Versjon fra zoomforelesning
04.11.2020

ISTx1003: Statistisk læring og data science

Zoom-forelesning 3: Klassifikasjon

Mette Langaas, IMF/NTNU. 04.11.2020

NTNU

Plan: Klassifikasjon

1. Hva er klassifikasjon?
2. Læringsmål, pensum og læringsressurser
3. Tre datasett: trening, validering og testing
4. k-nærmeste nabo (kNN): en intuitiv metode
5. Forvirringsmatrise og feilrate for å evaluere metoden
6. Logistisk regresjon
7. Har vi dekket pensum?
8. Etter timen

(Inkludert 15 minutters pause)

Klassifikasjon

- tilordne en ny observasjon til en av flere kjente klasser
- lage en klassifikasjonsregel
- estimere sannsynligheten for at en ny observasjon tilhører de ulike klassene

kategorisk variabel

respons

$$(x_{1i}, x_{2i}, \ldots, x_{pi}, y_i)$$

forklaringsvariabler kovariater

uavhengige observasjoner

$$i = 1, \ldots, n$$

Hvilke tall er håndskrevet på brevet?

60 000 håndskrevne siffer

Nytt scannet bilde av håndskrevet siffer → hvilket siffer?

piksler i bildet kovariater

"0–9" = 10 klasser

Klassifikasjon

respons

$$(x_{1i}, x_{2i}, \ldots, x_{pi}, y_i)$$

forklaringsvariabler

Hvem vant tennis-matchen? Vant spiller 1: $y=1$ ja! $y=0$ nei

kovariater: forskjell i serveers x_1

dobbeltfeil x_2

uprovede feil x_3

respons

Hvordan kan du automatisk si om en epost er spam?

kovariater: avsender, ord i eposten

respons: $y=1$ spam

$y=0$ ikke-spam

Kommer en gitt kunde til å betale tilbake lånet sitt? $y=1$ ja: $y=0$ nei:

kovariater: inntekt, saldo, alder, kredittkortrente ...

Læringsmål: klassifikasjon

- kunne forstå hva klassifikasjon går ut på og
- kjenne igjen situasjoner der klassifikasjon vil være en aktuell metode å bruke
- kjenne begrepene treningssett, valideringssett og testsett og forstå hvorfor vi lager dem og hva de skal brukes til
- vite hva en forvirringsmatrise er, og kjenne til begrepene feilrate og nøyaktighet (accuracy),
- forstå tankegangen bak *k*-nærmeste nabo-klassifikasjon, og hvordan k kan velges
- kjenne til modellen for logistisk regresjon, og kunne tolke de estimerte koeffisientene
- forstå hvordan vi utfører klassifikasjon i Python
- **kunne besvare oppgave 2 av prosjektoppgaven på en god måte!**

Læringsressurser klassifikasjon

Slå opp når du lurer på noe etter å ha sett videoene

Kompendium: Klassifikasjon (pdf og html)

Videoer: (2 stk)

1. Klassifikasjon - introduksjon og k-nærmeste-nabo-klassifikasjon (10:58 min)
2. Klassifikasjon - logistisk regresjon (14:17 min)

~~3. (15:20 min)~~

Zoom: slides med notater

Pensum er definert som

"**svarene på det du blir spurt om på prosjektoppgaven**”

og de kan du finne ved å bruke læringsressursene.

Oppgave 2: Klassifikasjon. 6 spørsmålspunkt.

trening/validering/test

hvorfor hvordan bruke

viktig å tenke på

Forvirringsmatrise og feilrate

velg mellom to modeller eller metoder

evaluere modell

velge hyperparameter

tolke plott

boksplott histogram kryssplott korrelasjon

hva ser vi etter når vi vil lage en god klassifikasjonsregel?

k-nærmeste nabo (kNN)

forstå metoden

velge k

logistisk regresjon

tolke estimerte koeffisienter

hypotesetest og p-verdi

velge mellom modeller

Pensum: hva er spurt om i Oppgave 2

Syntetisk eksempel

jeg har laget dataene (x_1, x_2 har ingen fysisk betydning)

kategorisk variabel

respons

$$(x_{1i}, x_{2i}, \ldots, x_{pi}, y_i)$$

forklaringsvariabler

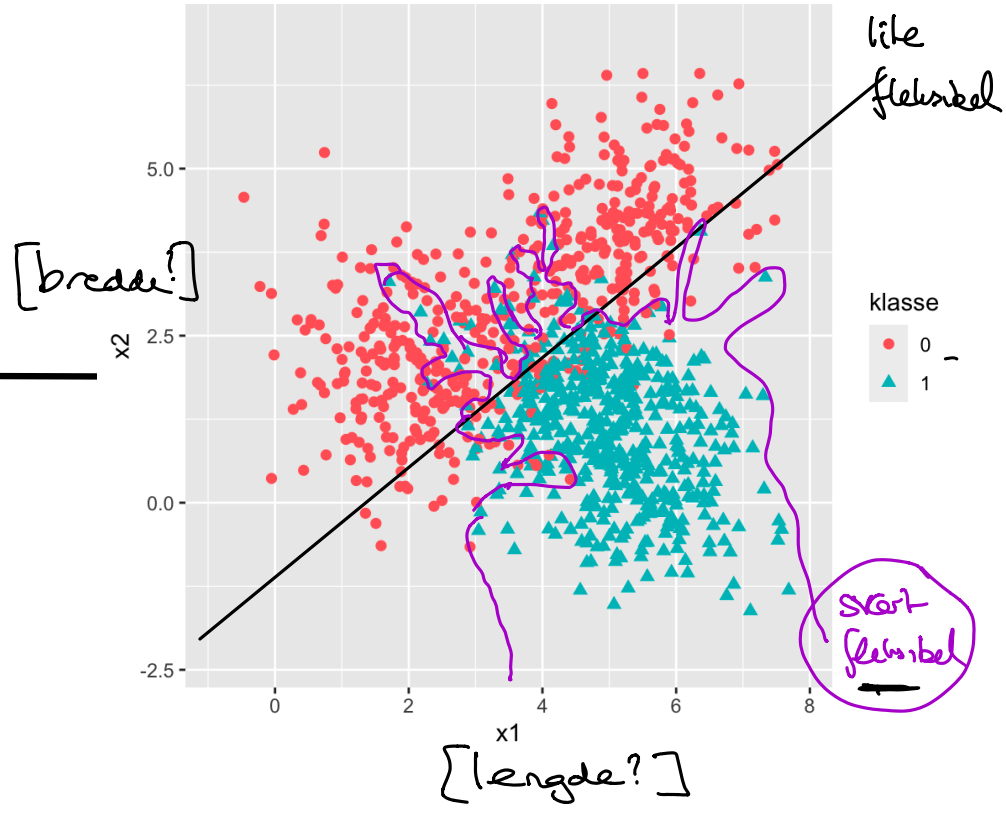
respons: 0 eller 1

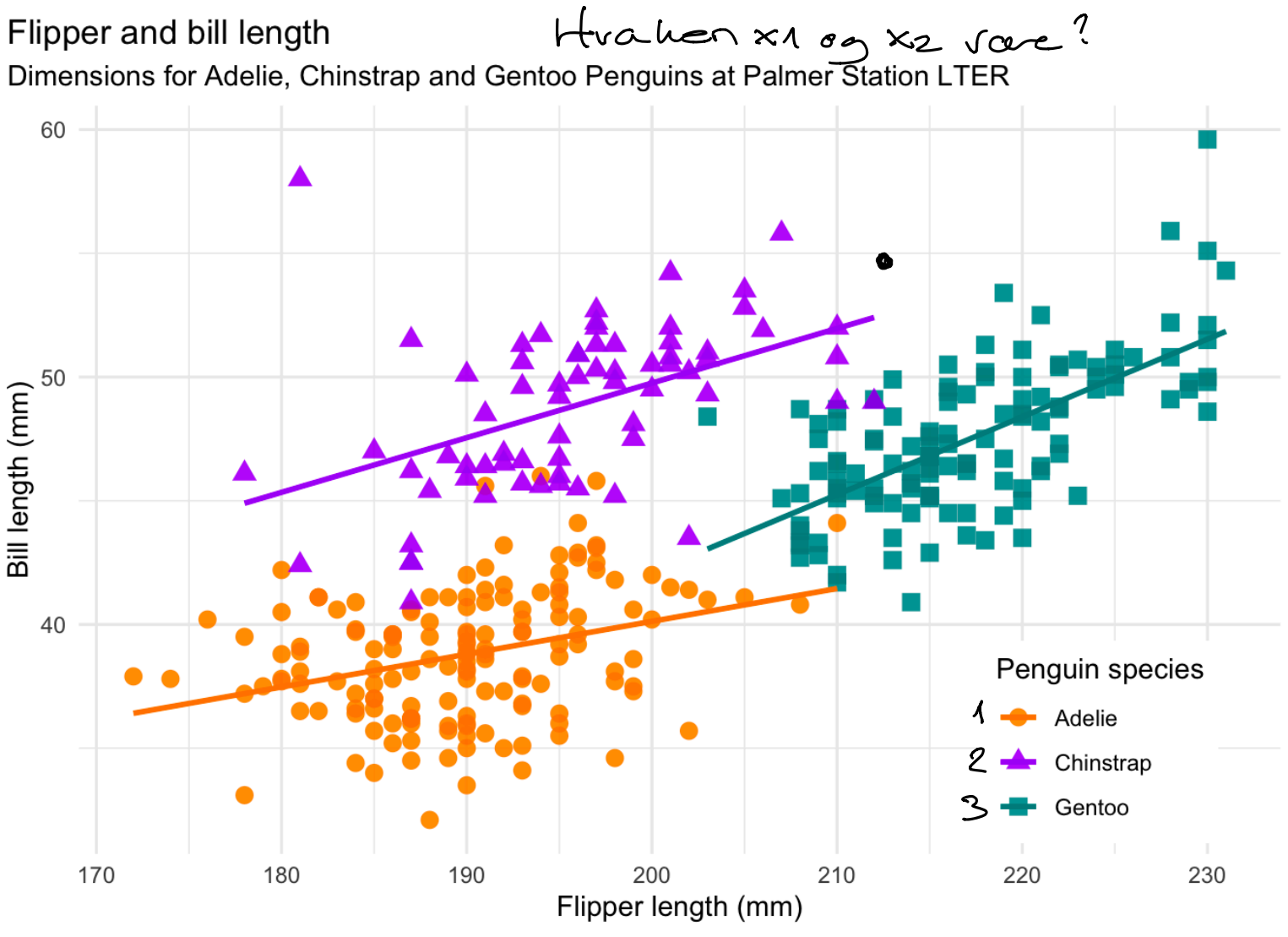
rød turkis

forklaringsvariabler:

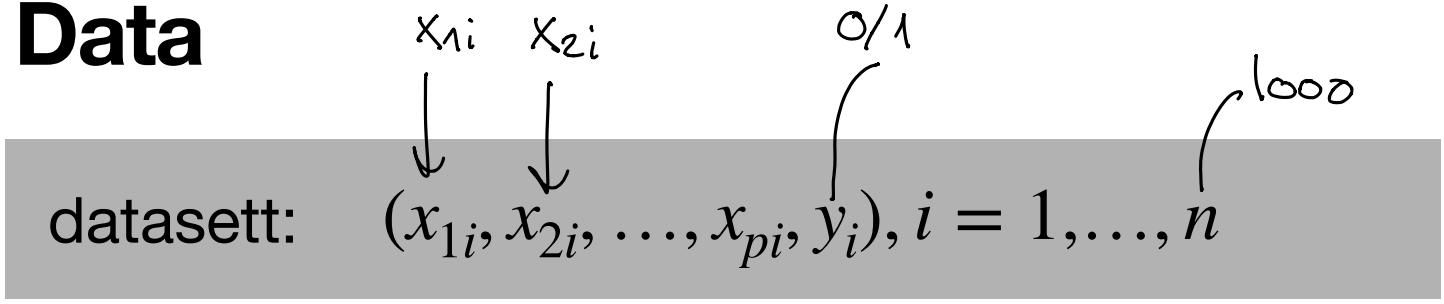
- x1
- x2

Hva vil beste klassifikasjonsgrense være?

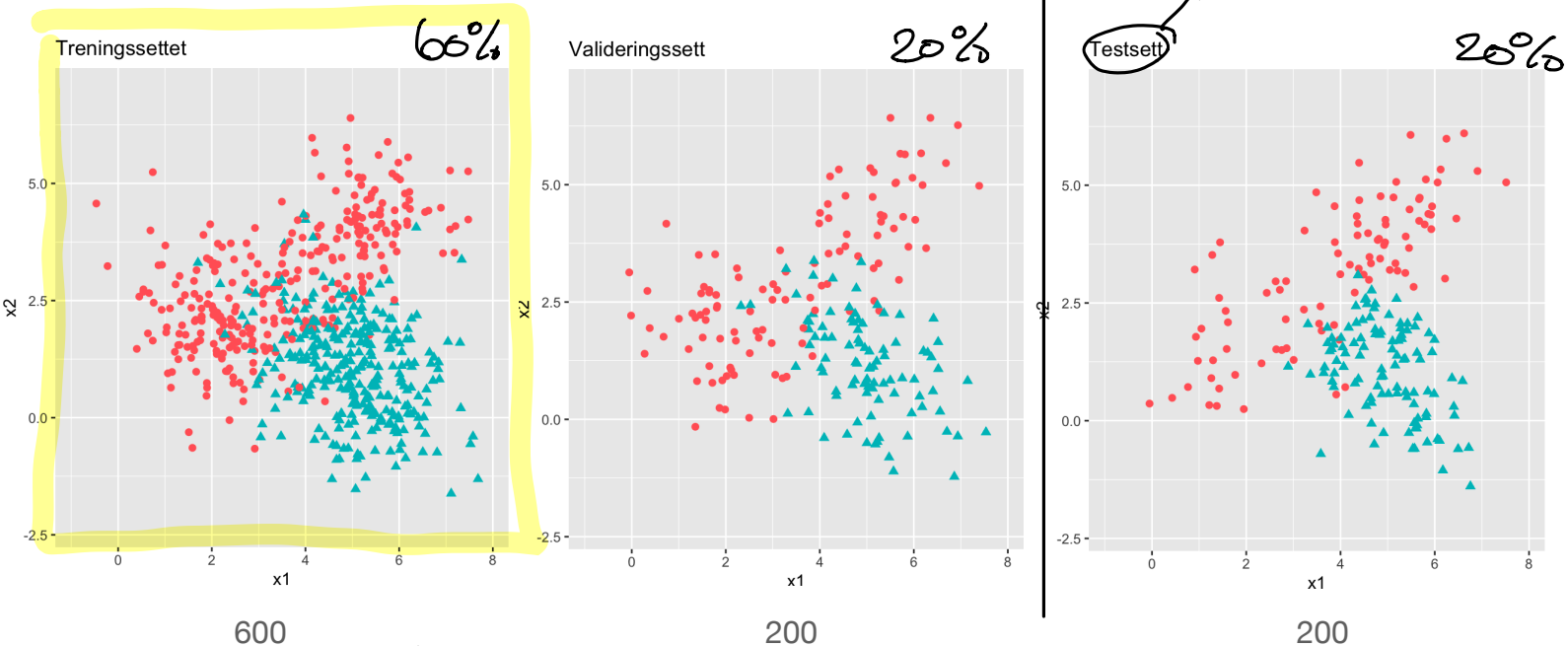



Flipper and bill length
Hvaher x1 og x2 vare?
Dimensions for Adelie, Chinstrap and Gentoo Penguins at Palmer Station LTER
60
50
40
Bill length (mm)
170
180
190
200
210
220
230
Flipper length (mm)
Penguin species
1 Adelie
2 Chinstrap
3 Gentoo
https://github.com/allisonhorst/palmerpenguins

Data


x_{1i}
x_{2i}
0/1
1000
datasett: $(x_{1i}, x_{2i}, \ldots, x_{pi}, y_i), i = 1, \ldots, n$
Hvor fleksibel vil vi ha klassegrensen?
Hvor god er metoden på nye data?
60%
20%
20%
treningssett
valideringssett
testsett
Lage klassifikasjonsregel
Velge modell eller hyperparametere (k)
Evaluere regelen på fremtidige data

Trening-, validering- og testsett for syntetiske data



Data er delt tilfeldig inn i tre datasett — de ser veldig like ut

Vi kan passe på at andelen av hver klasse er den samme i alle tre datasett.

k-nærmeste-nabo-klassifikasjon

Bruker **bare** treningsdataene.

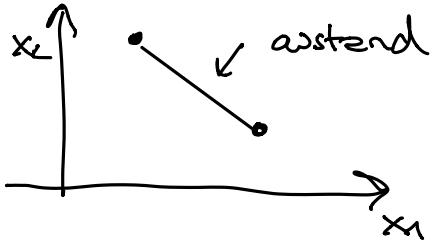
Algoritmen:

1) Ny observasjon: $(x_{10}, x_{20}, \ldots, x_{p0})$. Hvilken klasse bør denne klassifiseres til?
2) Finn de ***k* nærmeste naboene** til observasjonen i treningssettet.
3) Sannsynligheten for at den nye observasjonen tilhører klasse 1 anslår vi er andelen av de *k* nærmeste naboene som har tilhører klasse 1. Ditto for de andre klassene.
4) Klassen til den nye observasjonen er den som har størst sannsynlighet. Det blir det samme som å bruke flertallsavstemming.

k-nærmeste-nabo-klassifikasjon

Hva betyr nærmest?

Vi vil bruke
euklidsk avstand



Hvilke verdier kan k ha?

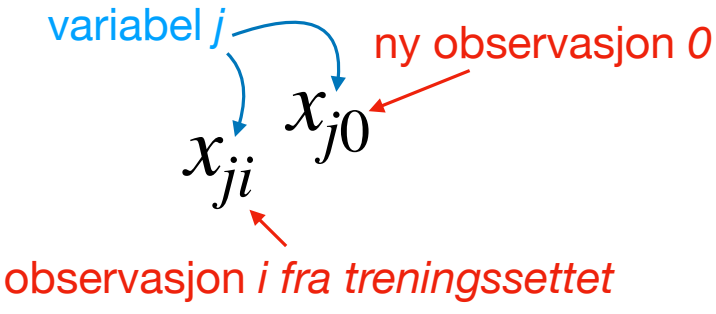
$k = 1, 2, 3, \ldots, 600$

↑
antall observasjoner
i treningsdatasett

Hvordan bestemmer vi k?

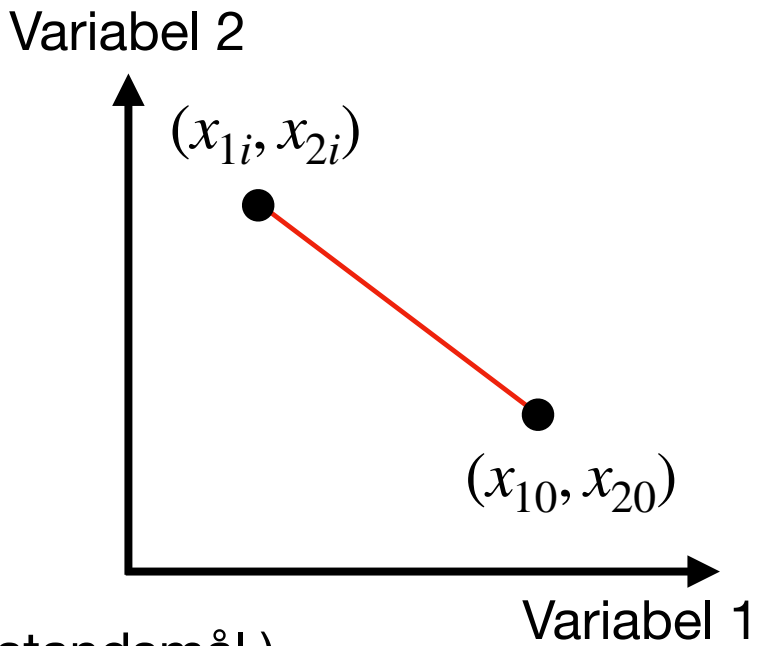
Hva betyr nærmest?

Nærmest er definert ved å bruke euklidsk avstand.



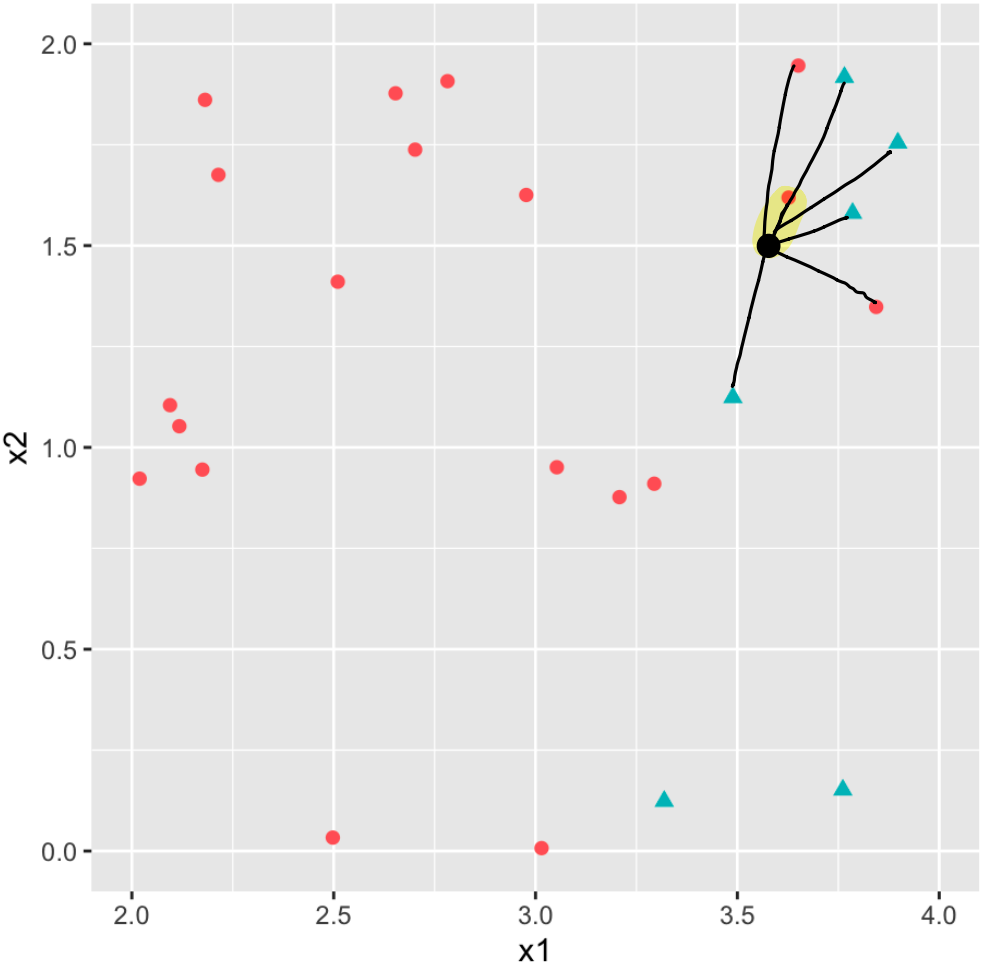
Euklidsk avstand

$$D_E(i,0) = \sqrt{\sum_{j=1}^{p} (x_{ji} - x_{j0})^2}$$



(Det er mulig å bruke andre avstandsmål.)

Nærmeste naboer



Både røde og turkise naboer:

k=1 : rød

k=2 : uavgjort ← velger ikke partalls k

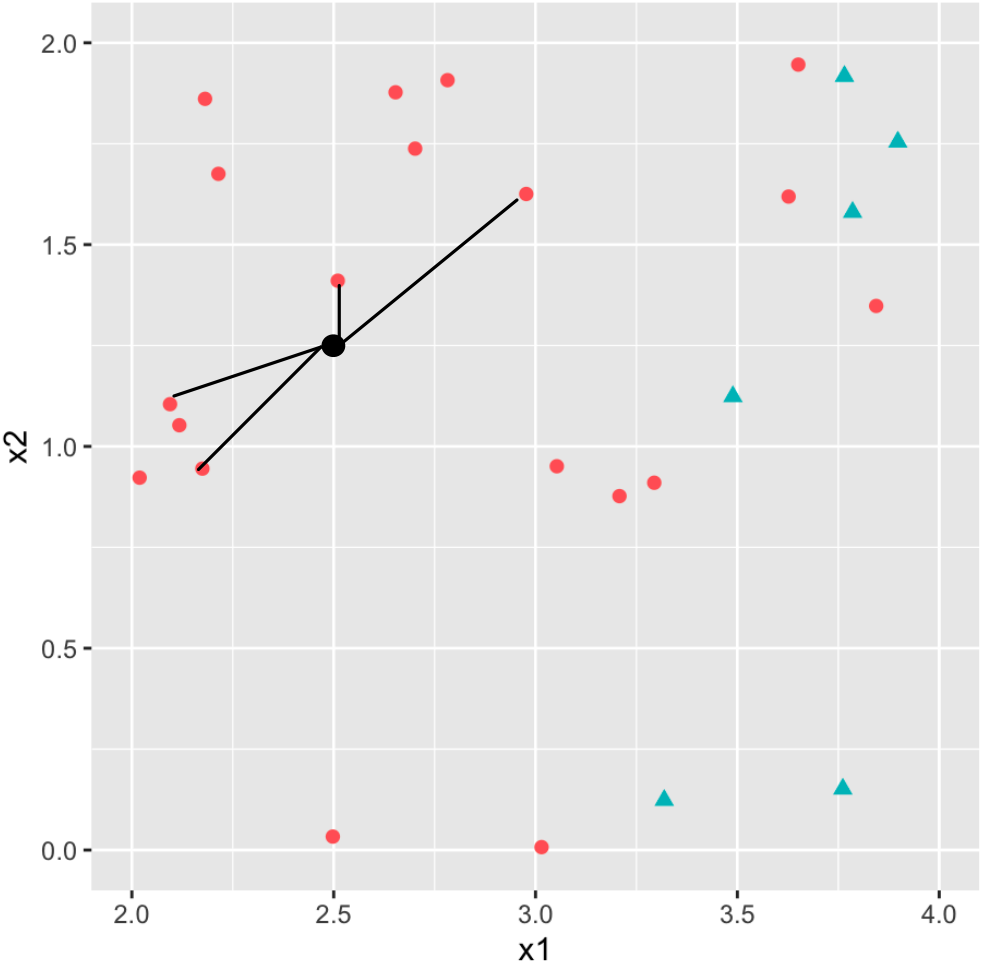
k=3 : rød

k=5 : turkis

k=7 : turkis

Det har mye å si hva k er.

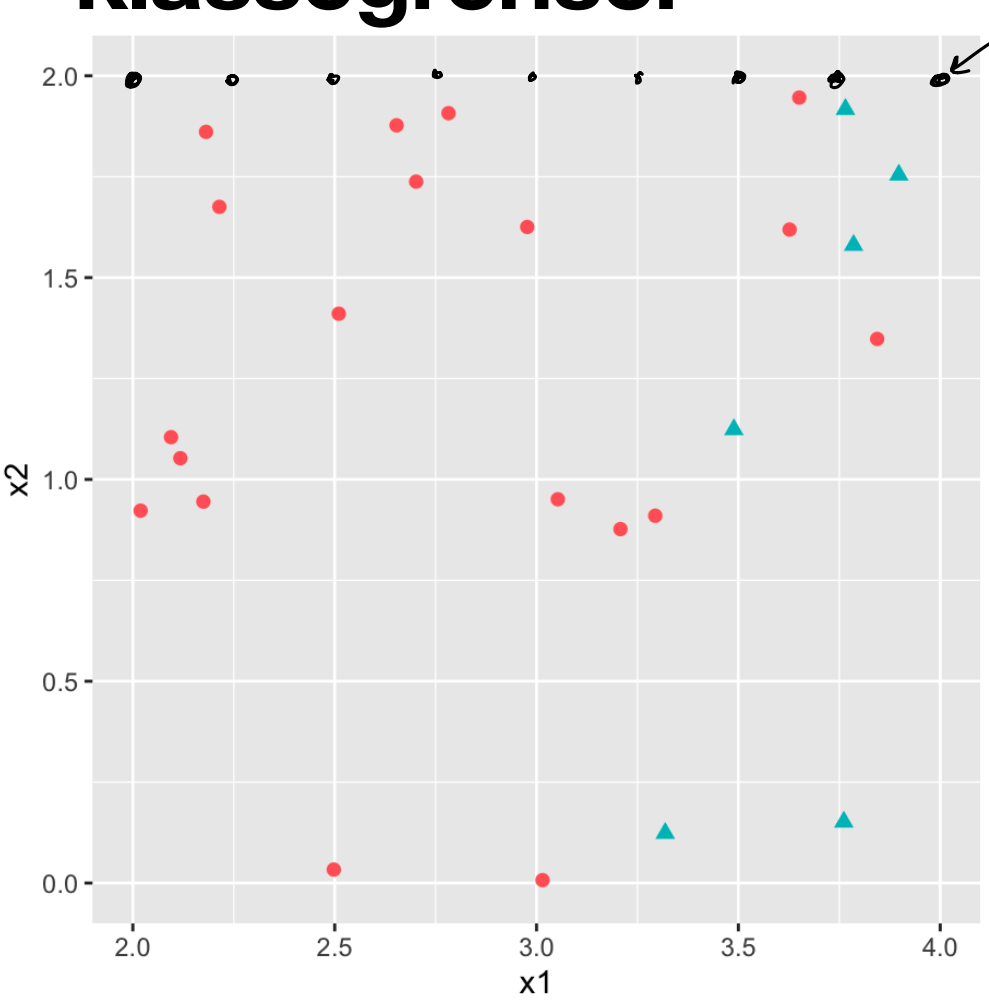
Nærmeste naboer



Den sorte observasjonen har røde naboer → blir klassifisert som rød

Det har lite å si hva k er.

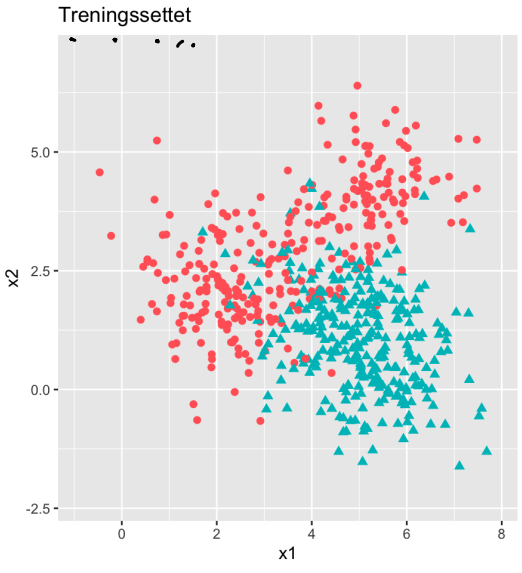
Nærmeste naboer - tegne klassegrenser

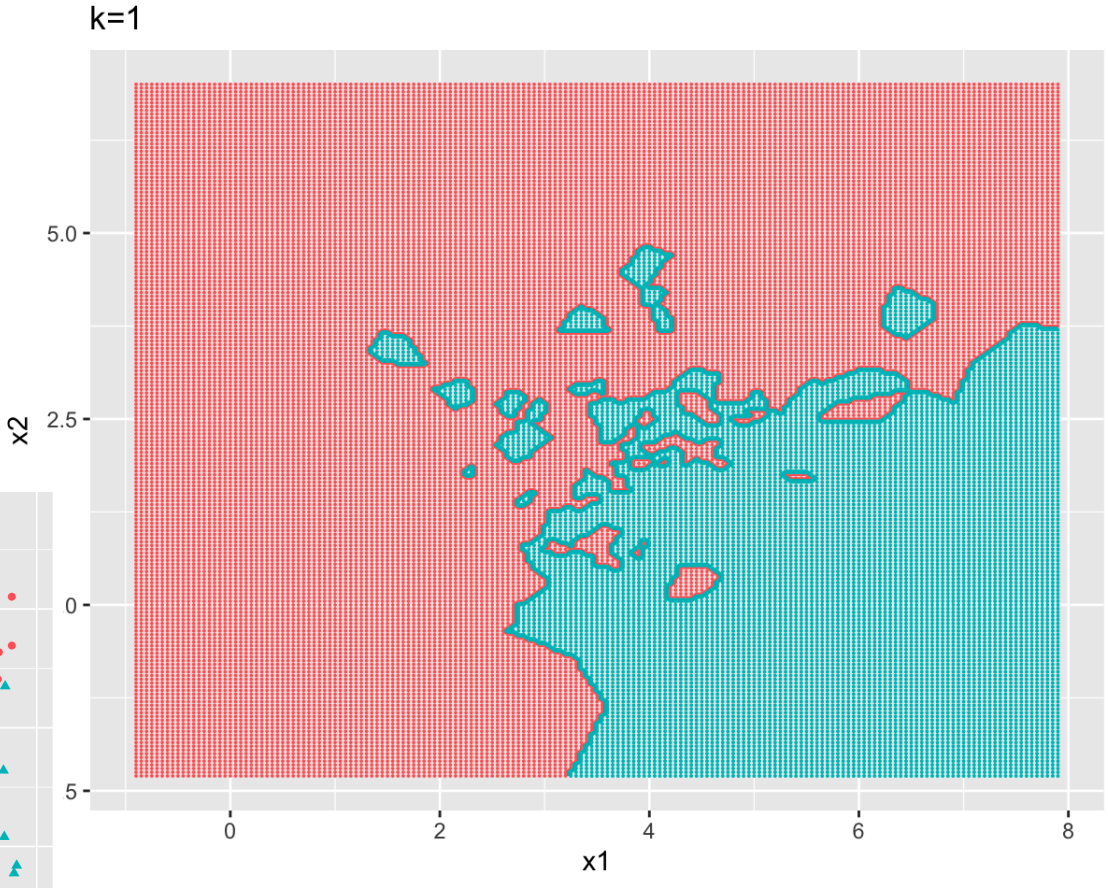


Lage rutenett

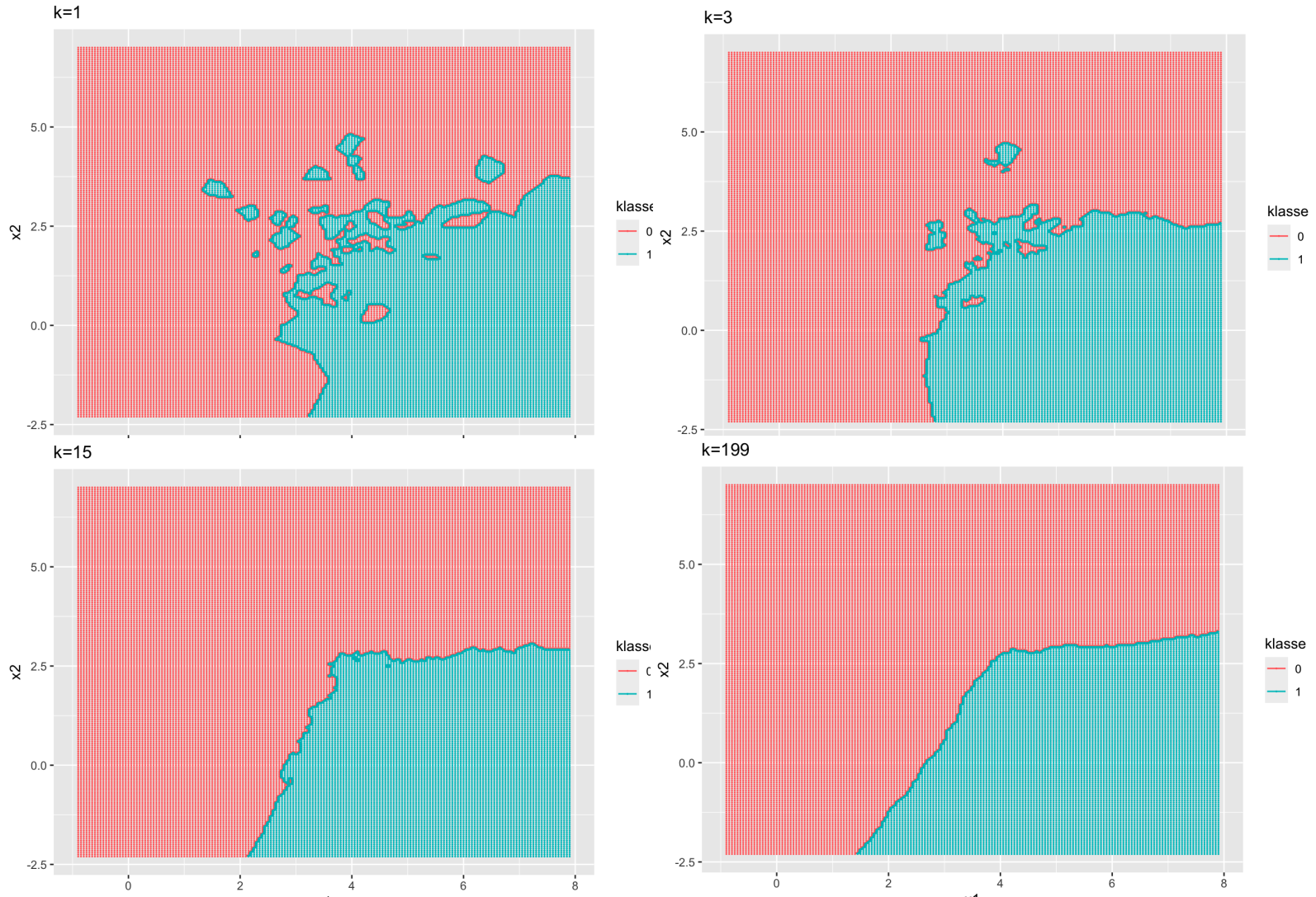
klassifisere hvert punkt i rutenettet som rød (0) eller turkis (1)

Hvordan ser klassegrensene ut?





Hvordan ser klassegrensene ut?



Hvordan velger vi *k*?

Vi må ikke velge k for liten – da blir det alt for fleksible grenser

→ Inn med valideringssettet
→ se hvorden de blir klassifisert

Hvordan velger vi *k*?

Se på hvor hver obs i valideringssettet er, og den blir feilklassifisert

Forvirringsmatrise med to klasser:

	Sann klasse 0 sann rød	**Sann klasse 1** sann turkis
Predikert klasse 0 klassifisert rød	Sann negativ (TN)	Falsk negativ (FN)
Prediktert klasse 1 klass. turkis	Falsk positiv (FP)	Sann positiv (TP)

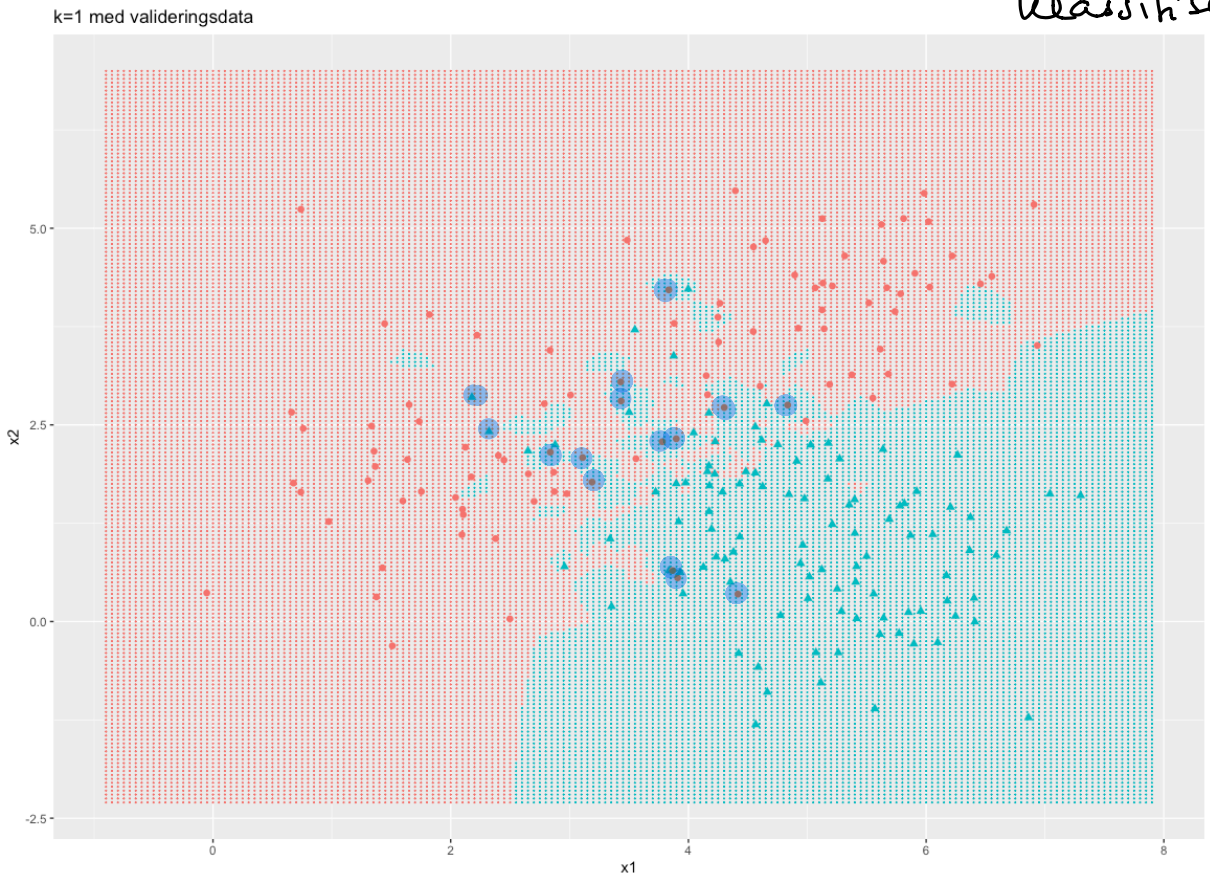
Feilrate: andel feilklassifiserte observasjoner

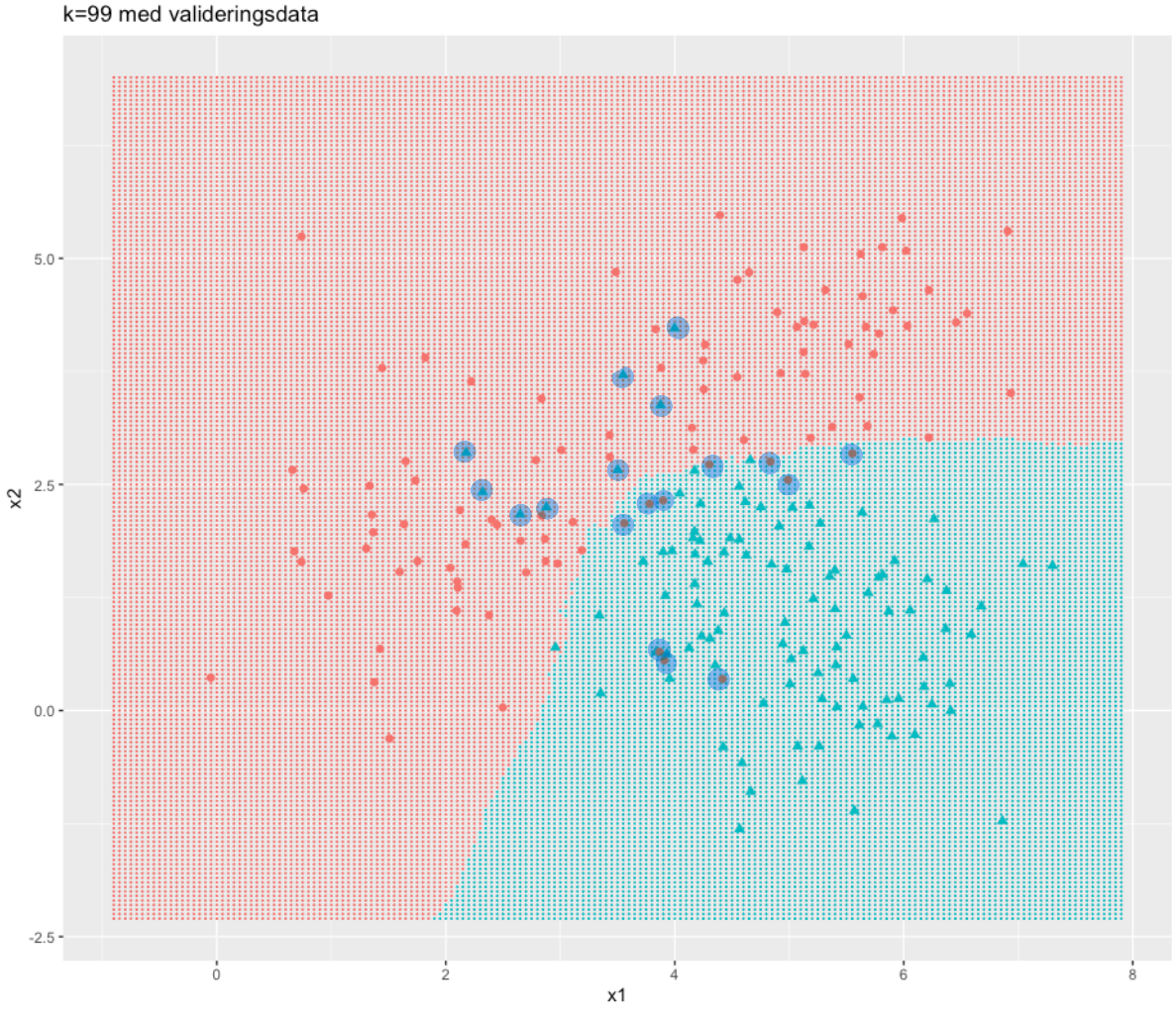
Vi velger den *k* som minimerer feilraten på valideringssettet.

Hvordan velger vi *k*?

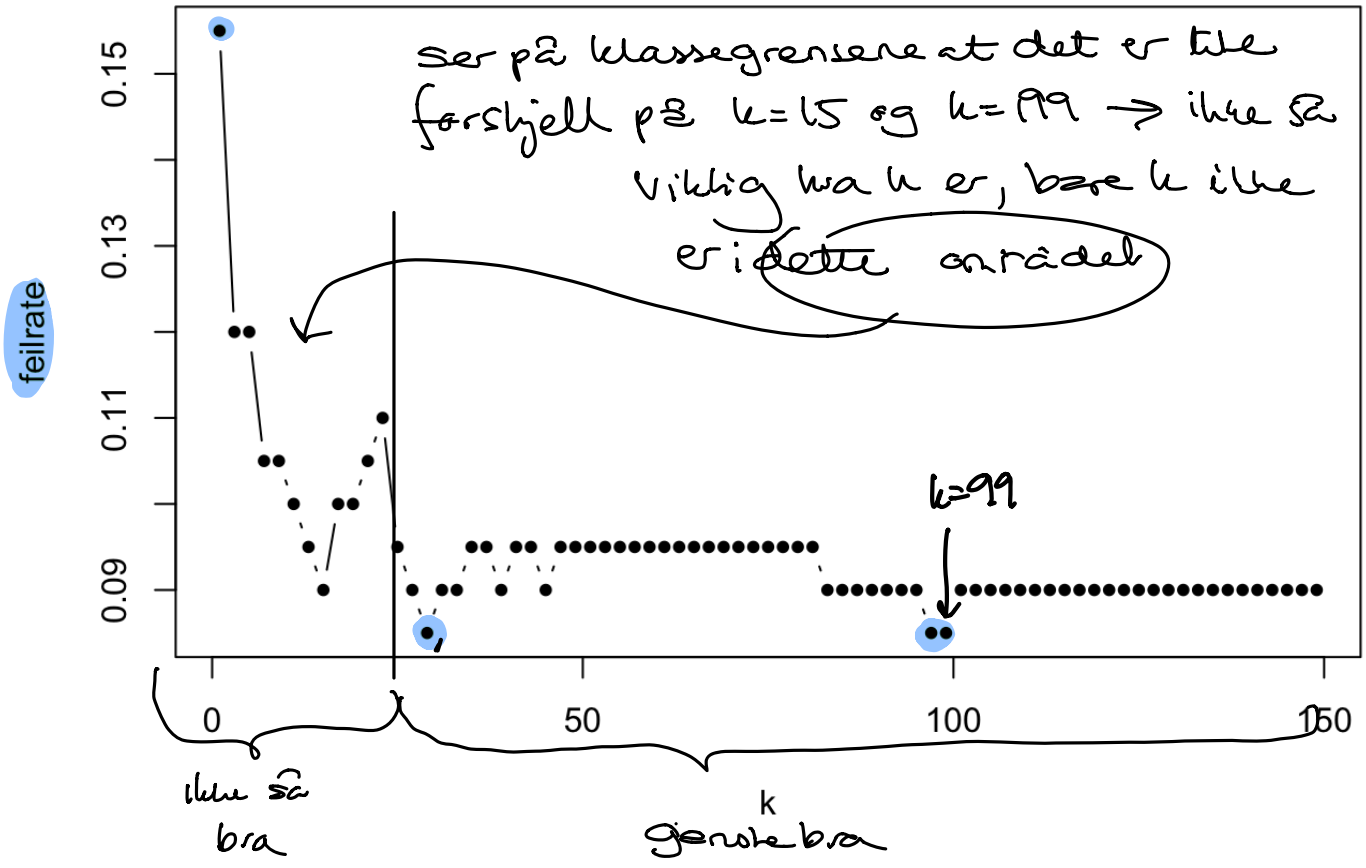
Prøver og tell antall gale klassifiseringer på valideringsdatasettet

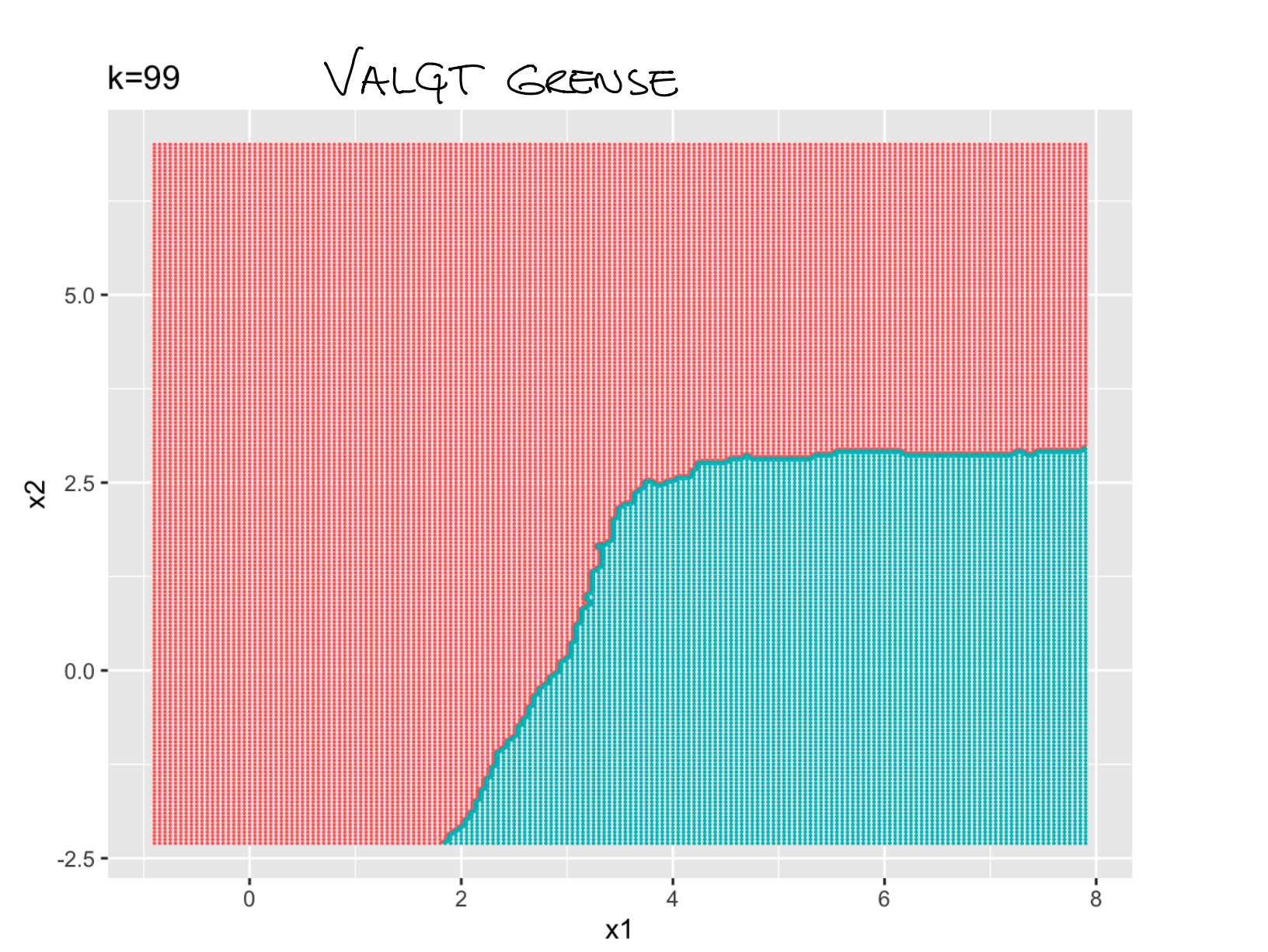
● = gal

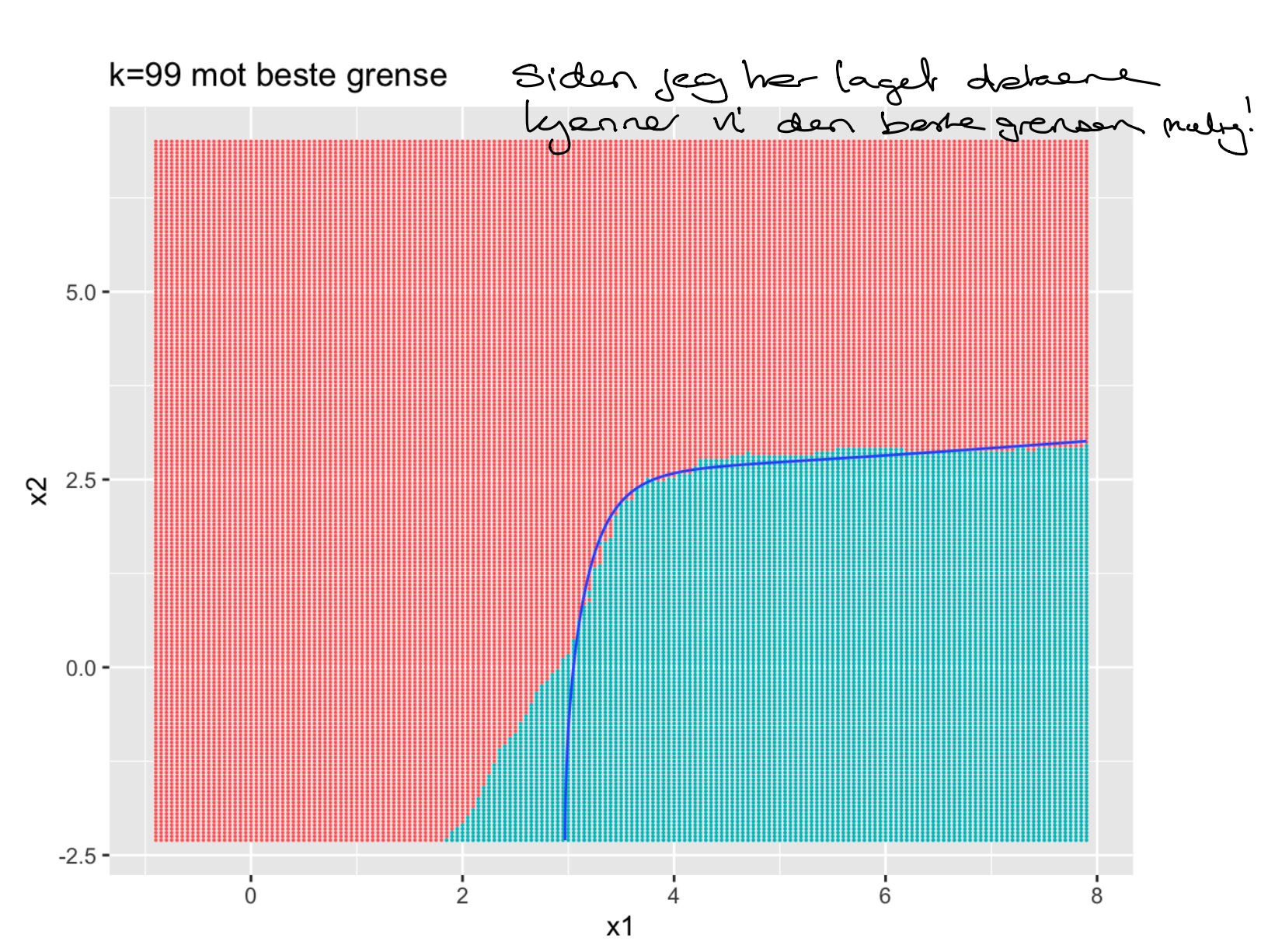


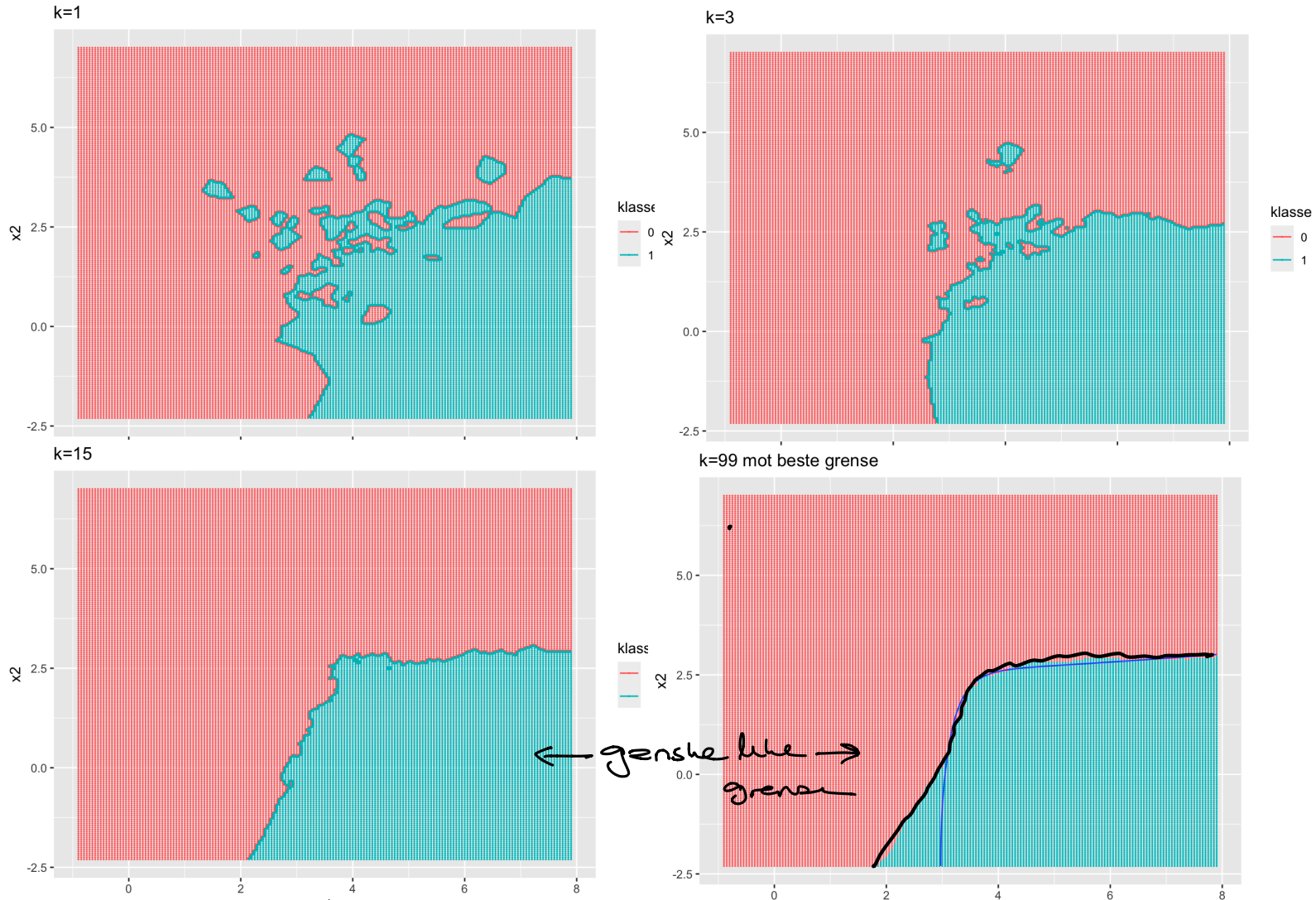

k=99 med valideringsdata
5.0
2.5
0.0
-2.5
x2
0
2
4
6
8
x1

Feilrate i valideringssettet



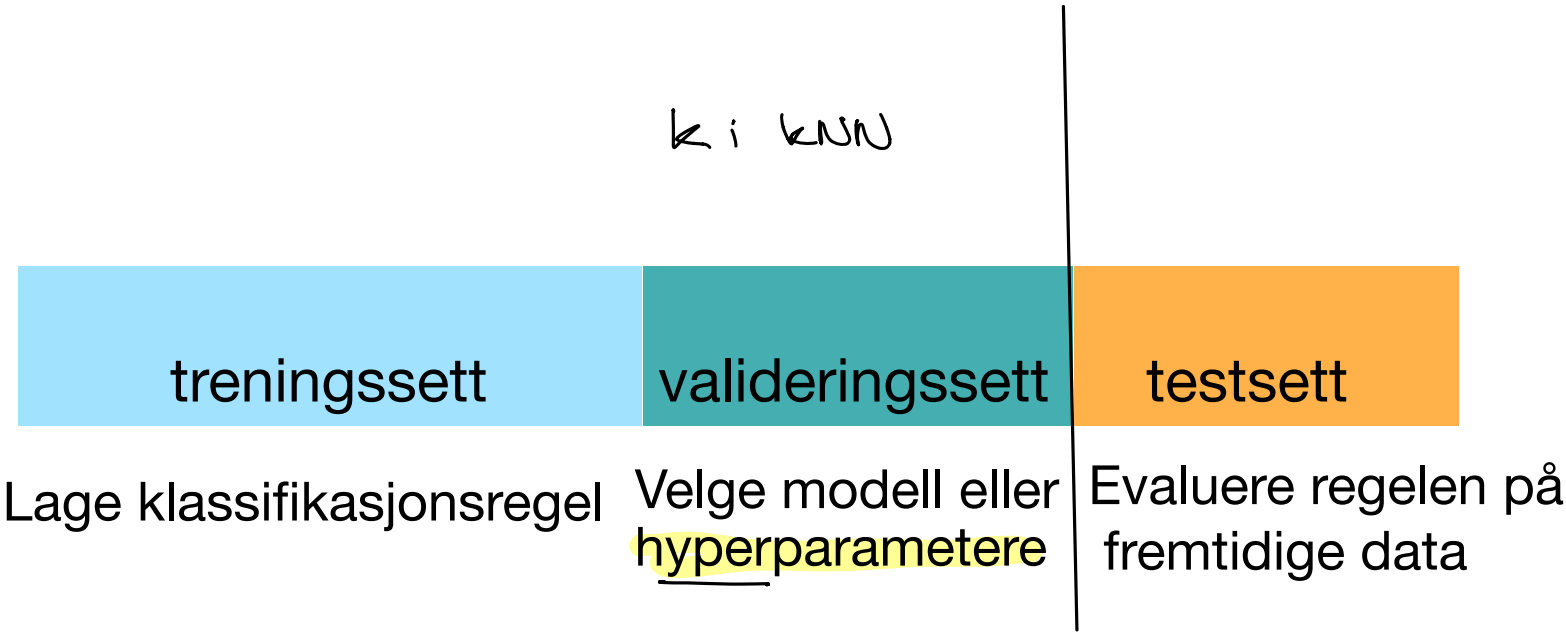

k=99
VALGT GRENSE
5.0
2.5
0.0
-2.5
x2
0
2
4
6
8
x1

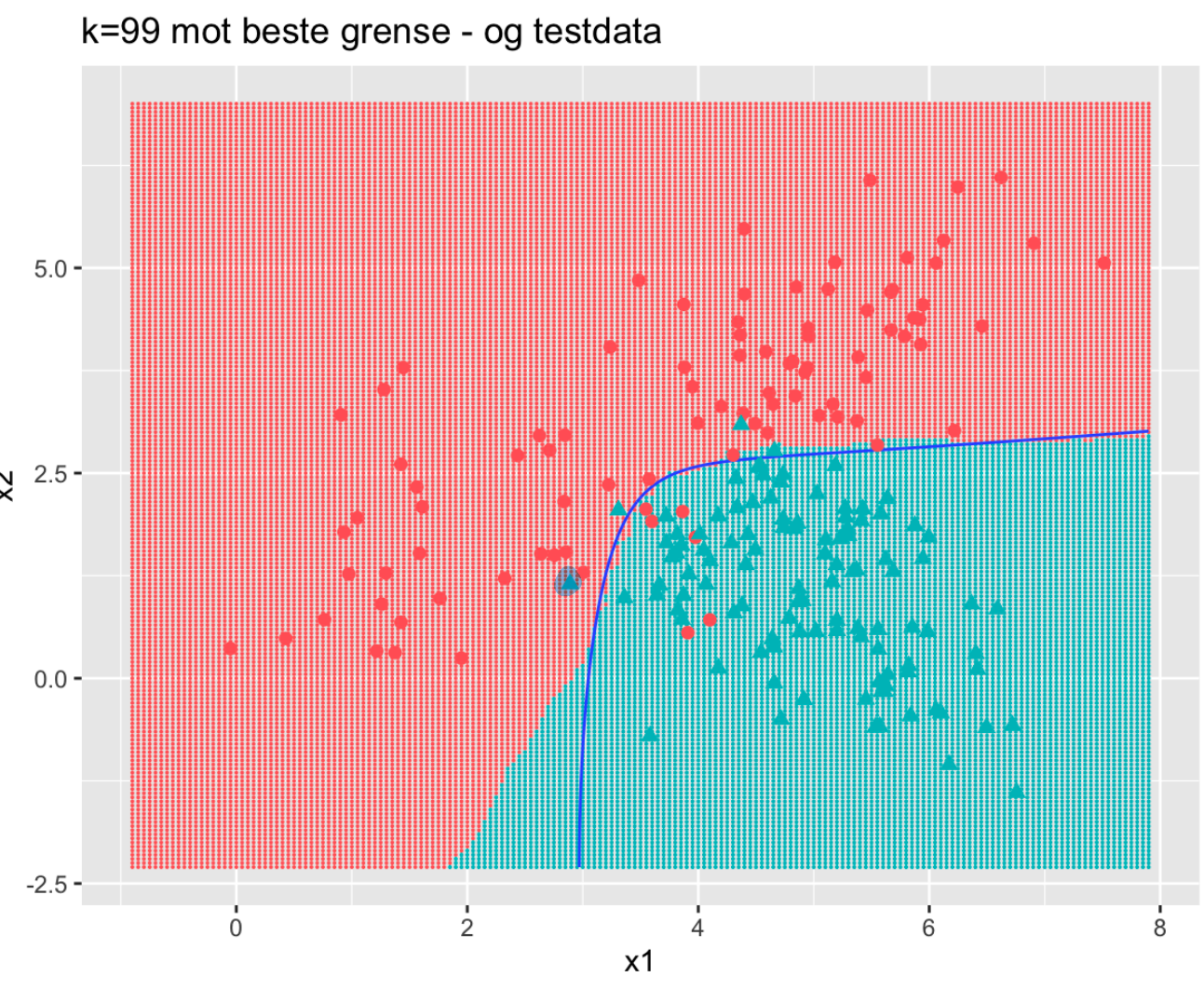

k=99 mot beste grense
Siden jeg her laget datasene
lyenner vi den beste grensen meldy!
5.0
2.5
0.0
-2.5
x2
0
2
4
6
8
x1


k=1
k=3
k=15
k=99 mot beste grense
x2
klasse
0
1
5.0
2.5
0.0
-2.5
0
2
4
6
8
grense like grense

Data

datasett: $(x_{1i}, x_{2i}, \ldots, x_{pi}, y_i), i = 1, \ldots, n$





Feilrate på testsettet: 5.5%

K-nærmeste nabo klassifikasjon i Python

```
knaboer = np.arange(1,199,step=2)

val_feilrate = np.empty(len(knaboer))

for i,k in enumerate(knaboer):
    knn = KNeighborsClassifier(n_neighbors=k,p=2)
    knn.fit(df_tren[['x1','x2']], df_tren['y'])
    val_feilrate[i] = 1-knn.score(df_val[['x1','x2']], df_val['y'])
```

k = 1,3,5,...,199

0,0,0,...,0

1,3,5,...

Eukl. avstand

feilrate på valideringssettet

K-nærmeste nabo klassifikasjon i Python

Plott feilrate

```
plt.title('k-NN for ulike verdier av antall naboer k')
plt.plot(knaboer, val_feilrate, label='Feilrate på valideringssettet')
plt.xlabel('Antall naboer k'); plt.ylabel('Feilrate')
plt.show()
```

Velg k

```
mink_valfeilrate = knaboer[np.where(val_feilrate == val_feilrate.min())]
print(mink_valfeilrate[0])
```

Sett opp klassifikator

```
bestek=99
knn = KNeighborsClassifier(n_neighbors=bestek,p=2)
```

Sett inn beste k her – også på prosjektet

Feilrate på testsett

```
knn.fit(df_tren[['x1','x2']], df_tren['y'])
print("Feilrate kNN:", 1-knn.score(df_test[['x1','x2']], df_test['y']))
```

trening/validering/test

hvorfor

hvordan bruke

viktig å tenke på

Forvirringsmatrise og feilrate

evaluere modell

velg mellom to modeller eller metoder

velge hyperparameter

tolke plott

boksplott histogram
kryssplott korrelasjon

hva ser vi etter når vi vil lage en god klassifikasjonsregel?

k-nærmeste nabo (kNN)

forstå metoden

velge k

= snahetom!

= gjenstår

logistisk regresjon

tolke estimerte koeffisienter

hypotesetest og p-verdi

velge mellom modeller

Pensum: hva er spurt om i Oppgave 2

Plan: Klassifikasjon

1. Hva er klassifikasjon?
2. Læringsmål, pensum og læringsressurser
3. Tre datasett: trening, validering og testing
4. k-nærmeste nabo (kNN): en intuitiv metode
5. Forvirringsmatrise og feilrate for å evaluere metoden ↓

---

6. Logistisk regresjon
7. Har vi dekket pensum?
8. Etter timen

Hit nå

(Inkludert 15 minutters pause)

Klassifikasjon med logistisk regresjon

Kan bare håndtere 2 klasse

- 1 : float glass
- 0 : ikke-floatglass

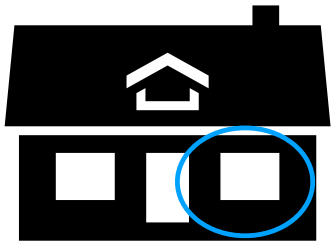
"finne" klassegrense og forstå regelen vi lager

$\hat{\beta}$

x_1: Al

x_2: RI = brytningsindeks

Glass på åsted



respons

vindusglass av type float

vindusglass av type ikke-float

vektprosent oksid av Al

brytningsindeks (RI)

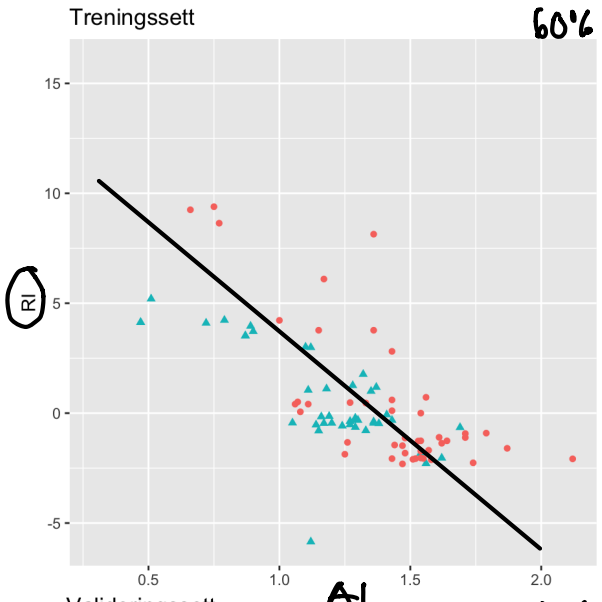
forklaringsvariabler

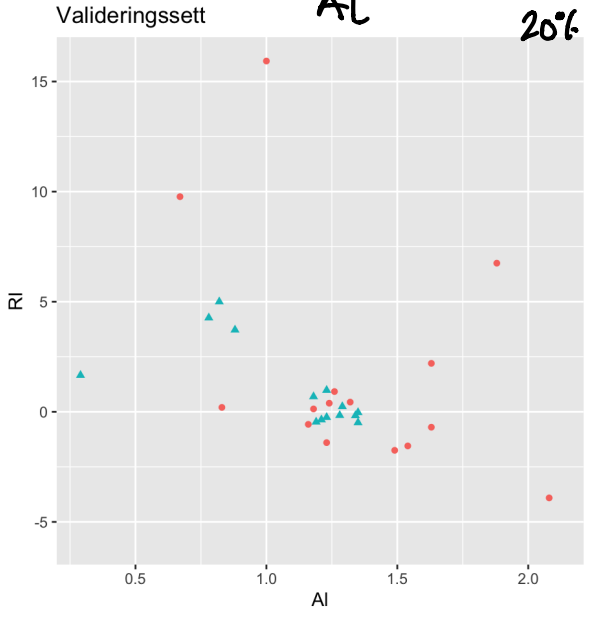
Trening-, validering- og testsett for glasskår-dataene

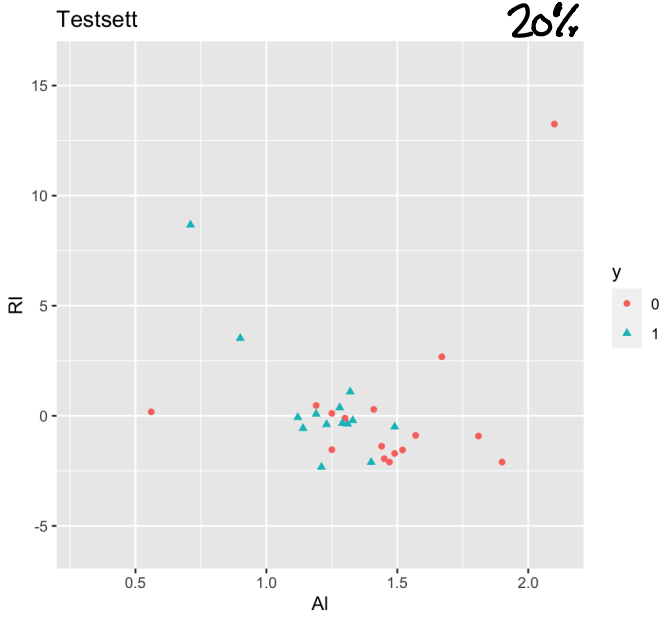
▲ 1: float

● 0: ikke-float

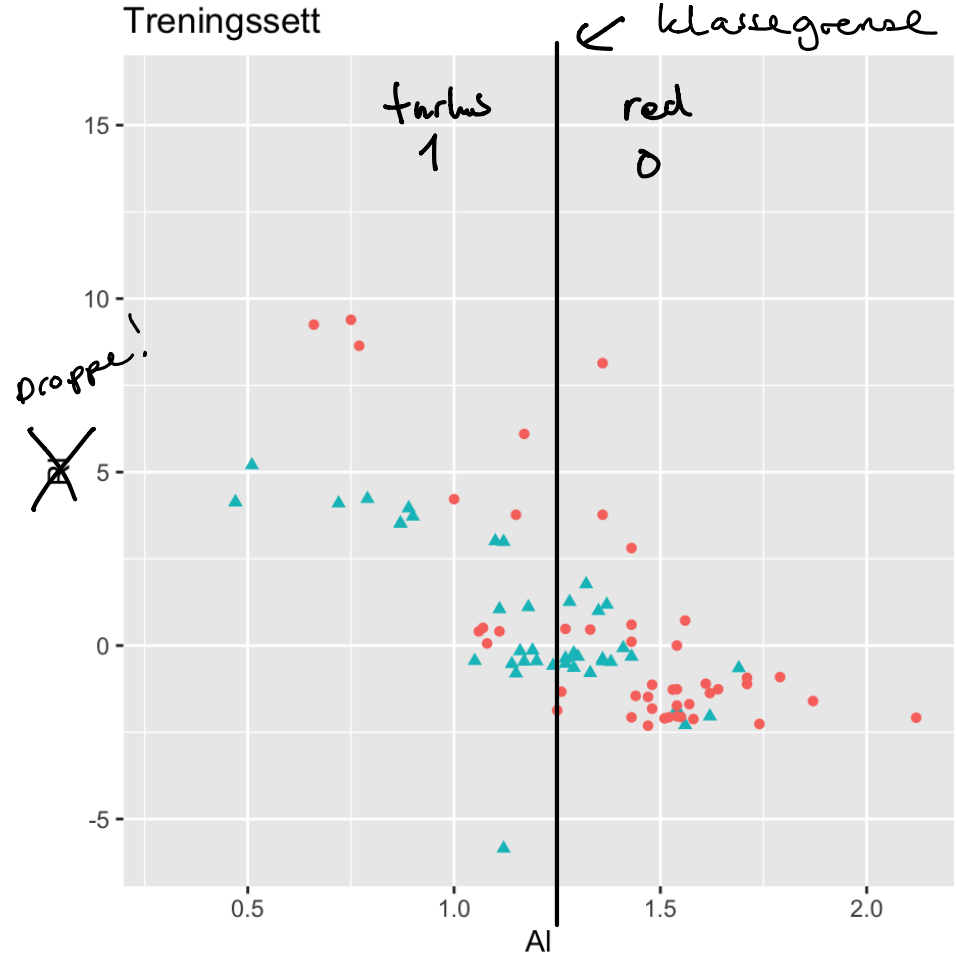
NB: vi har lite data!

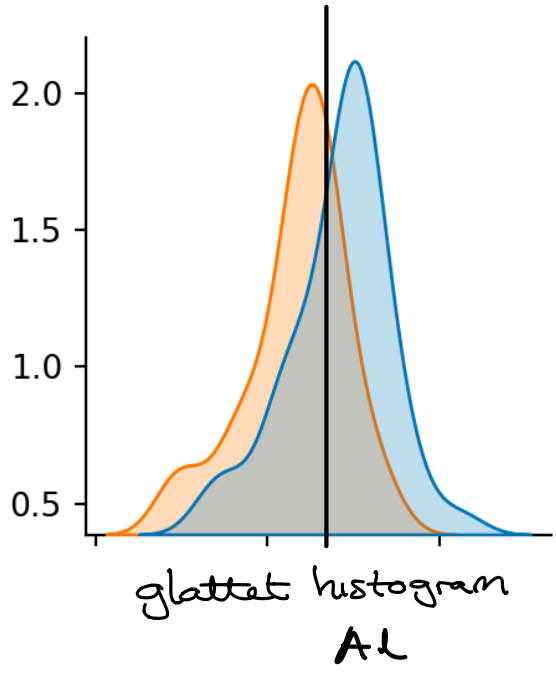






Først: kan vi bruke bare Al til å skille mellom float og ikke-float-glass?

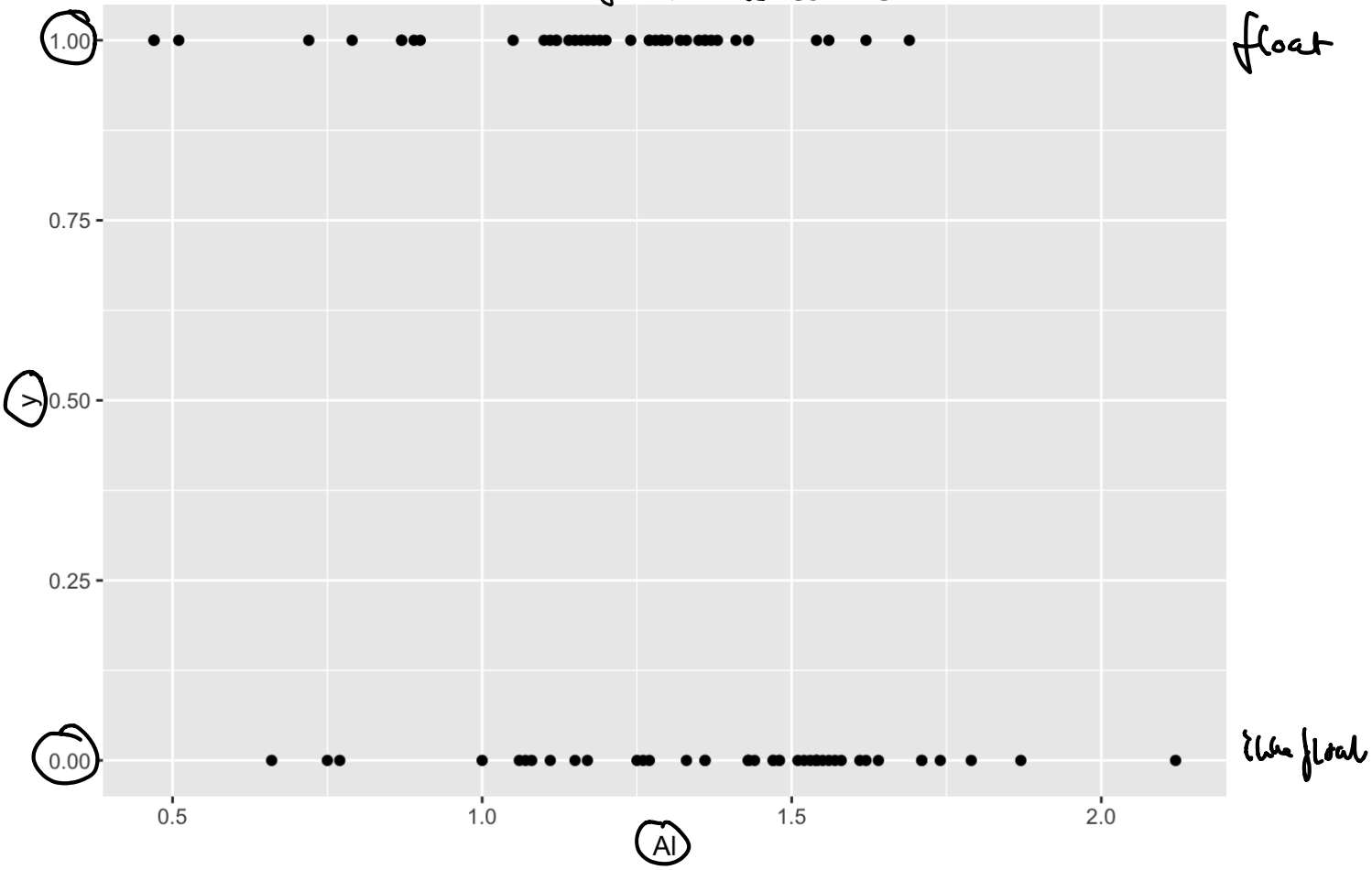




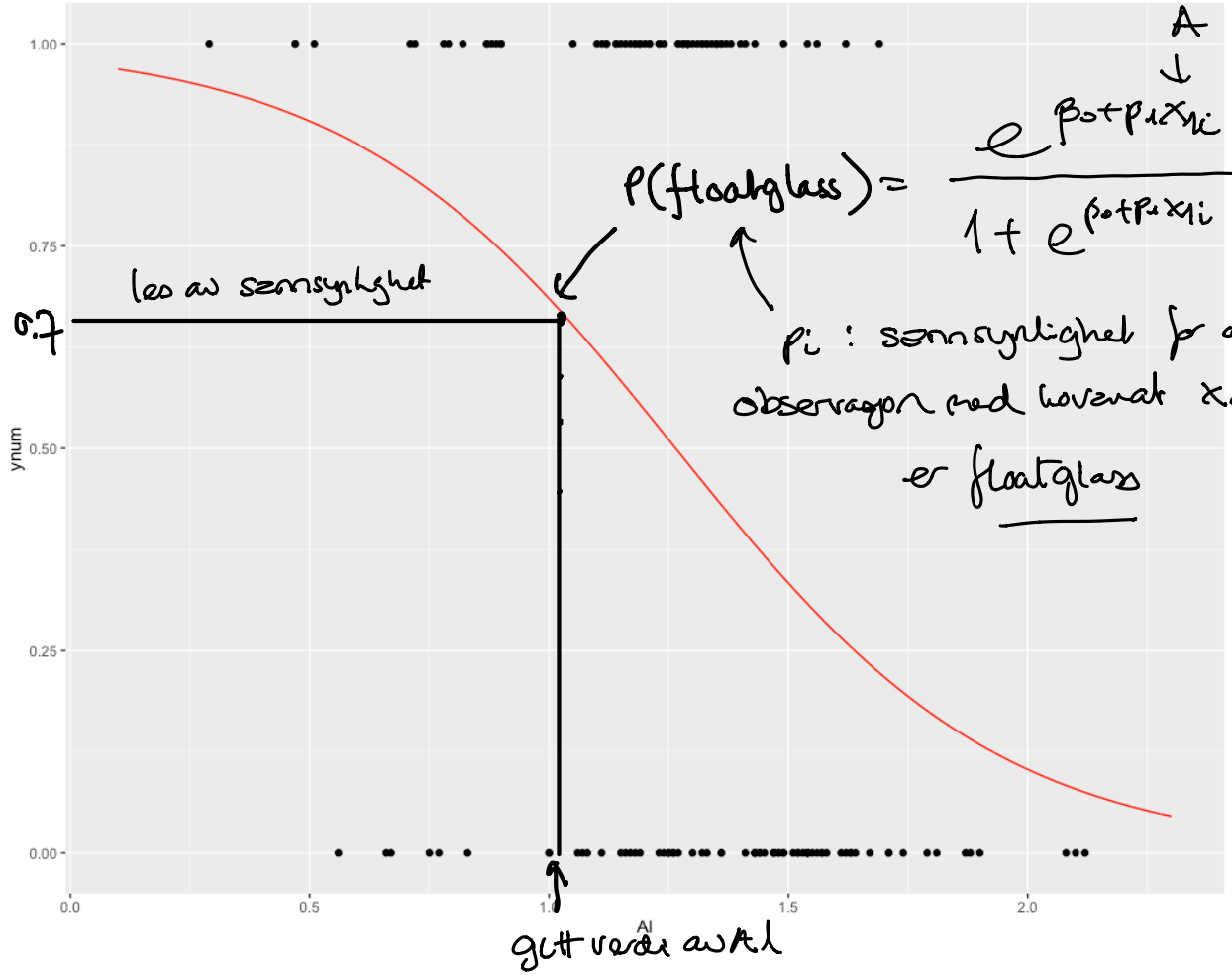
Kryssplott ← dette var vårt viktigste plott i regresjon —

nå er y ikke kontinuerlig

klassifikasjon med to klasser



Enkel logistisk regresjon



Enkel logistisk regresjon

Vi antar at responsen Y_i er binomisk fordelt

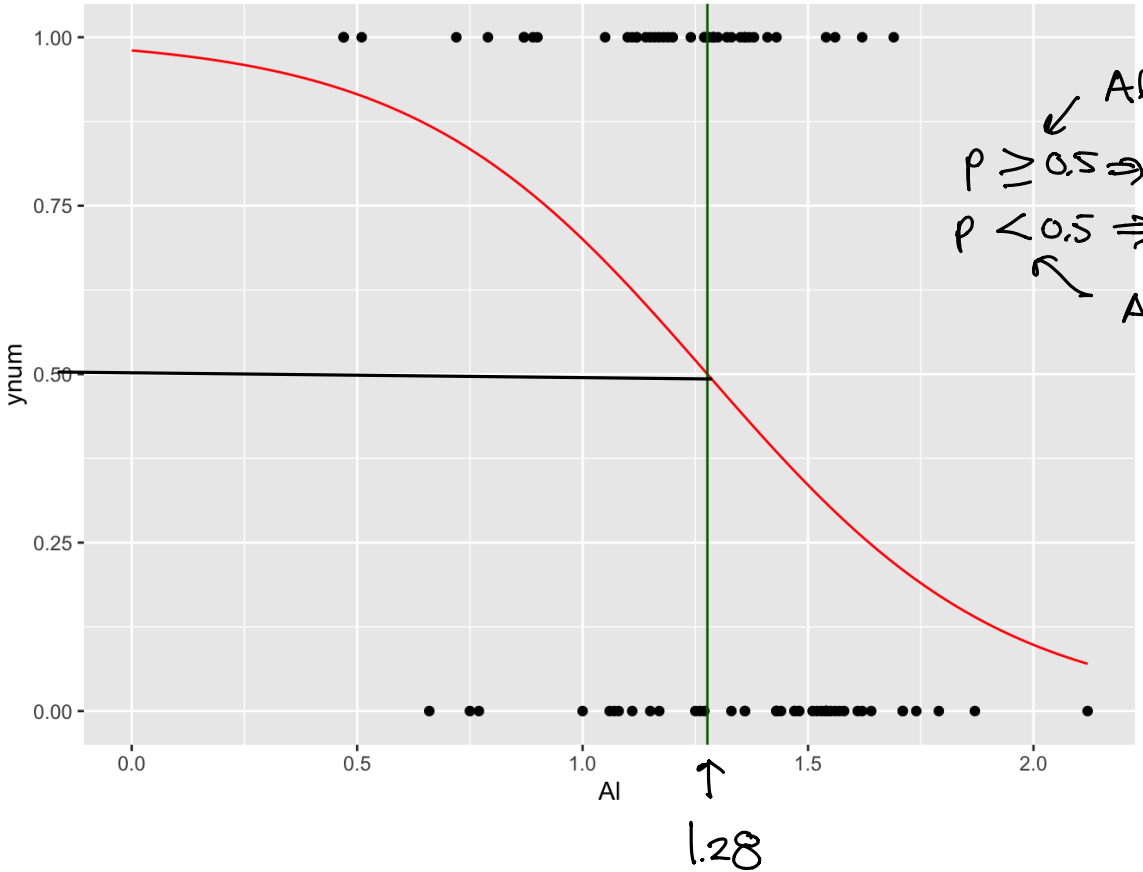
- med 1 forsøk
- suksessannsynlighet p_i ← for (fellesdelen) hadde alle observasjoner samme p_i — nå er p_i avhengig av Al

$$Y_i = \begin{cases} 1 \text{ med sannsynlighet } p_i, \\ 0 \text{ med sannsynlighet } 1 - p_i \,. \end{cases}$$

Videre kobler vi p_i sammen med forklaringsvariablen med en *logistisk funksjon*

$$p_i = \frac{e^{\beta_0 + \beta_1 x_{1i}}}{1 + e^{\beta_0 + \beta_1 x_{1i}}}$$

Klassifikasjonsregel



Al ≤ 1.28

$p \geq 0.5 \Rightarrow$ floatglass

$p < 0.5 \Rightarrow$ ikke floatglass

Al > 1.28

Enkel logistisk regresjon

Vi ser ikke på denne delen av utskriften

Logit Regression Results

Dep. Variable:	y	No. Observations:	87
Model:	Logit	Df Residuals:	85
Method:	MLE	Df Model:	1
Date:	Tue, 13 Oct 2020	Pseudo R-squ.:	0.1204
Time:	21:55:53	Log-Likelihood:	-52.997
converged:	True	LL-Null:	-60.252
Covariance Type:	nonrobust	LLR p-value:	0.0001394

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept ($\hat{\beta}_0$ =)	3.9141	1.232	3.176	0.001	1.499	6.330
Al ($\hat{\beta}_1$ =)	-3.0656	0.924	-3.317	0.001	-4.877	-1.254

sannsynlighet for floatglass

$$\hat{p}_i = \frac{e^{3.9 - 3.06\,Al}}{1 + e^{3.9 - 3.06 \cdot Al}}$$

$\Downarrow$

formen på kurven

$\hat{SE}(\hat{\beta}_j)$

$$\frac{\hat{\beta}_j - 0}{\hat{SE}(\hat{\beta}_j)}$$

$H_0: \beta_j = 0$ vs $H: \beta_j \neq 0$

p-verdier

95 % konf. int. for β_j

Tolkning av $\hat{\beta}_1$: odds

Odds er definert som ratioen mellom sannsynlighet for suksess og fiasko: $\frac{p_i}{1-p_i}$

p_i	0.1	0.2	0.5	0.7	0.9
odds	0.11	0.25	1	2.3	9

Resultat

Hvis vi øker verdien til kovariaten fra x_{i1} til $x_{i1}+1$ så blir den nye oddsen lik den gamle multiplisert med $e^{\hat{\beta}_1}$.

Dette er ranskelig å forstå!

Enkel logistisk regresjon

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	3.9141	1.232	3.176	0.001	1.499	6.330
Al	$\hat{\beta}_1$= -3.0656	0.924	-3.317	0.001	-4.877	-1.254

$e^{-3.0656} = 0.05$

Hvis Al øker med 1 $\Rightarrow$
odds ganges med 0.05.
Det blir mye mindre sannsynlig at det er floatglass ved høy Al

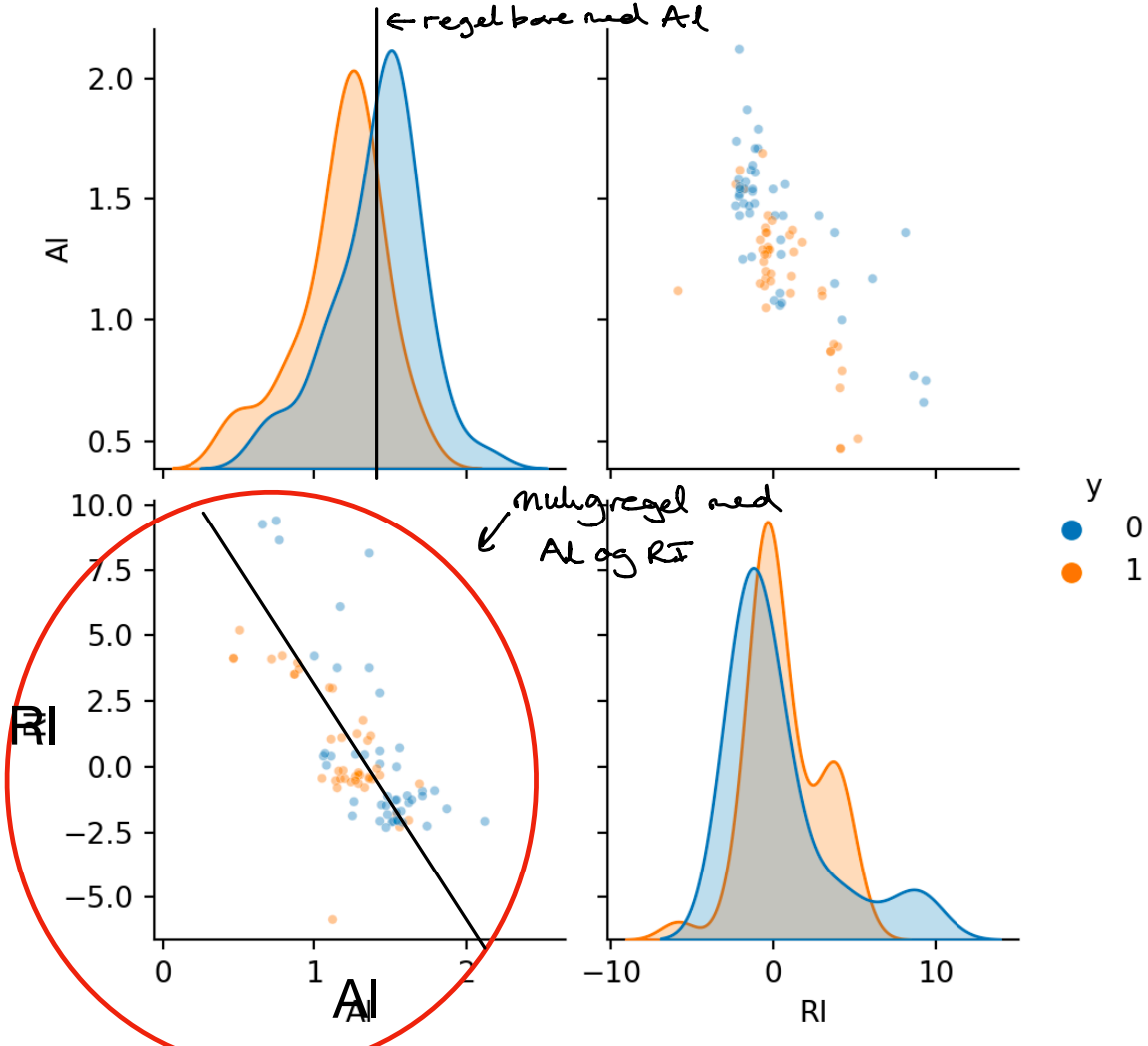
$H_0: \beta_1 = 0$ vs $H_1: \beta_1 \neq 0$

p-verdi 0.001 < 0.05 (sign.nivå)

Forkaster H_0 $\Rightarrow$

Al er viktig for å lage klassifikasjonsgrense.

Glass på åsted: Al og RI



Logistisk regresjon

Nytt: $p_i = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i}}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i}}}$

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	8.4824	2.203	3.851	0.000	4.165	12.800
Al ($\hat{\beta}_1$)	-6.4327	1.643	-3.915	0.000	-9.653	-3.212
RI ($\hat{\beta}_2$)	-0.4473	0.156	-2.874	0.004	-0.752	-0.142

$e^{\hat{\beta}_1} = 0.062$

$e^{\hat{\beta}_2} = 0.64$

$H_0: \beta_1 = 0$ mot $H_1: \beta_1 \neq 0$: p-verdi 0.000 → forkast H_0 → Al bør være med i modellen

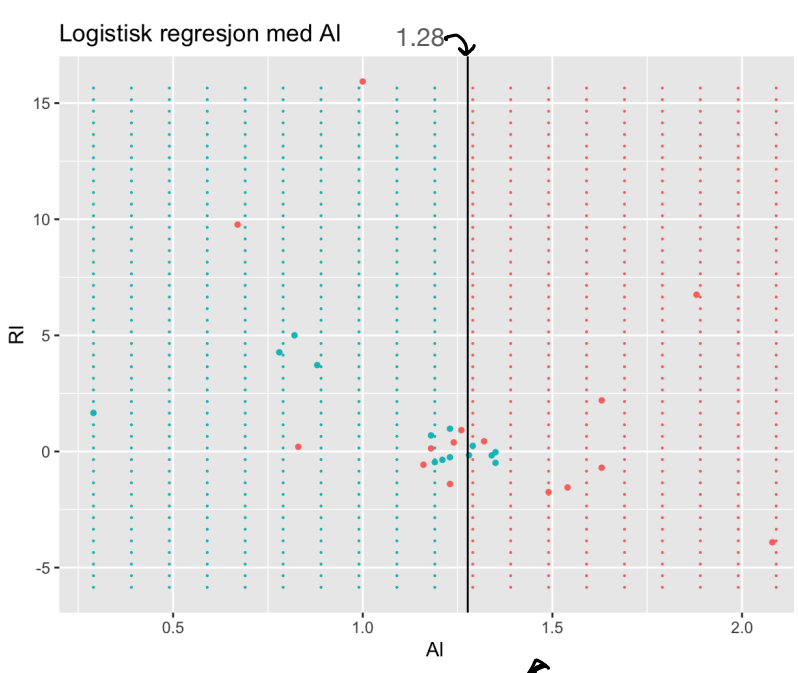
$H_0: \beta_2 = 0$ mot $H_1: \beta_2 \neq 0$: p-verdi: 0.004 → forkast H_0 → Ri bør være med i modellen

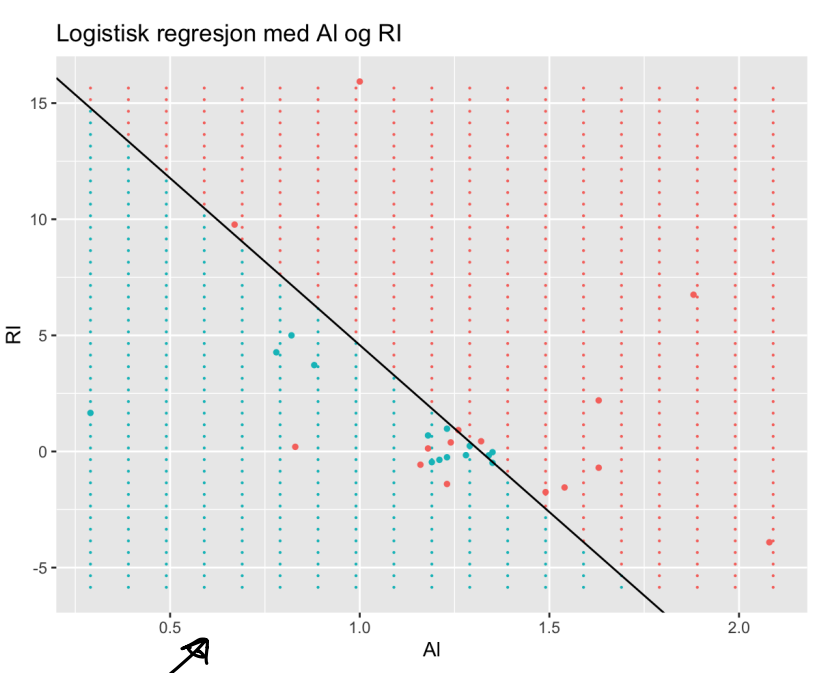
Men - hva med valideringssettet?

Skal vi velge modell med Al eller med Al og Ri?

Hvilken modell har lavest feilrate på valideringssettet?

(PGL)





tell opp antallet feilklassifiserte i valideringsdataen

Store prikker = valideringsdata

Små prikker = treningsdata

neste side

Hvor god er klassifikasjonsregelen?

Forvirringsmatrise med to klasser:

	Sann klasse 0	**Sann klasse 1**
Predikert klasse 0	Sann negativ (TN)	Falsk negativ (FN)
Predikert klasse 1	Falsk positiv (FP)	Sann positiv (TP)

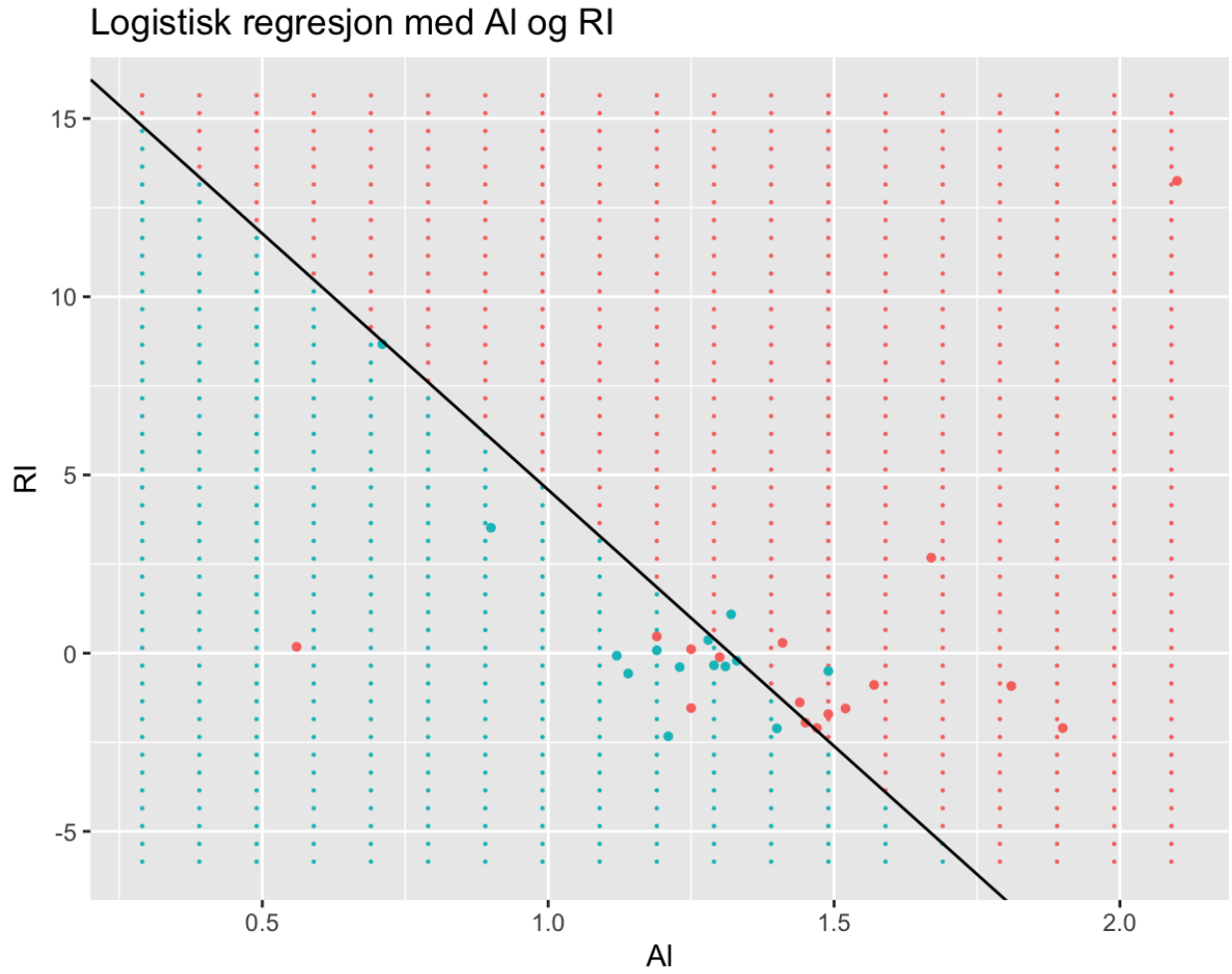
Feilrate: andel feilklassifiserte observasjoner

Bruker valideringssett for modelvalg:

1) Feilrate modell med bare AI: 0.45
2) Feilrate modell med AI og RI: 0.24

Velger modell med AI og RI!

Klassegrense og feilrate testsett



Feilrate for testsettet 27%

Logistisk regresjon i Python

```
formel='y ~ AI+RI'
modell = smf.logit(formel,data=df)
resultat = modell.fit()
resultat.summary()
print(np.exp(resultat.params))
```

```
cutoff = 0.5
test_pred = resultat.predict(exog = df_test)
y_testpred = np.where(test_pred > cutoff, 1, 0)
y_testobs = df_test['y']
print("Feilrate:", 1-accuracy_score(y_true=y_testobs,
y_pred=y_testpred,normalize=True))
```

trening/validering/test

hvorfor

hvordan bruke

viktig å tenke på

Forvirringsmatrise og feilrate

evaluere modell

velg mellom to modeller eller metoder

velge hyperparameter

tolke plott

boksplott histogram kryssplott korrelasjon

hva ser vi etter når vi vil lage en god klassifikasjonsregel?

k-nærmeste nabo (kNN)

forstå metoden

velge k

logistisk regresjon

tolke estimerte koeffisienter

hypotesetest og p-verdi

velge mellom modeller

Pensum: hva er spurt om i Oppgave 2

Videre denne og neste uken

- Ta med dere gruppa deres på digital veiledning!
- Jobb med Oppgave 2 på prosjektet!
- Still korte spørsmål i Forumet
- og lange spørsmål på digital veiledning mandag 8.15-10 og tirsdag 12.15-14 + egne tider Gjøvik.

Forberedelser til neste ukes zoom-time

- **Lenker og info er under uke 46: Klyngeanalyse**
- Se på videoene om Klyngeanalyse og K-gjennomsnitt (vent med videoen om hierarkisk klyngeanalyse) og bla i kompendiet, og
- titt på Prosjektoppgaven Oppgave 3 og tenk på om du har et bilde du vil analysere

Prosjektoppgaven

Vi ser hvor informasjonen ligger på Blackboard og hvordan melde seg på gruppe.
Snarvei til Bb 1003: https://s.ntnu.no/ist1003

Vi ser på prosjektoppgaven på https://s.ntnu.no/isthub

- Oppgave 1: Regresjon. 8 spørsmålspunkt.
- Oppgave 2: Klassifikasjon. 6 spørsmålspunkt.
- Oppgave 3: Klyngeanalyse. 6 spørsmålspunkt

Jobbe med prosjektoppgaven

Uke- tema- oppgave på prosjektet

Uke 43: Enkel lineær regresjon:
Oppgave 1: Q1.1-Q1.4

Uke 44: Multippel lineær regresjon:
Oppgave 1: Q1.5-Q1.8

Uke 45: Klassifikasjon
Oppgave 2

Uke 46: Klyngeanalyse
Oppgave 3