UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Software Practicals
Winter Semester 2023/24

Data Science Group
Heidelberg University
October 18, 2023

Slides Online

The slides are available on our webpage
https://ds.ifi.uni-heidelberg.de/teaching/current/

Organization

Outline

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

- Overview of topics (today)
  - Send application for a topic until **Monday, October 23, 1pm**
  - Assignment of topics by October 25
- First milestone (before Christmas break)
  - Prototype / part of software
  - Summary of research (literature and related systems/tools)
  - Further milestones in agreement with supervisor
- End of practical (mid/end February)
  - Code has to be in local Gitlab of the database group
  - Presentation / demo of practical and software (10-12 minutes)
  - Report / documentation as Gitlab document (README.md)

Application

- Apply directly to supervisor via mail
  - Program of study, semester of study, matriculation number
  - List relevant course experience, including course grades
  - List other experience:
    - Side projects you are working on
    - “Anwendungsgebiet” / Application Field
    - Job and project experience
  - Send your tentative schedule and milestones for the practical
  - Group work is not possible!
- It is recommended to apply for multiple topics (e.g., “top-3 list”)

Application is binding!
Don’t apply if you don’t want to do the practical!

Deadlines

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

- In general: biweekly meetings with supervisor
- Presentation: end of February 2024
- Report & Gitlab upload: end of February 2024
- No extension possible

<u>Not finished = failed (grade 5,0)!</u>

Assessment

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

- Credit points (Leistungspunkte)
  - Beginners Practical (IAP, 2 CP + 4 FÜK) [Bachelor students]
    - workload: 180 h (~1 ½ days/week)
  - Advanced Practical (IFP, 8 CP)
    - workload: 240 h (~2 days/week)
- Grading based on
  - code (readability, structure, functionality; code in local Gitlab)
  - documentation (README.md, code comments, documentation in Gitlab)
  - commitment and self-reliance
  - cool ideas!!
- IMPORTANT
  - talk to / communicate with your advisor (at least biweekly meetings)

Supervisors

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

- Michael Gertz (MG)
  gertz@informatik.uni-heidelberg.de
- Satya Almasian (SA)
  almasian@informatik.uni-heidelberg.de
- Jayson Salazar (JS)
  salazar@informatik.uni-heidelberg.de
- John Ziegler (JZ)
  ziegler@informatik.uni-heidelberg.de
- Ashish Chouhan (AC)
  chouhan@informatik.uni-heidelberg.de
- Nicolas Reuter (NR)
  reuter@informatik.uni-heidelberg.de

Project Topics

AP = Advanced Topic

BP = Beginners Topic (for BSc students)

Overview of Topics

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

1. Quantity and Concept Extraction with ChatGPT, **BP** (Gertz/Almasian)
2. Form Extraction from OCRed Documents, **AP** (Gertz)
3. Graph Library Benchmark, **AP** (Ziegler)
4. Visual Benchmark of Dimensionality Reduction for Big, Sparse Graphs **AP** (Salazar)
5. Acquisition, Analysis and NER on OPS Codes **AP/BP** (Salazar)
6. Package Integration for NER in Patient Records **BP/AP** (Salazar)
7. Unlocking Legal Insights, **BP** (Gertz/Chouhan)
8. Concept Exploration UI, **AP** (Chouhan)
9. Temporal Evolution of Legal Documents, **BP/AP** (Chouhan)
10. Table Structure Recognition with Ruling Lines, **AP/BP** (Reuter)
11. Creating a PDF Table Annotation Tool, **AP** (Reuter)
12. Creating a Domain-Aware PDF Table Corpus, **AP** (Reuter)

BP: Quantity and Concept Extraction with ChatGPT (MG/SA)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- Sentences from news articles tagged with quantity information

Tasks:

- Use ChatGPT to extract quantity information for a sentence, containing <value, unit, change, concept>
- “Apple hires 200 people”
  → value=200, unit=people, change=equal, concept=Apple.

Subtasks:

- Examine ChatGPT’s ability to standardize values, normalize unit, and find relevant nouns (concepts)
- Build a pipeline that gets sentences and outputs quantity information

Languages / Tools:

- Python

AP: Form Extraction from OCRed Documents (MG)

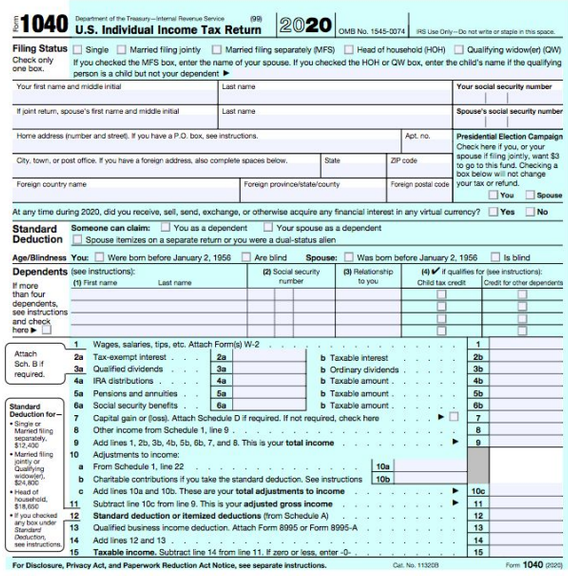
UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- Collection OCRed forms, e.g., receipts, bills, table like structure
- Popular libraries and tools to perform AI-based layout analysis

Tasks:

- Evaluate quality of different tools, frameworks and libraries
- Build framework that allows (visual) comparison of different tools



Form **1040** Department of the Treasury—Internal Revenue Service (99) **U.S. Individual Income Tax Return** **2020** OMB No. 1545-0074 IRS Use Only—Do not write or staple in this space.

Filing Status Check only one box. ☐ Single ☐ Married filing jointly ☐ Married filing separately (MFS) ☐ Head of household (HOH) ☐ Qualifying widow(er) (QW)
If you checked the MFS box, enter the name of your spouse. If you checked the HOH or QW box, enter the child's name if the qualifying person is a child but not your dependent ▶

Your first name and middle initial | Last name | **Your social security number**
If joint return, spouse's first name and middle initial | Last name | **Spouse's social security number**
Home address (number and street). If you have a P.O. box, see instructions. | Apt. no. | **Presidential Election Campaign** Check here if you, or your spouse if filing jointly, want $3 to go to this fund. Checking a box below will not change your tax or refund. ☐ You ☐ Spouse
City, town, or post office. If you have a foreign address, also complete spaces below. | State | ZIP code
Foreign country name | Foreign province/state/county | Foreign postal code

At any time during 2020, did you receive, sell, send, exchange, or otherwise acquire any financial interest in any virtual currency? ☐ Yes ☐ No

Standard Deduction **Someone can claim:** ☐ You as a dependent ☐ Your spouse as a dependent
☐ Spouse itemizes on a separate return or you were a dual-status alien

Age/Blindness **You:** ☐ Were born before January 2, 1956 ☐ Are blind **Spouse:** ☐ Was born before January 2, 1956 ☐ Is blind

Dependents (see instructions): | **(1)** First name Last name | **(2)** Social security number | **(3)** Relationship to you | **(4)** ✔ if qualifies for (see instructions): Child tax credit | Credit for other dependents
If more than four dependents, see instructions and check here ▶ ☐

Attach Sch. B if required.	1	Wages, salaries, tips, etc. Attach Form(s) W-2		1
	2a	Tax-exempt interest 2a	b Taxable interest	2b
	3a	Qualified dividends 3a	b Ordinary dividends	3b
	4a	IRA distributions 4a	b Taxable amount	4b
	5a	Pensions and annuities 5a	b Taxable amount	5b
	6a	Social security benefits 6a	b Taxable amount	6b
	7	Capital gain or (loss). Attach Schedule D if required. If not required, check here ▶ ☐		7
	8	Other income from Schedule 1, line 9		8
	9	Add lines 1, 2b, 3b, 4b, 5b, 6b, 7, and 8. This is your **total income** ▶		9
Standard Deduction for— • Single or Married filing separately, $12,400 • Married filing jointly or Qualifying widow(er), $24,800 • Head of household, $18,650 • If you checked any box under Standard Deduction, see instructions.	10	Adjustments to income:		
	a	From Schedule 1, line 22 10a		
	b	Charitable contributions if you take the standard deduction. See instructions 10b		
	c	Add lines 10a and 10b. These are your **total adjustments to income** ▶		10c
	11	Subtract line 10c from line 9. This is your **adjusted gross income** ▶		11
	12	**Standard deduction or itemized deductions** (from Schedule A)		12
	13	Qualified business income deduction. Attach Form 8995 or Form 8995-A		13
	14	Add lines 12 and 13		14
	15	**Taxable income.** Subtract line 14 from line 11. If zero or less, enter -0-		15

For Disclosure, Privacy Act, and Paperwork Reduction Act Notice, see separate instructions. Cat. No. 11320B Form **1040** (2020)

Subtasks:

- Familiarize yourself with document layout analysis, LayoutLM

Languages / Tools:

- Python, Hugging Face

AP: Graph Library Benchmark (JZ)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Given:

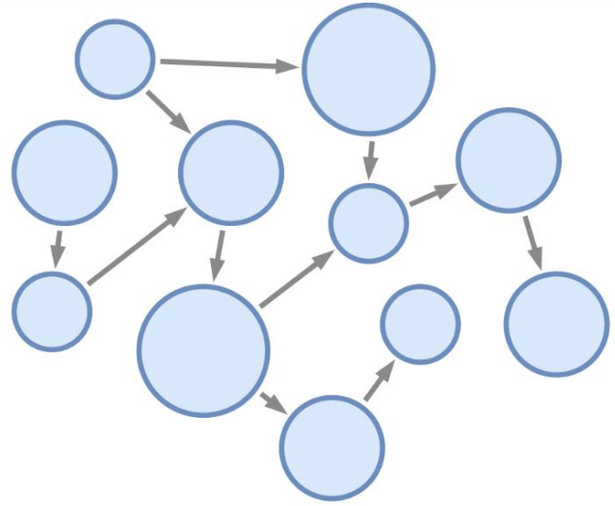
- Various open-source graph libraries
- Past benchmarks exist, e.g., here

Task:

- Conduct performance benchmark of different graph libraries

Subtasks:

- Prepare benchmark dataset(s)
- Create “lab” environment
- Special focus on *network dynamics*



Languages / Tools:

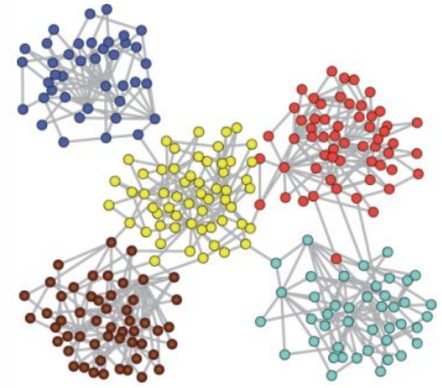
- Python, graph-tool, igraph, NetworkX, …

AP: Visual DimRed Benchmark for Graphs AP (JS)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

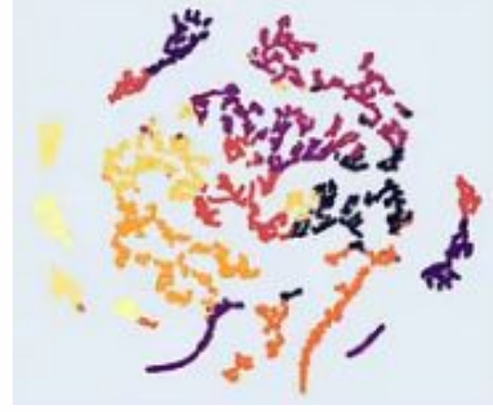
Given:

- Graphs can be built off almost any sort of text domain data, but they also grow ***quickly*** and are ***sparse***.
- Representing, grouping and visualizing their labels as well as properties is of crucial importance in our group



Tasks:

- Generate a synthetic graph dataset based on given, well-defined properties from real-world data examples.
- Build an application that allows a user to poll graph data, brush(filter) it and visualize it based on two chosen algorithms (e.g. PCA, t-SNE and UMAP).



Languages / Tools:

- Python, Javascript (Cytoscape.js+Svelte), Apache AGE (or Neo4J)

BP(AP): Acquisition, Analysis and NER on OPS Codes (JS)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

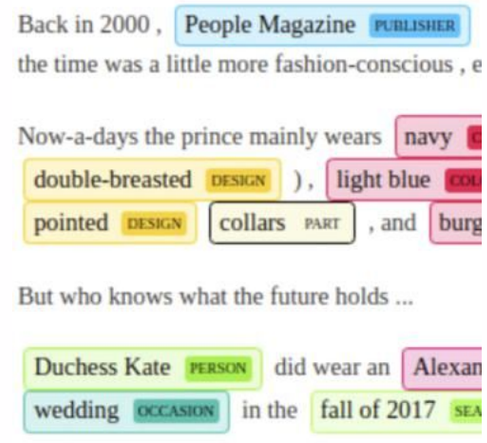
Given:

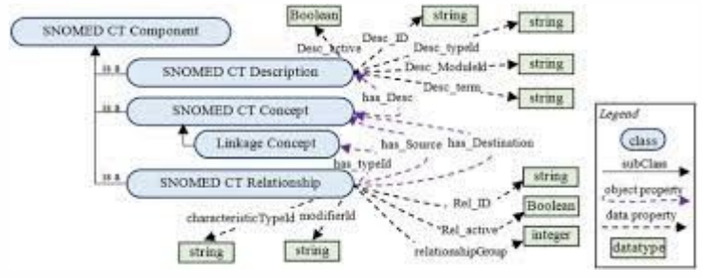
- Python library for Medical Thesaurus Correlation (MedKEET)
- Access to OPS-Database and related medical thesauri

Federal Institute for Drugs and Medical Devices

Tasks:

- Acquire and analyze the structure and contents of the OPS dataset.
- Extend MedKEET to process and annotate (naively) OPS entries





Languages / Tools:

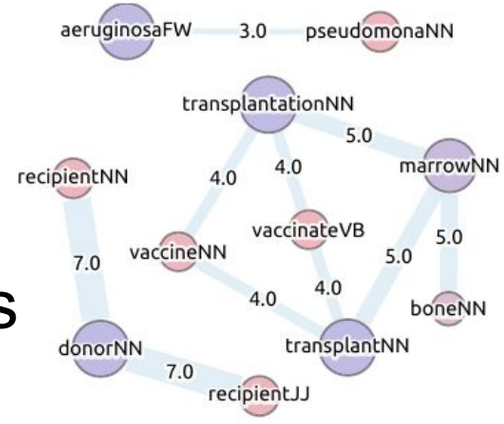
- Python, Pandas, Neo4J

BP(AP): Package Integration for NER in Patient Records (JS)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- PyCoNet, a prototype package aimed towards the extraction of concept relationships from raw (medical) text.
- FHIRPACK, an open source FHIR data Python processing toolkit
- Both developed jointly in the DBS and the UK-Essen



Tasks:

- Integrate and extend both packages so is-relationships (NER) present in raw, semi-structured patient records can be extracted and stored seamlessly

Languages / Tools:

- Python, PostgreSQL

BP: Unlocking Legal Insights (MG/AC)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Given:

- People post information about legal domain on LinkedIn (see, e.g., Martin Ebers)

Tasks:

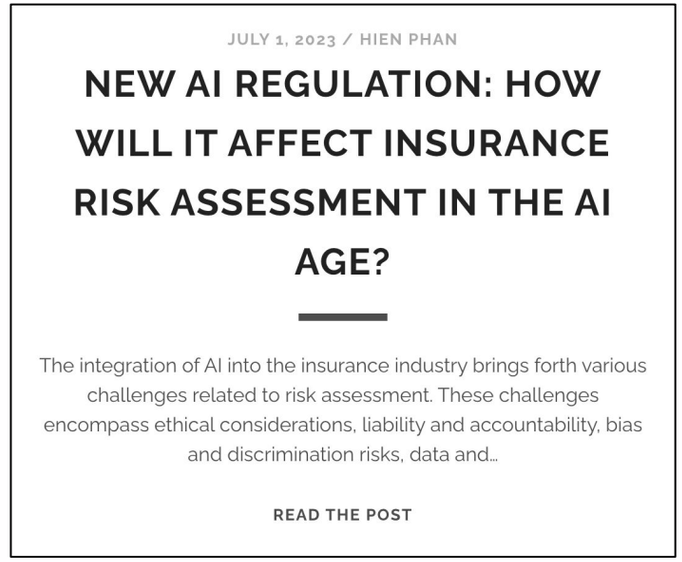
- Extract articles, documents, and posts related to legal domain and store it in OpenSearch

Subtasks:

- Realize backend to store extracted information using OpenSearch
- Analyze dataset and compute basic statistics

Languages / Tools:

- Python, OpenSearch

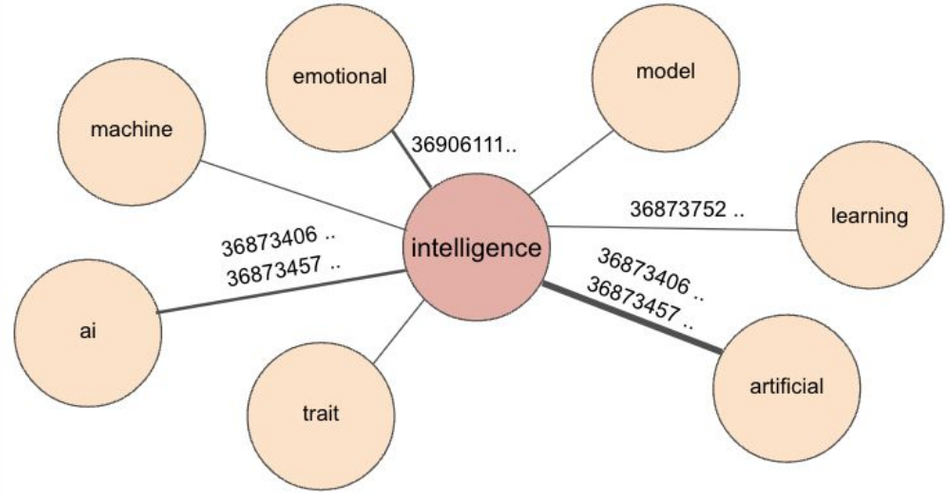


Link

AP: Concept Exploration UI (AC)

Given:

- API provides weighted word co-occurrence network for concept exploration in Pubmed abstracts



Tasks:

- Implementation of UI to explore concept

Subtasks:

- Handle API access
- Visualization of networks

Languages / Tools:

- Python, Sveltekit, eCharts, Cytoscape.js

BP/AP: Temporal Evolution of Legal Documents (AC)

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Given:

- Consolidated text from EUR-Lex website (HTML)

Tasks:

- Fetch consolidated text information
- **AP:** develop frontend to search and explore evolution of regulations

►B

AGREEMENT

on the withdrawal of the United Kingdom of Great Britain and Northern Ireland from the European Union and the European Atomic Energy Community

(OJ L 029 31.1.2020, p. 7)

Amended by:

		Official Journal		
		No	page	date
►M1	DECISION No 1/2020 OF THE JOINT COMMITTEE ESTABLISHED BY THE AGREEMENT ON THE WITHDRAWAL OF THE UNITED KINGDOM OF GREAT BRITAIN AND NORTHERN IRELAND FROM THE EUROPEAN UNION AND THE EUROPEAN ATOMIC ENERGY COMMUNITY of 12 June 2020	L 225	53	14.7.2020
►M2	DECISION No 3/2020 OF THE JOINT COMMITTEE ESTABLISHED BY THE AGREEMENT ON THE WITHDRAWAL OF THE UNITED KINGDOM OF GREAT BRITAIN AND NORTHERN IRELAND FROM THE EUROPEAN UNION AND THE EUROPEAN ATOMIC ENERGY COMMUNITY of 17 December 2020	L 443	3	30.12.2020

EUR-Lex
Access to European Union law

Subtasks:

- Design and implement document store based on OpenSearch and PostgreSQL

Languages / Tools:

- Python; SPARQL; OpenSearch; PostgreSQL

BP/AP: Table Structure Recognition with Ruling Lines (NR)

Given:

- FinTabNet dataset of annotated PDFs with Financial Report Tables

	2002		2001	
	High	Low	High	Low
1st Quarter	$ 59.46	$ 48.32	$ 64.00	$ 45.75
2nd Quarter	59.65	49.54	59.55	50.26
3rd Quarter	55.34	42.30	60.04	50.23
4th Quarter	44.37	34.14	53.61	42.30

Task:

- Implement a method that uses ruling lines to extract the structure of complex tables (mapping cells to the correct headers)

U.S. Pension Benefits			Non-U.S. Pension Benefits		
Fiscal Year			Fiscal Year		
2015	2014	2013	2015	2014	2013
$ 104	$ 107	$ 104	$ 60	$ 54	$ 43
105	97	94	33	29	27
(160)	(141)	(128)	(41)	(35)	(33)
—	1	(1)	—	1	1
65	85	71	12	11	8
$ 114	$ 149	$ 140	$ 64	$ 60	$ 46

Subtasks:

- Identify tables with rulings in the dataset and extract the ruling lines
- Recognize if a ruling belongs to a heading and assign it to it
- Use the position and size of the lines to extract the table structure

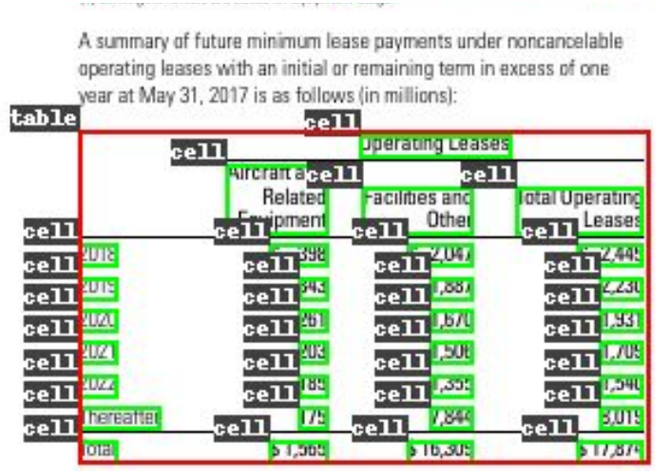
Languages / Tools:

- Python, pdfplumber, (Docker)

AP: Creating a PDF Table Annotation Tool (NR)

Given:

- Reference annotations: FinTabNet dataset
- Possible starting point: table extraction tool from previous practical



Task:

- Create a tool with a simple UI for annotating (complex) tables

Subtasks:

- Extract the tables from PDFs (camelot and tabula-py can be used)
- Create a UI with which the extracted tables can be edited to get a proper annotation
- Convert the annotations to the target format (see FinTabNet)

Languages / Tools:

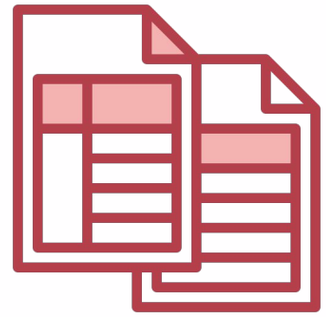
- Python, JavaScript (e.g. node.js, Svelte), Docker

AP: Creating a Domain-Aware PDF Table Corpus (NR)

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Given:

- A very large dataset of PDFs, e.g. CC-MAIN-2021-31-PDF-UNTRUNCATED



Tasks:

- Create a subset of the dataset containing only PDFs with tables
- Classify the PDFs by domain/content, language (and table type)

Subtasks:

- Implement a pipeline for table detection, metadata extraction and PDF classification
- Basic data cleanup: find duplicates and spam/useless documents using the extracted data
- Optional: Implement a method to crawl new PDFs with tables

Languages / Tools:

- Python, PostgreSQL, (Docker, OpenSearch)

Slides Online

The slides are available on our webpage
https://ds.ifi.uni-heidelberg.de/teaching/current/