Tekst som data: *M*achine *L*earning i praksis

Jeppe Fjeldgaard Qvist

Hvem er jeg og hvad kan jeg?

Jeppe Fjeldgaard Qvist
Postdoc, Ph.d

Kandidat i **Samfundsfag** og **Geografi**

Center for Comparative Welfare Studies (CCWS)
Forsknings- og undervisningsprofil speciliseret i **kvantitative metoder** og (social) **data science**

- *Kvantitativ tekstanalyse*
- *Udvikling af IT*

~~Data på nettet. Hvad er mulighederne?~~

- *Interessen var en introduktion til nye data at arbejde med*
- *I kan med Google eller ChatGPT nemt finde et script til skraping, der virker. MEN for at få noget brugbart data ud kræver det:*
  - *Forholdsvis bred viden om websider og hvordan de er opbygget*
  - *Forholdsvis bred viden om forskellige kodesprog; ét til scarping, ét til websider*
  - *Bred viden om Regular Expressions (Regex)*
- *Sociale medier og nyhedsmedier er i vidt omfang blevet **lukket for scraping***
- *Af de sider, der er tilladt at scarpe, eksisterer der allerede store klargjorte datasæt tilgængelige på nettet. Disse egner sig meget bedre til intro-forløb*

Tekst som data: Hvordan tænker vi det og hvordan kan vi arbejde med det?

Hvad skal vi nå denne halvanden time?

Min tilgang:
> *Hvordan ville jeg tilrettelægge et undervisningsforløb med en gymnasieklasse?*
> *Et forløb egnet til tværfagligt arbejde; her er det oplagt at kombinere samfundsfag, sprogfag, matematik*

1. Afmystificering af **ML** af **AI** som det overordnede formål og tema
2. Tekst som data (skridtet *efter* scraping)
3. Klassifikation og `superviseret Machine Learning` (**SML**)
4. Præsentation af *plug-and-play* `Python` kode i Google Colab
5. Kort diskussion af videre arbejde med metoden og koden

(D)et klassiske Machine Learning problem

Tekst-klassificering

- Sentiment analysis
- Forfatter/afsender klassifikation
- Spam identifikation
- Politisk/ideologisk klassifikation
- Tematisk klassifikation
- *Hvilken som helst* **gensidigt udelukkende kategorier** *i kan forestille jer…*

Superviseret Machine Learning 1

Den menneskelige komponent

- Data som **input-output-par**
  - *Input* (features): Et stykke tekst.
  - *Output* (labels): Den **korrekte klassifikation** af teksten.

Input (feature/Tekst)	Output (label)
Dette er en god dag	Positiv
Jeg er så træt af alting	Negativ
Vejret er fantastisk i dag	Positiv
Jeg har mistet min telefon	Negativ

`Superviseret Machine Learning` 2

- **Træningsfase**
  - Modellen trænes på `testdata`, hvor den tilstræber at minimere fejlen mellem det klassificerede output og det faktiske output.
- **Valideringsfase**
  - Efter modellen er **vurderet** til at være trænet tilfredsstillende, anvendes den på `test` data, som modellen ikke har set før. Men vi–mennesket–har de rigtige kategorier på, for at vurdere ***generaliserbarheden***. (Igen en menneskelig kompenent.)
- **Anvendelse**
  - Modellen "slippes fri" ...

`Superviseret Machine Learning` 3

For

- Let at evaluere (dog ikke **neurale netværk, deep learning**)
- Hvis vores opgave er "simpel" og klart defineret er metoden effektiv

Imod

- Kræver relativt store mængder kvalitativt-kodet data; dyrt og tidskrævende (**mturk**)
- Risiko for *overfitting* og lav generaliserbarhed

Tekst som data

- `ordforråd`

  $V = \{\text{ord}_1, \text{ord}_2, \ldots, \text{ord}_n\}$

- Repræsentation af tekst: `bag-of-words`, `n-grams` (der eksisterer mere avanceret repræsentation)

$$\begin{pmatrix} \text{ord } 1 & 1 \\ \text{ord } 2 & 0 \\ \text{ord } 3 & 3 \\ \text{ord } 4 & 1 \\ \text{ord } 5 & 4 \\ \text{ord } 6 & 5 \\ \dots & \dots \\ \text{ord n} & n \end{pmatrix}$$

Input (feature/Tekst)	Output (label)	af	alting	dag	dette	en	er	fantastisk	god	har	jeg	min	mistet	så
Dette er en god dag	Positiv	0	0	1	1	1	1	0	1	0	0	0	0	0
Jeg er så træt af alting	Negativ	1	1	0	0	0	1	0	0	0	1	0	0	1
Vejret er fantastisk i dag	Positiv	0	0	1	0	0	1	1	0	0	0	0	0	0
Jeg har mistet min telefon	Negativ	0	0	0	0	0	0	0	0	1	1	1	1	0

> **Spg.:** Kan i gennemskue hvad **Maskinen** tager som input og "lærer" på; herunder hvad det aktuelle problem ved dataen er?

"Optimering" og "generalisering" af tekstdata: *Trimming og stemming*

- Ordstammer (**stemming**)
- Små bogstaver
- Følgende ord er fjernede, fordi de er *særligt fremkomne* og derfor ikke meningsgivende i vores data: *Dette*, *er*, *en*, *jeg*, *så*, *af*, *har*, *min*, *i* (**trimming**)

Rå tekst	Bearbejdet
God	god
Dag	dag
Træt	træt
Alting	alt
Vejret	vejr
Fantastisk	fantast
Mistet	mist
Telefon	telefon

Klargjort input: Maskinlæsbar tekst

Input (feature/Tekst)	Output (label)	dag	god	træt	alt	vejr	fantast	mist	telefon
Dette er en god dag	Positiv	1	1	0	0	0	0	0	0
Jeg er så træt af alting	Negativ	0	0	1	1	0	0	0	0
Vejret er fantastisk i dag	Positiv	1	0	0	0	1	1	0	0
Jeg har mistet min telefon	Negativ	0	0	0	0	0	0	1	1

> Der ligger en grundlæggende antalelse om sammenhængen mellem ord *indenfor* teksterne. Hvilken?

Hvilke algoritmer kan tage sådanne et input

Listet efter uformel sværhedsgrad og komplesitet

- Lineær regression
- Logistisk regression
- **Naive Bayes** (tie)
- Beslutningstræer (tie)
- Støttevektormaskiner
- Neurale netværk, deep learning

Naive Bayes Classifier: Et populært værktøj til klassifikationsproblemet

- Relativ let at implementere; effektiv til klart definerede klassifikationer; *hurtig!*
- **En probabilistisk model klassifikationsmodel**.
  - Er det spam, er teksten positiv, er talen venstreorienteret, …
  - Ikke begrænset til 2 kategorier; men jo flere kategorier, des mere træningsdata

Naive Bayes Classifier: hvad gør den?

$$P(C \mid x_1, x_2, \ldots, x_n) = \frac{P(C) \cdot P(x_1, x_2, \ldots, x_n \mid C)}{P(x_1, x_2, \ldots, x_n)}$$

Hvor,

- $P(C \mid x_1, x_2, \ldots, x_n)$ er sandsynligheden for klassen C givet ordene $x_1, x_2, \ldots, x_n$ (**posterior sandsynlighed**).
- $P(C)$ er den forudgående sandsynlighed for klassen C (**prior sandsynlighed**, baseret på datasættet).
- $P(x_1 , x_2 , \ldots, x_n \mid C)$ er sandsynligheden for, at ordene $x_1, x_2, \ldots, x_n$ forekommer givet klassen C (**betinget sandsynlighed**).
- $P(x_1 , x_2 , \ldots, x_n)$ er sandsynligheden for, at ordene $x_1, x_2, \ldots, x_n$ forekommer (**normalisering**).

Naive Bayes Classifier: output

Input (feature/Tekst)	Model Sandsynlighed for Positiv	Model Sandsynlighed for Negativ	Model Output (label)	For-kodet Klassifikation
Dette er en god dag	0.85	0.15	1	1
Jeg er så træt af alting	0.10	0.90	0	0
Vejret er fantastisk i dag	0.80	0.20	1	1
Jeg har mistet min telefon	0.95	0.05	1	0

Positiv $= 1$, **Negativ** $= 0$

Naive Bayes Classifier: forklaring 1

Ord	Sandsynlighed for Positiv	Sandsynlighed for Negativ
dag	0.60	0.20
god	0.70	0.10
træt	0.10	0.80
alt	0.05	0.75
vejr	0.65	0.15
fantast	0.80	0.10
mist	0.05	0.85
telefon	0.05	0.85

$$P(C \mid x_1, x_2, \ldots, x_n) \propto P(C) \cdot \prod_{i=1}^{n} P(x_i \mid C)$$

Naive Bayes Classifier: forklaring 2

- Sandsynligheden for at en tekst er "**Positiv**" er baseret på hvor ofte et ord med høj sandsynlighed forekommer i positive tekster i forhold til alle forekomster.
- Vice versa for sandsynligheden for at en tekst er "**Negativ**".
  - "god": Har en høj sandsynlighed (0.70) for at *indikere* en positiv tekst.
  - "træt": Har en høj sandsynlighed (0.80) for at *indikere* en negativ tekst.
  - "mist" og "telefon": Har meget lav sandsynlighed for positivitet og høj sandsynlighed for negativitet.

"fantastisk dag":

- **Ord**, V: $x_1 = \text{fantast*}, \;\; x_2 = \text{dag}$
- **Forudgående sandsynligheder**: $P(\text{Positiv}) = 0.5, \;\; P(\text{Negativ}) = 0.5$
- **Betingede sandsynligheder for Positiv**: $P(\text{fantastisk} \mid \text{Positiv}) = 0.8, \;\; P(\text{dag} \mid \text{Positiv}) = 0.6$
- **Betingede sandsynligheder for Negativ**: $P(\text{fantastisk} \mid \text{Negativ}) = 0.1, \;\; P(\text{dag} \mid \text{Negativ}) = 0.2$
- **Posterior sandsynligheder**:

  $P(\text{Positiv} \mid \text{fantastisk}, \text{dag}) \propto 0.5 \cdot 0.8 \cdot 0.6 = 0.24$
  $P(\text{Negativ} \mid \text{fantastisk}, \text{dag}) \propto 0.5 \cdot 0.1 \cdot 0.2 = 0.01$
  Modellen vælger POSITIV (1), da den har den højeste sandsynlighed.

Spørgsmål; derefter Workshop i Google Colab

https://tinyurl.com/gymdagen