

ML-assisted ISCO-coding in the Danish LFS

Daniel F. Gustafsson (dfg@dst.dk)



Agenda

1. Background
2. Model and results
3. Experiment
4. Next steps

ISCO-coding in Denmark

- Data for the Danish LFS is collected throughout the year
 - Around 15.000 interviews each quarter
- Information on current/previous occupation collected as free text
 - Around 7,000 cases to be given 4-digit ISCO-08 codes each quarter

?

What is your position/job title? 

Data Analyst

?

Please, describe the main tasks of your job 

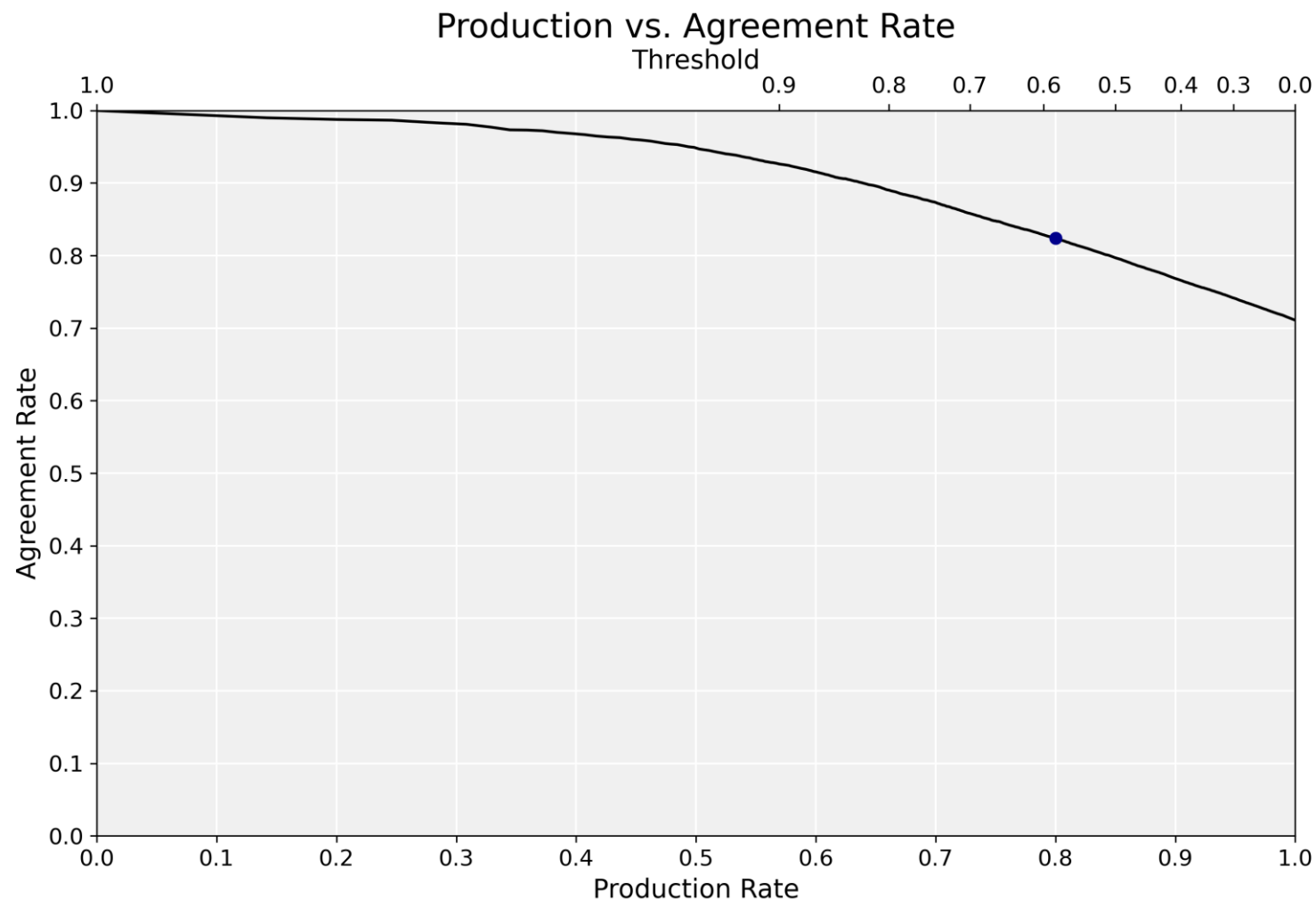
(For example, answer the phone, input forms, telephone sales, paint walls. Perhaps state what the work involves)

My tasks include collecting and analyzing data, preparing statistical reports, ensuring data quality, and supporting their publication.

Classification model

- We wanted to create a ML model, that assisted our coders
 - Increase consistency
 - Increase efficiency
 - Make it easier for newer employees to assist with coding
- Settled for a RoBERTa multi-classification model
 - Trained on free texts from the LFS in Danish and English (2013-2023)
 - For each respondent, outputs probability of all ISCO codes
 - Delivers an accuracy around 70-75 percent and top-6 accuracy above 90 percent

Accuracy at different thresholds



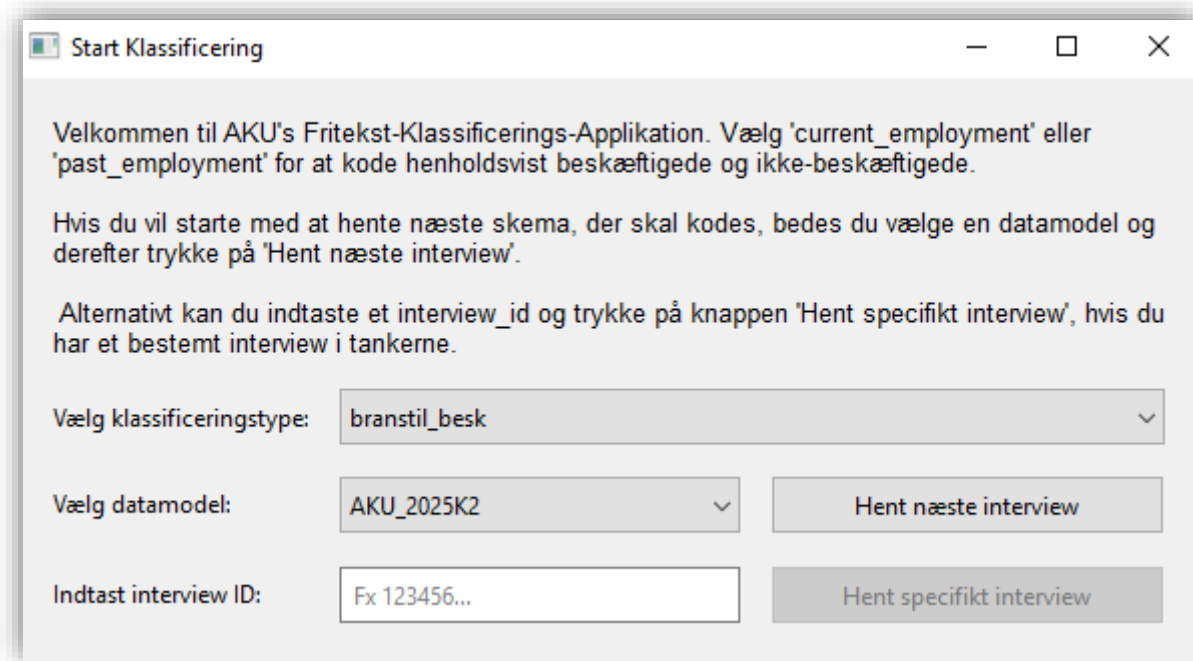
Data processing flow

- Daily batch of data loaded each morning
- Text responses moved to classification table in DB
- Auto-assignments made based on:
 - From look-up
 - Tax registers (if predicted by model)
 - High predicted probability (not implemented)
- Remaining cases labeled as 'ready for coding'

Application development

- We needed an application that:
 - Collects and presents all relevant information
 - Helps the coder identify the most suiting code
 - Saves the result to the database
- Opted for a PyQt-based solution in Python, using a Postgres database, that:
 - Fetches the next interview in the queue
 - Collects data from:
 - Interview
 - Registers
 - Prediction
 - Allows the user to save the information to the database

Application design



The screenshot shows a window titled "Start Klassificering" with standard Windows window controls (minimize, maximize, close). The window contains the following text and controls:

Velkommen til AKU's Fritekst-Klassificerings-Applikation. Vælg 'current_employment' eller 'past_employment' for at kode henholdsvis beskæftigede og ikke-beskæftigede.

Hvis du vil starte med at hente næste skema, der skal kodes, bedes du vælge en datamodel og derefter trykke på 'Hent næste interview'.

Alternativt kan du indtaste et interview_id og trykke på knappen 'Hent specifikt interview', hvis du har et bestemt interview i tankerne.

Vælg klassificeringstype:

Vælg datamodel:

Indtast interview ID:

Application design

Fritekstklassificering - Nuværende beskæftigelse

Registerdata

PERSON_ID: [redacted]
INTERVIEW_ID: [redacted]
AKUIPNR: 1006569
AUDD TEKST: [redacted]
BRANCHE_REG: 461100
BRANCHE_REG_TXT: [redacted]

Spørgeskemadata

Beskstatus.B2Status: 1
Beskstatus.loenmodtager: 1
Branstil.B2persled: 2
Branstil.B2Gade: [redacted]
Branstil.B2Navn: [redacted]
by: [redacted]
personage: 46
persongender: 2
postnummer: [redacted]
Arbtid.B2Midt: 2
Socstat.B4Soc: 1

Branche i hovedjob

klass_var: B2VirkD
svar_bdt: [redacted]
svar_bdt_detaljeret: [redacted]
klass_kode: 4611
klass_metode: 5
reg_kode: 4611

Stilling i hovedjob

klass_var: B2StilD
svar_bdt: [redacted]
svar_bdt_detaljeret: [redacted]
klass_kode: None
klass_metode: 0
reg_kode: None

Branche i bijob

Annuller

4110 - Almindeligt kontorarbejde
2422 - Arbejde med virksomhedsstrategi/policy
3343 - Chef- og direktionssekretærarbejde
2619 - Andet juridisk arbejde
2421 - Analytikerarbejde inden for ledelse og virksomhedsorganisation
3359 - Andet arbejde med forvaltning og udøvelse af lovgivning
2131 - Arbejde inden for biokemi, biologi, botanik, zoologi og beslægtede områder
0110 - Militært arbejde på officersniveau
0210 - Militært arbejde på befalingsmandsniveau (underofficersarbejde)
0310 - Militært arbejde, øvrige rangordener

Implementation results

- Implemented fully for our two coders in 2025Q1
- Took a bit of getting used to, but coding speed has increased overall
 - Especially after adding productivity features, like tooltips and hotkeys
- Fewer interviews to be coded manually
 - Could be reduced further with automatic assignment of probable codes

	2024q1	2025q1	
	Actual	Actual	Assigned if $p > 0.85$
Lookup	1,819	1,734	1,734
Register data, automatically assigned	0	2,423	2,423
ML-assigned	0	0	1,071
Manually checked/coded	5,351	2,640	1,569
Total	7,170	6,797	6,797

Experiment – Design

- As part of a thesis project, an experiment was conducted, in order to test the effect of coding with and without ML-assistance
- Coders were asked to recode around 200 cases, now without model suggestions
- Finally, they were asked to evaluate the deviations and select the best fit
- Allows us to analyze how the model influences the decisions of coders

Experiment - Results

- Coders select top cases more often when presented (67% -> 69%)
- Coders agree more often when suggestions are presented (65% -> 68% at 1-digit level)
- In cases of ambiguity, coders tend to think that the more probable code is more suitable
- Model suggestions reduce coding time (49 -> 41 seconds on average)

Next steps

- Ordering of predictions
- Decide on automatic assignment strategy
- Decide on re-training of the model
- Expanding model with auxiliary variables

Thank you for your attention!