

What are we asking from **Tabular Data?**

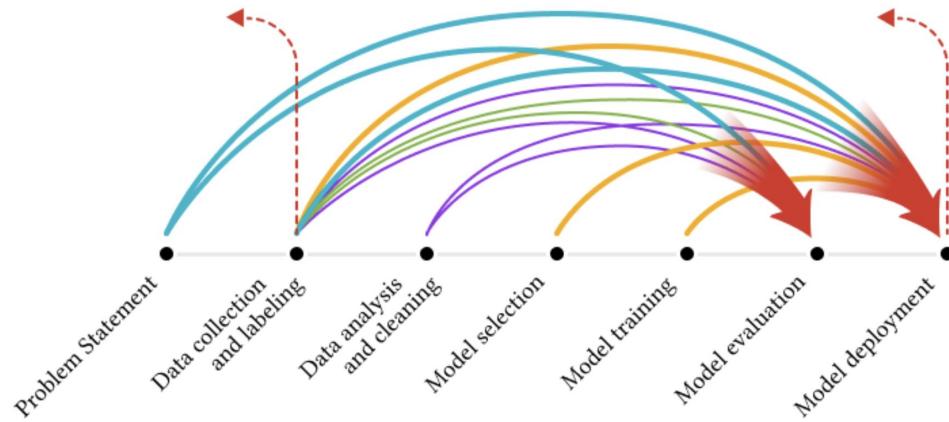
Madelon Hulsebos

6 December, Copenhagen

AI for Tabular Data workshop @ EurIPS 2025

As a data scientist,
in the “real world”,
I realized 3 things...

Realization 1: most of my work was data work



Data science = “80% data work, 20% model work”

“Everyone wants to do the model work, not the data work”: Data Cascades in High-Stakes AI,
Sambasivan et al., 2021

Realization 2: everyone is doing the same....

Medical Insurance Cost Prediction

Notebook Input Output Logs Comments (1)

```
In [1]:  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import StandardScaler, OneHotEncoder, OrdinalEncoder  
from sklearn.impute import SimpleImputer  
from sklearn.linear_model import LinearRegression, RidgeCV, ElasticNetCV, LogisticRegression  
from sklearn.ensemble import RandomForestRegressor, Random ForestClassifier  
from sklearn.compose import ColumnTransformer  
from sklearn.pipeline import make_pipeline, Pipeline  
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error, confusion_matrix, classification_report  
import warnings  
  
warnings.filterwarnings('ignore')
```

```
In [2]:  
df=pd.read_csv('/kaggle/input/medical-insurance-cost-prediction/medical_insurance.csv')  
df.head()
```

```
Out[2]:  


| person_id | age   | sex | region | urban_rural | income   | education | marital_status | employment_status | household_size | ...<br>liver_c |     |   |
|-----------|-------|-----|--------|-------------|----------|-----------|----------------|-------------------|----------------|----------------|-----|---|
| 0         | 75722 | 52  | Female | North       | Suburban | 22700.0   | Doctorate      | Married           | Retired        | 3              | ... | 0 |
| 1         | 80185 | 79  | Female | North       | Urban    | 12800.0   | No HS          | Employed          | 3              | ...            | 0   |   |
| 2         | 19865 | 68  | Male   | North       | Rural    | 40700.0   | HS             | Married           | Retired        | 5              | ... | 0 |
| 3         | 76700 | 15  | Male   | North       | Suburban | 15600.0   | Some           | Married           | Self-employed  | 5              | ... | 0 |
| 4         | 92992 | 53  | Male   | Central     | Suburban | 89600.0   | Doctorate      | Married           | Self-employed  | 2              | ... | 0 |


```

5 rows x 54 columns

```
In [3]:  
df.shape
```

```
Out[3]:  
(100000, 54)
```

Insurance XGBRegressor Model

Copied from Bhavya Motiyani (+420, 99)

Notebook Input Output Logs Comments (0)

```
In [1]:  
from sklearn.compose import ColumnTransformer  
from sklearn.pipeline import Pipeline  
from sklearn.impute import SimpleImputer  
from sklearn.preprocessing import OneHotEncoder, RobustScaler  
from xgboost import XGBRegressor  
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import StandardScaler, OneHotEncoder, OrdinalEncoder
```

over, and over,
and over again...

```
E0000 80:00:1762338430.9876468 13 cuda.blas.cc:1418] Unable to register cubLAS factory: Atte  
mpting to register factory for plugin cubLAS when one has already been registered
```

```
In [2]:  
def make_mi_scores(X, y, discrete_features):  
    mi_scores = mutual_info_regression(X, y, discrete_features=discrete_features)  
    mi_scores = pd.Series(mi_scores, name="MI Scores", index=X.columns)  
    mi_scores = mi_scores.sort_values(ascending=False)  
    return mi_scores
```

```
In [3]:  
df = pd.read_csv('/kaggle/input/medical-insurance-cost-prediction/medical_insurance.csv', index_col  
='person_id')
```

```
In [4]:  
target_columns = ['annual_premium', 'monthly_premium',  
'total_claims_paid', 'avg_claim_amount', 'annual_medical_cost']
```

Medical Insurance Cost Prediction

Notebook Input Output Logs Comments (1)

```
In [1]:  
# This Python 3 environment comes with many helpful analytics libraries installed  
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python  
# For example, here's several helpful packages to load  
  
import numpy as np # linear algebra  
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)  
  
# Input of files are available in the read-only "../input/" directory  
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the i  
nput directory  
  
import os  
for dirname, _, filenames in os.walk('../kaggle/input'):  
    for filename in filenames:  
        print(os.path.join(dirname, filename))  
  
# You can write up to 300B to the current directory (/kaggle/working/) that gets preserved as output  
when you create a version using "Save & Run All"  
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current  
version  
  
#kaggle/input/medical-insurance-cost-prediction/medical_insurance.csv
```

```
In [2]:  
import pandas as pd  
import numpy as np  
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import LabelEncoder, StandardScaler  
from sklearn.ensemble import RandomForestRegressor  
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error  
import matplotlib.pyplot as plt  
import seaborn as sns  
import warnings  
warnings.filterwarnings('ignore')
```

```
In [3]:  
df = pd.read_csv('/kaggle/input/medical-insurance-cost-prediction/medical_insurance.csv')  
pd.set_option('display.max_columns', None)  
df.head()
```

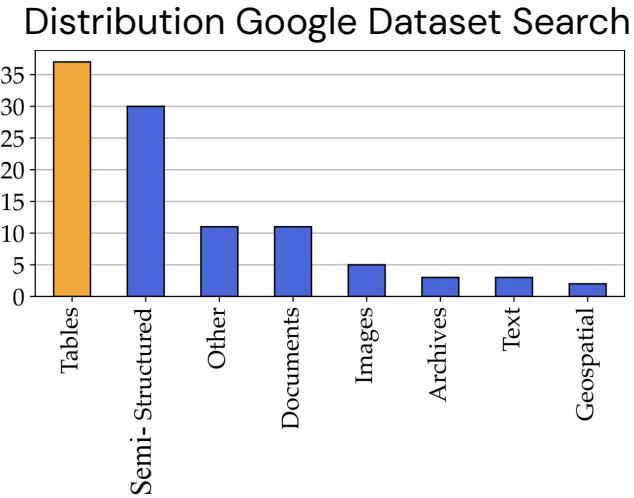
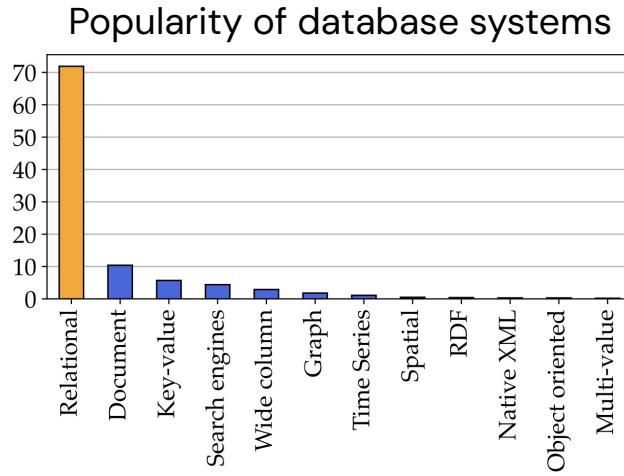
```
In [4]:  


| person_id | age   | sex | region | urban_rural | income   | education | marital_status | employment_status | household_size | dependents |   |
|-----------|-------|-----|--------|-------------|----------|-----------|----------------|-------------------|----------------|------------|---|
| 0         | 75722 | 52  | Female | North       | Suburban | 22700.0   | Doctorate      | Married           | Retired        | 3          | 1 |


```

There must be latent patterns to *learn* (data, code, etc).

Realization 3: tables prevail in the org data landscape



For a reason: **tables serve high-value decisions**

Surprisingly, “tables” ignored as modality in neural AI



You
what to include in speech with PhD graduation committee after defense ceremony?

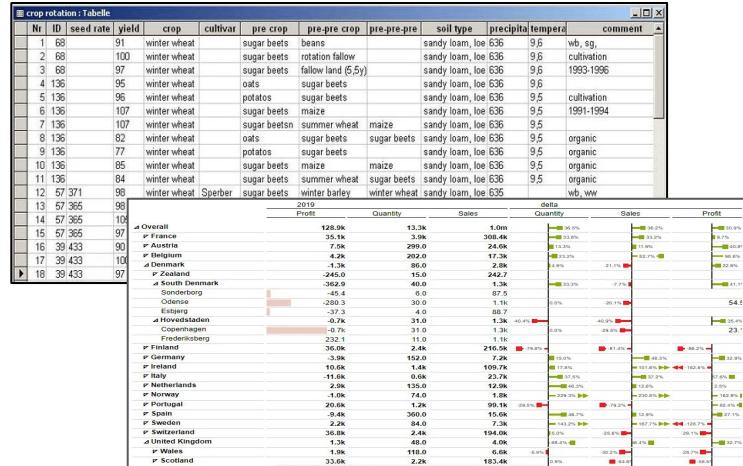
ChatGPT
After successfully defend
gratitude and appreciate
throughout your academic
journey.

1. Acknowledgment of the
committees individually
during the defense process.
2. Express Gratitude: Express
mentors, colleagues, friends
throughout your PhD journey.
3. Reflect on the challenges
you've faced and the
challenges you've overcame.
4. Highlight Contributions:
Impact in your field or
work and its relevance.

```
sentiments.ts  ↗ write_sq.go  ↗ parses_expenses.py  ↗ addresses.rb

1 //!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 async function isPositive(text: string): Promise<boolean> {
7   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
8     method: "POST",
9     body: `text=${text}`,
10    headers: {
11      "Content-Type": "application/x-www-form-urlencoded",
12    },
13  });
14
15  const json = await response.json();
16
17  return json.label === "pos";
18}
```

Copilot



Text, Images, Code...

Tables?!

Did a PhD on table semantics, but had a larger vision.

The Table Representation Learning workshop @ NeurIPS 2022 was born.

We received what we expected:

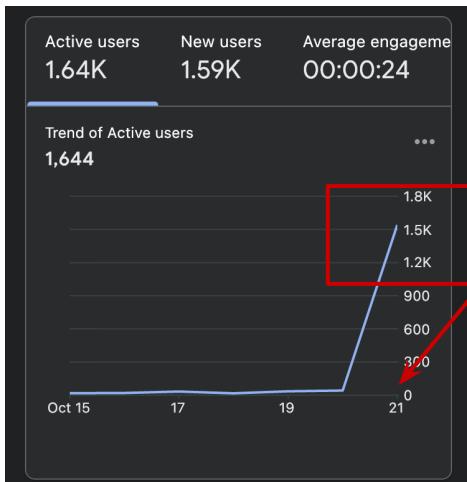
- Tabular QA / text-to-SQL,
- Synthetic data generation,
- Data preparation, etc.

And... neural models for predictive tabular ML.



Anecdotes on neural models for tabular predictive ML

- Received neural predictive ML papers, rejected from main ML confs(?)
- Loud “pro-XGBoost” camp vs. small “pro-neural models” camp.
- Betted on pre-trained neural models for tables in ‘18: let’s empower vision!
- And, 1 paper intro’d a pre-trained cross-table model: TabPFN. It went viral:



Frank's TabPFN tweet

Congrats team TabPFN,
TabICL, ConTexTab, etc for
pushing through and
heading leaderboards!

Tables weren't really cool in AI

But something happened in a tiny room in New Orleans at TRL @ NeurIPS 2022.

Great vibes, a wildly diverse community,
trying to connect the dots.

Tables were Back.



Fast-forward to 2025.

Tabular AI is the “new hot topic” (quote CV researcher)

Tabular AI is Europe-led  (but let's diversify).

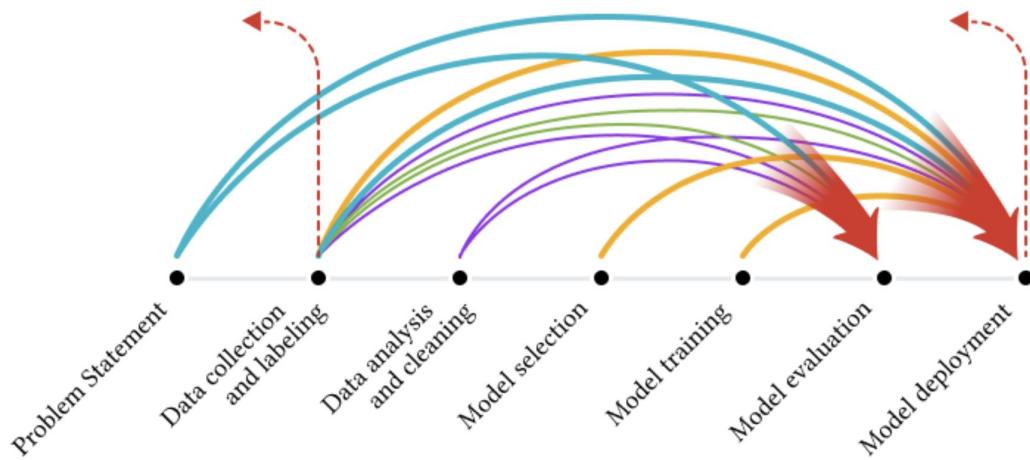
From an anonymous peer:

- > is there no tab workshop this year at main conf bc you all took to eurips?
- > either way wish I could be at the workshop :)

And we're only just beginning.

So, what are we asking from tabular data?

Tabular pipelines are multi-faceted



*Replace “model” with “tool/dashboard” and it’s BI.

LLM-enthusiasts make believe that DS is solved...

The screenshot shows a user interface for a 'Data Science Agent'. At the top, there's a purple header bar with a lab flask icon, the text 'Data Science Agent', and a button labeled 'Experiment'. Below the header, there are two menu items: 'README.md' and 'Playground', with 'Playground' being underlined. The main content area displays a file named '2018_Central_Park_Squirrel_Census_-_Squirrel_Data_20240501.csv'. A descriptive text below the file name reads: 'Analyze the proportion of adult and juvenile animals in the census data. Are there any spatial patterns in age distribution?'. A callout box highlights the file name again. At the bottom, there's a grey footer bar with the word 'Plan' and a dropdown arrow icon.

It's not

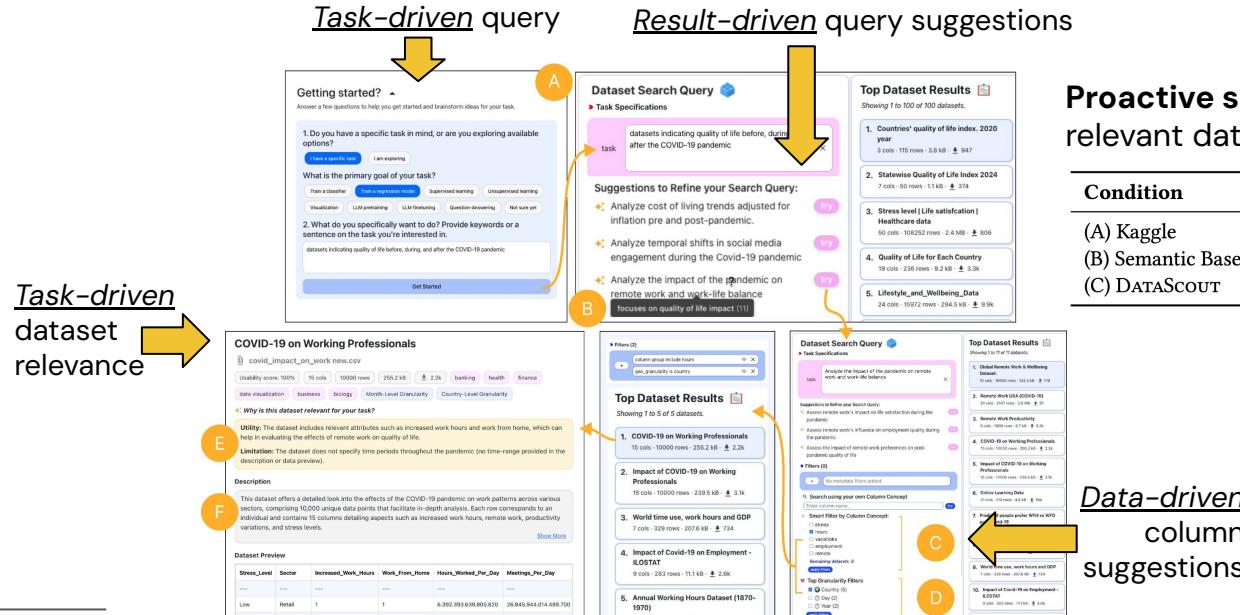
Model	SDE (69 tasks)		PM (28 tasks)		DS (14 tasks)	
	Success	Score	Success	Score	Success	Score
Closed model APIs						<i>API-based API</i>
Claude-3.5-Sonnet	30.43	38.02	35.71	51.31	14.29	21.70
Gemini-2.0-Flash	13.04	18.99	17.86	31.71	0.00	6.49
GPT-4o	13.04	19.18	17.86	32.27	0.00	4.70
Gemini-1.5-Pro	4.35	5.64	3.57	13.19	0.00	4.82
Amazon-Nova-Pro-v1	2.90	6.07	3.57	12.54	0.00	3.27
Open-weight models						<i>Open-weights</i>
Llama-3.1-405b	5.80	11.33	21.43	35.62	0.00	5.42
Llama-3.3-70b	11.59	16.49	7.14	19.83	0.00	4.70
Qwen-2.5-72b	7.25	11.99	10.71	22.90	0.00	5.42
Llama-3.1-70b	1.45	4.77	3.57	15.16	0.00	5.42
Qwen-2-72b	2.90	3.68	0.00	7.44	0.00	4.70

(TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks, Xu et al., 2024)

It all starts
with retrieving
the right data

What table to use for my task?

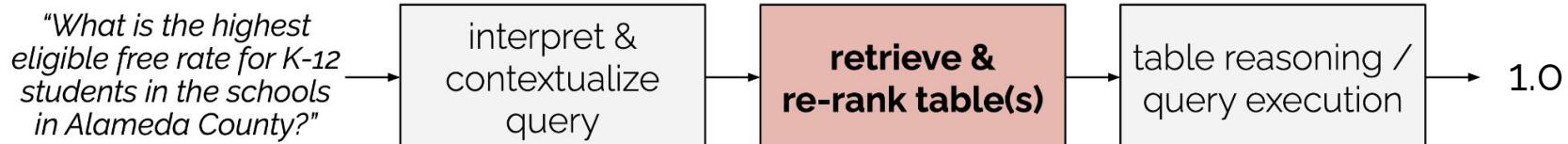
In 2025, we get “AGI” but *it still takes weeks to find the right dataset.*



Proactive search makes it easier to find more relevant datasets, with higher success rate.

Condition	Ease-of-use	Relevance	# Successes
(A) Kaggle	$\mu=3.08; \sigma=0.51$	$\mu=3.25; \sigma=1.05$	7 of 12
(B) Semantic Baseline	$\mu=3.75; \sigma=0.45$	$\mu=3.25; \sigma=0.86$	6 of 12
(C) DATA(SCOUT)	$\mu=4.75; \sigma=0.45$	$\mu=3.67; \sigma=0.78$	10 of 12

How to query tables end-to-end? ~RAG over tables



How to best retrieve tables in end-to-end tabular QA? Some findings:

- BM25 **less effective than for text**, requires highly descriptive metadata.
- Retrieval over generated summaries helps (but tricky for enterprise-y tables)
- **Row-level retrieval most effective for relational DBs**, but infeasible (many, large tables).

There's a very long road ahead. Imagine predictive questions over data lakes?

TARGET: benchmarking Table Retrieval for Generative Tasks
Ji, X., Parker, G., Parameswaran, A., Hulsebos, M., 2024

Metadata Matters in Dense Table Retrieval
Gomm, D., Hulsebos, M., 2025

UC Berkeley

CWI

Are We Asking the Right Questions?

In end-to-end tabular data analysis, we desire queries that perfectly express the ***insight need***: the **data**, the **operation**, and **output**.

Such queries are **platonic**, we won't see them.

But we need to understand **queries for e2d tabular data analysis** to build better systems and eval capabilities. (*join Daniel's talk for more*)

Tables are complex,
they require context

What is this table about?

Nr	ID	seed rate	yield	crop	cultivar	pre crop	pre-pre crop	pre-pre-pre	soil type	precipita	tempera	comment
1	68	91	winter wheat		sugar beets	beans			sandy loam, loe 636	9,6	wb, sg,	
2	68	100	winter wheat		sugar beets	rotation fallow			sandy loam, loe 636	9,6	cultivation	
3	68	97	winter wheat		sugar beets	fallow land (5,5y)			sandy loam, loe 636	9,6	1993-1996	
4	136	95	winter wheat		oats	sugar beets			sandy loam, loe 636	9,6		
5	136	96	winter wheat		potatos	sugar beets			sandy loam, loe 636	9,5	cultivation	
6	136	107	winter wheat		sugar beets	maize			sandy loam, loe 636	9,5	1991-1994	
7	136	107	winter wheat		sugar beets	summer wheat	maize		sandy loam, loe 636	9,5		
8	136	82	winter wheat		oats	sugar beets	sugar beets		sandy loam, loe 636	9,5	organic	
9	136	77	winter wheat		potatos	sugar beets			sandy loam, loe 636	9,5	organic	
10	136	85	winter wheat		sugar beets	maize			sandy loam, loe 636	9,5	organic	
11	136	84	winter wheat		sugar beets	summer wheat	sugar beets		sandy loam, loe 636	9,5	organic	
12	57 371	98	winter wheat	Sperber	sugar beets	winter barley	winter wheat		sandy loam, loe 635		wb, ww	
13	57 365	98	winter wheat	Sperber	potatos	sugar beets	summer barli	sandy loam, loe 635			cultivation, weed	
14	57 365	105	winter wheat	Sperber	sugar beets	maize	maize		sandy loam, loe 635		1987-1992	
15	57 365	97	winter wheat	Sperber	sugar beets	winter wheat	sugar beets		sandy loam, loe 635			
16	39 433	90	winter wheat	Okapi	summer barley				sandy loam, loe 690	8,5	oats, cultivation, weec	
17	39 433	100	winter wheat	Okapi	oats				clay, silt	690	8,5	1982-1986
18	39 433	97	winter wheat	Okapi	winter wheat				clay, silt	690	8,5	

Pre crop?

Pre-pre crop?

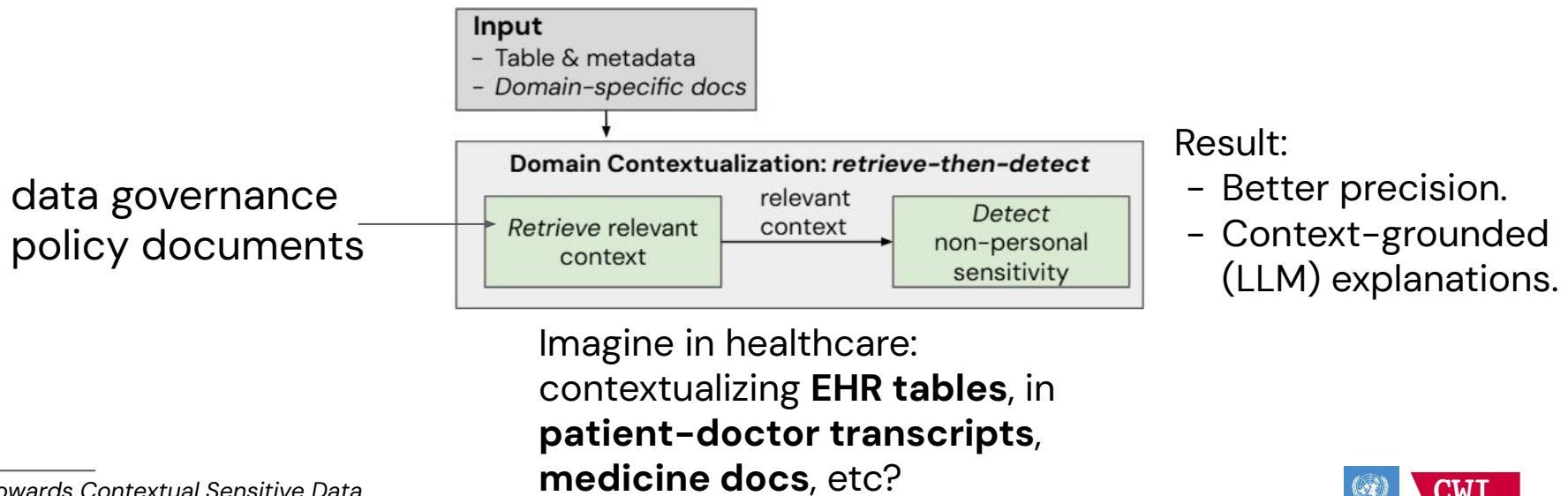
Pre-pre-pre?

A useful signal for data viz, data prep (integration, missing val mech.), ML, etc.

Sherlock: (tiny!) neural model surfaces **table semantics** by column embeddings.

Is this data sensitive?

Understudied. Beyond fixed types (pii) sensitivity depends on context.
Requires ***context beyond the table***.



But eventually,
we want that insight

What is the answer to my question?

Typical questions that drive decisions:

- Analytical questions → e.g. what has happened?
- Predictive queries → e.g. what might happen?

Some stats from Gael's keynote at TRL @NeurIPS '24 resonated.

Pypi #downloads last month (updated):

- **Scikit-learn: 189,875,197**
- **Pandas: 502,255,215**

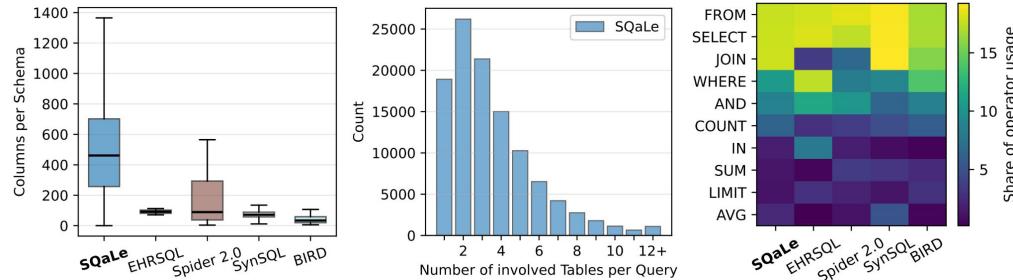
Analytical questions: we can use SQL

Why enter the *text-to-SQL* game?

- Saturated (but not solved), everyone just throwing LLMs at it.
- SQL made for humans not machines: it's too flexible, different abstraction?

Then, Cornelius (PhD) convinced me that we need **SQaLe to specialize.**

- Large text-to-SQL dataset
- 500K+ (question, sql, schema)
- Grounded in *real* schemas



SQaLe: A large text-to-SQL corpus grounded in real schemas,
Wolff, Gomm, and Hulsebos, 2025.

CWI

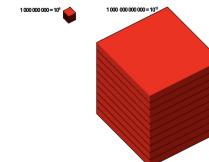
Specialization over general-purpose

Initial Qwen model *fine-tuned* for text-to-SQL on SQuaLe:



0.5 **billion** params vs 1.8 **trillion** params

*Find Cornelius in the poster session for more.



Executability is not accuracy, sure.

"For each customer in the gold or platinum loyalty tier who has placed at least two delivered orders in the last six months, return their full name, email, city and state, the total number of delivered orders, the total amount they paid based only on captured payments, their most frequently ordered product category in that period, and whether any of the products they ordered currently have low inventory in any warehouse (defined as quantity on hand minus quantity reserved less than 10), sorted by total captured payment amount from highest to lowest."

Realistic phrasing? Maybe not: inherited from source.
But look at that complex SQL! Promising start.

```
WITH gold_platinum_customers AS
  (SELECT c.id,
         c.first_name,
         c.last_name,
         c.email,
         c.city,
         c.state
    FROM customers c
   JOIN orders o ON c.id = o.customer_id
   WHERE c.loyalty_tier IN ('gold',
                           'platinum')
        AND o.status = 'delivered'
        AND o.delivered_at >= date('now', '-6 months')
  GROUP BY c.id
 HAVING COUNT(*) >= 2),
 order_details AS
 (SELECT gpc.id,
        gpc.first_name,
        gpc.last_name,
        gpc.email,
        gpc.city,
        gpc.state,
        o.order_number,
        SUM(p.amount) AS total_paid,
        p.status AS payment_status
   FROM gold_platinum_customers gpc
  JOIN orders o ON gpc.id = o.customer_id
  LEFT JOIN payments p ON o.id = p.order_id
  WHERE o.status = 'captured'
        AND o.delivered_at >= date('now', '-6 months')
        AND o.status IN ('delivered',
                          'completed')
  GROUP BY gpc.id,
          o.id),
 product_category AS
 (SELECT od.id,
        od.first_name,
        od.last_name,
        od.email,
        od.city,
        od.state,
        p.category,
        ROW_NUMBER() OVER (PARTITION BY od.id
                           ORDER BY COUNT(*) DESC) AS rn
   FROM order_details od
  JOIN order_items oi ON od.order_number = CAST(oi.order_id AS TEXT)
  JOIN products p ON oi.product_id = p.id
  GROUP BY od.id)
SELECT gpc.first_name,
       gpc.last_name,
       gpc.email,
       gpc.city,
       gpc.state,
       COUNT(od.order_number) AS total_orders,
       SUM(od.total_paid) AS total_captured_payment_amount
  FROM gold_platinum_customers gpc
 JOIN order_details od ON gpc.id = od.id
 WHERE od.payment_status = 'captured'
  LEFT JOIN product_category pc ON od.id = pc.id
  AND pc.rn = 1
 WHERE pc.category IS NOT NULL
 GROUP BY gpc.id
 ORDER BY total_captured_payment_amount DESC;
```

What should we ask from tabular data, next?

Generalization versus specialization?

Deriving insights from tabular data to inform decisions is an inherently multi-faceted problem, **perfect for smaller specialized models**, but still looking for the right level of abstraction, which may vary.

Eventually tabular systems are a mix and match:

- Databases and other tools for analytical queries, data prep, etc.
- Some ML models for statistical reasoning
- Some human, document, and LLM contextualization
- Some agentic capabilities
- Some fluid interfaces that connect things together

Still, many open questions

What is a table? What is a relation? How to deal with different representations?

What data do we encounter in practice?

What queries are asked? What can and should be asked? → check Daniel's talk

How to contextualize tabular data w/ domain knowledge from docs / human input?

How do we want to enable interaction with tabular data?

What is the ideal scope for foundation models? When to generalize vs specialize?

How do we go full-cycle from data collection to decision? What can we "agentize"?

Where does the community stand?

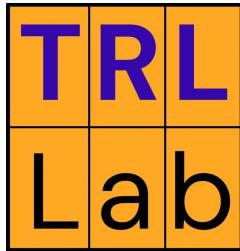
Excitement:

- **Tabular AI** has gained momentum – it's recognized widely!
- Academic / VC funding is flowing.
- There's much more to **explore and exploit**.

♥ **Tabular AI community!**

You, too? We're looking for committed members to foster the community.
Reach out if you want to contribute, stay tuned for more initiatives :).

Reach out



Madelon Hulsebos
Faculty



Xue (Effy) Li
Postdoctoral researcher



Daniel Gomm
PhD researcher



Cornelius Wolff
PhD researcher



Jan Henrik Bertrand
Research student (ELLIS MSc Honours Program)



Wojciech Kosiuk
Research student (ELLIS MSc Honours Program)

madelon@cwi.nl

<https://www.madelonhulsebos.com>

<https://trl-lab.github.io/>