IN1140: Introduksjon til språkteknologi

Forelesning #8

Samia Touileb

Universitetet i Oslo

20. oktober 2021

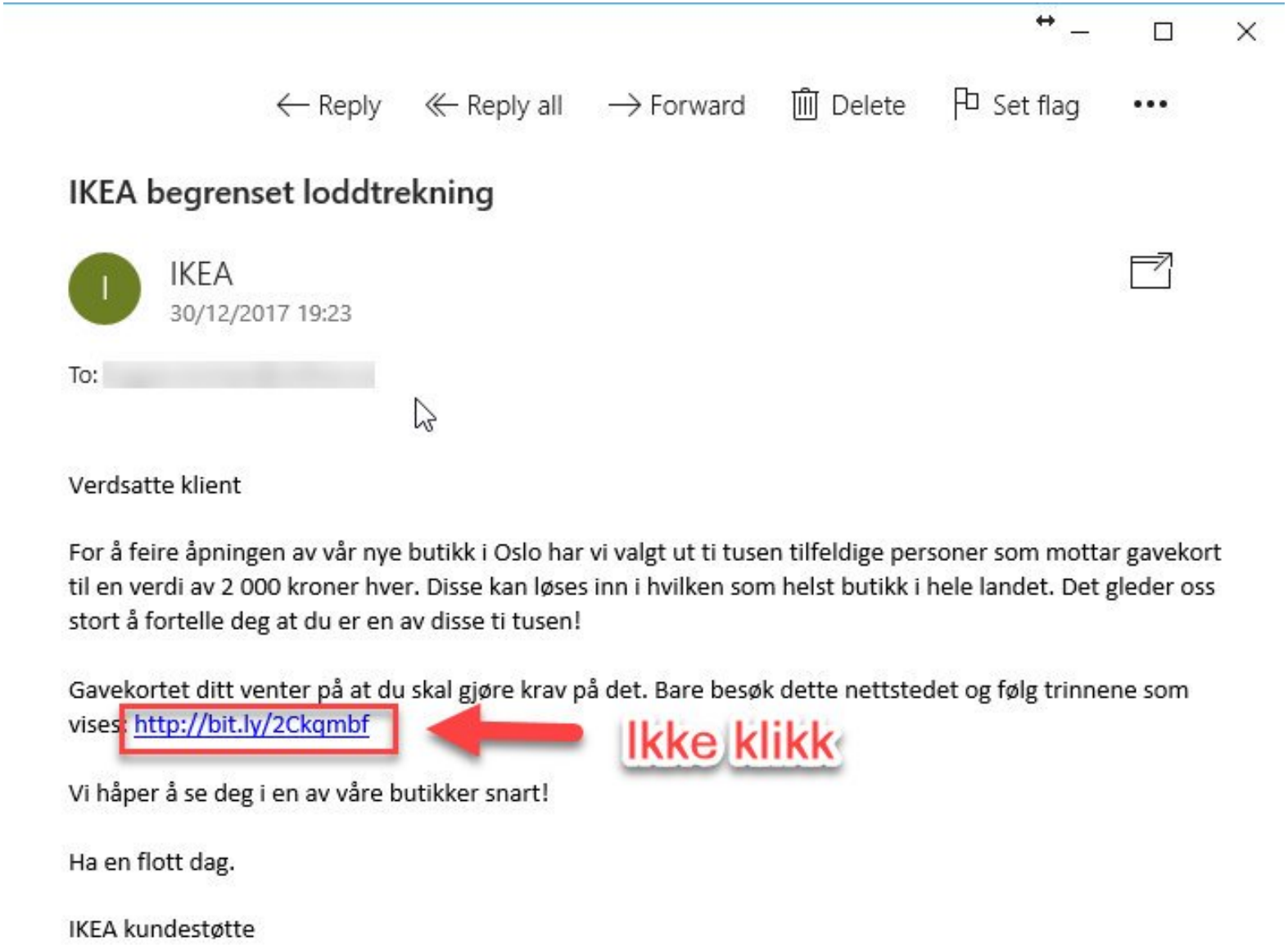


I dag skal vi se på:

- Maskinlæring og Klassifisering
  - Naive Bayes klassifisering

Maskinlæring og Klassifisering

Spam? Ikke spam?



https://twitter.com/UlvBjerkan/status/947280876479664128

Positive eller Negative?

- En varm og velopplagt bok. +
- Forfatteren burde ha fått bedre manushjelp av forlaget sitt. –
- Oppskrytt dameroman. –
- Denne boken er et hån mot underholdningssjangeren. –
- Dette er kanskje årets beste barnebok. +

Hentet fra NoReC korpuset `https://github.com/ltgoslo/norec`

Hva handler artikkelen om?

BBC: Sjansene for en brexitavtale mindre etter nattens forhandlinger

Det pågikk intense brexitforhandlinger mellom Storbritannia og EU i natt. Men det ble ikke noe gjennombrudd. Sjansene for en avtale er blitt mindre, sier britiske regjeringskilder til BBC.



Veggmaleri av Boris Johnson på en bygning i London
FOTO: HANNAH MCKAY



Øystein Heggen
Journalist

Publisert i dag kl. 10:30
Oppdatert for én time siden

Kategori?

- Politikk
- Sport
- Underholdning
- Kultur
- ...

`https://www.nrk.no/urix/forhandlinger-om-brexit-i-natt-_-uten-losning-1.14744008`

Tekstklassifisering

Automatisk tildeling av kategorier, tema, sjangere ...

- Spam-gjenkjenning.
- Sentiment analyse.
- Identifikasjon av forfattere/forfatterskap.
- Alder- og kjønnsidentifikasjon.
- Språkgjenkjenning.
- ...

Tekstklassifisering

- Input:
  - et dokument d
  - et forhåndsdefinert sett med klasser C = $\{c_1, c_2, ..., c_n\}$
- Output:
  - tildele en klasse c

Manuell klassifisering?

- Regelbasert klassifisering basert f.eks. på ord:
  - Spam: inneholder “dårlig språk” AND “nevner ukjent person” AND “Klikk her!”
- Regler definert av eksperter $\rightarrow$ gode resultater.

Manuell klassifisering?

- Regelbasert klassifisering basert f.eks. på ord:
  - Spam: inneholder “dårlig språk” AND “nevner ukjent person” AND “Klikk her!”
- Regler definert av eksperter $\rightarrow$ gode resultater.
- Men vanskelig å lage og vedlikeholde slike regler.

Statistisk klassifisering

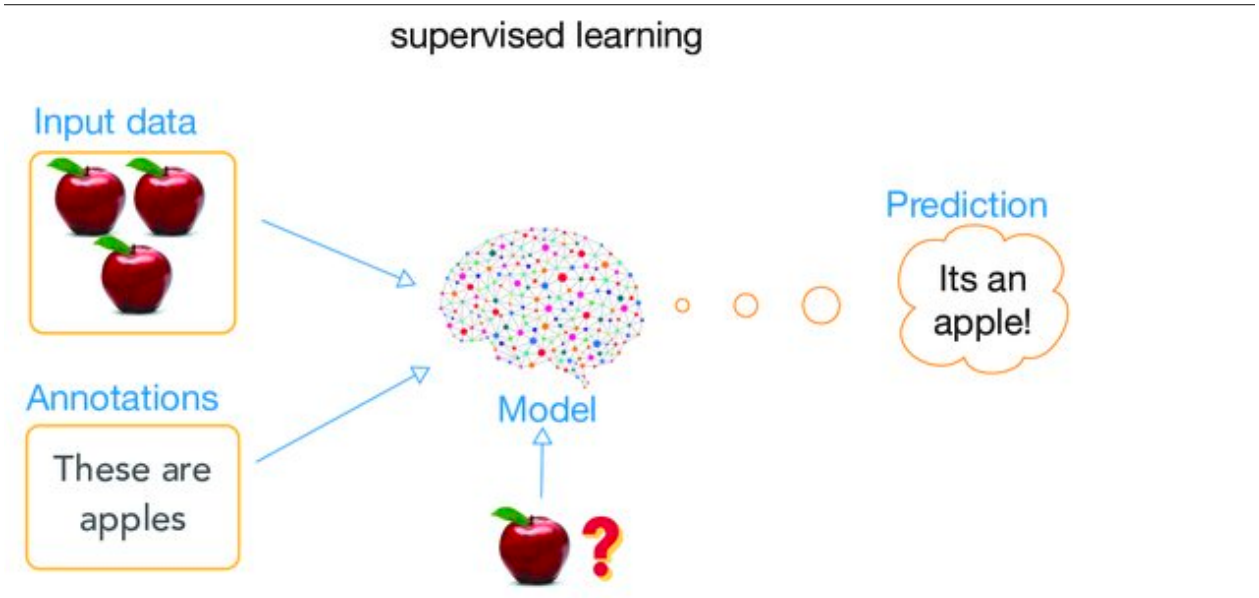
- Sentral metode innenfor maskinlæring.
- Automatisk avgjøre hvilken kategori en observasjon tilhører.
- Basert på annotert treningsdata: observasjoner der kategorien er kjent.
- Supervised klassifisering: klassifisering som benytter annotert treningsdata.

Supervised Machine Learning

- Input:
  - et dokument d
  - et forhåndsdefinert sett med klasser C = $\{c_1, c_2, ..., c_n\}$
  - et treningsett m av manuelt annoterte dokumenter $(d_1, c_1), ..., (d_m, c_m)$

- Output:
  - en trent klassifiserer y: d $\rightarrow$ c

Supervised Machine Learning



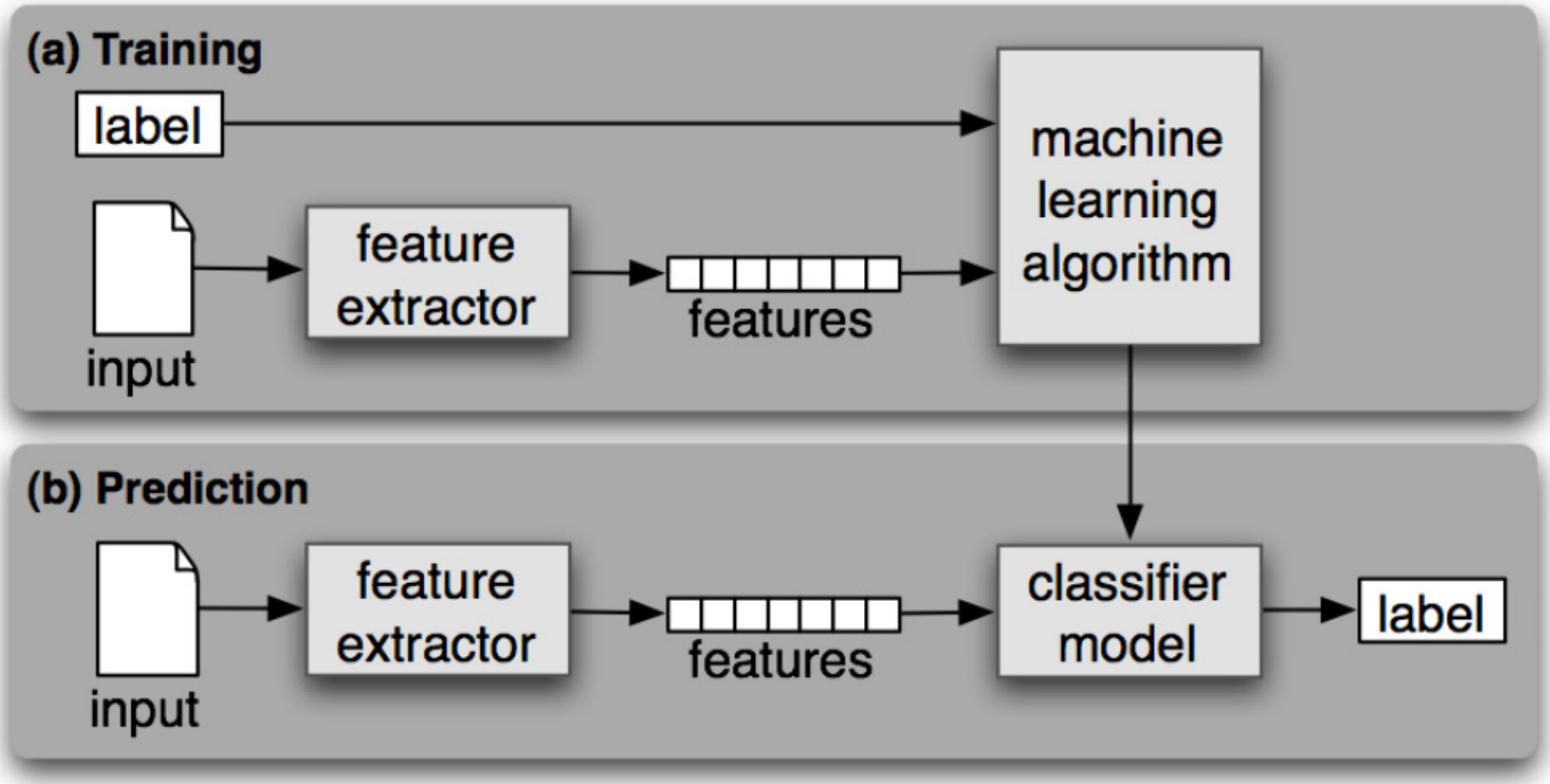
https://www.researchgate.net/publication/329533120/figure/fig1/AS:702267594399761@1544445050584/Supervised-learning-and-unsupervised-learning-Supervised-learning-uses-annotation_W640.jpg

Supervised Machine Learning

Tre “splits”:

- Train: data for å trene modellen.
- Dev: data for å evaluere modellen underveis.
- Test: data for å evaluere den beste modellen etter trening og evaluering på dev.

Statistisk klassifisering



Sentiment Analyse

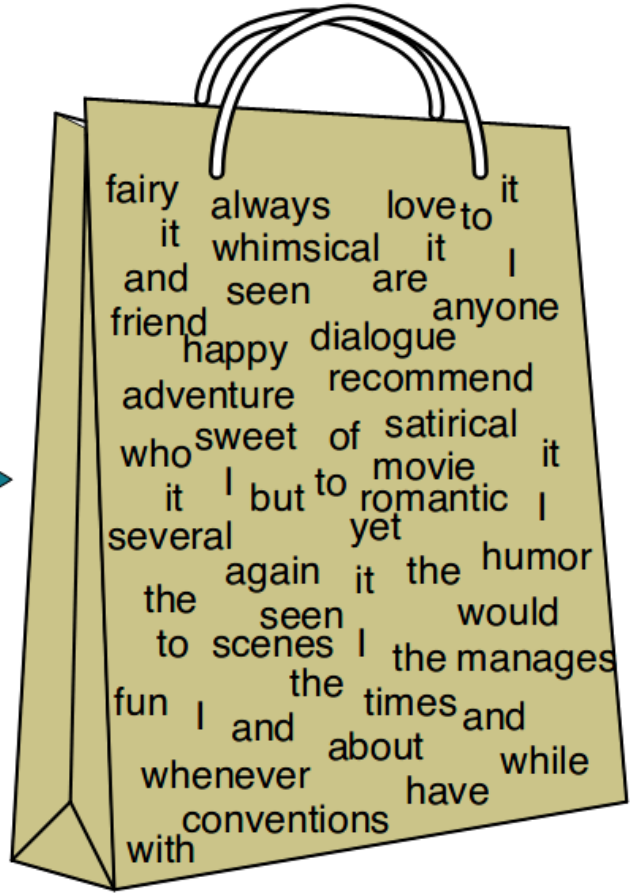
- Første skritt består i å finne og å hente ut **trekk** ("features") fra treningsdataene.

- Eksempel:
  - En varm og velopplagt bok. POS
  - Forfatteren burde ha fått bedre manushjelp av forlaget sitt. NEG
  - Oppskrytt dameroman. NEG
  - Denne boken er et hån mot underholdningssjangeren. NEG
  - Dette er kanskje årets beste barnebok. POS

- Hvilke **trekk** ("features") kan vi bruke for å skille mellom de positive og negative anmeldelser?

Trekk: Bag-of-words

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

https://web.stanford.edu/~jurafsky/slp3/4.pdf

Andre mulige trekk

- Ordklasser.
- Bi-grams, tri-grams, n-grams av ord eller karakterer.
- Top N mest frekvente ord.
- Syntaktiske funksjoner.
- ...

Klassifisering

- Gitt treningsdataene og trekkvektorene, kan en rekke forskjellige maskinlæringsalgoritmer brukes til å trene en klassifiserer.

Klassifisering

- Gitt treningsdataene og trekkvektorene, kan en rekke forskjellige maskinlæringsalgoritmer brukes til å trene en klassifiserer.

- Her skal vi se på **Naive Bayes**-klassifisering:
  - Statistisk klassifiserer.
  - Bruker informasjon om ord i konteksten.
  - Enkel og naiv metode basert på Bayes regelen.

Naive Bayes

- **Hovedantagelse**:
  for å finne en klasse $\hat{c}$ (hentet fra alle mulige klasser C) for en trekkvektor $\overrightarrow{f}$ må vi beregne den mest sannsynlige klassen, gitt vektoren

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \; P(c|\overrightarrow{f})$$

Naive Bayes

- **Hovedantagelse**:
  for å finne en klasse $\hat{c}$ (hentet fra alle mulige klasser C) for en trekkvektor $\overrightarrow{f}$ må vi beregne den mest sannsynlige klassen, gitt vektoren

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \; P(c|\overrightarrow{f})$$

- Men det er problematisk å trene direkte: "sparse data"-problemet (alltid finnes både ord og sekvenser vi ikke har sett.)

Naive Bayes

- **Hovedantagelse**:
  for å finne en klasse $\hat{c}$ (hentet fra alle mulige klasser C) for en trekkvektor $\overrightarrow{f}$ må vi beregne den mest sannsynlige klassen, gitt vektoren

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \; P(c|\overrightarrow{f})$$

- Men det er problematisk å trene direkte: "sparse data"-problemet (alltid finnes både ord og sekvenser vi ikke har sett.)

- Kan bruke **Bayes teorem**

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes regel

- Betinget sannsynlighet:

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

Bayes regel

- Betinget sannsynlighet:

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

- Produktsetningen:

$$P(A,B) = P(A|B)P(B)$$

Bayes regel

- Betinget sannsynlighet:

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

- Produktsetningen:

$$P(A,B) = P(A|B)P(B) = P(B|A)P(A)$$

Bayes regel

- Betinget sannsynlighet:

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

- Produktsetningen:

$$P(A,B) = P(A|B)P(B) = P(B|A)P(A)$$

- Bayes regel:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Naive Bayes

- Omformulering ved Bayes teorem:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \ \frac{P(\overrightarrow{f}|c)P(c)}{P(\overrightarrow{f})}$$

Naive Bayes

- Omformulering ved Bayes teorem:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \frac{P(\overrightarrow{f}|c)P(c)}{P(\overrightarrow{f})}$$

- Men fremdeles ikke nok data for denne beregningen...
- Bryte opp trekkvektoren og se på individuelle trekk i kombinasjon med klasser.
- En **uavhengighetsantagelse**: trekk er uavhengige av andre trekk

$$P(\overrightarrow{f}|c) \approx \prod_{j=1}^{n} P(f_j|c)$$

Naive Bayes

- Naive Bayes klassifiserer

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \; P(c) \prod_{j=1}^{n} P(f_j|c)$$

- vi **trener** klassifisereren ved å beregne sannsynligheter fra et korpus (MLE – Maximum Likelihood Estimation)

Naive Bayes

- To sannsynligheter:
  1. prior-sannsynligheten for klassen $P(c)$

$$P(c) = \frac{N_c}{N_{doc}}$$

N_c = antall dokumenter i treningsdataen som er i klassen c
N_{doc} = total antall dokumenter

  2. sannsynligheten for individuelle trekk $P(f_j|c)$

$$P(f_j|c) = \frac{count(f_j, c)}{count(c)}$$

Naive Bayes

- For å beregne P(f_j|c) kan vi anta at et trekk er ett ord som finnes i dokumentets bag of words, og kan derfor heller beregne følgende:

$$P(w_i|c) = \frac{count(w_i, c)}{\sum_{w \in V} count(w, c)}$$

Naive Bayes

- For å beregne P(f_j|c) kan vi anta at et trekk er ett ord som finnes i dokumentets bag of words, og kan derfor heller beregne følgende:

$$P(w_i|c) = \frac{count(w_i, c)}{\sum_{w \in V} count(w, c)}$$

- Med andre ord:

  Antall ganger ordet w_i forekommer i alle dokumenter av klasse c, delt på antall ord som finnes i alle dokumenter av klasse c.

  V her representerer hele vokabularet, altså mengden av alle ord (types) fra alle klasser.

Naive Bayes – problem?

- La oss anta at vi prøver å finne sannsynligheten for ordet “fantastisk” gitt klassen “positiv”.

Naive Bayes – problem?

- La oss anta at vi prøver å finne sannsynligheten for ordet "fantastisk" gitt klassen "positiv".

- Ordet "fantastisk" blir ikke brukt i noen av dokumentene som er klassifisert som positive.

$$P(fantastisk|positiv) = \frac{count(fantastisk, positiv)}{\sum_{w \in V} count(w, positive)} = 0$$

Naive Bayes – problem?

- La oss anta at vi prøver å finne sannsynligheten for ordet "fantastisk" gitt klassen "positiv".

- Ordet "fantastisk" blir ikke brukt i noen av dokumentene som er klassifisert som positive.

$$P(fantastisk|positiv) = \frac{count(fantastisk, positiv)}{\sum_{w \in V} count(w, positive)} = 0$$

- Dette er problematisk! Siden alle sannsynligheter blir multiplisert med hverandre, å få 0 her vil lede til at sannsynligheten for hele klassen vil bli lik 0.

Naive Bayes – problem?

- Enkleste måte å løse dette på: bruke smoothing/glatting.

Naive Bayes – problem?

- Enkleste måte å løse dette på: bruke smoothing/glatting.
- Bruke add-one (Laplace) smoothing/glatting.

$$P(w_i|c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} (count(w, c) + 1)} = \frac{count(w_i, c) + 1}{(\sum_{w \in V} (count(w, c)) + |V|}$$

Naive Bayes – problem?

- Hva om det finnes ord i testdataen som ikke finnes i vårt vokabular V?

Naive Bayes – problem?

- Hva om det finnes ord i testdataen som ikke finnes i vårt vokabular V?
  - Ignorer dem, og slett dem fra testdataen, og ikke beregn sannsynligheter for dem.

Eksempel – NB for sentiment klassifisering

	Kat	Dokument
Train	+	en varm og velopplagt bok
	+	dette er kanskje årets beste barnebok
	−	forfatteren burde ha fått bedre manushjelp av forlaget sitt
	−	oppskrytt dameroman
	−	denne boken er et hån mot underholdningssjangeren
Test	?	boken og forfatteren er oppskrytt

Eksempel – NB for sentiment klassifisering forts.

Vi tar en Kahoot!

Gå på `www.Kahoot.it`

Eksempel – NB for sentiment klassifisering forts.

- Begynne med å regne ut prior-sannsynligheten for klassene $P(-)$ og $P(+)$.
- Husk at vi beregner $P(c)$ slik:

$$P(c) = \frac{N_c}{N_{doc}}$$

N_c = antall dokumenter i treningsdataen som er i klassen c
N_{doc} = total antall dokumenter

Eksempel – NB for sentiment klassifisering forts.

- Begynne med å regne ut prior-sannsynligheten for klassene $P(-)$ og $P(+)$.
- Husk at vi beregner $P(c)$ slik:

$$P(c) = \frac{N_c}{N_{doc}}$$

N_c = antall dokumenter i treningsdataen som er i klassen c
N_{doc} = total antall dokumenter

- $P(-)$ og $P(+)$ er da:

$$P(-) = \frac{3}{5} \qquad P(+) = \frac{2}{5}$$

Eksempel – NB for sentiment klassifisering forts.

- Vi beregner sannsynlighetene for hvert ord i testsetningen, altså: “boken”, “og”, “forfatteren”, “er”, “oppskrytt”. Vi bruker smoothing/glatting.

- Vi må da beregne sannsynligheten for individuelle trekk $P(f_j|c)$ altså :

$$P(w_i|c) = \frac{count(w_i, c) + 1}{(\sum_{w \in V} (count(w, c)) + |V|}$$

Eksempel – NB for sentiment klassifisering forts.

$$P(\text{boken}|-) = \frac{1+1}{18+28}$$

$$P(\text{boken}|+) = \frac{0+1}{11+28}$$

$$P(\text{og}|-) = \frac{0+1}{18+28}$$

$$P(\text{og}|+) = \frac{1+1}{11+28}$$

$$P(\text{forfatteren}|-) = \frac{1+1}{18+28}$$

$$P(\text{forfatteren}|+) = \frac{0+1}{11+28}$$

$$P(\text{er}|-) = \frac{1+1}{18+28}$$

$$P(\text{er}|+) = \frac{1+1}{11+28}$$

$$P(\text{oppskrytt}|-) = \frac{1+1}{18+28}$$

$$P(\text{oppskrytt}|+) = \frac{0+1}{11+28}$$

Eksempel – NB for sentiment klassifisering forts.

$$P(-)P(S|-) = \frac{3}{5} \times \frac{\overbrace{2 \times 1 \times 2 \times 2 \times 2}^{16}}{46^5} = 0,000000047 = 4.7 \times 10^{-8}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{\overbrace{1 \times 2 \times 1 \times 2 \times 1}^{4}}{39^5} = 0,000000018 = 1.8 \times 10^{-8}$$

$$P(-)P(S|-) > P(+)P(S|+)$$

$\Longrightarrow$ Den blir derfor klassifisert som *negativ*.

Optimere Naive Bayes for sentiment analyse



- Forekomsten av ord er vikitgere enn frekvensen: telle hvert ord i hvert dokument kun én gang.
- Håndtere negasjon:
  - I really like this movie. +
  - I really did't like this movie. −
  - Don't dismiss this film, doesn't really get us bored. +
  - Veldig enkel og naiv måte å løse dette på er å legge til prefikset `NOT_` før hvert ord som blir negert:
    - didn't like this movie , but I liked the actors .
    - didn't `NOT_like NOT_this NOT_movie` , but I liked the actors.
- Ikke nok treningsdata?
  - Bruk et sentimentleksikon.
  - Legge til et trekk til Naive Bayes. Telle hver gang et ord er fra det positive eller negative leksikonet.

Evaluering

- Husk! Vi har annotert treningsdata + annotert testsett.
- Testsettet brukes med Naive Bayes uten klassene.
- Klassene fra testsettet brukes som gullstandard (gold standard) for å sjekke om Naive Bayes har angitt rikitg klasse.

Evaluering

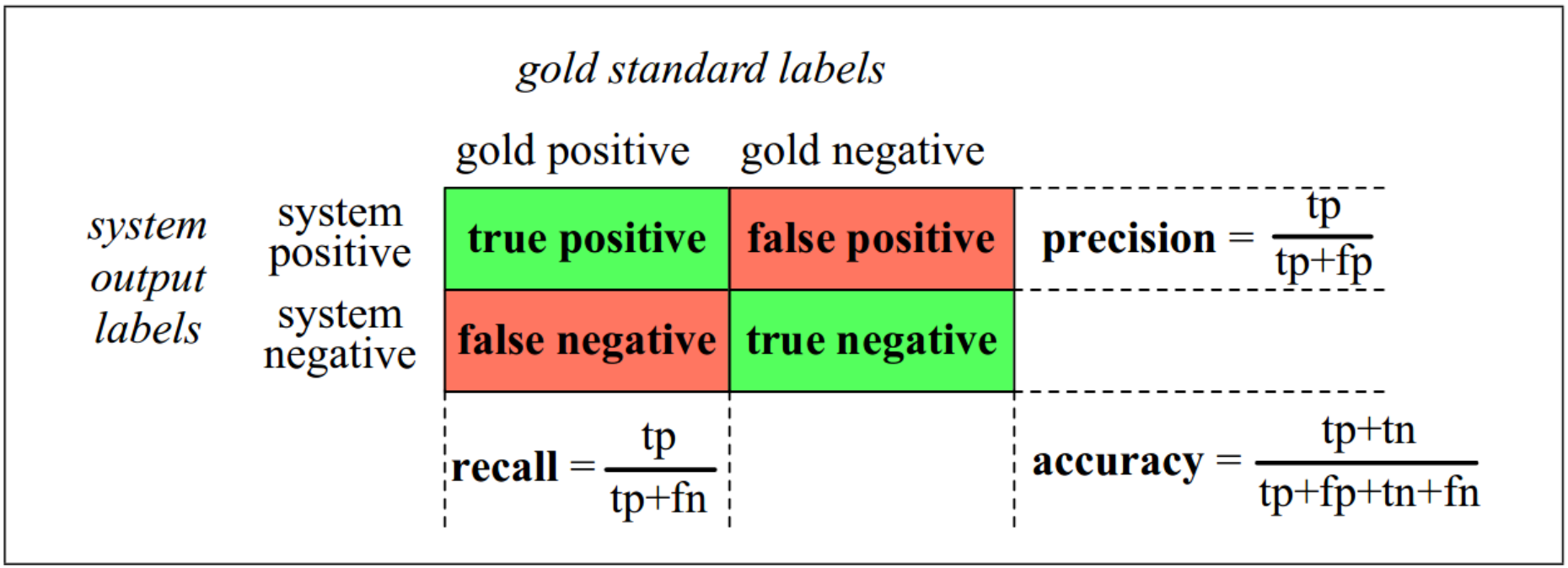
- Husk! Vi har annotert treningsdata + annotert testsett.
- Testsettet brukes med Naive Bayes uten klassene.
- Klassene fra testsettet brukes som gullstandard (gold standard) for å sjekke om Naive Bayes har angitt rikitg klasse.
- Hvordan kan vi vite om vår Naive Bayes klassifiserer riktig?
- Hvordan kan vi evaluere den?

Evaluering

- Lage contingency tabell.
- Regne ut accuracy, precision, recall, og F_1.

Evaluering

- Lage contingency tabell.
- Regne ut accuracy, precision, recall, og F_1.



$$F_1 = \frac{2PR}{P+R}$$

tp = true positive, fp = false positive, tn = true negative, fn = false negative, P = precision, R = Recall

https://web.stanford.edu/~jurafsky/slp3/4.pdf

Eksempler med Python

Se filene:

- `doc_classification.py`
- `doc_classification_BOW.py`