DESTATIS
Statistisches Bundesamt

Text Classification Projects at Destatis

Use-Case 1 Household Budget Survey: *Ariane Lestrade, Bogdan Levagin, Dr. Jerome Olsen*

Use-Case 2 NACE-Classification: *Susanne Wegner, Julius Weißmann*

AIML4OS Project – WP10 – June 2025

Destatis
Statistisches Bundesamt

Agenda

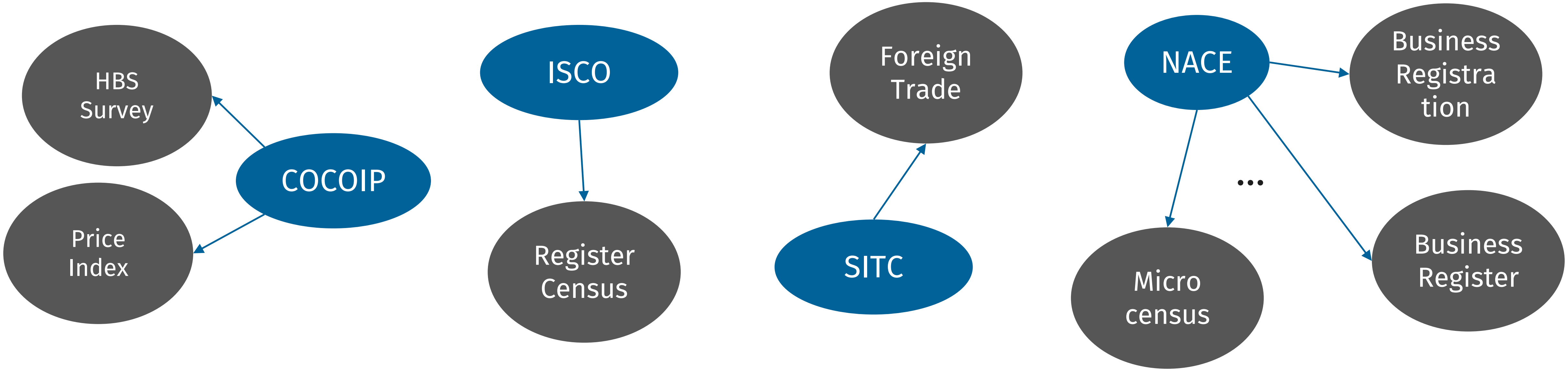
(1) Text Classification Projects at Destatis

(2) Focus Use-Case 1 - German Household Budget Survey for COCOIP-Classification

(3) Focus Use-Case 2 – Business Registrations for NACE-Classification

Text Classification Projects at Destatis

Text Classification using International/National Classification Systems



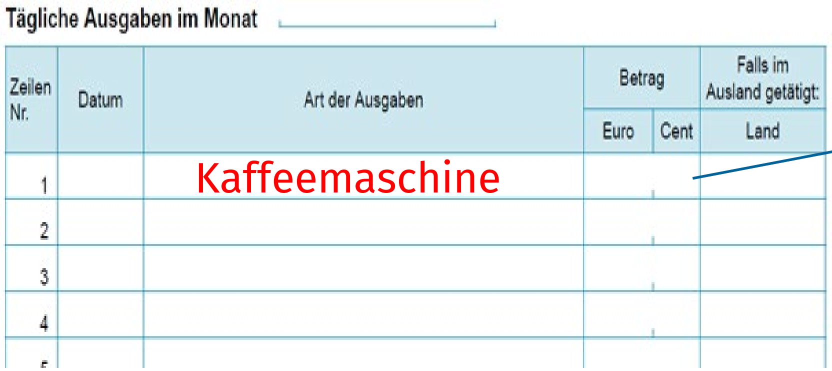
Key challenges to production — two potential solutions

Use-Case COCOIP: Using a cloud-oriented platform hosted centrally by Destatis (Cloudera)

Use-Case NACE: Using a decentralized infrastructure, hosted by individual state statistical offices, using more conventional, server-based setups.

Focus Use-Case 1 - COICOP

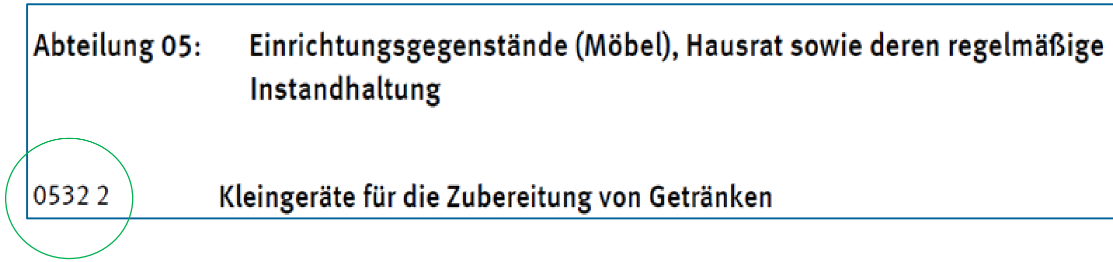
HBS Survey - Need for Automatization



Tägliche Ausgaben im Monat

Zeilen Nr.	Datum	Art der Ausgaben	Betrag		Falls im Ausland getätigt:
			Euro	Cent	Land
1		Kaffeemaschine			
2					
3					
4					

COCOIP-Classification



German HBS 2023: 5 Mio records were expected to be classified!

Initial Situation for the Project

Initial Situation

Migration to Spark: Current Results

Build on the current ML system using light preprocessing, TF-IDF, and traditional classifiers.

COCOIP-Classification	F1-Score	Accuracy	Duration (Min)	Laufzeit für Fine Tuning	RAM Peak
R-Server – Last Production Run (RF mit Ranger)	0.80	0.90	~ 30 Min	Only on data subsample possible.	~200 GB
R-Server - Tests with LR (Scikit-Learn)	Does not converge, Error.		Computation takes too long, too risky for production.		15 GB
Spark Algorithm (LR with MLLib)	0.78	0.88	~ 3 – 5 Min	~ 50 Min for Gridsearch with 18 different combinations of parameters.	~40 GB
Tests with RF (MLLib)	RAM-Error, not well fitted for multi-class problems				

Experiment with fastText

- We used fastText for embeddings and classification task.
- Better results, while training fastText on our own dataset, instead of pretrained fastText models.
- fastText obtains better results, especially with covariates, but seems not as robust on new difficult data.

COCOIP-Classification	Test Set		Stresstest Dataset	
	F1-Score	Accuracy	F1-Score	Accuracy
Spark Algorithm (LR with MLLib)	0.78	0.88	0.29	0.32
fastText (with covariates)	0.89	0.94	0.16	0.19
fastText (without covariates)	0.80	0.90	0.23	0.30

ML in production for HBS Survey
Human-in-the-Loop ML Process

How many errors can the ML system be allowed?

*Performance: **89%** Accuracy on Test Set using only ML alone.*

Manual review - Balancing Data Quality with Staffing Capacity

***Improved** Performance: **96%** accuracy on the test set when all records below a **55%** score are manually reviewed.*

Operational Model in Production

If the score is ≥ 55%, the ML prediction is saved with an "ML" flag.

If the score is < 55%, the ML prediction is saved with a "recommender" flag and manually checked by nomenclature experts.

Key Data from the Latest Production Run

2,726,145 processed records from **50,923** Households.

- **333,860** records comes from app users.
  (App uses a search algorithm — only entries that couldn't be automatically classified remain, category "Other")
- **2,393,479** records from paper users.

That's about half as many codable entries as originally expected (5 million).

Classification outcomes:

73% had a score ≥ 55% (ML automatic classification only)

27% were manually reviewed via **Recommender**

Focus Use-Case 2 - NACE

Overview

(1) Use case: Current situation

(2) New search solution with ML

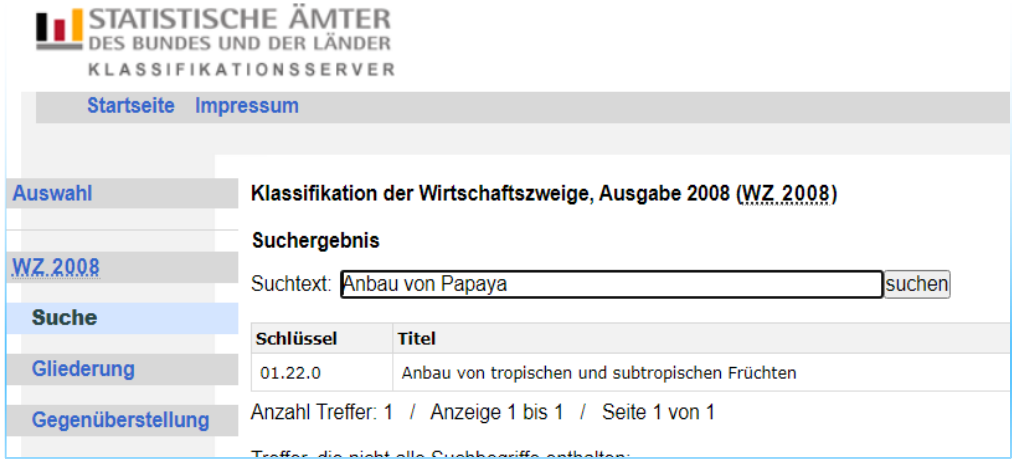
(3) Implementation: Architecture, Pipeline, Performance

(4) Outlook

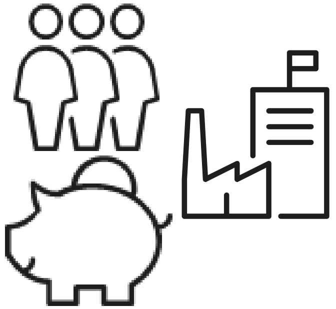
Current situation

Tool: **KlassServer**

- Rule based legacy search
- Hard to maintain
- UI and API



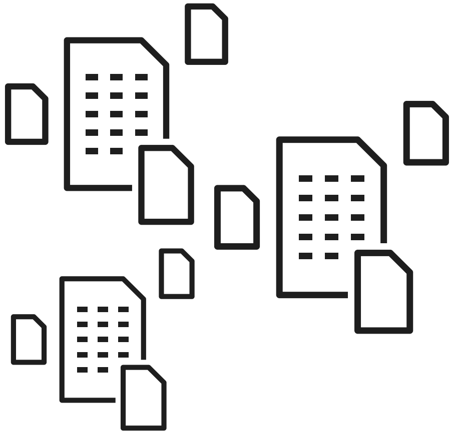
Many users in and outside of Destatis



Low satisfaction rate

☹

A wealth of datasets manually labelled over the years



https://www.klassifikationsserver.de/klassService/

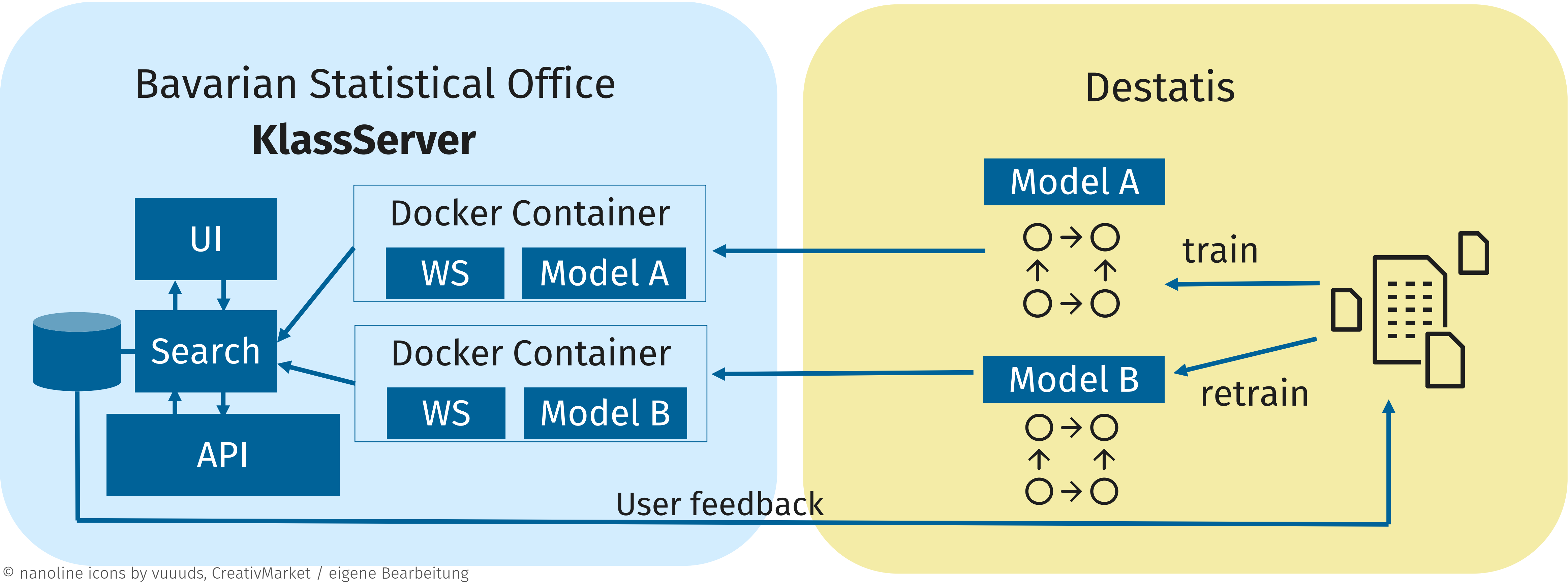
New NACE search using Machine Learning

Goals

- Improve the current search
- Be able to quickly implement new models
- (Partial) automatization of the labelling process in various statistics via API

New search UI

Model Hosting Architecture



Data and Pipeline

Dataset

- Business registration data (classification scheme NACE rev. 2)
- Ca. 5 mio. observations
- 833 classes on the 5th level

Tested algorithms

- Algorithms:
  - Naïve Bayes
  - Logistic Regression
  - SVM
  - Random Forest
  - Neural Network
  - BERT

Current Pipeline

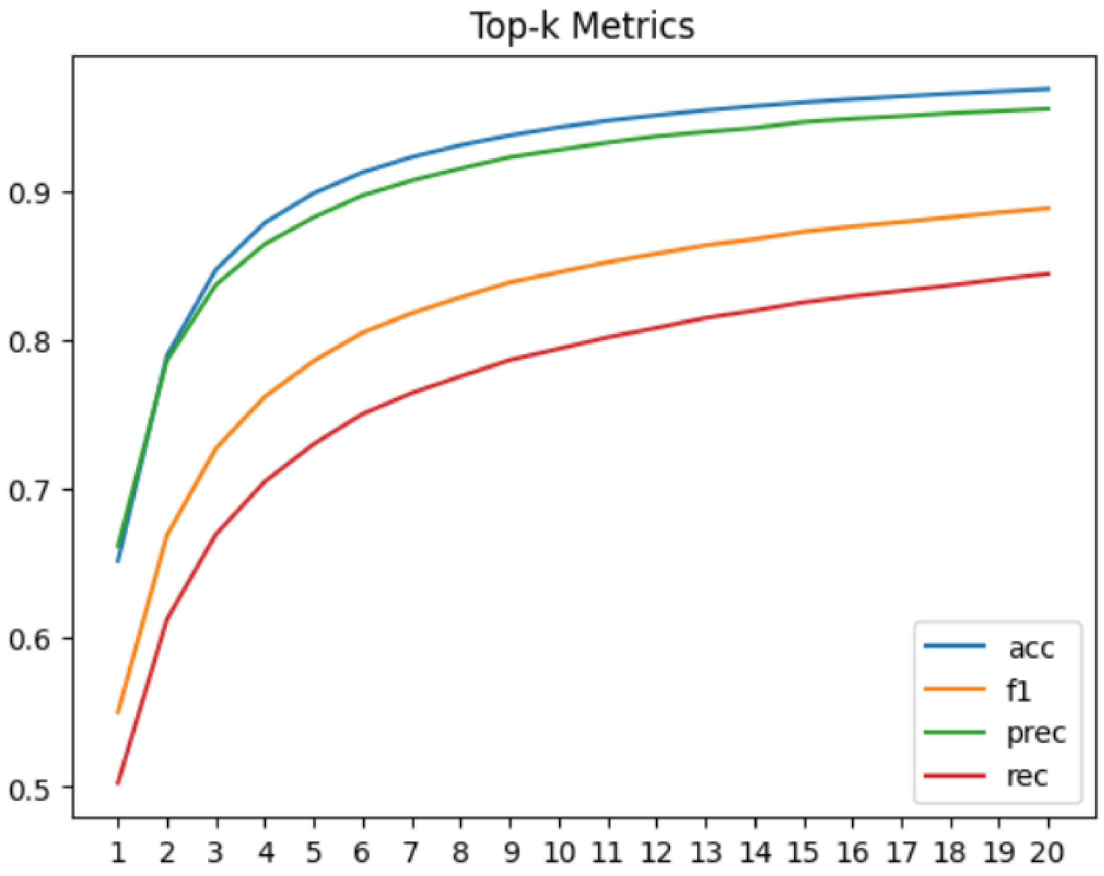
- Algorithm:
  - Sklearn Logistic Regression
- Preprocessing
  - Lowercase
  - Umlaut standardization
  - Use lowest possible hierarchy level
- Vectorization
  - N-Grams (3,5)
  - Hashing Vectorizer (2*13 features)

Current Results

Top 1 results on all levels

Level	F1	Accuracy	Precision	Recall
2	0.63	0.79	0.71	0.59
3	0.47	0.72	0.55	0.43
4	0.40	0.69	0.47	0.36
5	0.53	0.65	0.66	0.50

Top-20 results on the 5th level



Next steps

(1) Improve the model (ongoing)

(2) Finish development of UI and API

(3) Develop a model for NACE rev 2.1

(4) Implement automation workflows into statistical processes

(5) Develop and integrate AI models for other classifications into the KlassServer

Contact

Statistisches Bundesamt

www.destatis.de

www.destatis.de/kontakt

C13 - Ariane Lestrade

Ariane.Lestrade@destatis.de

C13 - Bogdan Levagin

Bogdan.Levagin@destatis.de

C13 – Susanne Wegner

Susanne.wegner@destatis.de

C13 – Julius Weißmann

Julius.Weissman@destatis.de