

Lynkurs i matte + Logistisk regresjon (del 2)

Pierre Lison
plison@nr.no

IN2110: Språkteknologiske metoder
27. februar 2024



Oblig 1b

► Hust Oblig 1b! **Innleveringsfrist:** 18. mars, kl. 23:59

► To deler:

1. Logistisk regresjon

2. Sekvensmodeller

Angstgefühlen	'ʔan,kstge:fylən	→	tysk
réaionalisations	ʁeʒionalizasiɔ̃	→	fransk
tilbøriligste	tɪl'bø:ɭɪkstə	→	norsk
ين اي بكف	fakabeɪa:nijj	→	arabisk
果	kɑʊ˥˥	→	mandarin
justifiable	dʒ'ʌstɪf, aɪəbəl	→	engelsk

= Gitt den *fonetiske transkripsjonen* av et ord, modellen skal predikere *hvilket språk* ordet hører til

“Den formelle eieren av krokodillene er René Hedegaard i Krokodillezoo i Danmark”



“Den formelle eieren av krokodillene er <PER>René Hedegaard</PER> i <ORG>Krokodillezoo</ORG> i <GPE>Danmark</GPE>”

= Gitt en tekst, modellen skal finne ut de *navngitte entitetene*



På en mandag
Inga Strømke
Førsteamanuensis, Norwegian Open AI Lab, NTNU

Kjære student: KI-verktøy lurar oss begge, men det er deg det går ut over



Med KI-verktøy
kommer de
flinke til å bli
enda flinkere, og
de dårlige
kommer til å bli
flere

jobben min lærer jeg opp to vidt forskjellige vesener: studenter innen datateknologi - og kunstig intelligens-modeller (KI). Og selv om det er noe forskjeller mellom dere, er det jammen noen likheter også.

Både mennesker og KI-modeller trenger erfaringer å lære fra og en vurdering av læringsutbyttet. Men forskjellene mellom dere gjør det mye enklere å vurdere læringen til en KI-modell enn en menneskelig student.

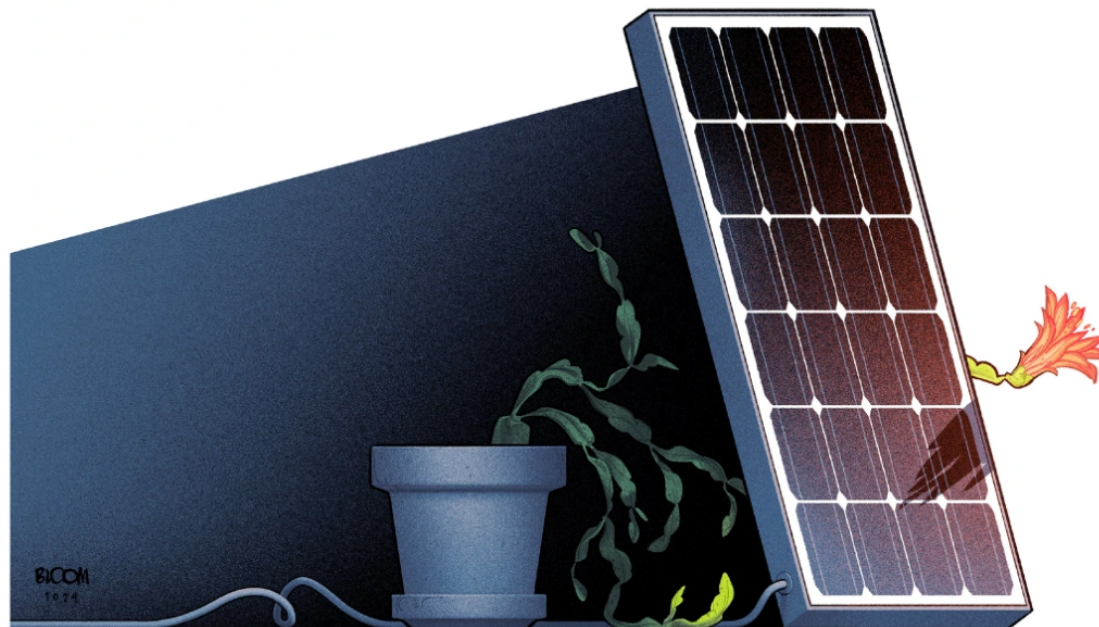
KI-modeller trenger ikke å motiveres, og det er utvetydig hvilken oppgave de skal utføre. ChatGPT skal skrive tekst. Alpha-zero skal spille sjakk. Dall-E skal lage bilder. AlphaFold skal beregne proteinstrukturer.

Dere studenter skal derimot tegne dere evner som kan brukes til å løse problemer vi ikke aner hva kommer til å være, i fremtiden.

Dere skal fungere i et samfunn i stadig endring, bidra til å løse enorme problemer som klimakrisen og energimangel, opprettholde et velfungerende demokrati og leve et lykkelig liv. Ja, og så skal dere fungere i en digital verden der hva som helst kan være falskt og KI-generert.

Heldigvis er det ikke jeg som skal lære dere alt dette. Men til sammen kommer årene dere tilbringer i utdanningssystemet til å prege alle disse områdene.

Kronen på verket etter endt utdanning



Tegning: Christian Bloom

er en grad, og den oppnår dere etter lopende å ha blitt vurdert gjennom hele utdanningen.

Umulig å bevise. Men hvordan skal jeg vurdere læringen deres nå som det er umulig for meg å vite om dere har brukt KI-verktøy til å skrive alt fra kode til rapporter?

For ja, det er bokstavelig talt umulig å bevise at noen har brukt generativ KI til å skape en tekst, en kodesnutt, et bilde eller en lydfil med mindre KI-verktøyet selv lager et vannmerke.

Det er langt ifra alle verktøyene som lager vannmerker, og pr. i dag er disse dessuten enkle å fjerne. Videre holder det ikke om en lærer eller sensor sier «det er 99 prosent sannsynlig at denne teksten er skrevet av ChatGPT». Skal du ta noen for juks, må du kunne påvise det med sikkerhet.

Når vi snakker om KI-verktøy i utdanningen, handler 90 prosent av diskusjonen om vurdering. Det er synd, for det viktigste for meg er at dere lærer. Men rart er det ikke, for vurderingen forteller meg hva dere har lært - i alle fall innenfor dagens system.

Stor samfunnsomveltning. Kunstig intelligens har særlig i løpet av de siste to årene markert seg som en disruptiv teknologi, altså en teknologi som fører til stor samfunnsomveltning. En interessant egenskap ved disruptive hendelser er at ting som før var implisitte, blir eksplisitte.

Norge under pandemien er et godt eksempel på dette. Det ble åpenbart at skolen er en viktig sosial arena og hvor avhengige vi er av utenlandske arbeidere.

Disse implisitte faktumene ble brått eksplisitte og krevde at vi forholdt oss til dem. Det samme gjør kunstig intelligent teknologi med skole- og utdanningssektoren vår nå.

Vi har lenge brukt vurdering av studenter produkter for å måle læring. Den implisitte antagelsen om at evnen til å skape et produkt er en god målestørrelse på kunnskapen studenten har tilegnet seg, er ikke lenger gyldig når studenten bruker KI-verktøy.

Målet med høyere utdanning. En interessant likhet mellom mennesker og KI-modeller som lærer, er at begge gjør hva enn som gir belønning. Og KI-sikkerhetsforskning har pekt på at blind optimalisering mot belønning kan føre til enorme problemer.

Et tullede tankeeksperiment som illustrer dette, innebærer en intelligent vaskerobot som får belønning så lenge bollen holdes ren. Vaskerobotten oppdager raskt at barn og hunder fører til mest urenheter og bestemmer seg derfor for å kvitte seg med både barn og hunder. Det eksplisitte målet om en ren bolig oppnås, men på bekostning av noe langt viktigere: det implisitte målet om at ingen i husstanden skal ofres.

Det er når det eksplisitte målet står i konflikt med det implisitte målet at ting blir virkelig spennende.

Utdanningssektoren lider, som Tore Wig skriver i Morgenbladet, «under et økende fokus på gjennomstrømming og avlagte studiepoeng». Uansett hvor mye jeg fokuserer på at studentene mine skal lære, handler utdanningssektorens eksplisitte mål om vurdering.

Vi skal lage flest mulig studenter med høyest mulig grader. Men hva er egentlig målet med høyere utdanning: er det å dele ut kunnskap eller å dele ut mastergrader?

Inntil nylig var en godt skrevet tekst et tegn på skriveferdigheter og en kodesnutt et tegn på programmeringsevner. Slik er det ikke lenger. Utdanningssektoren må ta inn over seg hvor store endringer som potensielt kan bli nødvendige for å motivere til læring og muliggjøre vurdering av tilegnet kunnskap.



Hva er egentlig målet med høyere utdanning: er det å dele ut kunnskap eller å dele ut mastergrader?

Bruke verktøyene riktig. Kjære student. Du tilhører den første generasjonen mennesker som aldri kommer til å være bedre enn verktøyene våre. Du kommer aldri til å ha lest like mye tekst som ChatGPT er trent på, til å kunne skape et like stort mangfold bilder like raskt som Dall-E, eller ha sett like mange eksempler på programkode som Copilot.

Dette kan være nedslående og demotiverende ved første øyekast, men det betyr også at potensialet for problemløsning og skaping aldri har vært større. Du tilhører den første generasjonen problemløsere som ikke vil være begrenset av menneskelige intelligens.

Hvis du bruker verktøyene riktig, kan du få et læringsutbytte som kun undervisning med privatlærer kan måle seg med. Men hva som ligger i «bruke riktig», er det ingen som vet ennå. Dette må vi som samfunn finne ut, og prosessen vil være umulig å gjøre risikofri.

Men én ting er jeg sikker på: Jo mindre kunnskap du har på et område, desto større er sannsynligheten for at et KI-verktøy vil gi deg illusjonen av å ha lært noe uten at du egentlig har lært noe som helst.

Rosker i samfunnets rotter. Jeg er ikke bekymret for at verdens professorer skal bli dårligere i fagene sine på grunn av kunstig intelligens. Jeg er også lite bekymret for at master- og doktorgradstudenter skal la KI-verktøy gjennomføre vitenskapelige eksperimenter for dem.

Men jeg er fryktelig bekymret for at første- og annetårsstudenter skal tro at de har lært noe idet diverse KI-verktøy hjelper dem litt for godt gjennom øvinger og innleveringer.

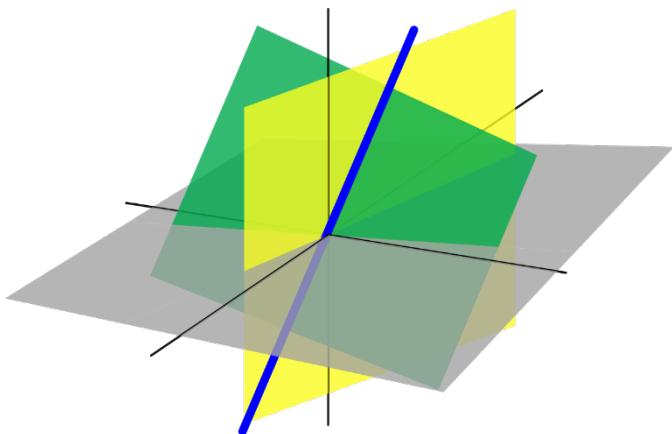
Med KI-verktøy kommer de flinke til å bli enda flinkere, og de dårlige kommer til å bli flere.

Disruptive hendelser rosker i samfunnets rotter. Tilgangen til intelligent teknologi krever at vi reflekterer over hva vi egentlig prøver å oppnå med målene vi har satt oss. For å klare dette må vi innse at det ikke bare handler om verktøyene vi får tilgang til, men hva vi blir gjennom å bruke dem.

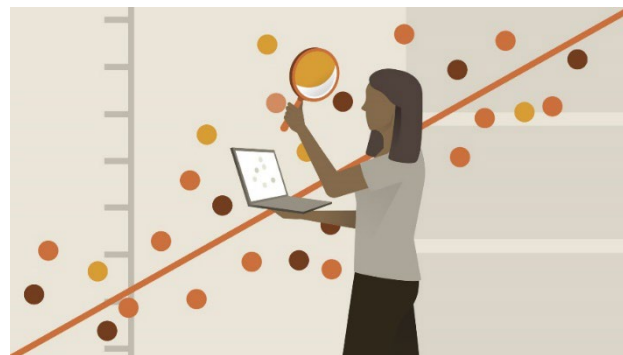


Jeg er fryktelig bekymret for at første- og annetårsstudenter skal tro at de har lært noe idet diverse KI-verktøy hjelper dem litt for godt gjennom øvinger og innleveringer

I dag skal vi snakke om 2 temaer:

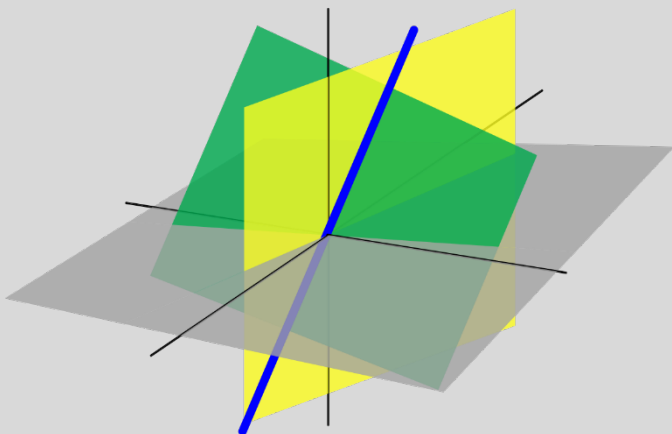


Lynkurs i lineær algebra
og sannsynlighetsteori

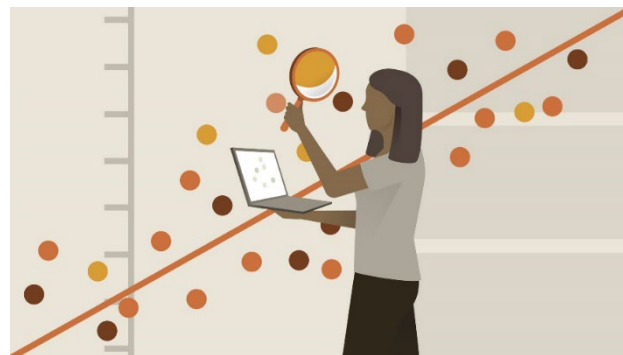


Logistisk regresjon
(del 2): optimering,
regularisering og
forklarbarhet

I dag skal vi snakke om 2 temaer:



Lynkurs i lineær algebra
og sannsynlighetsteori



Logistisk regresjon
(del 2): optimering,
regularisering og
forklarbarhet

Lynkurs i lineær algebra

"Unfortunately, no one can be told what the Matrix is. You have to see it for yourself."

- Morpheus, "The Matrix"



Matematiske objekter

► **Skalar:** et enkelt tall, f.eks. -3.56

► **Vektor:** en rekke tall (av lengde m), f.eks. $\begin{bmatrix} -3.4 \\ 6.2 \\ \dots \\ 0.3 \end{bmatrix}$ Lengde m

► **Matrise:** en tabell (med m rad og n kolonner):

Antall kolonner n $\xrightarrow{\hspace{2cm}}$

Antall rad m $\downarrow$

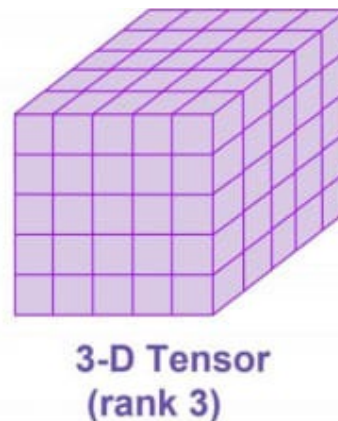
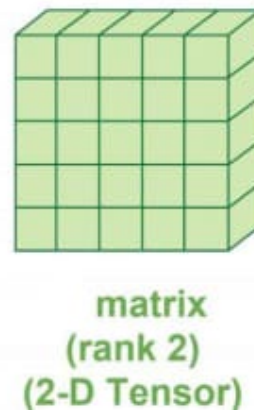
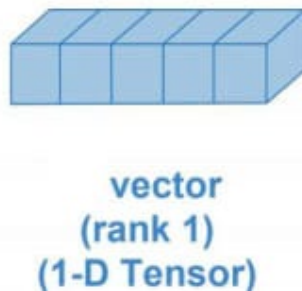
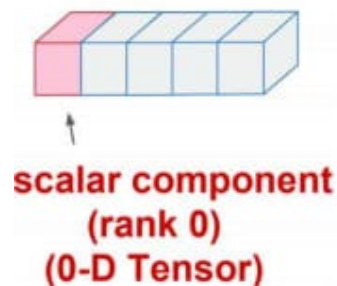
-1.4	4.5	...	2.3
8.4	0.2	...	0.5
$\vdots$	$\vdots$	$\ddots$	$\vdots$
2.8	1.3	...	-4.9

Celle (2,n)

Matematiske objekter

Man kan generalisere disse til **tensorer**:

- En skalar er en tensor med 0 dimensjon
- En vektor er en tensor med 1 dimensjon
- En matrise er en tensor med 2 dimensjoner
- Osv.



Matematiske objekter

- ▶ I maskinlæring bruker man tensorer (vektorer, matriser, osv.) for å representere både **data** og **modelparametre**
- ▶ **Eksempel:** en video (uten lyd) = en 4-dimensjonal tensor :
 - Dimensjon 1 er tid (med "sampling" på en viss frekvens, f.eks. 50 frames/sek)
 - Dimensjon 2 og 3 er X- Y-akse i bildene
 - Dimensjon 4 er RGB verdiene



Blue					
Green					
Red					
	255	134	93	22	
	255	134	202	22	2
255	231	42	22	4	30
123	94	83	2	92	124
34	44	187	92	14	142
34	76	232	124	14	
67	83	194	202		

Operasjoner

► Addisjon:
$$\begin{bmatrix} -3.4 \\ 6.2 \\ \dots \\ 0.3 \end{bmatrix} + \begin{bmatrix} -0.4 \\ -1.2 \\ \dots \\ 2.4 \end{bmatrix} = \begin{bmatrix} -3.8 \\ 5.0 \\ \dots \\ 2.7 \end{bmatrix}$$

► Multiplikasjon med skalar:
$$2 * \begin{bmatrix} -3.4 \\ 6.2 \\ \dots \\ 0.3 \end{bmatrix} = \begin{bmatrix} -6.8 \\ 12.4 \\ \dots \\ 0.6 \end{bmatrix}$$

► Elementvis produkt:
$$\begin{bmatrix} -2 \\ 1 \\ \dots \\ 0.3 \end{bmatrix} \circ \begin{bmatrix} -0.4 \\ -1.2 \\ \dots \\ 2.4 \end{bmatrix} = \begin{bmatrix} 0.8 \\ -1.2 \\ \dots \\ 0.72 \end{bmatrix}$$

Elementvis
produkt $\neq$
matriseprodukt!
(mer om det
snart)

Prikkprodukt

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i$$

► Regneprosedyre:

- vi tar to vektorer (av samme lengde)

$$\mathbf{a} = \begin{bmatrix} -0.4 \\ -1.2 \\ 2.4 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} -3.4 \\ 6.2 \\ 0.3 \end{bmatrix}$$

$$-0.4 * -3.4 = 1.36$$

$$-1.2 * 6.2 = 5.0$$

$$2.4 * 0.3 = 0.72$$

- For hver posisjon i **ganger vi** a_i og b_i
- Og til slutt **legger vi sammen** alle tallene:
$$= (-0.4 * -3.4) + (-1.2 * 6.2) + (2.4 * 0.3) = -5.36$$

► Prikkproduktet er nyttig for å bl.a.:

- Sammenligne vektorer (Egil sine forelesninger)
- Representere vektete summer (som i logistisk regresjon)

Matriseprodukt

- ▶ Prikkprodukt er mellom vektorer, men det finnes en lignende operasjon for matriser eller tensorer
- ▶ Matriseproduktet er definert mellom to matriser A og B hvor A har dimensjon (m,n) og B er (n,p)



$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Antall kolonner i A må være lik antall rader i B!



$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{bmatrix}$$

Regneprosedyre for matriseprodukt

- ▶ Man tar en rad i A og en kolonne i B (altså to vektorer), og beregne **prikkproduktet** mellom de to
- ▶ Vi gjentar operasjon for alle kolonner 1,...,p i B
- ▶ Og for alle radene 1,...,m i A

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

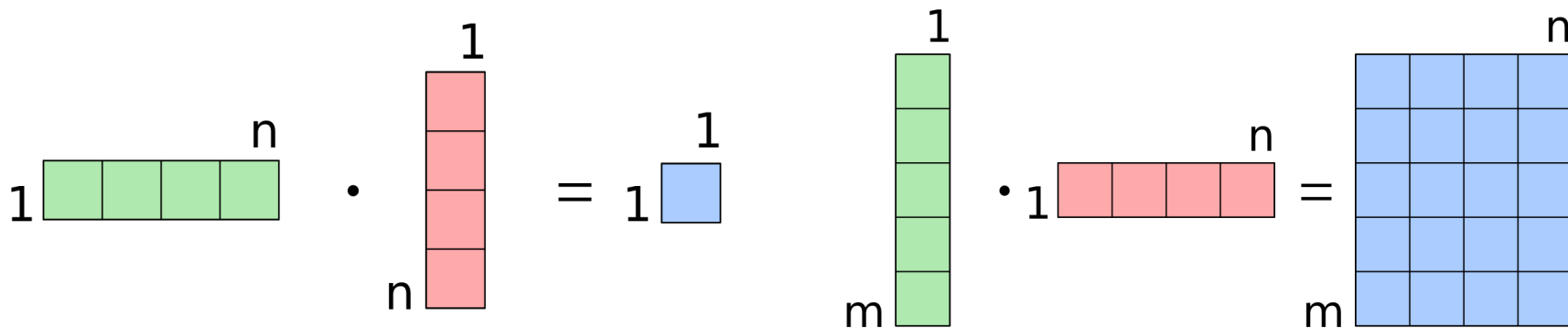
$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{bmatrix}$$

$$\rightarrow C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{bmatrix}$$

hvor hver $c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$
(altså prikkproduktet mellom raden a_i . og kolonnen b_j)

Matriseprodukt

- ▶ Prikkprodukt er en spesielt tilfelle av matriseprodukt (med kun én dimensjon)
- ▶ Merk at matriseproduktet *ikke* er kommutativ, dvs. at rekkefølgen av matrisen er viktig: $AB \neq BA$
- ▶ Retningen (vertikal eller horisontal) er også viktig:



Øvelser

- ▶ Gitt to matriser A og B:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad B = \begin{pmatrix} 5 & -1 \\ 1 & 0 \\ -2 & 3 \end{pmatrix}$$

- ▶ Spørsmål 1: Hva er dimensjon til $C = AB$?
- ▶ Spørsmål 2: Regn ut C

$$C = \begin{bmatrix} 1 & 8 \\ 13 & 14 \end{bmatrix}$$

Litt lineær algebra

Hvorfor er det nyttig?

- Vi kan skrive formelen for logistisk regresjon slik:

skjæringspunktet

$$P(y = 1) = \sigma(w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b)$$

Sannsynlighet for klassen $y=1$ $\sigma(z) = \frac{1}{1 + e^{-z}}$ Input $\mathbf{x}$ med n (numeriske) features

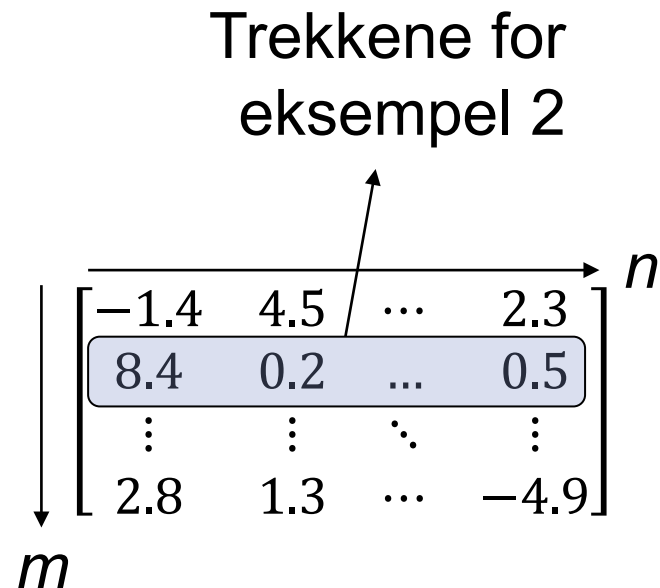
- Men det er lettere slik:

$$P(y = 1) = \sigma(\mathbf{w}^T \mathbf{x} + b) \quad \text{hvor } \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{bmatrix} \quad \text{and } \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

Litt lineær algebra

- ▶ Hvis vi har en datasett $\mathbf{X}$ med m eksempler, og hvert eksempel har n trekk, kan vi representere den som en matrise (m,n) :

Trekkene for eksempel 2



	-1.4	4.5	...	2.3
	8.4	0.2	...	0.5
	⋮	⋮	⋮	⋮
	2.8	1.3	...	-4.9

m n

- ▶ I obligen 1b vil dere trene en språkklassifikator, og treningsettet vil da representeres som et par $(\mathbf{X}, \mathbf{Y})$ hvor:
 - $\mathbf{X}$ er en inputmatrise (m,n) , hvor m er antall eksempler (ord), og n antall (fonetiske) trekk for hvert eksempel
 - $\mathbf{Y}$ er en vektor av lengde m som fastsetter outputklassen for hvert eksempel

Lynkurs i sannsynlighetsteori

- ▶ Hvorfor bruker vi sannsynligheter?
- ▶ Fordi de gjør det mulig å **uttrykke usikkerheter** og **resonnere seg fram** basert på usikker kunnskap
- ▶ Og usikkerhet er overalt i språkteknologi (og AI generelt)!



Pierre-Simon de Laplace
(1749-1827)

*«Probability is nothing but common sense
reduced to calculation»*

*«The true logic for this world is the
calculus of probabilities»*



James Clerk Maxwell
(1831-1879)

Sannsynligheter

- ▶ En *sannsynlighetsmassefunksjon* P (probability mass function, eller PMF) gir oss sannsynligheten knyttet til en bestemt verdi eller hendelse:
 - $P(\text{terning}=4) = 1/6$
 - $P(\text{ord}=\text{"kvantemekanikk"}) = 1.2\text{E-}12$
- ▶ Aksiomer:
 - $0 \leq P(X = x) \leq 1 \quad \forall \text{ mulige verdier } x \in X$
 - $\sum_{x \in X} P(X = x) = 1$

Dette her en veldig uformell innføring! En grundigere gjennomgang burde starte med *utfallsrom* og hvordan *tilfeldige variabler* kan defineres basert på disse

Felles og betingede sannsynligheter

- ▶ Vi må ofte jobbe med flere variabler samtidig, som for eksempel $P(ord_{t-1}="en", ord_t="maler")$
 - Det kalles en **felles sannsynlighet** (joint probability), og skrives $P(X, Y)$
 - Hvis de to hendelser er uavhengig, $P(X, Y) = P(X) P(Y)$
- ▶ Et annet begrep er den **betingede sannsynligheten** $P(X | Y)$ for X gitt en annen informasjon Y ("evidensen")
- ▶ Felles og betingede sannsynligheter er tett koblet:
$$P(x, y) = P(x | y) P(y) , \text{ eller omvendt: } P(x | y) = \frac{P(x, y)}{P(y)}$$

Marginalisering

- ▶ La oss si at vi har felles sannsynligheter $P(X, Y)$, og vi ønsker å beregne $P(X)$. Hvordan gjør vi det?
- ▶ Vi kan bruke **marginalisering** (også kalt "summing out"):

$$P(X) = \sum_{y \in Y} P(X, Y = y)$$

Øvelse

Sannsynligheten for regnvær i dag er 55% hvis man er i Bergen, og 30% hvis man er et annet sted i Norge

Hva er sannsynnighet for at det skal regne, gitt at det er 15% sjanse for at jeg befinner meg i Bergen i dag?

Bruk formler for å komme til svaret

Husk:

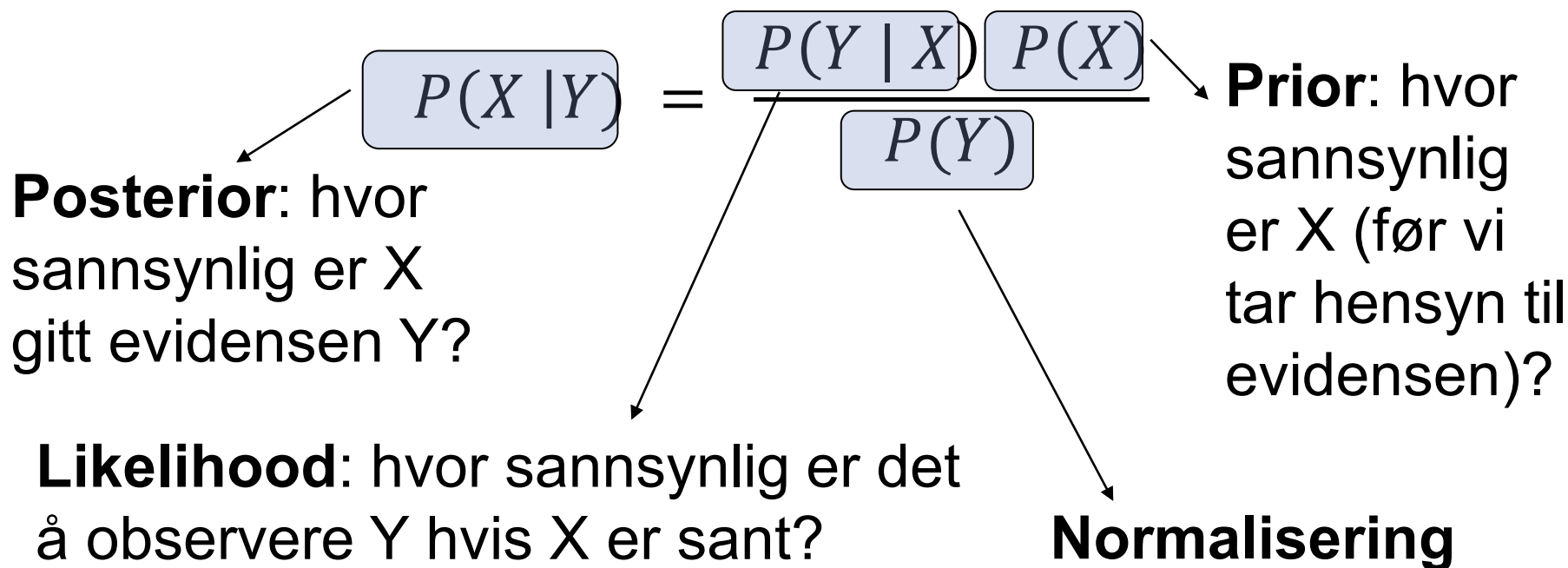
$$P(X, Y) = P(X | Y) P(Y)$$

Og:

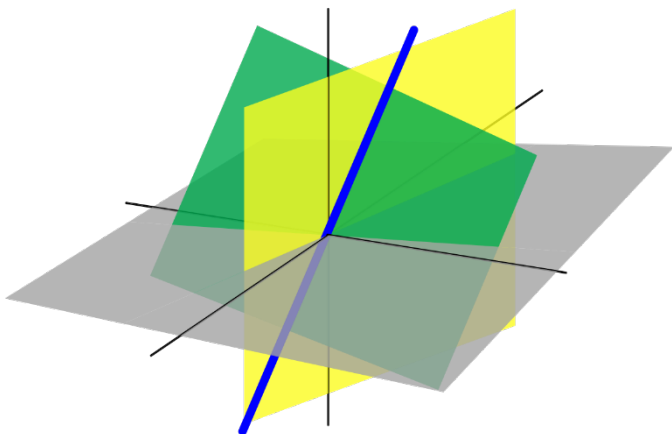
$$P(X) = \sum_{y \in Y} P(X, Y = y)$$

Bayes' setningen

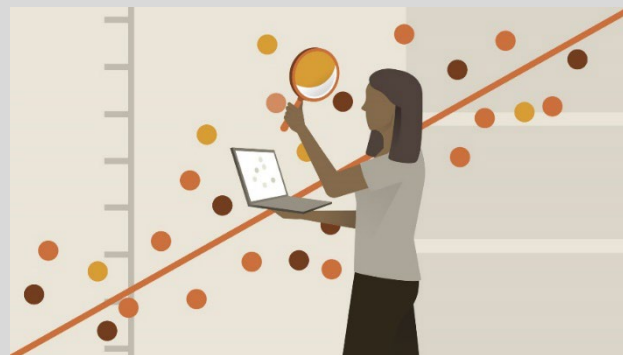
En siste operasjon som er sentralt i statistikk er Bayes' setning, som gjør det mulig å "snu" en betinget sannsynlighet:



I dag skal vi snakke om 2 temaer:



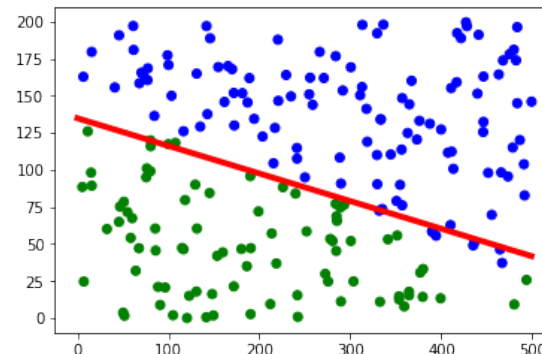
Lynkurs i lineær algebra
og sannsynlighetsteori



Logistisk regresjon
(del 2): optimering,
regularisering og
forklarbarhet

Trening

$$y = \sigma(\mathbf{w}^T \mathbf{x} + b)$$



- ▶ Hvordan velger man vektene $\mathbf{w}$ og skjæringspunktet b ?
- ▶ *Intuisjon*: vi ønsker å finne verdiene for $\mathbf{w}$ og b som minimere "feiler" på treningsett, basert på:
 1. En **tapsfunksjon** $L(\hat{y}, y)$ som angir hvor mye "feil" vi gjør hvis modellen gir sannsynlighet $\hat{y}$ som output mens den "ekte" verdi er y
 2. En **optimeringsalgoritme** som søker på verdiene for $\mathbf{w}$ og b som minimerer tapsfunksjonen på hele treningsett

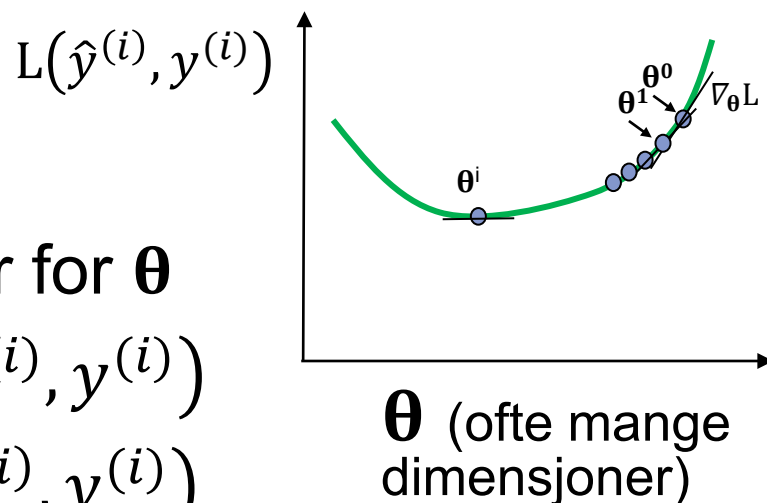
Gradient descent

1. Vi starter med tilferdige verdier for θ
2. Vi tar en treningseksempel $(x^{(i)}, y^{(i)})$
3. Vi beregner gradienten $\nabla_{\theta} L(\hat{y}^{(i)}, y^{(i)})$
4. Vi endrer deretter θ verdiene med en liten inkrement i den motsatte retningen av gradienten:

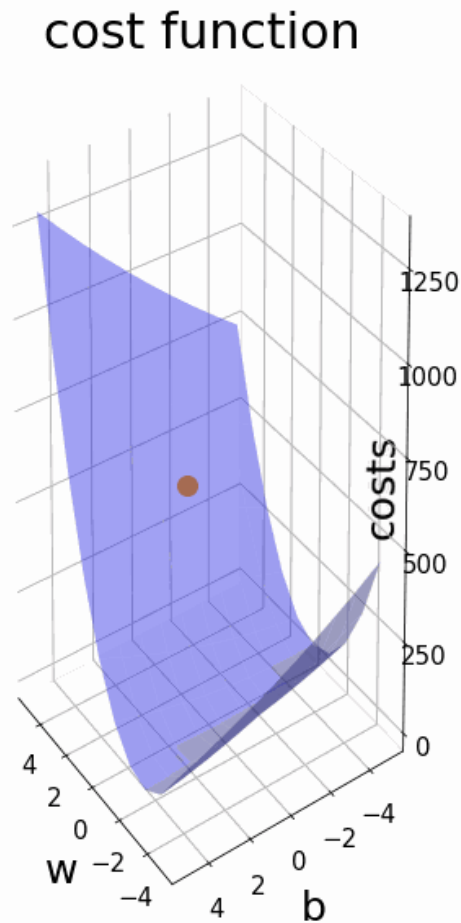
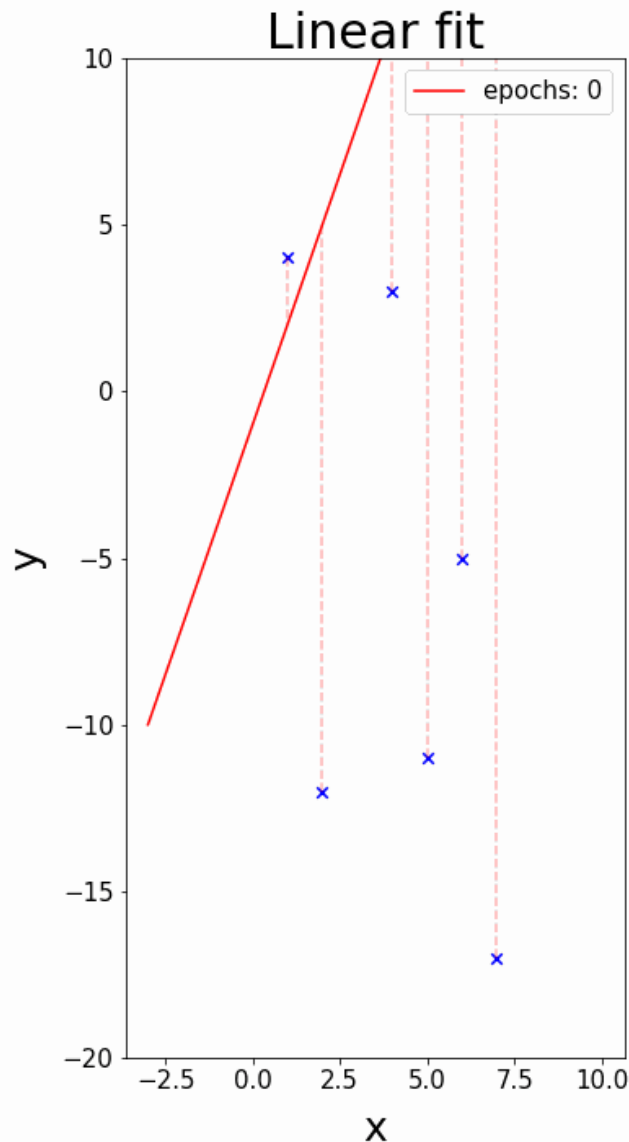
$$\theta^{t+1} = \theta^t - \eta \nabla_{\theta} L(\hat{y}^{(i)}, y^{(i)})$$

5. Vi gjentar for alle treningseksempelene, ofte flere ganger (epochs) inntil et minimum er nådd

η er *læringsraten*, og bestemmer størrelsen på “trinnene” ved hver endring



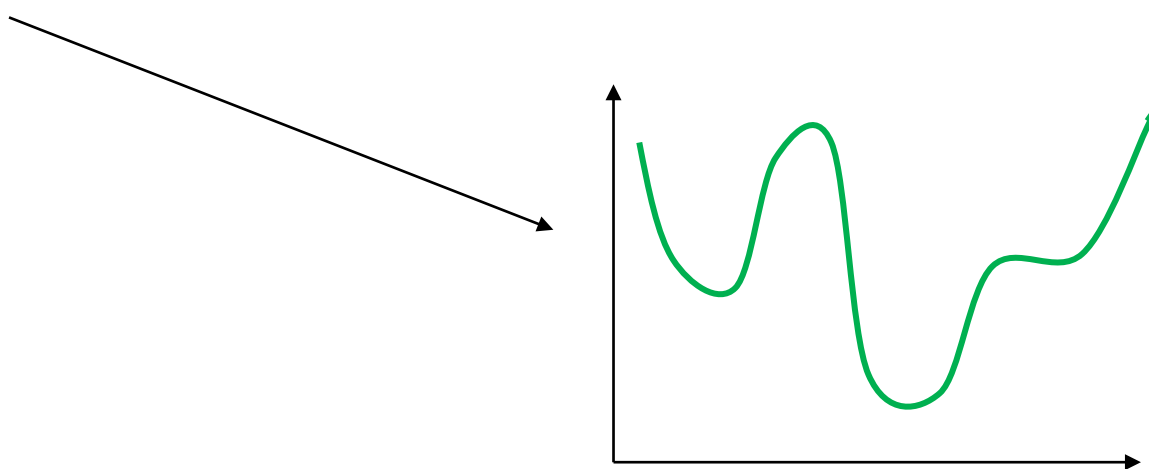
Eksempel (animert)



Vi justerer parameterne (her w og b) stegvis med *gradient descent*, slik at vi minimerer tapet på treningsett.

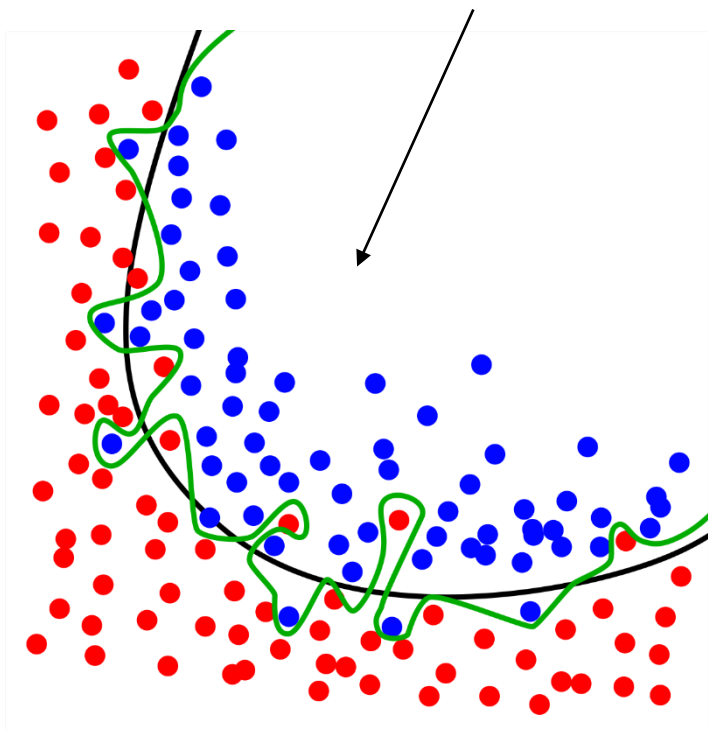
Gradient descent

- ▶ I logistisk regresjon med vanlige tapsfunksjoner som cross-entropy vet vi at det finnes kun én global minimum
 - Matematisk sett sier vi at funksjonen er **konveks**
- ▶ I mer kompliserte maskinlærings-modeller (slik som nevrale nettverk) kan vi derimot ha lokale minima, slik som her:



Overtrening?

- ▶ Overtrening (“overfitting”) er en av de største farene når man trener en maskinlæringsmodell
 - Man kan ha “perfekt” klassifisering på treningsett, men elendig ytelse på nye, ukjente datapunkter!

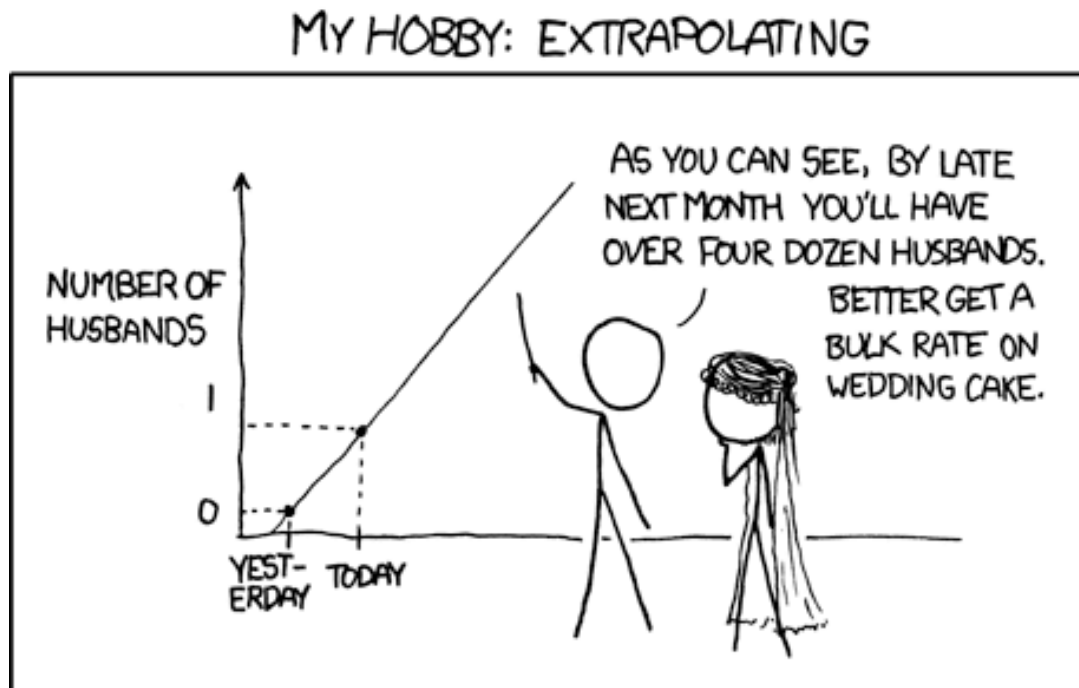


- ▶ Logistisk regresjon er relativt robust, men kan allikevel overtrenes når antall trekk er veldig høy
- ▶ Man kan redusere faren for overtrening ved å bruke *regularisering*

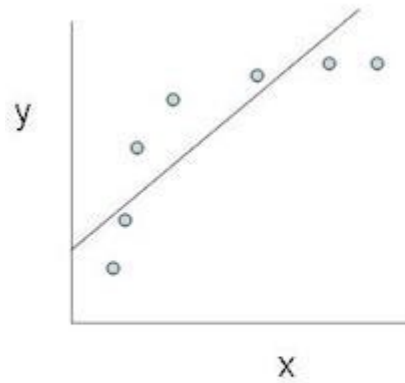
Overtrening?

Overtrening er særlig problematisk når man jobber med små datamengder

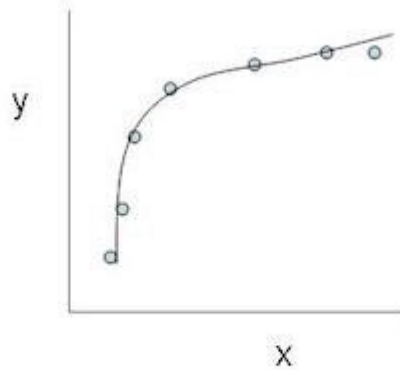
- Modellen må ekstrapolere ut fra få observasjoner



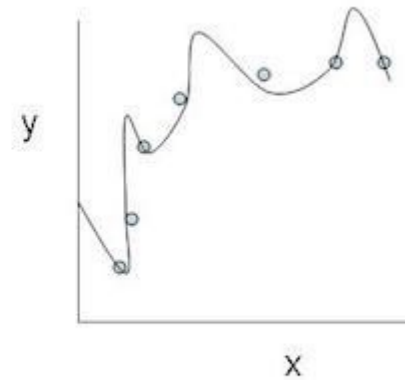
Overtrening?



Underfit



Just right



Overfit

Regularisering

Det finnes mange regulariseringsmetoder, men de fungerer ofte ved å “straffe” modeller som gir for mye vekt (positiv eller negativ) til enkelte trekk:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}; \boldsymbol{\theta}) + \alpha \sum_j^n \theta_j^2$$

Regulariseringsstyrke

Vi “straffer” modeller med høye parameterverdier

Regularisering

Ved å ta i bruk regularisering kan vi ta hensyn til *andre faktorer* enn bare å minimere tapsfunksjonen på treningsdata

- ▶ For eksempel å foretrekke enklere modeller
- ▶ Eller modeller som ikke stoler for mye på individuelle trekk (= hvis de gir høye parameterverdier til noen trekk)
- ▶ Eller modeller som er "sparse" og er i stand til å ignorere trekk som er lite relevante for klassifiseringen

→ Regularisering er et viktig "designvalg" i trening av maskinlæringsmodeller!

Forklarbarhet

- Tilbake til hovedformelen:

$$y = \sigma(\mathbf{w}^T \mathbf{x} + b)$$



- La oss anta vi har trent modellen på et dataset (=estimert gode verdier for $\mathbf{w}$ og b)
- Vi ønsker å **forstå** hva modellen har “lært”
- Stadig viktigere å utvikle “transparente” AI-modeller
 - GDPR: retten til forklaring ved automatiserte avgjørelser
 - Avdekke mulige skjevheter (*bias*) i modellen

Forklarbarhet

Eksempel: forskere utviklet en maskinlæringsmodell som skiller mellom ulv og husky:



Predicted: Wolf
True: Wolf



Predicted: husky
True: husky



Predicted: Wolf
True: Wolf



Predicted: Wolf
True: Wolf

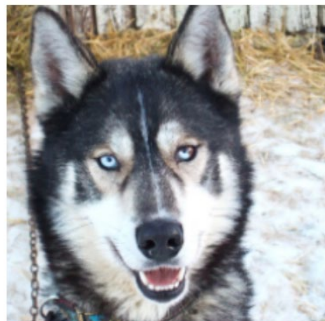


Predicted: husky
True: husky

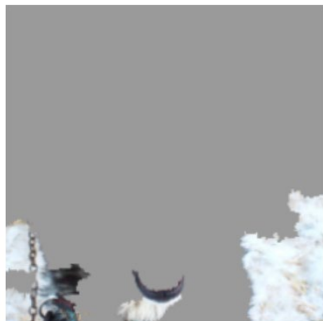


Predicted: Wolf
True: Husky

Den fungerte bra på treningsett, men de oppdaget noen rare feilklassifiseringer



(a) Husky classified as wolf



(b) Explanation

Når de prøvde å “tolke” modellen oppdaget de at modellen ignorerte dyret fullstendig og fokuserte kun på forekomst av snø i bakgrunnen!

[Ribeiro et al. (2016). "Why should I trust you?" Explaining the predictions of any classifier. *In Proceedings of SIGKDD.*]

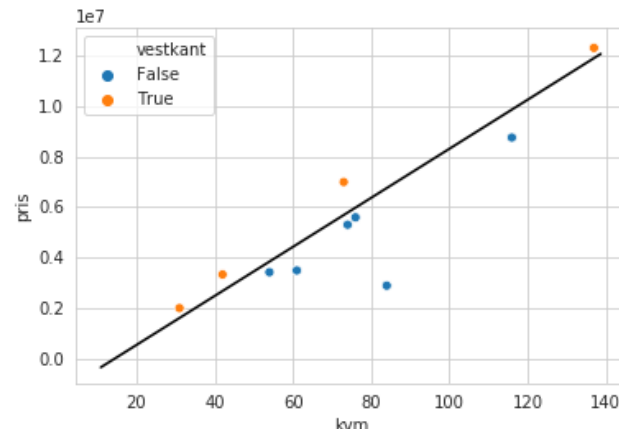
Forklarbarhet

$$y = \sigma(\mathbf{w}^T \mathbf{x} + b)$$

- ▶ Logistisk regresjon er heldigvis (relativt) lett å tolke
- ▶ For hver feature $x_i \in \mathbf{x}$ og tilsvarende vekt w_i :
 - Hvis w_i er positiv $\rightarrow$ *positiv* effect av x_i på y
(dvs. at $P(y=1|\mathbf{x})$ øker i samsvar med x_i)
 - Hvis w_i er negativ $\rightarrow$ *negativ* effekt av x_i på y
(dvs. at $P(y=1|\mathbf{x})$ minsker i samsvar med x_i)
 - Hvis w_i er null $\rightarrow$ ingen effect av x_i (kan forkastes)
- ▶ Jo *større* verdien (+ eller -), jo *større* effekten

Forklarbarhet

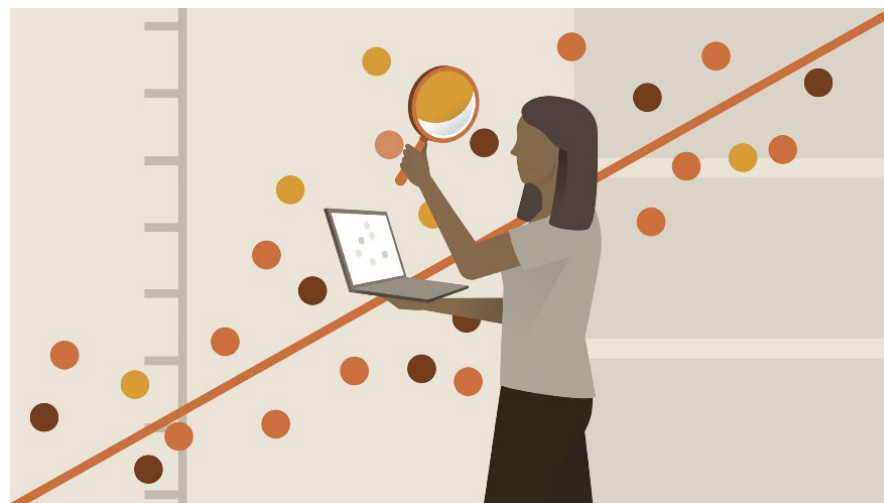
- Ta igjen vår logistisk regresjonsmode som predikerte om en bolig var på østkant eller vestkant ut fra antall kvm og pris
- Utfra vår treningsdata (og etter omskalering) finner vi at $w_{\text{kvm}} = -0.79$, $w_{\text{pris}} = 0.83$ og $b = -0.49$
- **Spørsmål:** i følge modellen, hvordan påvirker antall kvm og pris sannsynlighet til å holde på østkant eller vestkant?



Vi må omskalere de to variablene siden de har veldig forskjellige verdiområde (30-150 for kvm, 2.5-13 M for pris)

Viktig tips: det er alltid en god idé å omskalere variabler (i scikit-learn: `StandardScaler` eller `QuantileTransformer`)

Oppsummering



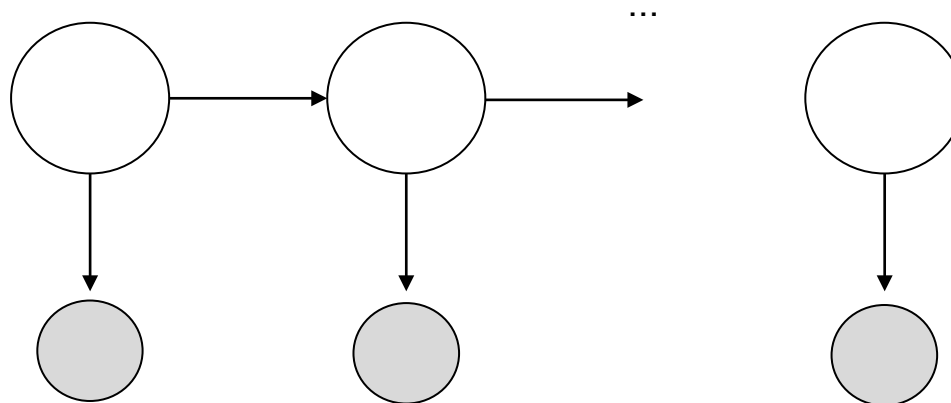
- ▶ **Logistisk regresjon:** en lineær, probabilistisk klassifikasjonsmodell
 - Binær (med sigmoid) eller multi-klasse (med softmax)
- ▶ Hører til de viktigste verktøyene i språkteknologi
 - Pålitelig, rask å trene, og forklarbar
 - Men begrenset til lineære kombinasjoner – kan ikke lære komplekse interaksjoner mellom features

Midtveiseevaluering

- ▶ Fint om dere kan fylle ut dette skjemaet:
<https://nettskjema.no/a/406434>
- ▶ **Frist:** mandag neste uke



Neste uke



Modellering av sekvenser:
Markovkjeder og Hidden Markov Models