

Module 4: Classification

ISTx1003 Statistisk læring og Data Science

Stefanie Muff, Institutt for matematiske fag

November xy, 2021

Plan for i dag

- Læringsmål og ressurser
- Hva er klassifikasjon
- Trening, validering og testing (3 datasett)
- k -nærmeste nabo (kNN): en intuitiv metode
- Forvirringsmatrise og feilrate for å evaluere metoden
- Logistisk regresjon

Læringsmål

- kunne forstå hva klassifikasjon går ut på og kjenne 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, valg av k
- 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

Læringsressurser

- **Kompendium:** Klassifikasjon (pdf og html, by Mette Langaas)
- **Korte videoer:** (by Mette Langaas)
 - Introduksjon og k -nærmeste nabo klassifiasjon (10:58)
 - Logistisk regresjon (14:17)
- **Disse slides** med notater

Klassifikasjon – hva er det?

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

For hver av de uavhengige observasjonene $i = 1, \dots, n$ har vi

- Forklарingsvariabler $(x_{1i}, x_{2i}, \dots, x_{pi})$
- En kategorisk responsvariablel y_i .

Eksempler

- Hvilke tall er håndskrevet på brev? Og hva er sannsynligheten for de ulike tallene (=klasser).

A grid of handwritten digits from 0 to 9, arranged in a 10x10 pattern. The digits are written in various styles and sizes, representing a sample of handwritten digits.

- “Spam eller ham?” spam filer: Klassifikasjon om en e-post er spam eller ikke.

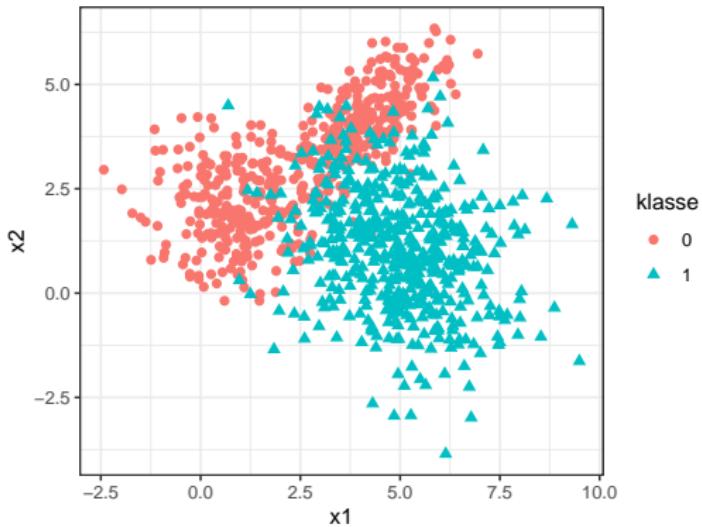
Binær: Respons variable er “ja”/“nei” (1/0).

- Kommer en gitt kunde til å betale tilbake lånet sitt?
- Prognose om noen blir syk (hjertesykdom, kreft...) og sannsynligheten for det.

Noe om tilsvarende oppgave i prosjektet

Syntetisk eksempel

- Et datasett med følgende struktur: (x_{1i}, x_{2i}, y_i) .



Spørsmål vi vil besvar: Hva vil beste klassifikasjonsgrense være?

Et eksempel til

Data

Datasett: $(x_{1i}, x_{2i}, \dots, x_{pi}, y_i)$

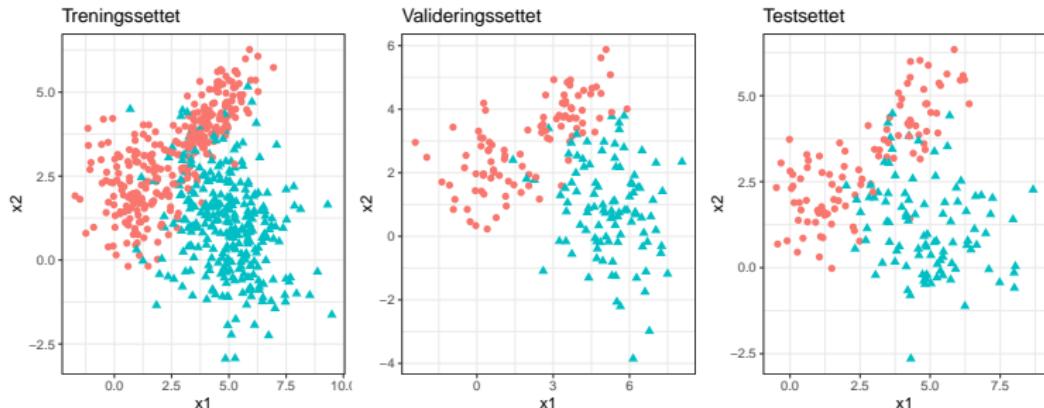
Vi må dele datasettet i tre deler:

- Treningsdata
- Valideringsdata
- Testdata



Hvorfor det?

Trening-, validering- og testsett for syntetiske data



k-nærmeste-nabo-klassifikasjon

For å finne klassifikasjonsreglen bruker vi bare treningsdataene.

Bruker bare treningsdataene. Algoritmen:

- 1) Ny observasjon: $x_0 = (x_{1,0}, x_{2,0}, \dots, x_{p,0})$. Hvilken klasse bør denne klassifiseres til?
- 2) Finn de k nærmeste nablene til observasjonen i treningssettet.
- 3) Sannsynligheten for at den nye observasjonen tilhører klasse 1 anslår vi er andelen av de k nærmeste nablene 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.

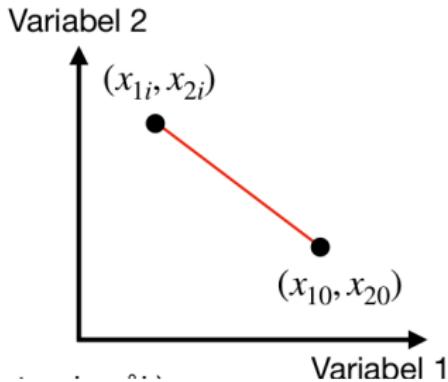
k -nærmeste-nabo-klassifikasjon

Tre spørsmål:

- Hva betyr “nærmest”? Vi trenger en definisjon av avstand.
- Hvilke verdier kan k ha?
- Hvordan bestemmer vi k ?

Hva betyr nærmest?

Nærmest er definert ved å bruke euklidsk avstand.

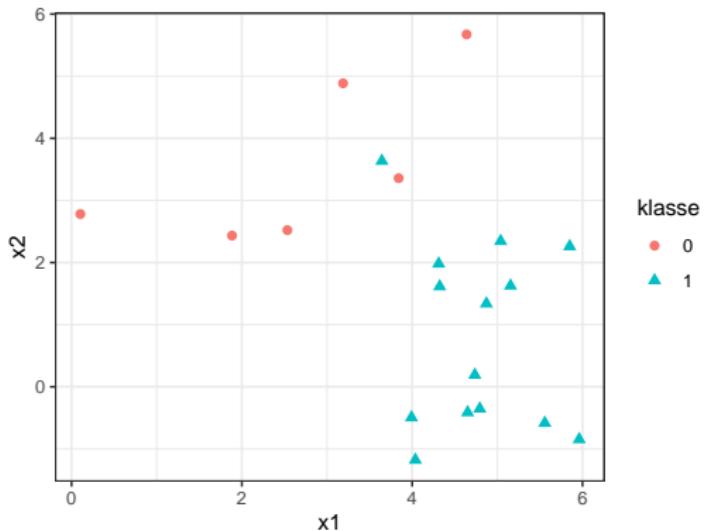


Euklidsk avstand:

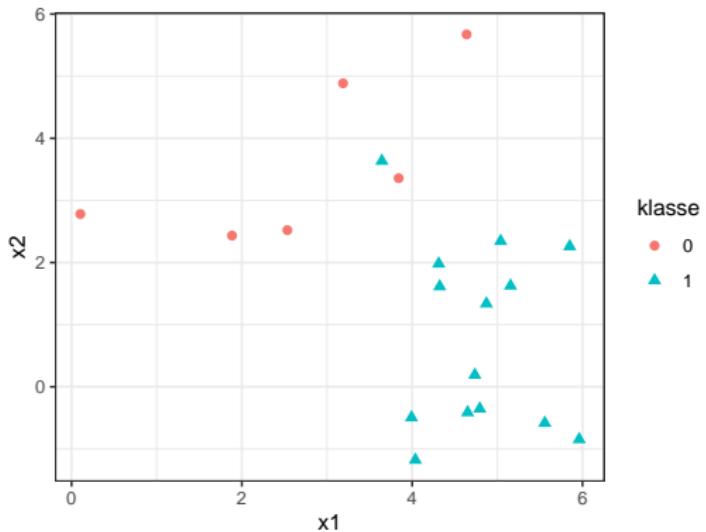
$$D_E(i, 0) = \sqrt{\sum_{j=1}^p (x_{ji} - x_{j0})^2}$$

Andre avstandsmål kan også brukes, men Euklidsk avstand er mest vanlig.

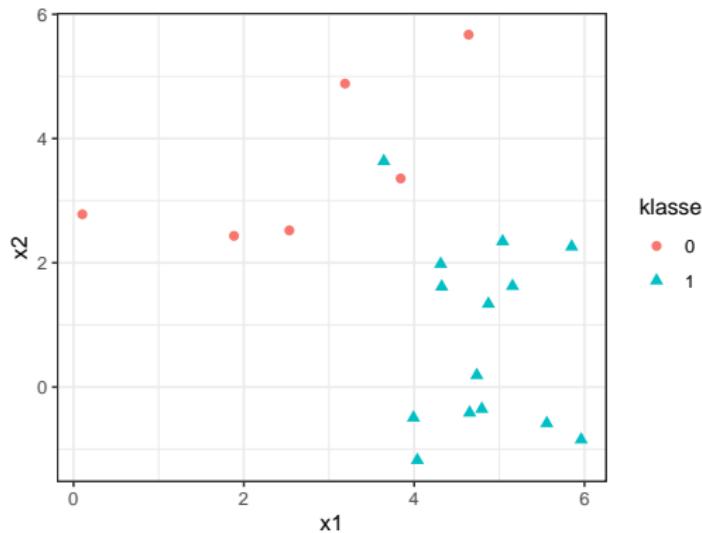
Nærmeste naboer



Nærmeste naboer

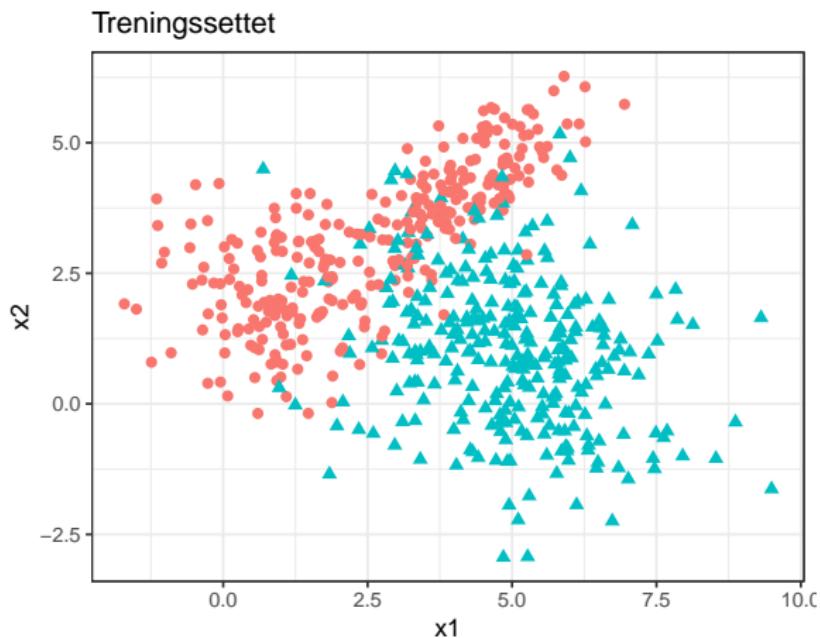


Tegn klassegrenser



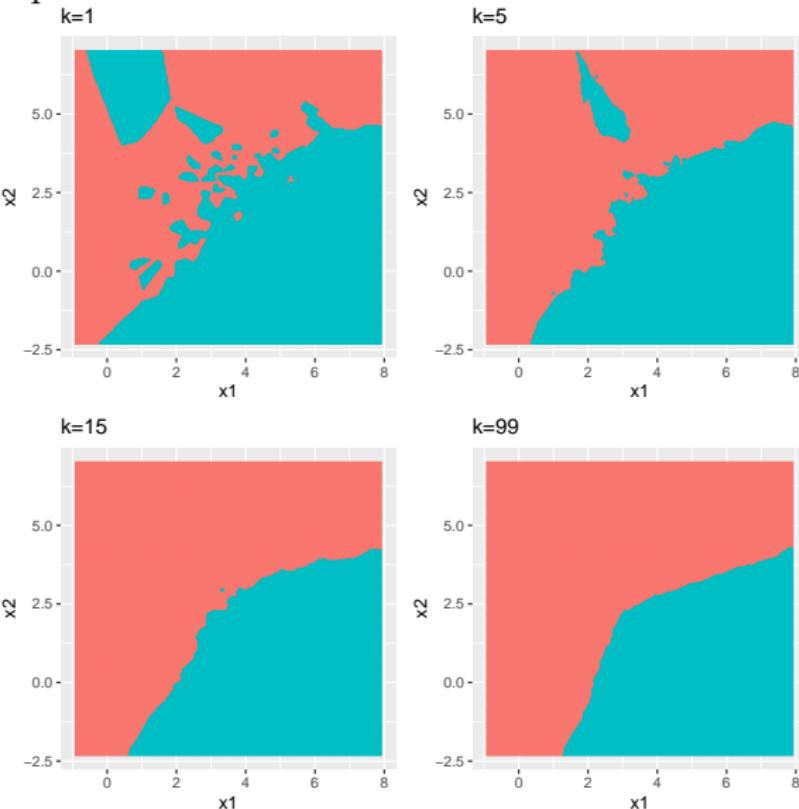
Hvordan ser klassegrensene ut?

Husk: Vi bruker treningssettet for å finne klassifikasjonsreglen:



Hvordan ser klassegrensene ut?

Svar: It depends!



Men vent... hvordan velger vi k da?

Vi så jo at

- Vi skal ikke velge k for liten, ellers blir grensen for fleksibel.

Men:

- Vi skal antakeligvis heller ikke velge k for stor, ellers kan bli for ufleksibel.

Det er noe som kalles en trade-off, og vi skal bruk valideringssettet for få en ide om kvaliteten i klassifikasjonen.

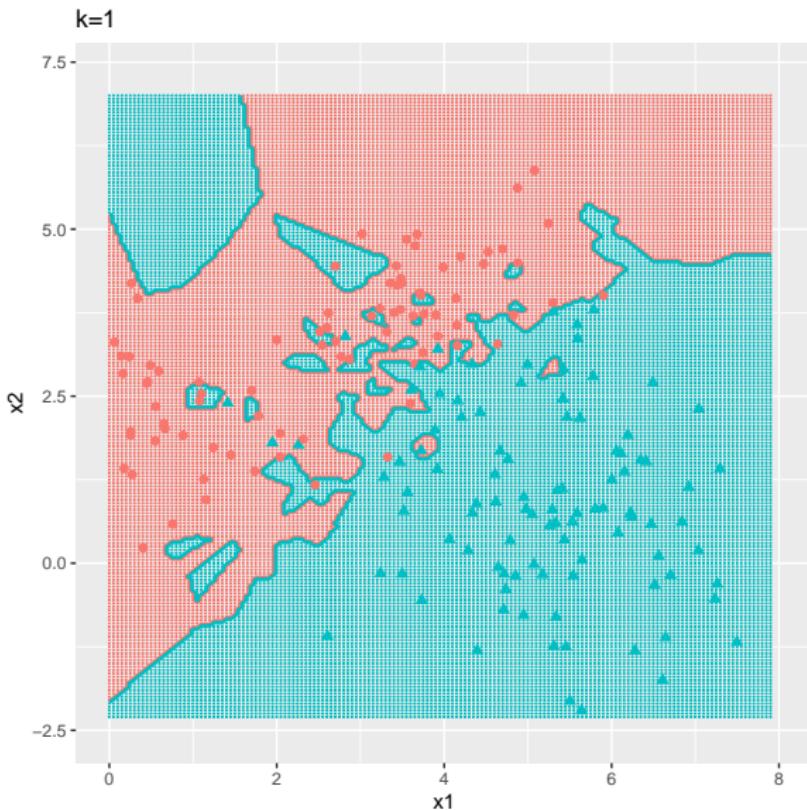
Forvirringsmatrise

Forvirringsmatrise med to klasser:

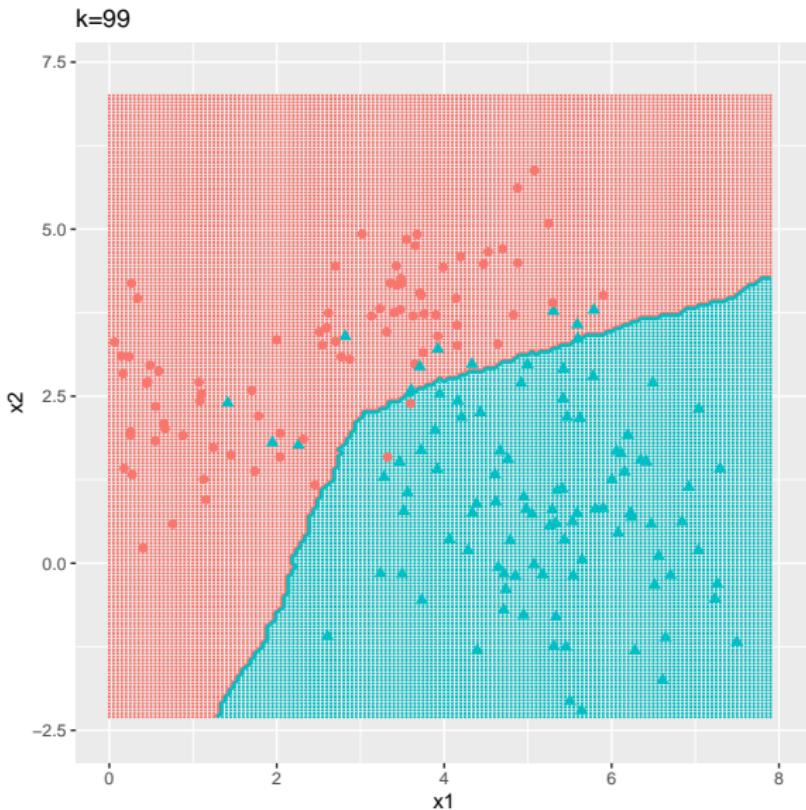
- **Feilrate:** andel feilklassifiserte observasjoner.

Derfor: Vi velger den k som minimerer feilraten på *valideringssettet* (ikke trainingssettet!).

Marker og tell antall gale klassifiseringer på valideringssettet ($k = 1$):

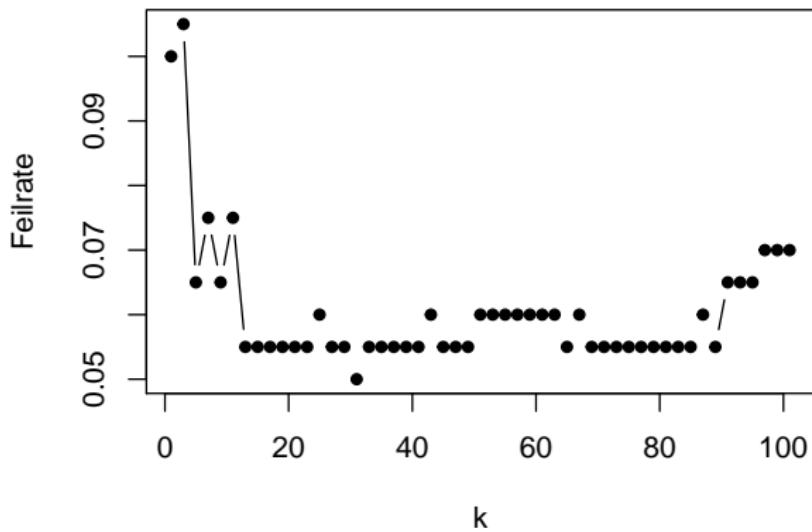


Igjen, men nå med $k = 99$:

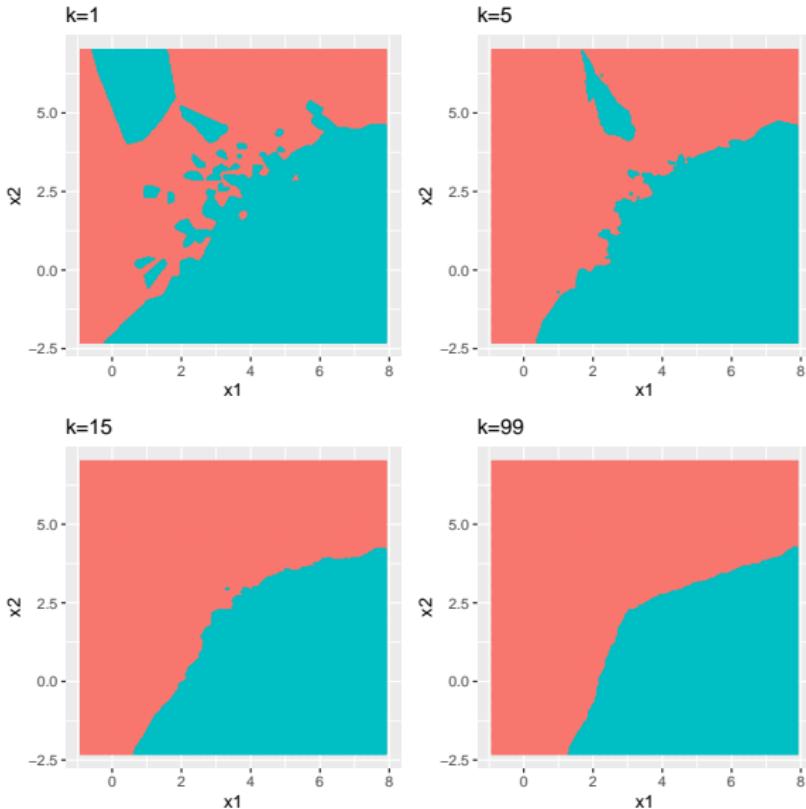


Feilrate i valideringssettet

Det kan nå gjøres med alle $k = 1, 3, \dots, k \leq n$. De ser slik ut:



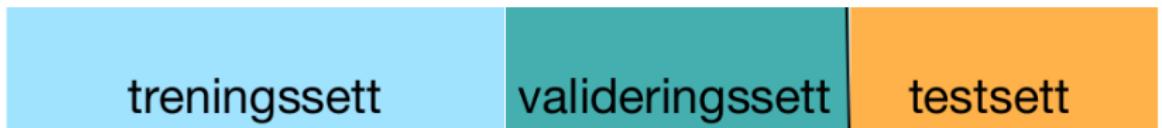
Klassegrensene ser ganske likt ut for $k = 15$ og $k = 99$:



Data

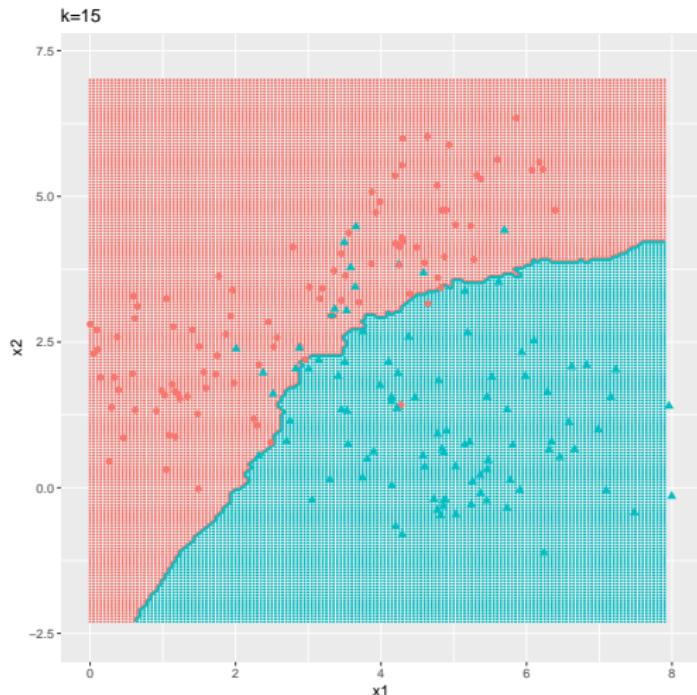
Husk at vi hadde tre deler i datasettet:

- Treningssett: For å lage en klassifikasjonsregel
- Valideringssett: For å finne på optimale hyperparametere
- Testsett: For å evaluere regelen på fremtidige data



Nå kan vi bruk testsettet for å finne feilraten.

Feilraten på testsettet for KNN med $k = 15$ (egentlig er $k = 31$ best, men det gjør ikke en stor forskjell):



Feilraten er 0.09.

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'])
```

Todo (adapt to the code they need in oppgave 2)

Plott feilrate:

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

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']))

Pensum for oppgave 2 – hvor er vi nå?

- trening/validering/test
- hvorfor
- hvordan bruke
- viktig å tenke på
- tolke plott
- boksplott histogram
- kryssplott korrelasjon
- hva ser vi etter når vi vil lage en god klassifikasjonsregel?
- Forvirringsmatrise og feilrate
- evaluere modell
- velg mellom to modeller eller metoder
- velge hyperparameter
- k -nærmeste nabo (kNN)
- forstå metoden
- velge k
- logistisk regresjon
- tolke estimerte koeffisienter
- hypotesetest og p-verdi
- velge mellom modeller

Plan for i dag

- Læringsmål og ressurser
- Hva er klassifikasjon
- Trening, validering og testing (3 datasett)
- k -nærmeste nabo (kNN): en intuitiv metode
- Forvirringsmatrise og feilrate for å evaluere metoden
- Logistisk regresjon

Logistisk regresjon

- Kan bare handtere *to kasser* $y_i \in \{0, 1\}$.
- Vi antar at Y_i har en **Bernoulli fordeling** med suksessannsynlighet p_i , derfor:

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

- **Mål:** For forklaringsvariabler $(x_{1i}, x_{2i}, \dots, x_{pi})$ vi vil estimere $p_i = \Pr(y_i = 1 \mid x_1, \dots, x_p)$.

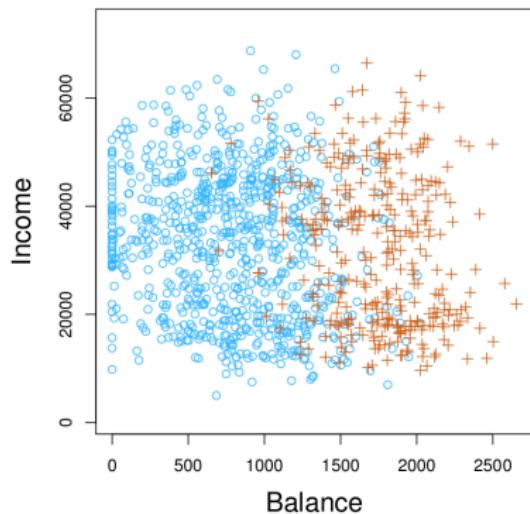
Eksempel: Kredittkort data

Datasettet **Default** er tatt fra her:

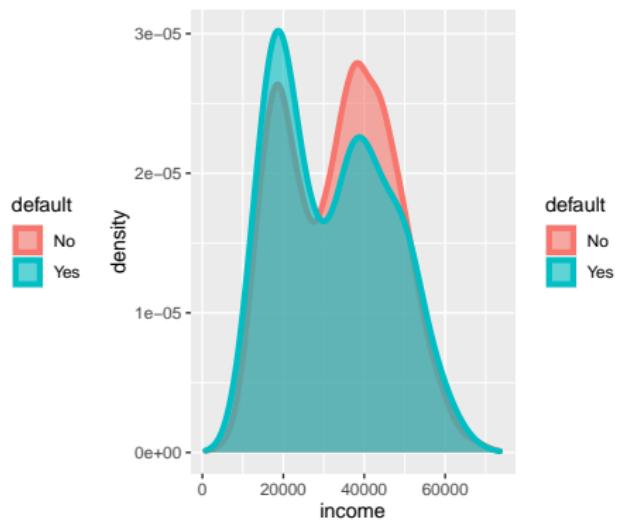
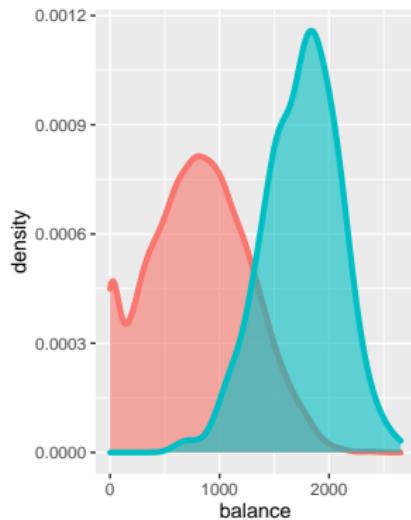
<https://rdrr.io/cran/ISLR/man/Credit.html>

Mål : forutsi om en person ikke betaler kredittkortregning (“person defaults”), avhengig av årsinntekten (income) og balansen på kredittkortet (balance).

Orange: default=yes, blue: default=no.



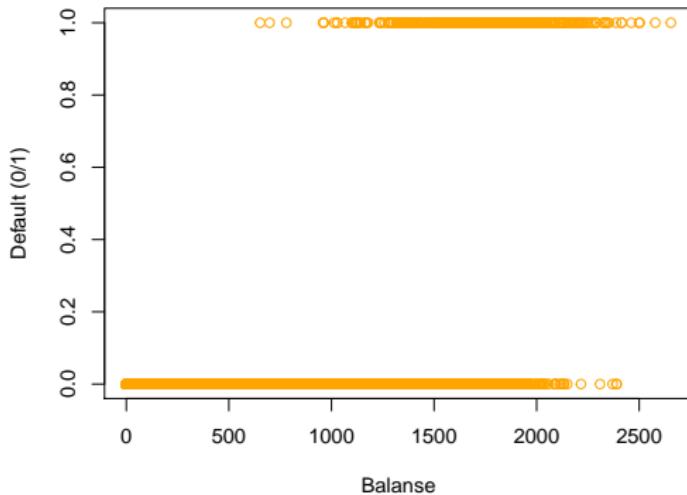
Det ser ut som at “Balance” er en ganske god forklarende variabel til Default (nei/ja).



Kan vi bare bruke linear regresjon for binær klassifikasjon?

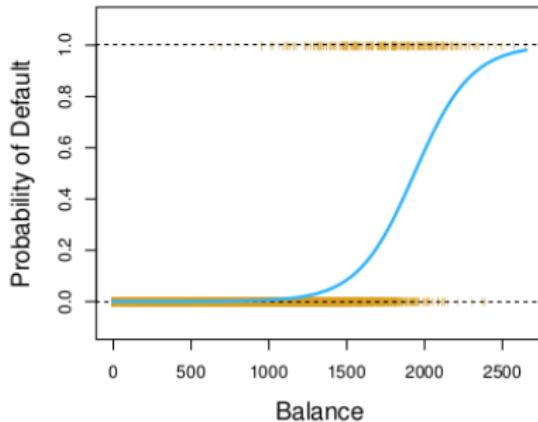
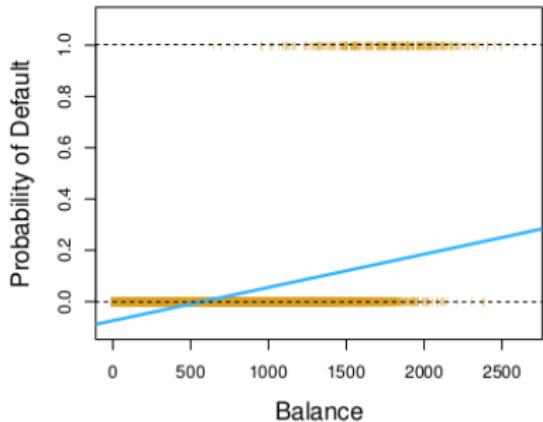
For en binær responsvariabel $Y = \text{yes}$ or no , og forklaringsvariabler X bruker vi vanligvis en *dummy encoding* :

$$Y = \begin{cases} 0 & \text{if no ,} \\ 1 & \text{if yes .} \end{cases}$$

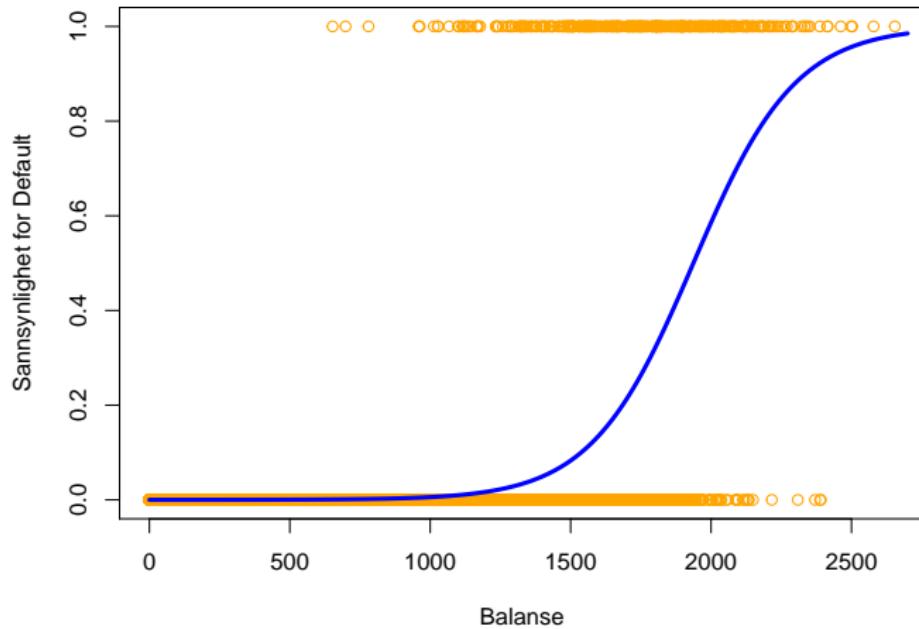


- Kunne vi da ikke bare formulere en linear regresjonsmodell for X vs. Y ?
- Vi kunne jo klassifisere response som “ja” (1) hvis $\hat{Y} > 0.5$.
- Problemet med linear regresjon: Vi kan forutsi $Y < 0$ eller $Y > 1$ med modellen, men en sannsynlighet er alltid mellom 0 og 1.

For kredittorddatasettet:



Vi trenger logistisk regresjon!



Enkel logistisk regresjon

- Vi antar at responsen Y_i er binomisk fordelt med suksessannsynlighet p_i
- **Ideen:** å koble p_i sammen med forklaringsvariablen med en logistisk funksjon:

$$p_i = \frac{\exp(\beta_0 + \beta_x x_{1i})}{1 + \exp(\beta_0 + \beta_x x_{1i})}$$