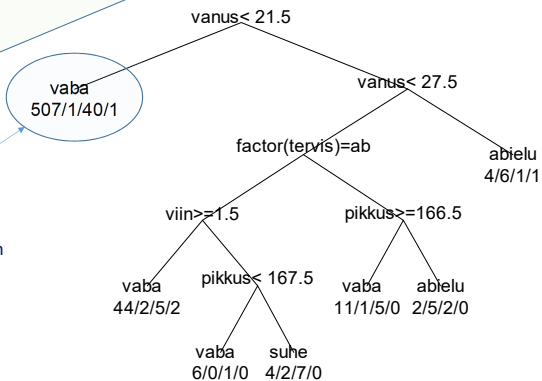


Otsusepuu kui prognoositaval
tunnusel on rohkem kui kaks
võimalikku väärtust

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



Prognoosimine otsustuspuid abil

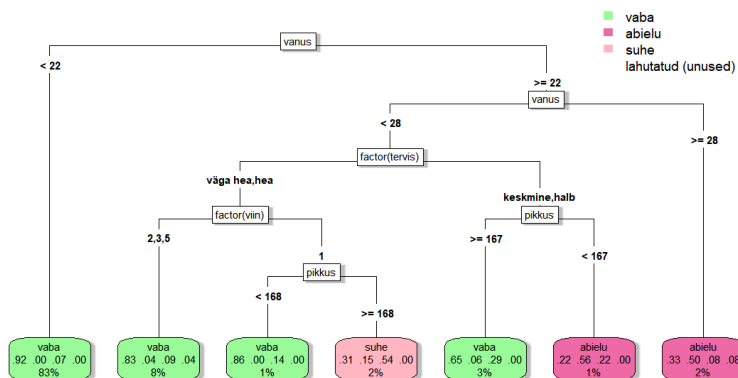
```
> predict(pere, newdata=data.frame(vanus=22, pikkus=161,
viin=1, tervis="väga hea"))
```

	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")
```

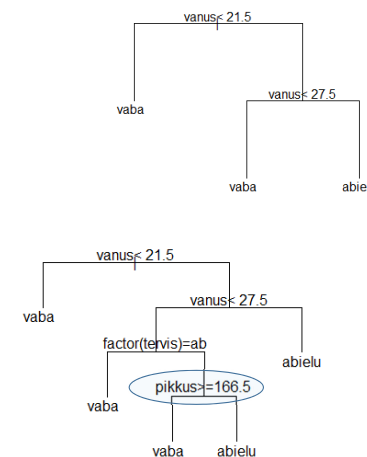
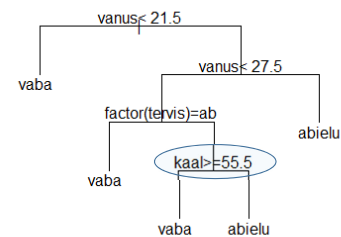
1
vaba

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



Palju puid = mets

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



Palju puid = mets

```
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,]

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

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

kasutame tunnuseid kus on vähe puuduvaid väärtuseid

vajadusel muuda juba siin osad tunnused faktortunnusteks!

jagame andmestiku treening ja testandmestikuks

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

Katsetame kui hästi „mets“ testandmete peal töötab

Palju puid = mets

```
> 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
```

Kolm tegelikult abielus olevat tudengit oleme ekslikult prognoosinud vallalisteks

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

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**2)
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: $P(\text{vaba} | \text{vanus}=25) = 0,59$

mudel 2: $P(\text{abielus} | \text{vanus}=25) = 0,14$

mudel 3: $P(\text{suhe} | \text{vanus}=25) = 0,24$

mudel 3: $P(\text{lahutatud} | \text{vanus}=25) = 0,24$

$$0,59 + 0,14 + 0,24 + 0,24 = \mathbf{1,21}$$

probleem