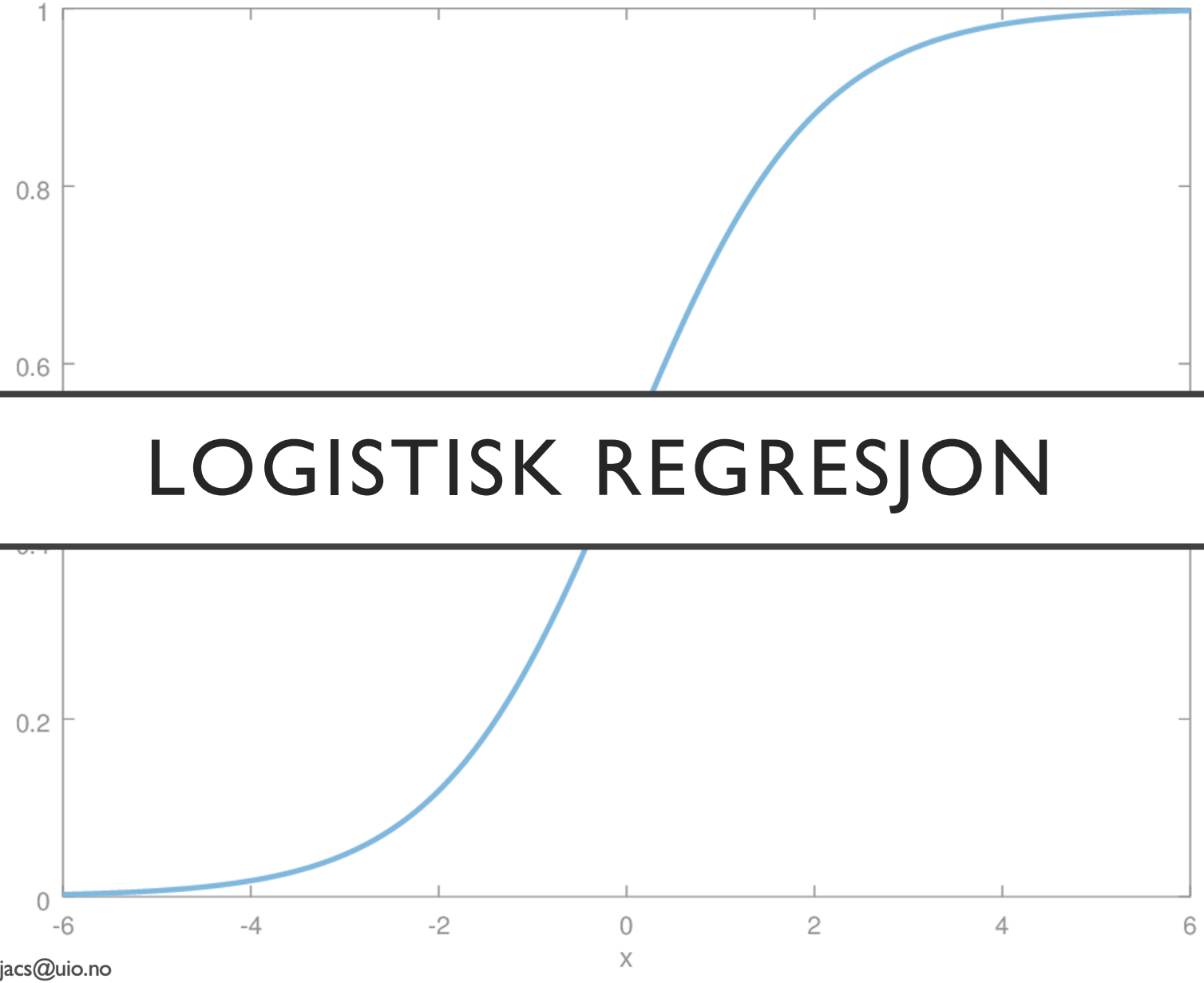


LOGISTISK REGRESJON

LINEÆR REGRESJON

Forsøker å finne en *rett linje* vi kan trekke som *decision boundary*

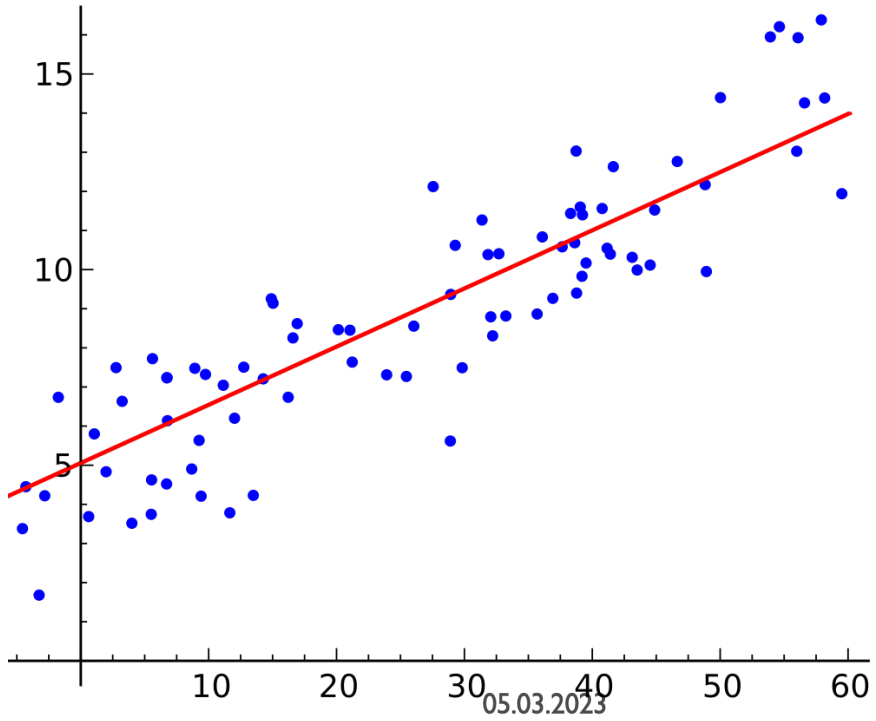
$$y = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$$

Output

Input med *n* numeriske verdier

Estimerte vekter i modellen

Skjæringspunktet: Verdien av y hvis alle trekk i x er 0



Men: formelen gir oss verdier mellom -uendelig og +uendelig

Vi vil ha verdier mellom 0 og 1

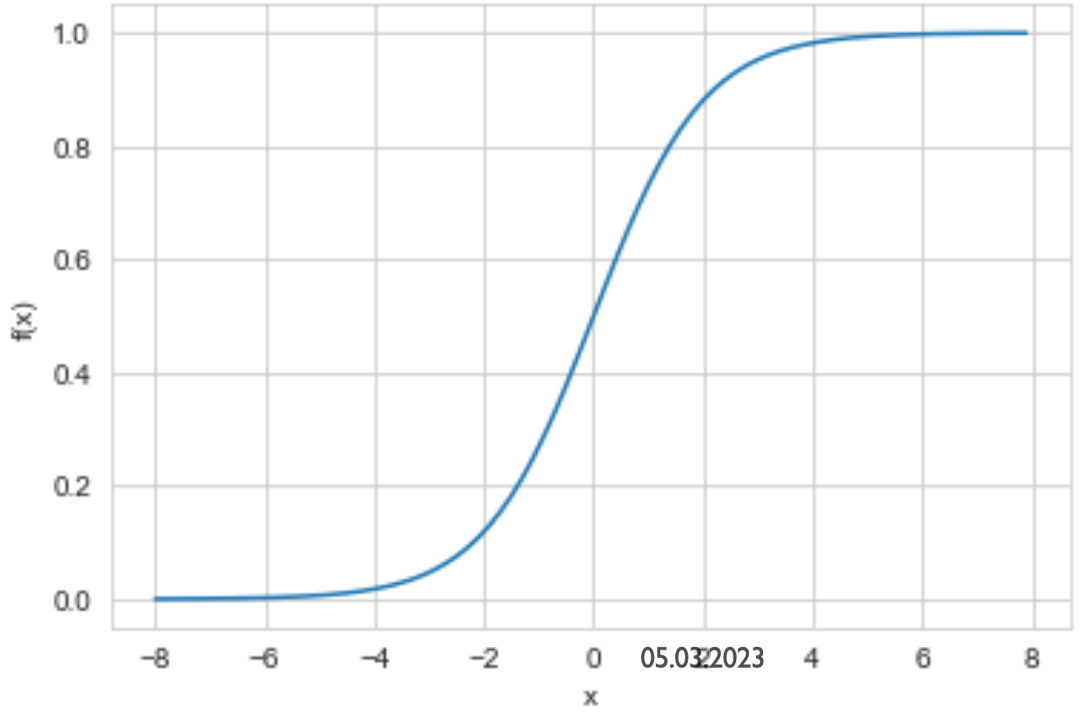
LOGISTISK REGRESJON

Klassifikasjonsmodell som gir oss en *sannsynlighet* for en outputklasse *y* gitt input *x*

For å gjøre om regresjonsresultatet til en sannsynlighet mellom 0 og 1, bruker vi *Sigmoid-funksjonen* σ:

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

$$\sigma = \frac{1}{1 + e^{-z}}$$



TRENE EN LOGISTISK REGRESJONSMODELL

Vi estimerer vektene *w* og skjæringspunktet *b* ved hjelp av et treningssett

- Vi ønsker at disse verdiene *minimerer* feil modellen gjør

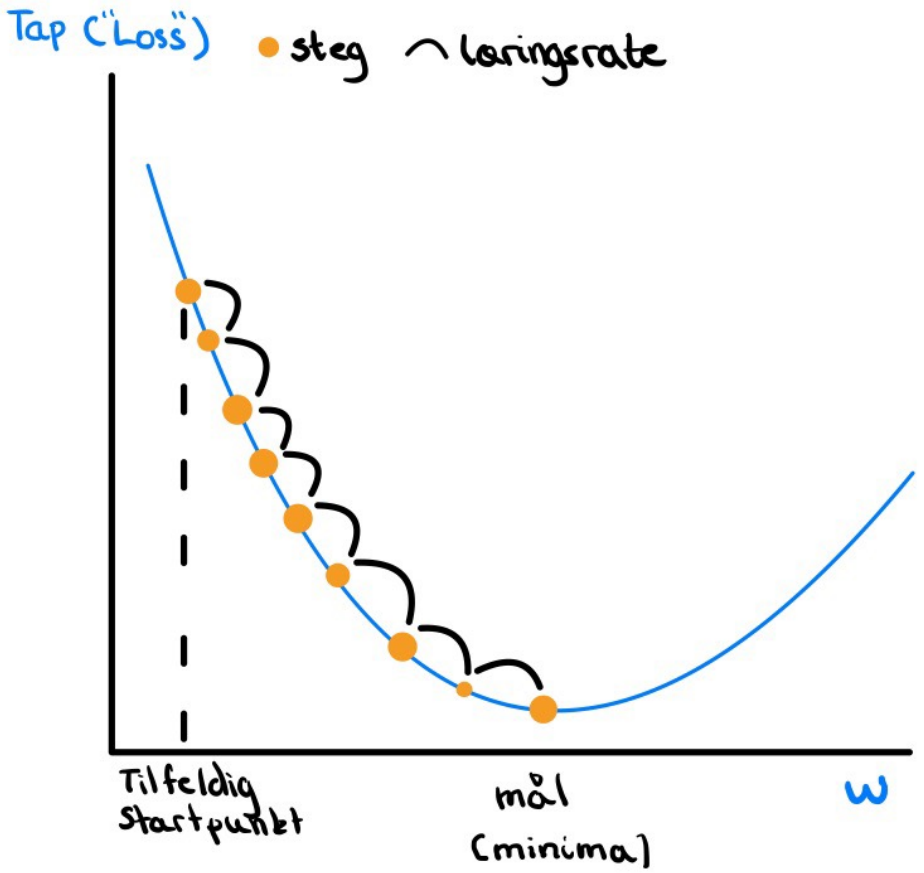
Vi må definere:

- En tapsfunksjon: Måler hvor mye modellen feiler
  - Cross Entropy
- En optimiseringsalgoritme: finner verdiene for *w* og *b* som minimerer tapsfunksjonen på treningssettet
  - Gradient Descent

GRADIENT DESCENT

Minimerer tapsfunksjonen: modellen feiler så lite som mulig

- Justerer *w* og *b* gradvis med *gradient descent*, slik at tapet minimeres



FORDELER & ULEMPER

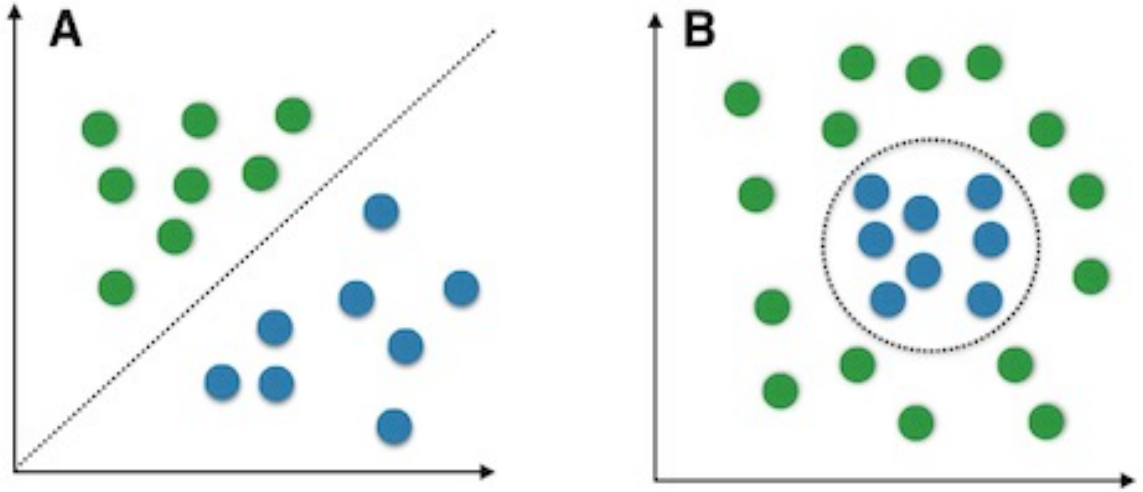
– En lineær modell vil ikke kunne skille mellom klassene i alle datatyper

\+ Kan trenes på *beskjedne datamengder*: Få parametere som må estimeres (*w* og *b*)

\+ Gode *generaliseringsevner*

\+ *Rask* og *skalerbar*: Kan skalere til store datasett uten problemer

\+ *Forklarbar*: Vi kan forstå hvilken del av input som påvirker modellens prediksjoner



VIDERE:

Intro til Pandas og Scikit-Learn

Ukesoppgaver

Oblig 1b