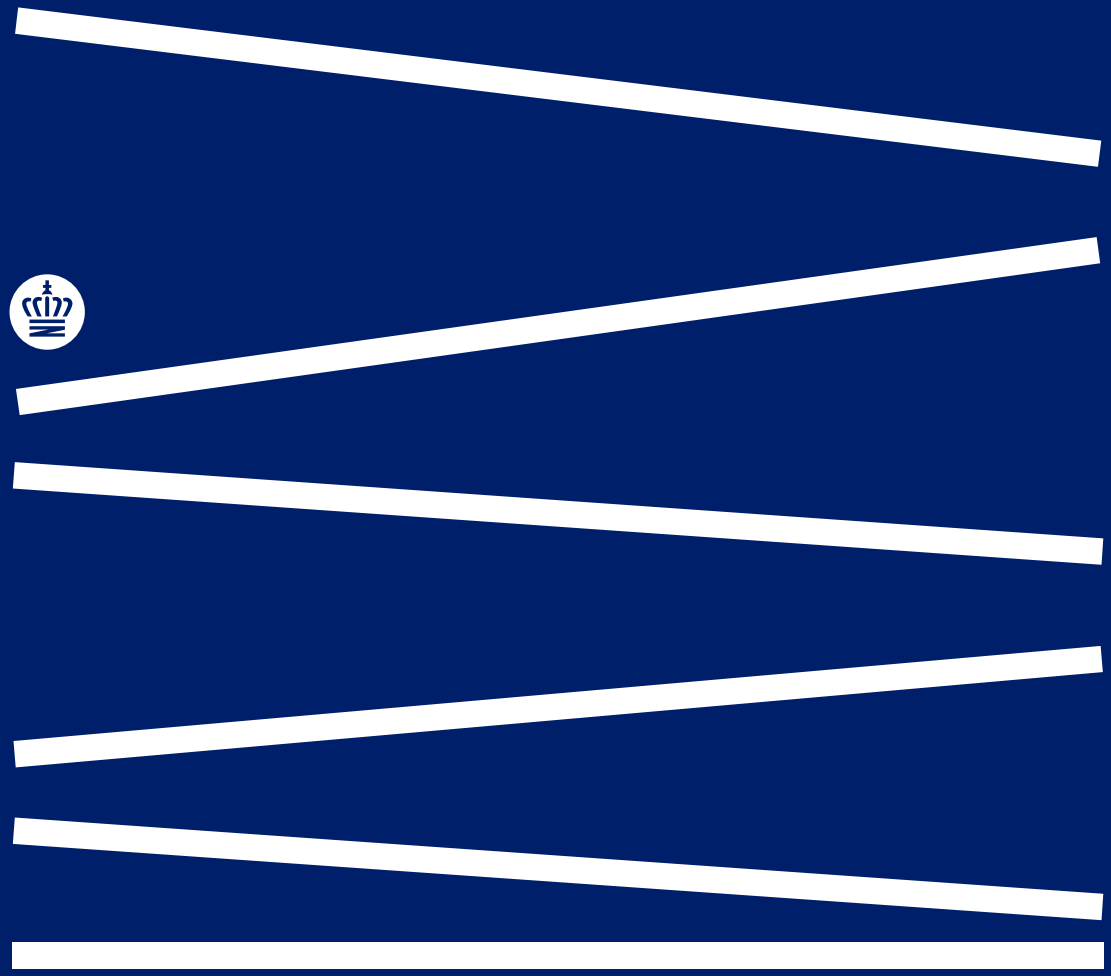
Identifying duplicates in a radio/TV archive

a data sprint from Master in Data Driven Organisational Development (MDO)

Bjarne Andersen
Head of Data
December 2023



DET KGL. BIBLIOTEK
Royal Danish Library

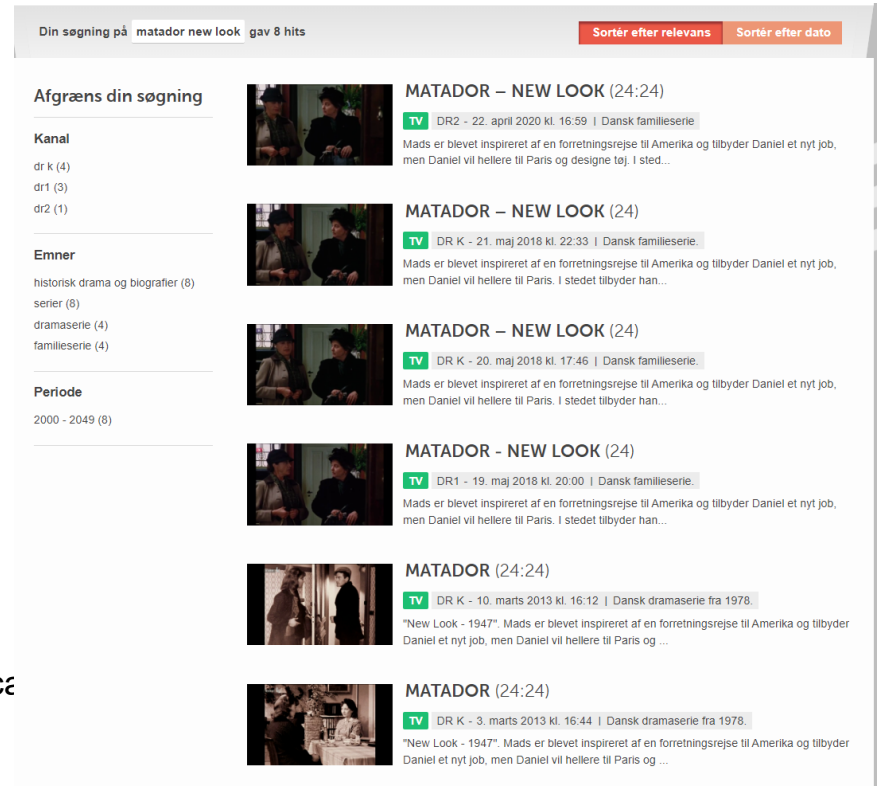
About me

- Started at the Royal Danish Library in 2001 as a java developer
- Worked a lot with web archiving in the early years
- Into Digital Preservation and repository systems
  - Co-founder of Open Preservation Foundation
  - Steering Commetee member of IIPC

- Head of the *it-development department* from 2008
- CIO from 2014 until 2021 where IT were merged into a larger *digital transformation division*

- Head of Data from 2021
  - Managing developers working with digital legal deposit and digital preservation
  - Responsible for IT architecture across the library
  - Co-responsible for cross institutional work within *data driven organisational development*

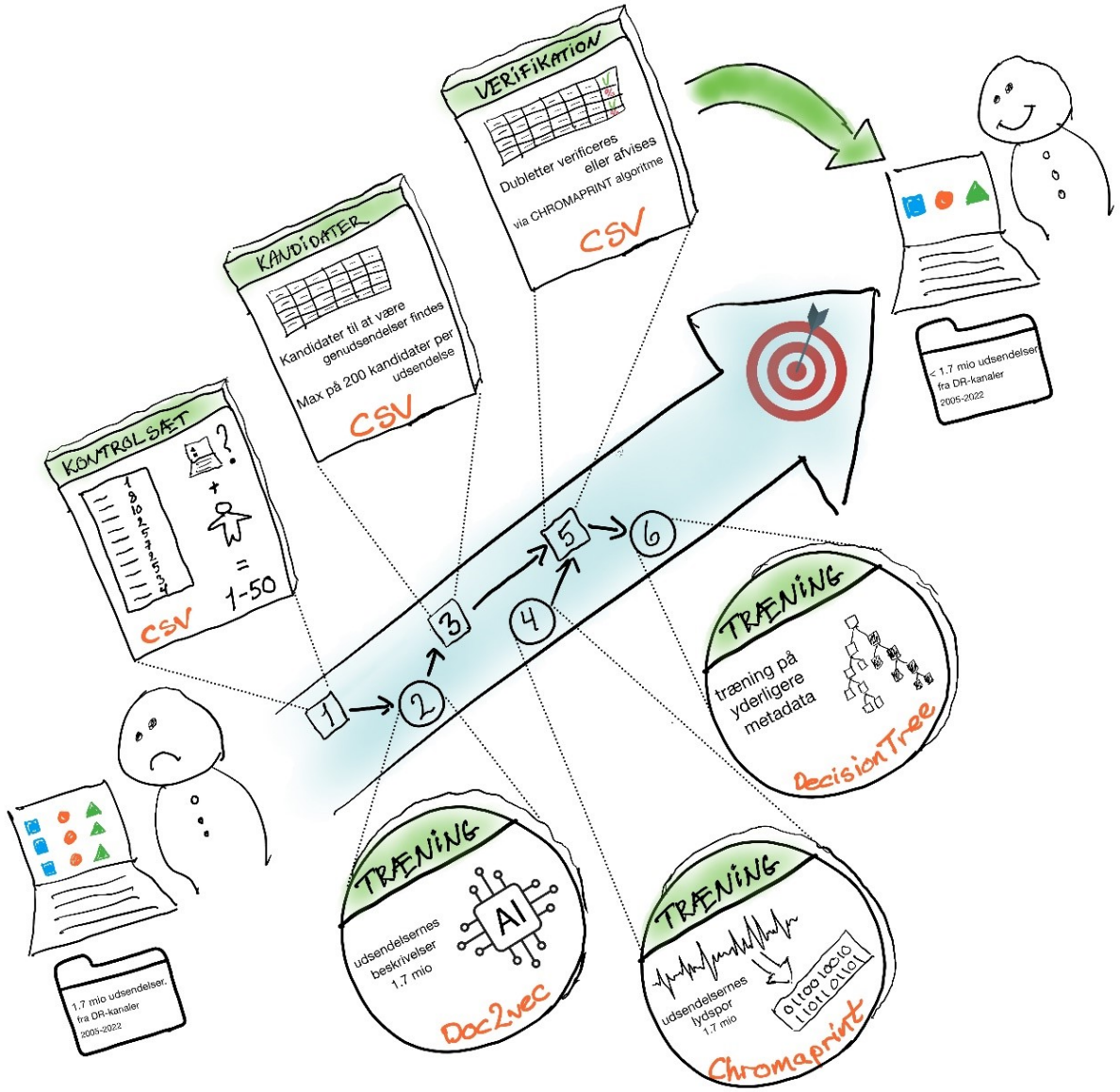
The Problem



- In the radio/TV archive we hold hundreds of thousands of duplicates (replays)
  - Danish Public Service Broadcaster (DR) estimates their replay rate to be around 50%
  - Programs replayed between 1 and 150 times
- When users search in our online portal they get multiple hits of the same program
- In this datasprint I wanted to investigate possibilities to use data (and metadata) to find duplicates (one program is a duplicate of another) and clusters of duplicates (many duplica of the same program)
  - Using metadata – how far can you go / how precise will it be?
  - Using the audio track – can you verify / reject the duplicate-status identified through metadata?
- I have been working with 1.7 mio programs from the national public service broadcaster (DR)
- I have analyzed 200Tbytes of audio tracks

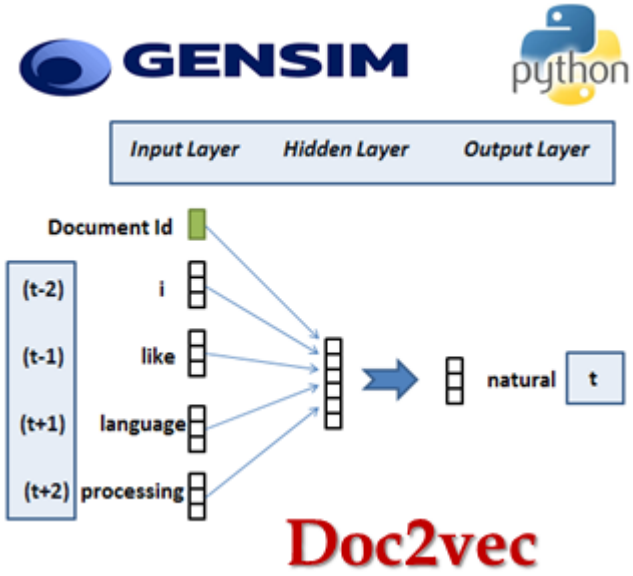
- During the data sprint an extra angle showed up – has to do with the automatic clipping of program-streams from the original 1-hour chuncks into program chuncks

Data Sprint Protocol



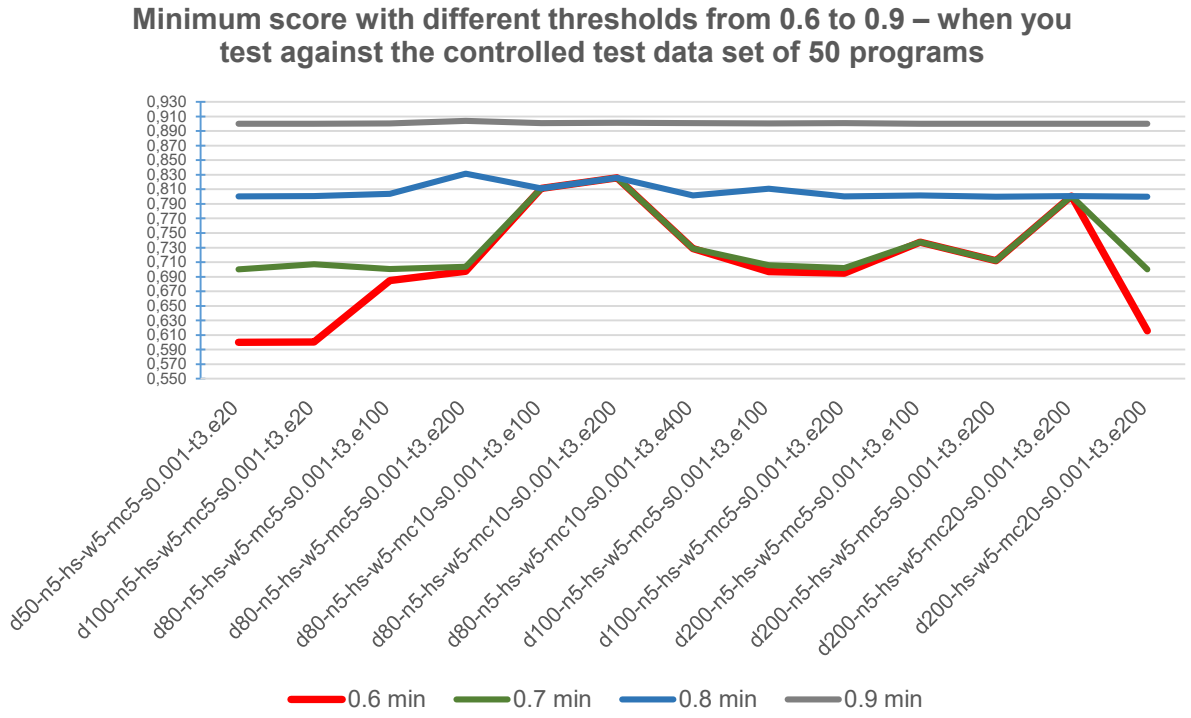
I’m sorry for the danish words in some of the graphics

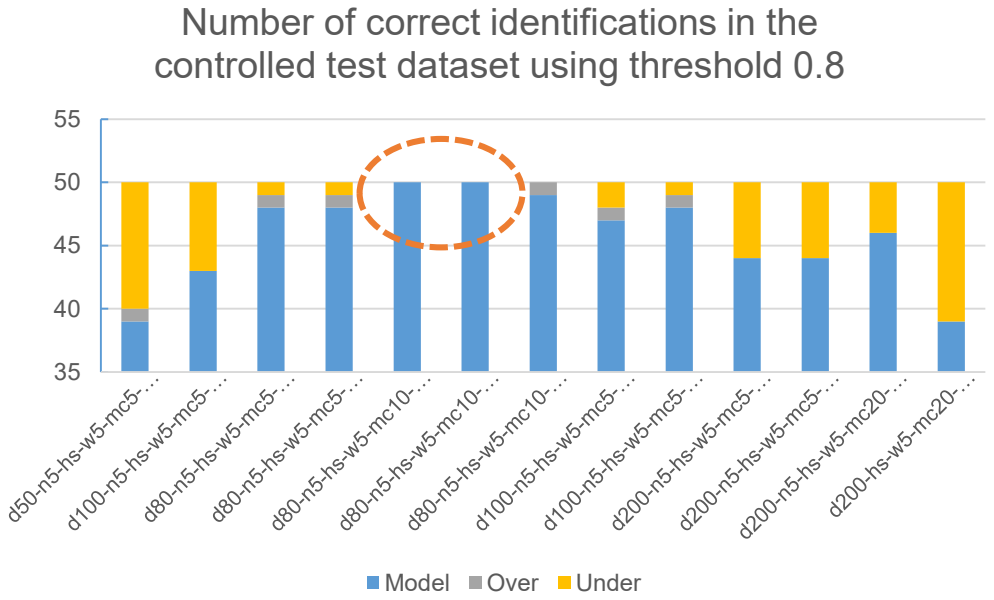
Training of a doc2vec algorithm



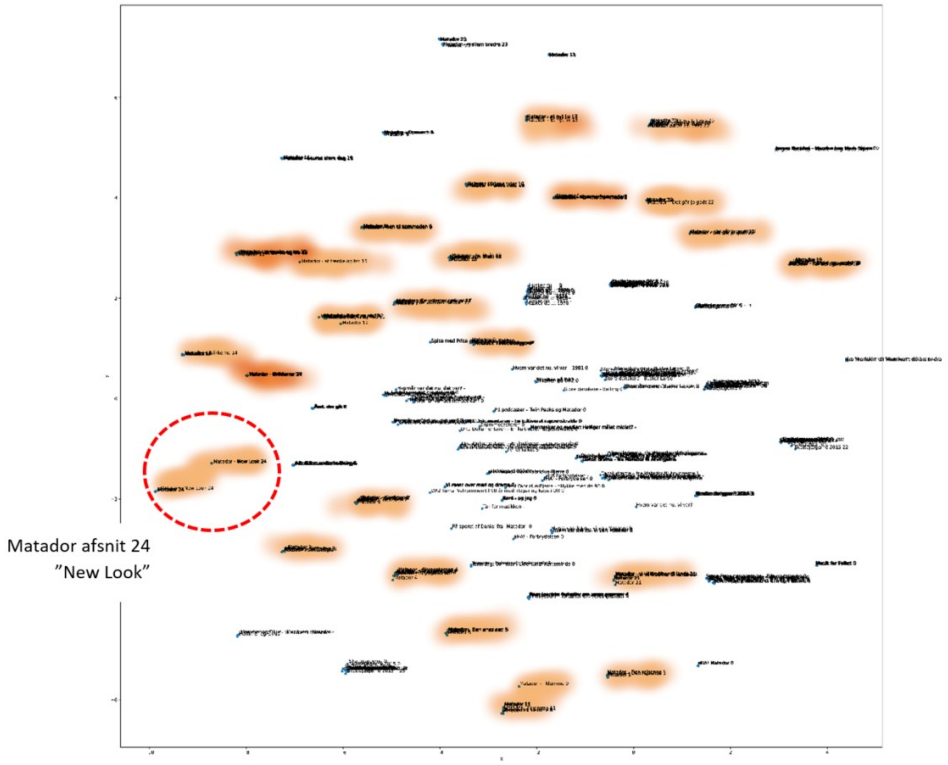
- Doc2vec is an algorithm that places a ”document” in a vector space of X dimensions to be able to identify closest neighbours and calculate the specific distance of two documents
- Example – 2 descriptions (metadata field) that looks like one another, but not completely identical
  - ”Mads er blevet inspireret af en forretningsrejse til Amerika og tilbyder Daniel et nyt job, men Daniel vil hellere til Paris og designe tøj. I stedet får Agnes buddet og takker ja. Senere har Daniel en ven med hjem fra Paris, og det skaber både vrede og forvirring.”
  - ”Mads er blevet inspireret af en forretningsrejse til Amerika og tilbyder Daniel et nyt job, men Daniel vil hellere til Paris. I stedet tilbyder han Agnes jobbet. Da Daniel senere har en ven med hjem fra Paris, går det op for Mads, at sønnen er homoseksuel, og han forstøder ham. Det får Ingeborg til at rejse væk i protest.”
  - https://www2.statsbiblioteket.dk/mediestream/tv/record/pvica_radioTV%3Adu%3Aecc03fe1-d5b5-4d15-9ae7-325fa63b1c17
  - https://www2.statsbiblioteket.dk/mediestream/tv/record/doms_radioTVCollection%3Auuid%3A74f937e2-487c-4bf7-beb7-1fa1513c4c02

Doc2vec performance





Doc2vec – from 80 to 2 dimensions (T-SNE)



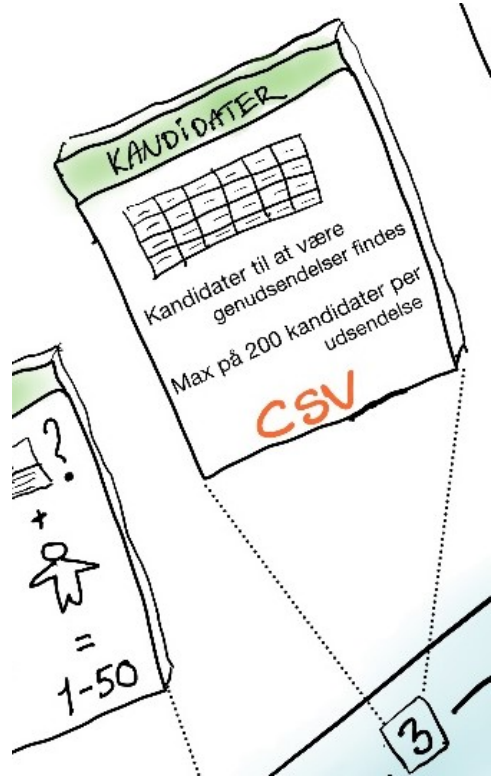
The illustration is made with demensionality reduction algorithm T-SNE – reducing the vector space from 80 to 2 dimensions

24 chapters of the Danish TV Series "matador" is colored with the orange background. Many duplicates of the same chapter is plotted right on top of each other (8-10 duplicates / replays) – but some chapters (e.g. chapter 24 New Look) is plottet close but not entirely aligned – this is showing, that the descriptions are not 100% identical but still placed near each other, even in a two dimensional vector space.

At the border and in the middle of the illustations programs matching the searc word "matador" is plotted – as shown these are placed distinct apart from the chapters of the TV Series

Identifying possible candidates to be duplicates based on program description

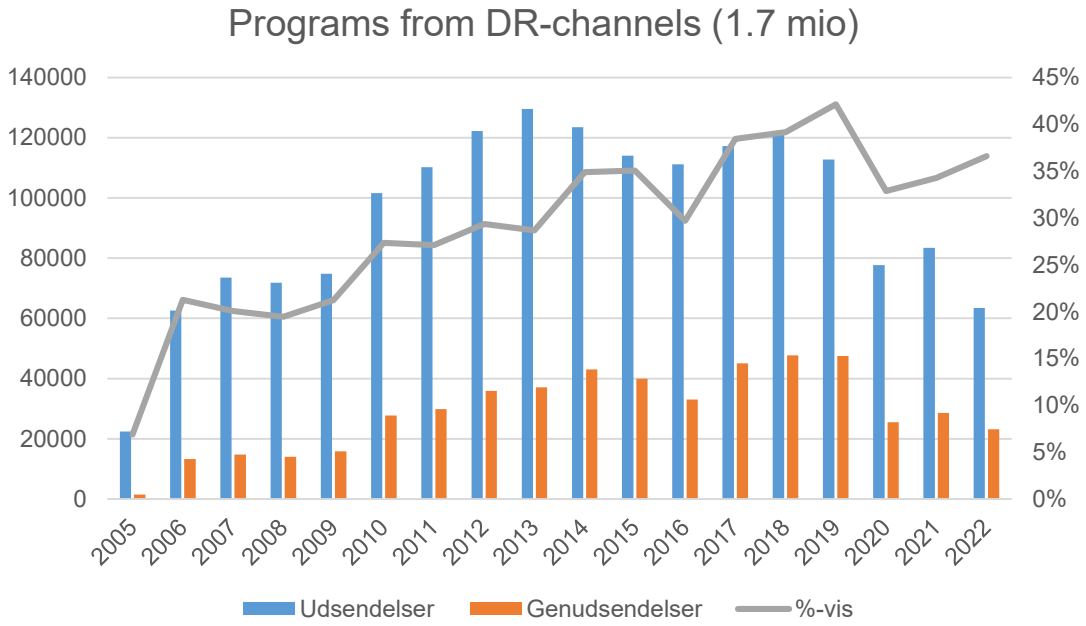
- For each 1.7 mio programs:
  - Identify all programs that based on the description field metadata have a doc2vec score over 0.8
  - This gives a first list of 15 mio candidate pairs (each pair is just two unique ID-numbers)
  - Since many duplicates are identified ”in both directions” this list can be uniqued into a list of around 10 mio candidates
  - The number is high because 1 program can be i a cluster with e.g. 50 duplicates (50+49+48+47+46…..)
  - I choose to limit to the following
    - Description field must be more that 5 words
    - Programs with more than 200 replays are filtered out
  - The big question: How do we verify of reject these duplicate candidates ?



Analysing audio tracks using chromaprint

— Chromaprint is an algorithm that can generate a "fingerprint" of a soundtrack and afterwards you can compare these fingerprints and get a figure for how similar they are.
— Chromaprint is a heavy CPU-using algorithm
  – Took approx. 14 days to generate fingerprints for the 1.7 million programs
  – Took more than 4 months to check the 10 million candidates
— Verifies whether 1 pair of programs from the candidates list are actually a duplicate by looking at whether the audio track overlaps by at least 30 seconds within a 2 minute window that runs across the two audio tracks (both tracks at once) – i.e. allows up to 90 seconds of difference between the two tracks relative to where they are identical
— This is where the problem with KB's "skewed clipping" is taken into accunt
  – We either cut according to Ritzau data = the planned broadcast times
  – Or we cut by TV meter data = the actual broadcast times
  – Even on DR's channels (without commercials) there can be longer slack of several minutes
— I had to adjust the chromaprint comparison by cutting chunks of 30 seconds of respectively one and the other audio file in turn – allowing up to 4 minutes of slack in total
  – It identified approx. 20% more duplicates among candidates

Verified duplicates

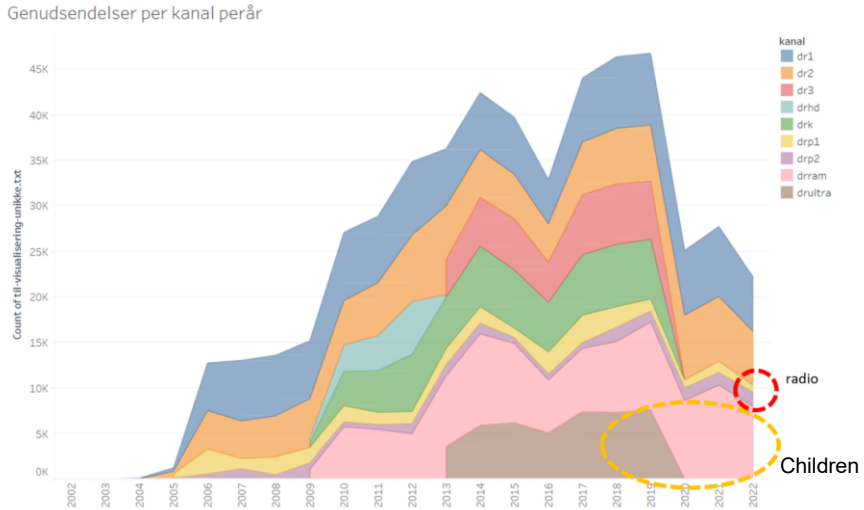


The largest jumps are linked to the DR-channel strategy / availability

2009: 3 new channels are launched: Ramasjang, DR K, DRHD

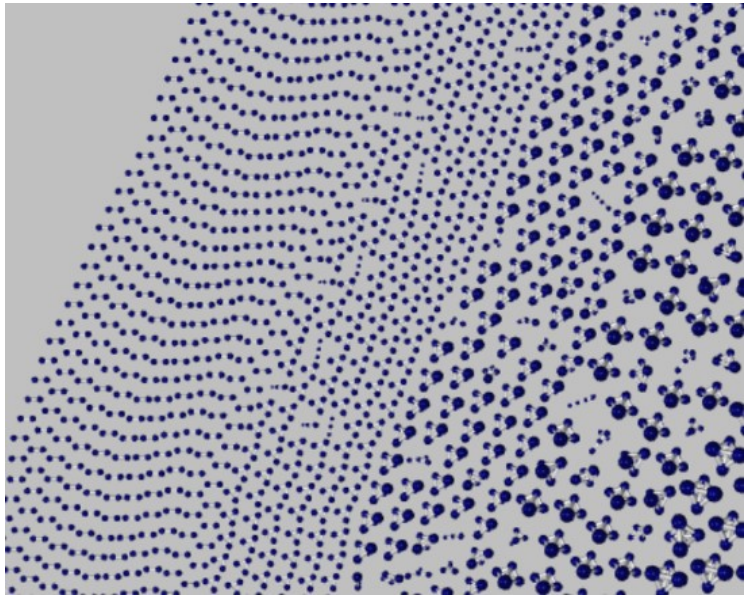
2013: DR Ultra is launched og DR3 replaces DRHD

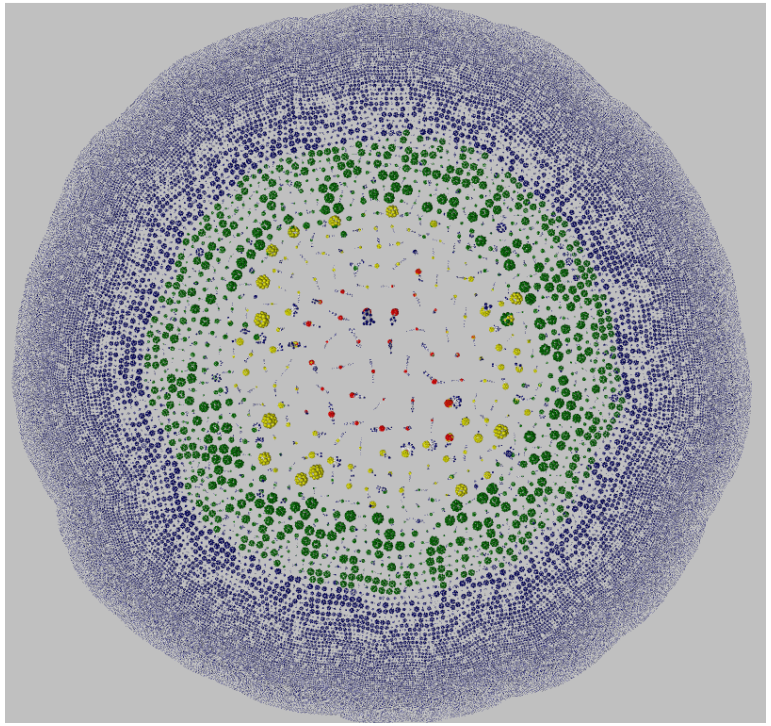
2020: 3 channels are shut down as broadcast: DRK, DR3, DR Ultra

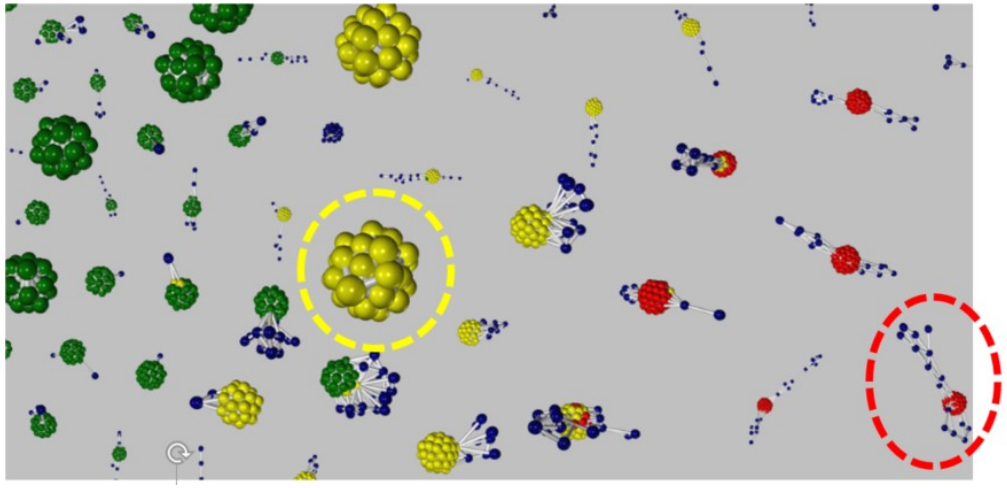


DET KGL. BIBLIOTEK
Royal Danish Library

Visualising duplicates



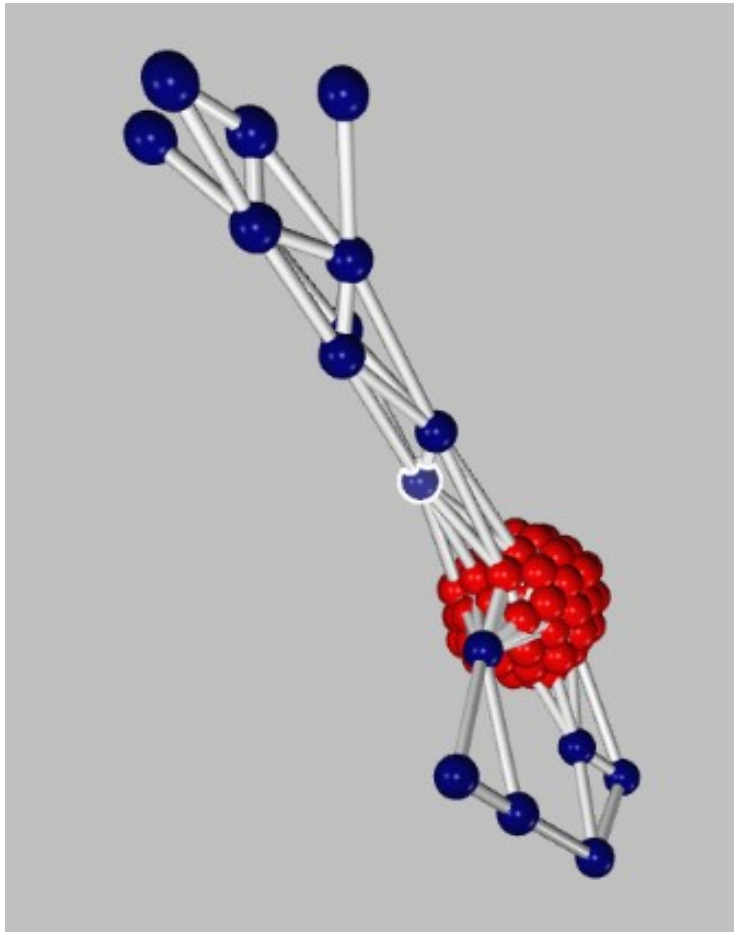




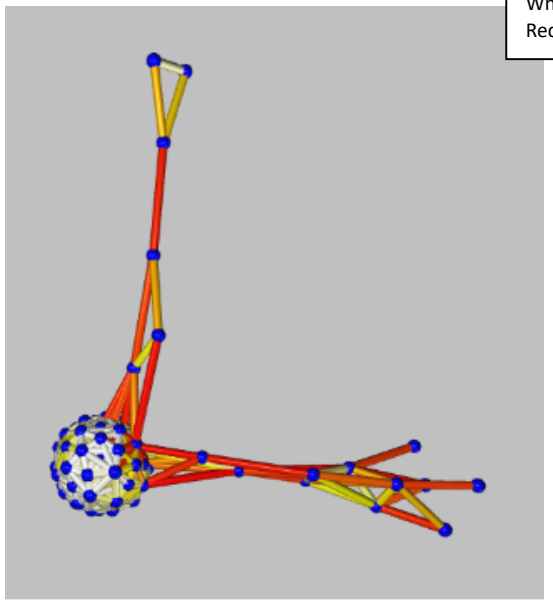
DET KGL. BIBLIOTEK
Royal Danish Library

Farvekodningen betyder: Blå noder har under 20 kanter, grønne noder har under 50 kanter, gule noder har under 100 kanter og røde noder har over 100 kanter

Visualising continued



Visualising the same cluster – this time with colours showing the chromaprint offset.
White = 0 secs
Red = 60 secs

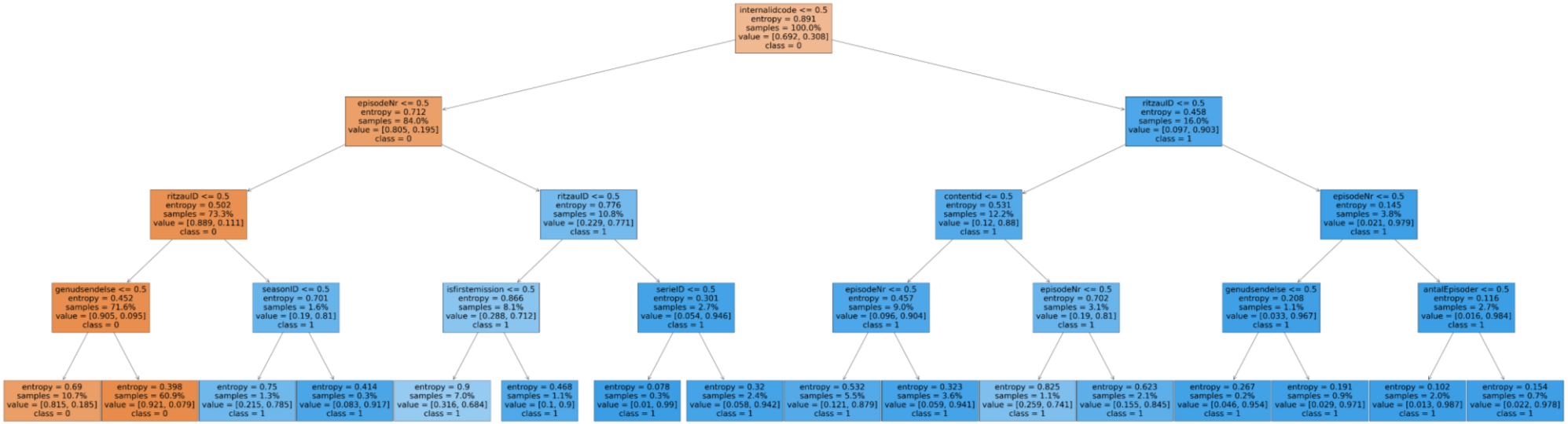


Node	Program actual start	Displacement from prev. Node (sec.)	Calculated offset from Chromaprint (sec.)
doms_radioTVCollection:uuid:1e134414-5582-4182-a0f4-1ece0be31bd0	0:54		
doms_radioTVCollection:uuid:250eee91-1c10-4df4-a8cd-3a6059998802	1:28	Ca. 34	34.5
doms_radioTVCollection:uuid:68e9c800-12f7-4aaa-99e0-4874b86e0292	2:23	Ca. 55	55.5
doms_radioTVCollection:uuid:36531d70-a40b-4aef-a357-520a5a497011	3:01	Ca. 38	38.5
doms_radioTVCollection:uuid:a116c204-5f3c-44ed-be30-2e47e70e7e96	3:49	Ca. 48	47.6

Did I find any duplicates? YES!

- Overall, 100,700 clusters were identified out of the total of 523,930 broadcasts that have 1 or more rebroadcasts.
- So the potential improvement in relation to Search without duplicates is that 423,230 duplicates/replays can be removed, which corresponds to 24.8% of all broadcasts in the analyzed data set
- So an average replay rate of around 5.2
- The most rebroadcast is "Stress af med DR3" (many episodes) - 134 confirmed replays in the cluster (out of over 150 candidates)
- Sharply followed by "Rosa fra Rouladegade" (many episodes) - all well over 100 replays

What can we do with metadata alone? - DecisionTree



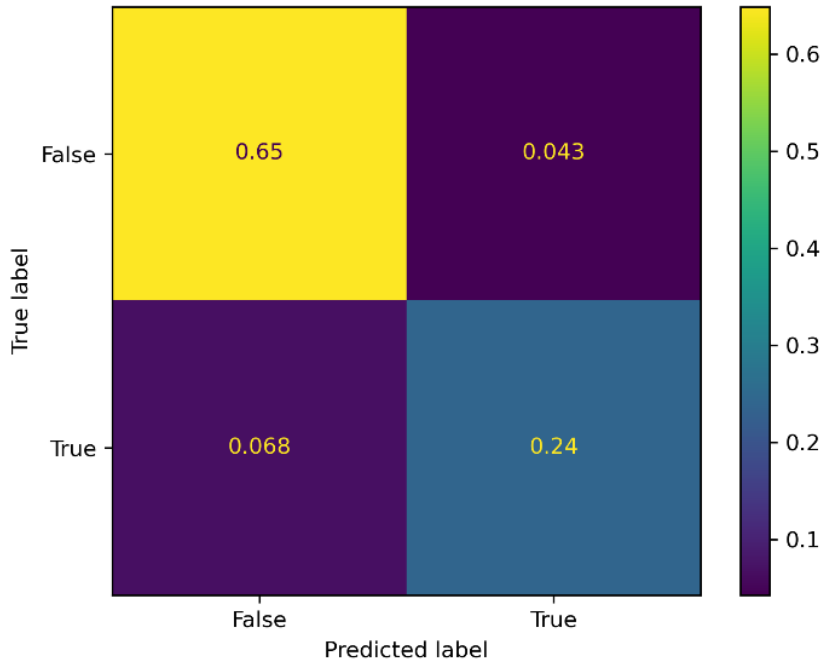
internalidcode <= 0.5
entropy = 0.891
samples = 100.0%
value = [0.692, 0.308]
class = 0

episodeNr <= 0.5
entropy = 0.712
samples = 84.0%
value = [0.805, 0.195]
class = 0

ritzauID <= 0.5
entropy = 0.458
samples = 16.0%
value = [0.097, 0.903]
class = 1

contentid <= 0.5
entropy = 0.531
samples = 12.2%
value = [0.12, 0.88]
class = 1

How precise are metadata itself?



Confusion matrix on the left expresses how many times the algorithm finds the correct answer "replay or non-replay".

As you can see, it identifies 24% replays as correct (True/True) while it also identifies 65% non-replays (False/False) correct.

The interesting thing is how many false negatives there are (a replay is not found even though it is there) and how many false positives there are (finds a replay that doesn't exist).

As can be seen, there are 6.8% false negatives and 4.3% false positives.

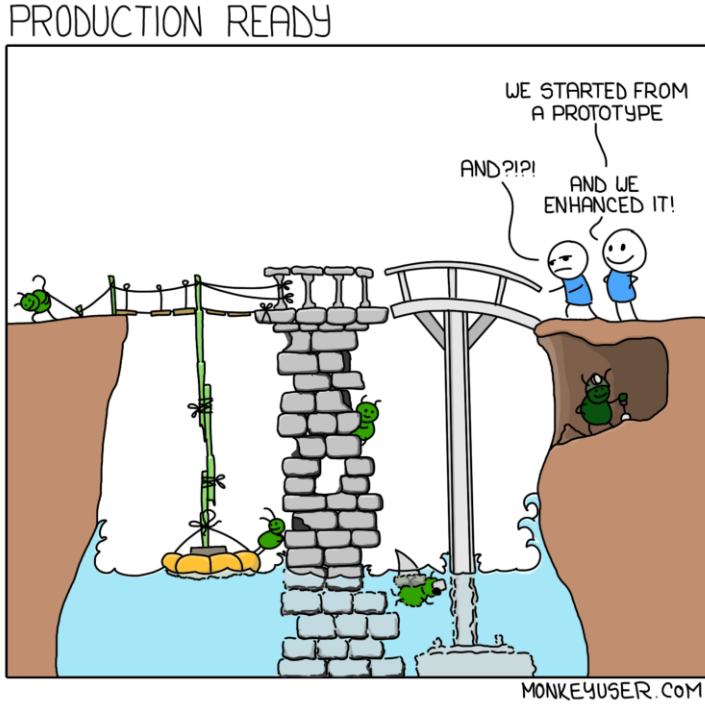
Depending on what you want to use the algorithm for, that performance may well be fine. For offering users to filter out replays in a search result, it is the 4.3% false positives that are most problematic.

Here, the search engine will erroneously filter out 4.3% of the search results.

The algorithm's weighted F1 score is calculated to be 0.8878, which is relatively high – so metadata in the archive can do a lot, but not everything.

1. Meta alone therefore finds 24% of rebroadcasts completely correct
2. Metadata alone finds the 65% non-retransmissions completely correct
3. The "problem" is the 6.8% rebroadcasts that do not exist
4. And what is particularly problematic is perhaps the 4.3% that are identified as rebroadcasts without being so – this figure is perhaps artificially high due to the skewed clippings.
5. It all depends on what you want to use this data for.

Next steps



- Can this be put into production?
- My strategy / method has a number of challenges
  - A number of choices along the way – use of description field – no more than 200 rebroadcasts – thresholds etc.
- It needs to mature – the algorithms can be further tweaked to give even better results
  - Workflows must be set up so that all new programs are taken into account in relation to all existing ones, i.e. incremental training of the algorithms.
- A lot of hardware must be used - especially for the first training on the entire archive - not so much after that
- A new algorithm must be built which identifies the "master broadcast" in a cluster of rebroadcasts - which version should the user actually be presented with?
- For clusters with many duplicates, it may be easy to take "one in the middle" - but for a small cluster with perhaps only 2 programs, it is difficult to know which one is "best"
- The user interface needs to be adapted to include this information in metadata and for the search logic to use the information for filtering ("..don't search duplicates...")
- So, unfortunately, it's not entirely ready for production
- BUT – the project has shown that there are plenty of opportunities in using data and algorithms to solve specific challenges in KB's archives.

Questions

DET KGL. BIBLIOTEK
Royal Danish Library