



Building an AI-assisted dashboard  
for your daily newsletter

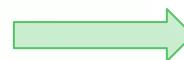
# Who we are



Moritz



Johannes



**first time to host a workshop  
- please feedback us! :)**



# Today we want to build this:

Settings

Topic selection

research and science

Newsletter selection

TLDR

Fetch recommendations

show session state

## Recommendations - 1/10

### Real World Recommendation System - Part 1

Training a collaborative filtering based recommendation system on a toy dataset is a sophomore year project in colleges these days. But where the rubber meets the road is building such a system at scale, deploying in production, and serving live requests within a few hundred milliseconds while the user is waiting for the page to load. To build a system like this, engineers have to make decisions spanning multiple moving layers like...

[Next recommendation](#)

## Similar Stories - 1/10

### Ultra-light liquid hydrogen tanks promise to make jet fuel obsolete (3 minute read)

Gloyer-Taylor Laboratories (GTL) has developed ultra-lightweight cryogenic tanks that have a 75% mass reduction compared with other aerospace cryotanks. A 12 kg tank from GTL is able to hold over 150 kg of hydrogen. The weight reduction means that hydrogen-fueled aircraft may be able to fly at least four times as far as comparable aircraft running on jet fuel while completely eliminating carbon emissions. It could also mean increased cargo or passenger capacity.

[Next similar](#)

## Story Browser

### A new and outlandish delivery drone concept can carry 100 pounds up to 80 miles (2 minute read)

Austria-based Cyclotech and Japanese delivery firm Yamato have partnered to create a concept delivery drone. The CCY-01 uses a thrust vectoring propulsion system developed by Cyclotech that enables it to land in confined spaces and handle challenging wind conditions. The drone is able to produce horizontal sideways thrust without tilting. The CCY-01 has a payload capacity of 99 lbs and it can fly up to 25 miles at speeds of around 80 mph. A video of the CCY-01 performing its first free flight is available in the article.

[View full newsletter](#)

[Get similar stories](#)

## Full Newsletter HTML

If you don't want to receive future editions of TLDR, please [click here to unsubscribe](#).

**TLDR**

Daily Update 2022-04-15

**SSH To Anywhere With Tailscale (Sponsor)**

No additional hardware to manage. No complicated firewalls. Tailscale keeps it simple & secure. [Learn more](#).

If you would like to sponsor TLDR, please let me know by replying to this email or check out our [sponsorship page](#)!

# Agenda for today

## **pairwise theory and practical content - all in Python!**

- |   |        |
|---|--------|
| 1. collecting data with oauth2 in GMail         | 15 min |
| 2. processing data (e.g. HTML and text parsing) | 30 min |
| 3. building training data for topic modelling   | 15 min |
| 4. applying transfer learning with transformers | 30 min |
| 5. <b>break</b>                                 | 10 min |
| 6. building the streamlit UI                    | 40 min |
| 7. building a minimal backend with FastAPI      | 40 min |

This is guided by a GitHub repository:

<https://github.com/code-kern-ai/datalift-summit>

# Expectation Management

**We (unfortunately) just have three hours.**

We are building something that is not perfect, but really fun.

We are going to cut corners, but will cover the basics.

Goals:

- Show you real-life data collection
- Code a little real-life example of an NLP application
- Introduce you to fastAPI and streamlit
- Give you many possibilities for extension

# Installing dependencies

**Please now go to the repository, and follow the installation instructions!**

We'll have a short theoretical input for oauth2 to bridge the installation time

# Data collection



Webpages, HTML



Social Media



Mail provider



Chats

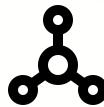
# Data collection



Webpages, HTML



selenium,  
requests +  
beautifulsoup



Social Media



APIs



Mail provider



oauth2



Chats



Bots, APIs

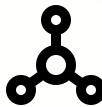
# Data collection



Webpages, HTML



selenium,  
requests +  
beautifulsoup



Social

always check the statement about  
data collection of the respective  
source (company, institution, ...)

APIs



Mail

!!!

oauth2



Chats



Bots, APIs

# Data collection



Mail provider



oauth2

# Diving into oauth2

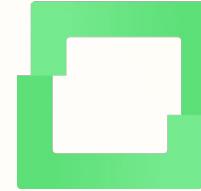
| Authentication                   | Authorization   |
|----------------------------------|---|
| I prove that I am who I say I am | I prove that someone else told me that I could access something |
| Can literally do anything        | Has scopes (e.g. reading mails, but not deleting them)          |
| Username and Password            | oauth2  |

e.g. Google doesn't allow 3rd party apps to authenticate as users any longer.  
They can only access via authorization.

# Diving into oauth2

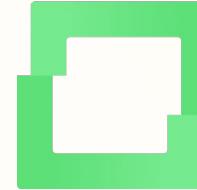
| Authentication  | Authorization   |
|---|---|
| <p><b>Log Into My Account</b></p> <div data-bbox="345 507 816 556"><a href="#"> Sign in with Google</a></div> <p>or</p> <div data-bbox="336 594 816 820"><div data-bbox="336 594 816 642"><input data-bbox="336 594 816 642" type="text"/><span data-bbox="777 610 796 631">i</span></div><div data-bbox="336 664 816 712"><input data-bbox="336 664 816 712" type="password"/><span data-bbox="777 680 796 702">i</span></div><div data-bbox="336 729 816 772"><span data-bbox="547 739 604 761">Log In</span></div><div data-bbox="336 783 489 804"><a href="#">Don't have an account?</a></div><div data-bbox="681 783 806 804"><a href="#">Forgot password?</a></div></div> <p><a href="#">Log in with SSO</a></p> | <p><b>Log Into My Account</b></p> <div data-bbox="1094 507 1584 556"><a href="#"> Sign in with Google</a></div> <p>or</p> <div data-bbox="1104 594 1584 772"><div data-bbox="1104 594 1584 642"><input data-bbox="1104 594 1584 642" type="text"/><span data-bbox="1545 610 1564 631">i</span></div><div data-bbox="1104 664 1584 712"><input data-bbox="1104 664 1584 712" type="password"/><span data-bbox="1545 680 1564 702">i</span></div><div data-bbox="1104 729 1584 772"><span data-bbox="1315 739 1372 761">Log In</span></div><div data-bbox="1104 783 1257 804"><a href="#">Don't have an account?</a></div><div data-bbox="1449 783 1574 804"><a href="#">Forgot password?</a></div></div> <p><a href="#">Log in with SSO</a></p> |

# Diving into oauth2



# Diving into oauth2

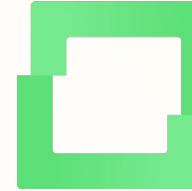
1. access my mails



# Diving into oauth2



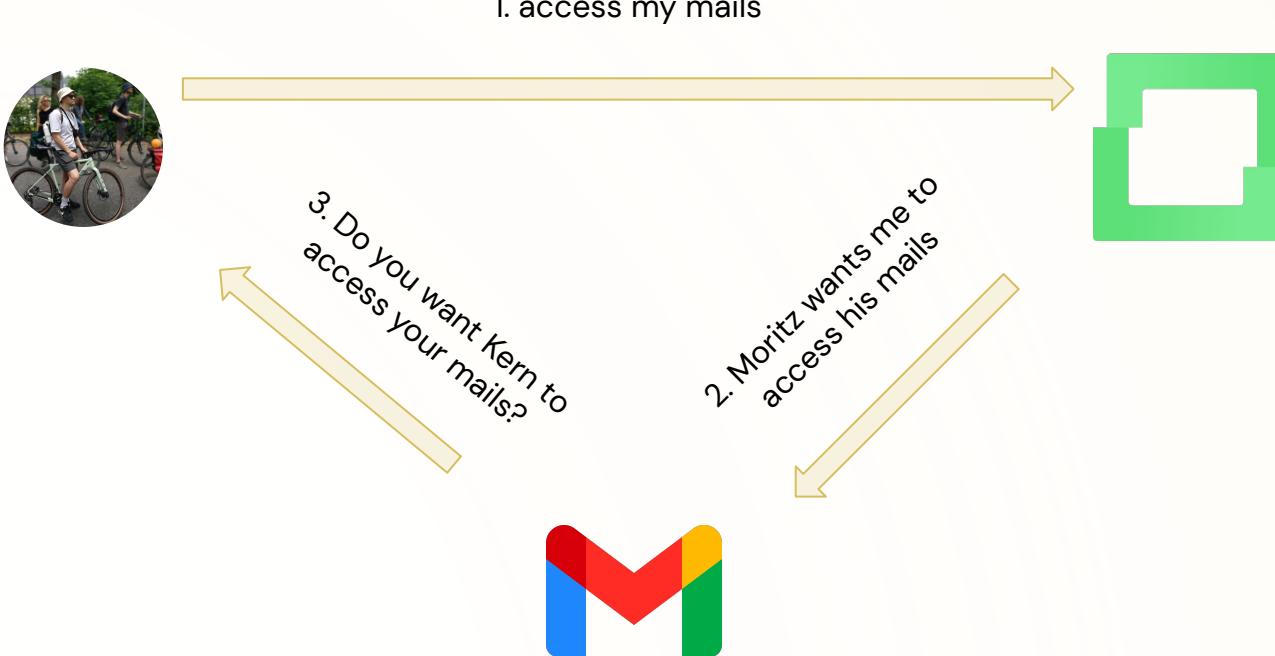
1. access my mails



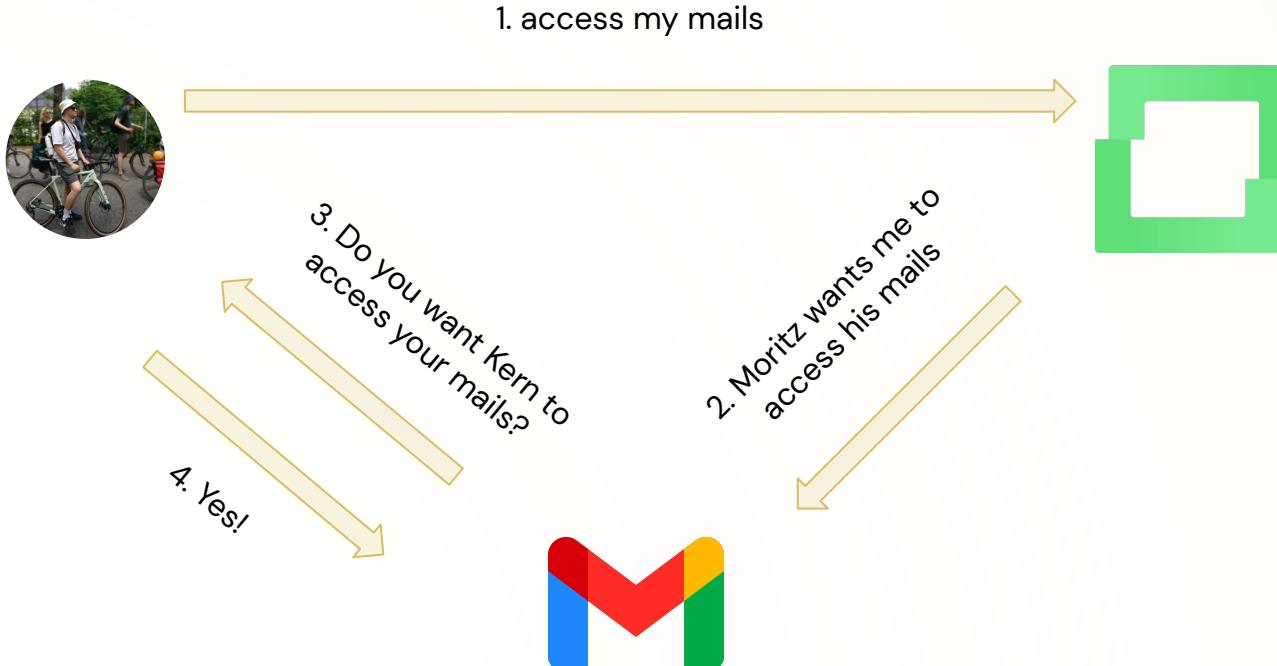
2. Moritz wants me to  
access his mails



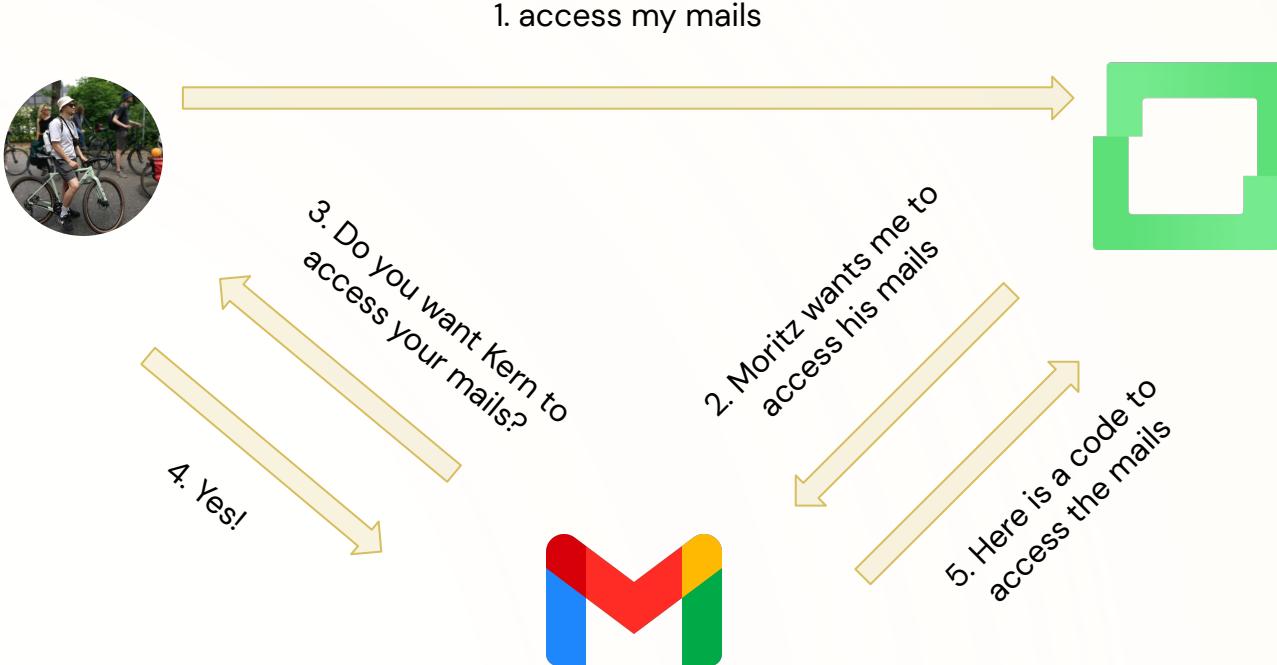
# Diving into oauth2



# Diving into oauth2



# Diving into oauth2



# Practical part 1

# Processing data

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html xmlns="http://www.w3.org/1999/xhtml">
<head> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> <title>Data Science Weekly - Issue 440</title></head> <body style="margin: 0; padding: 0; background-color: #;"> <center> <table align="center" border="0" cellpadding="0" cellspacing="0" height="100%" width="100%" id="bodyTable" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; margin: 0; padding: 0; background-color: #;border-collapse: collapse !important; height: 100% !important; width: 100% !important;"> <tr> <td align="center" height="100%" valign="top" width="100%" id="bodyCell" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; margin: 0; padding: 20px; border-collapse: collapse !important; height: 100% !important; width: 100% !important;"> <table border="0" cellpadding="0" cellspacing="0" id="templateContainer" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; width: 600px; border-collapse: collapse !important;"> <tr> <td align="center" valign="top" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; border-collapse: collapse !important;"> <!-- BEGIN PREHEADER --> <table border="0" cellpadding="0" cellspacing="0" width="100%" id="templatePreheader" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; border-top: 0; border-collapse: collapse !important;"> <tr> <td class="preheaderContent" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; color: #34495e; font-family: Tahoma; font-size: 9px; line-height: 125%; padding-top: 10px; padding-bottom: 10px; text-align: left; border-collapse: collapse !important;"> <span style="font-family: tahoma, verdana, segoe, sans-serif;"><span style="font-size: 11px;"><font color="#34495e">Curated news, articles and jobs related to Data Science.&nbsp;<br> <strong>Keep up with all the latest developments</strong></font></span></span></td> <!-- --> <td width="220" class="preheaderContent" style="padding-left: 20px; mso-table-lspace: 0pt; color: #34495e; font-family: Tahoma; font-size: 9px; line-height: 125%; padding-top: 10px; padding-bottom: 10px; text-align: left; border-collapse: collapse !important;"> <span style="font-size: 11px;"><span style="font-family: tahoma, verdana, segoe, sans-serif;">Email not displaying correctly?<br> <a href="https://datascienceweekly.us3.list-manage.com/track/click?u=71a2b2a38789d4d25b738462f&id=3d2f3344bc&e=7dcf46431f" target="_blank" style="color: #34495e; font-weight: bold; text-decoration: none;">View it in your browser</a></span></td> <!-- --> </tr> </table> <!-- END PREHEADER --> </td> </tr> <tr> <td align="center" valign="top" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; border-collapse: collapse !important;"> <!-- BEGIN HEADER --> <table border="0" cellpadding="0" cellspacing="0" width="100%" id="templateHeader" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; border-top: 10px solid #000000; border-bottom: 5px solid #000000; border-collapse: collapse !important;"> <tr> <td align="top" class="headerContent" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; color: #000000; font-family: Helvetica; font-size: 20px; font-weight: bold; line-height: 100%; padding-top: 20px; padding-bottom: 20px; text-align: center; border-collapse: collapse !important;"> <h1 style="display: block; font-family: Georgia; font-size: 26px; font-style: normal; font-weight: bold; line-height: 100%; letter-spacing: normal; margin-top: 0; margin-right: 0; margin-bottom: 10px; margin-left: 0; text-align: center; color: #34495e !important;"></h1> <h2 style="display: block; font-family: Tahoma; font-size: 20px; font-style: normal; font-weight: bold; line-height: 100%; letter-spacing: normal; margin-top: 0; margin-right: 0; margin-bottom: 10px; margin-left: 0; text-align: center; color: #34495e !important;">Issue #440<br> April 28 2022</h2> </td> </tr> </table> <!-- END HEADER --> </td> </tr> <tr> <td align="center" valign="top" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; border-collapse: collapse !important;"> <!-- BEGIN BODY --> <table border="0" cellpadding="0" cellspacing="0" width="100%" id="templateBody" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; border-top: 0; border-bottom: 0; border-collapse: collapse !important;"> <tr> <td align="top" class="bodyContent" style="mso-table-lspace: 0pt;mso-table-rspace: 0pt; color: #000000; font-family: Helvetica; font-size: 16px; line-height: 150%; padding-top: 40px; padding-bottom: 40px; text-align: left; border-collapse: collapse !important;"> <h2 style="line-height: 20px; display: block; font-family: Tahoma; font-size: 20px; font-weight: bold; letter-spacing: normal; margin-top: 0; margin-right: 0; margin-bottom: 10px; margin-left: 0; text-align: center; color: #34495e !important;"><font face="tahoma, verdana, segoe, sans-serif">Editor Picks</font></h2> &nbsp; <ul> <li> <font face="tahoma, verdana, segoe, sans-serif"><a href="https://datascienceweekly.us3.list-manage.com/track/click?u=71a2b2a38789d4d25b738462f&id=0b657fd1e9&e=7dcf46431f" target="_blank" style="color: #FF0000; font-weight: normal; text-decoration: none;">Beyond interpretability: developing a language to shape our relationships with AI</a><br> AI will continue becoming more complex, bigger, and smarter. Wouldn't it be nice if we could ask it questions to learn how and why it makes its predictions? Unfortunately, we don't have a good language to communicate with AI yet...[This post is based on the 2022 ICLR Keynote]...</font></li> </ul> <ul> <li> <font face="tahoma, verdana, segoe, sans-serif"><a href="https://datascienceweekly.us3.list-manage.com/track/click?u=71a2b2a38789d4d25b738462f&id=8330e19e35&e=7dcf46431f" target="_blank" style="color: #FF0000; font-weight: normal; text-decoration: none;">DALL&middot;E 2 and The Origin of Vibe Shifts</a><br> The point of this essay isn't about predicting next year's design trends. To me the more interesting thing is to understand the ecological process that generates those trends, seeing the true signaling function of visual design, and learning why some corporate status signaling is so effective and why some isn't. Projecting further out, it's about trying to picture a world where it's cheap and easy for anyone to generate just about any kind of image they want... To answer these questions, we're going to tap the most well-developed pool of knowledge on the use of costly signals and their evolution over time: biology....</font></li> </ul> <ul> <li> <font face="tahoma, verdana, segoe, sans-serif"><a href="https://datascienceweekly.us3.list-manage.com/track/click?u=71a2b2a38789d4d25b738462f&id=a74165086a&e=7dcf46431f" target="_blank" style="color: #FF0000; font-weight: normal; text-decoration: none;">Learning with not Enough Data Part 3: Data Generation</a><br> Part 3 of &ldquo;what if you don't have enough training data&rdo; series - touch base on creating more synthetic data by data augmentation or model generation, as well as some ideas on how to work with noisy labels (given synthetic data might not be fully correct)....</font></li> </ul> &nbsp; <hr> &nbsp; <h2 style="line-height: 20px; display: block; font-family: Tahoma; font-size: 20px; font-style: normal; text-decoration: none;">
```

# Processing data

| newsletter        | date                      | headline  | ... |
|-------------------|---------------------------|---|-----|
| datascienceweekly | 2022-04-28 23:21:02+00:00 | Beyond interpretability: developing a language to shape our relationships with AI |     |
| datascienceweekly | 2022-04-28 23:21:02+00:00 | DALL-E 2 and The Origin of Vibe Shifts  |     |
| datascienceweekly | 2022-04-28 23:21:02+00:00 | It's Our Moral Obligation to Make Data More Accessible                            |     |
| datascienceweekly | 2022-04-28 23:21:02+00:00 | Scientists Publish Breakthrough Study in Oreo-Splitting Research                  |     |
| Box of Amazing    | 2022-05-08 06:01:12+00:00 | Refind – Get smarter every day  |     |
| Box of Amazing    | 2022-05-08 06:01:12+00:00 | Inside Elon Musk's Big Plans for Twitter  |     |
| Box of Amazing    | 2022-05-08 06:01:12+00:00 | Influencer culture is everywhere — even in academia                               |     |

## **the “typical” steps to preprocess raw data**

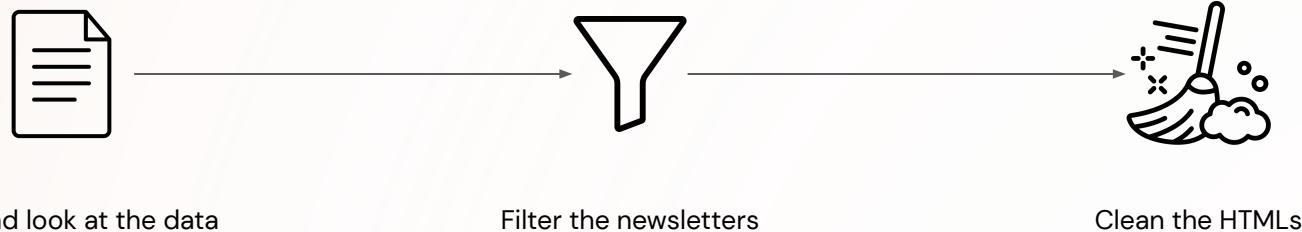
- loading: what data types do we have, how can we integrate them into one file?
- filtering: which newsletter sources do we want to keep for which time period?
- cleansing: what part of the HTML do we want to keep?
- splitting: how many stories does one newsletter contain?

for processing HTML data, we highly recommend BeautifulSoup!

(you'll see that in the practical part)

**processing for general data preparation and processing for ML can differ!**

## Practical part 2



# Building training data

**SL use case =  
algorithms + labeled data**

“

When a system isn't performing well, many teams instinctually **try to improve the code**. But for many practical applications, it's more effective instead to **focus on improving the data**.

”

Andrew Ng

Ex-Head Google Brain/Baidu AI, Founder Landing.AI Coursera, Professor Stanford

$$f(x) = y$$

model-centric

$$f(\underline{x}) = \underline{y}$$

data-centric

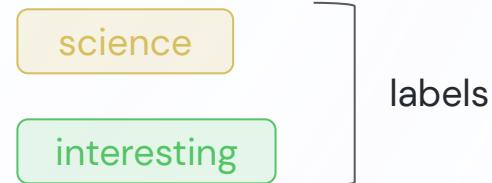
# Building training data



Scientists Publish Breakthrough  
Study in Oreo-Splitting Research

# Building training data

Scientists Publish Breakthrough  
Study in Oreo-Splitting Research



# Building training data

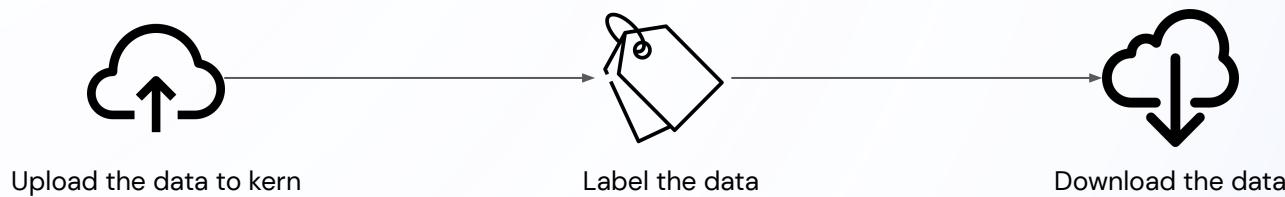


different open-source labeling tool options:

- Label Studio is the most diverse with custom HTML templates for labeling UIs
- Doccano is a great alternative, simple usage as Label Studio
- Kern AI is the most programmatic approach, focusing on NLP and semi-structured data (OS-release on 17.07)

All of them are great tools. Pick which you like the most 😊

## Practical part 3



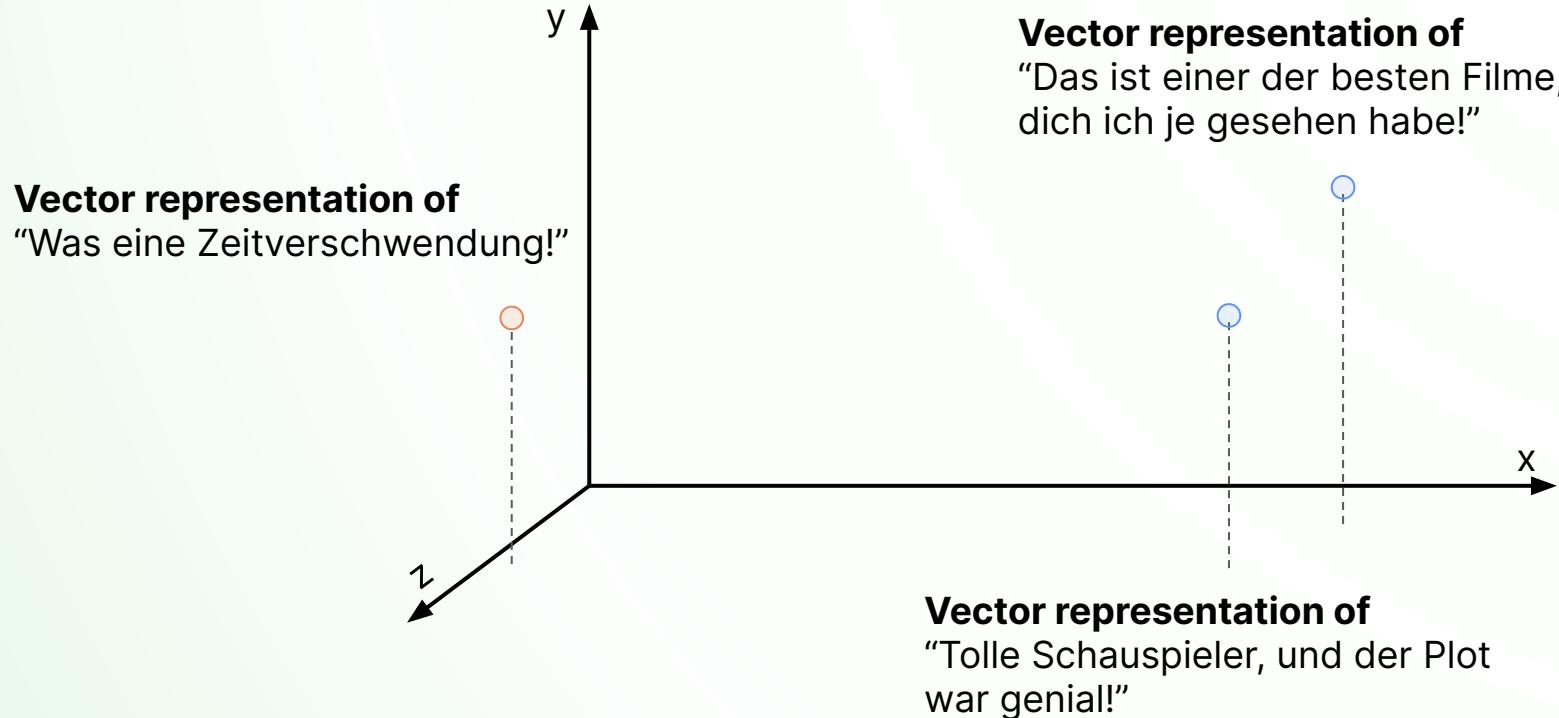
# Applying transfer learning

## integration of large-scale language models

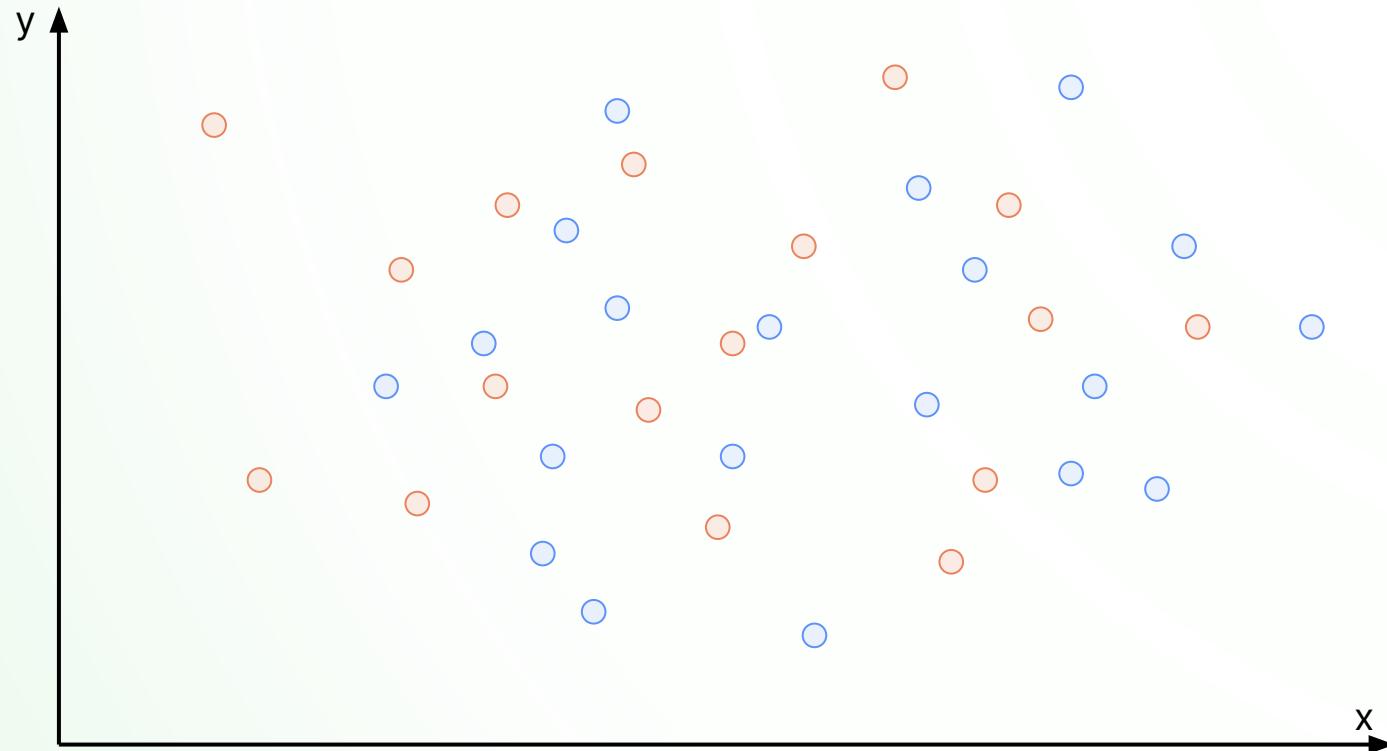
- goal: achieve great results with rather small amounts of training data
- 1. pull a transformer model from Hugging Face
- 2. create vector representations for your sentences (encoding/embedding)
- 3. train a logistic regression (or some other simple model) that takes these vectors as input

We will not finetune the model as this would take too much time!

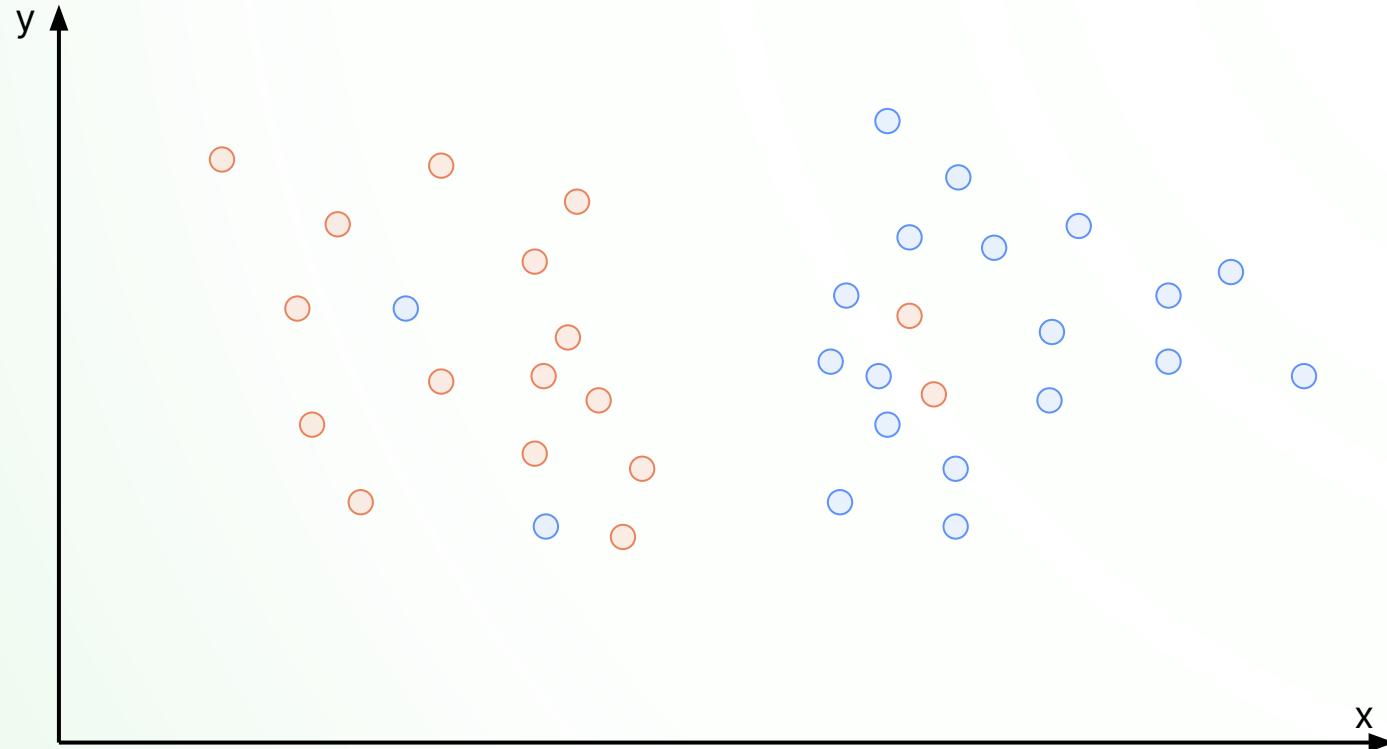
# Applying transfer learning



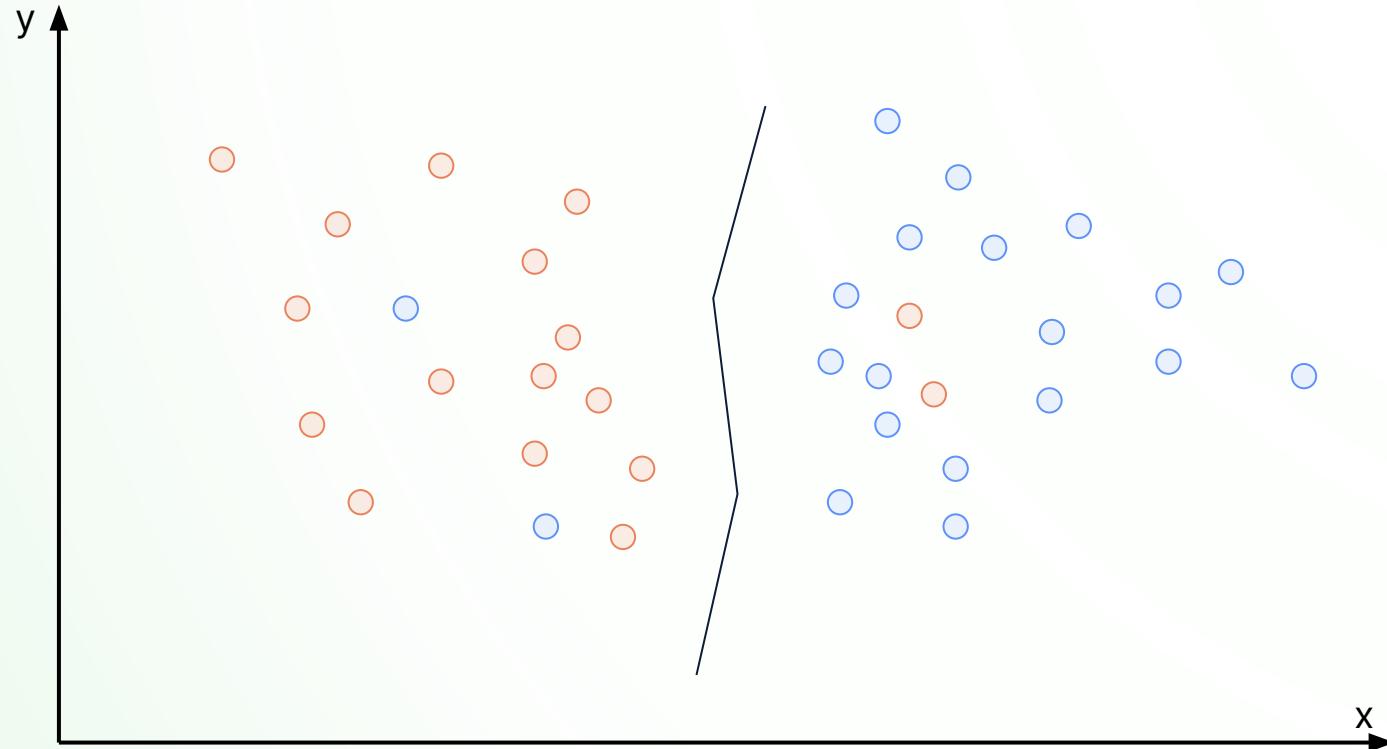
# Regular encoder



# Transformer model



# Transformer model

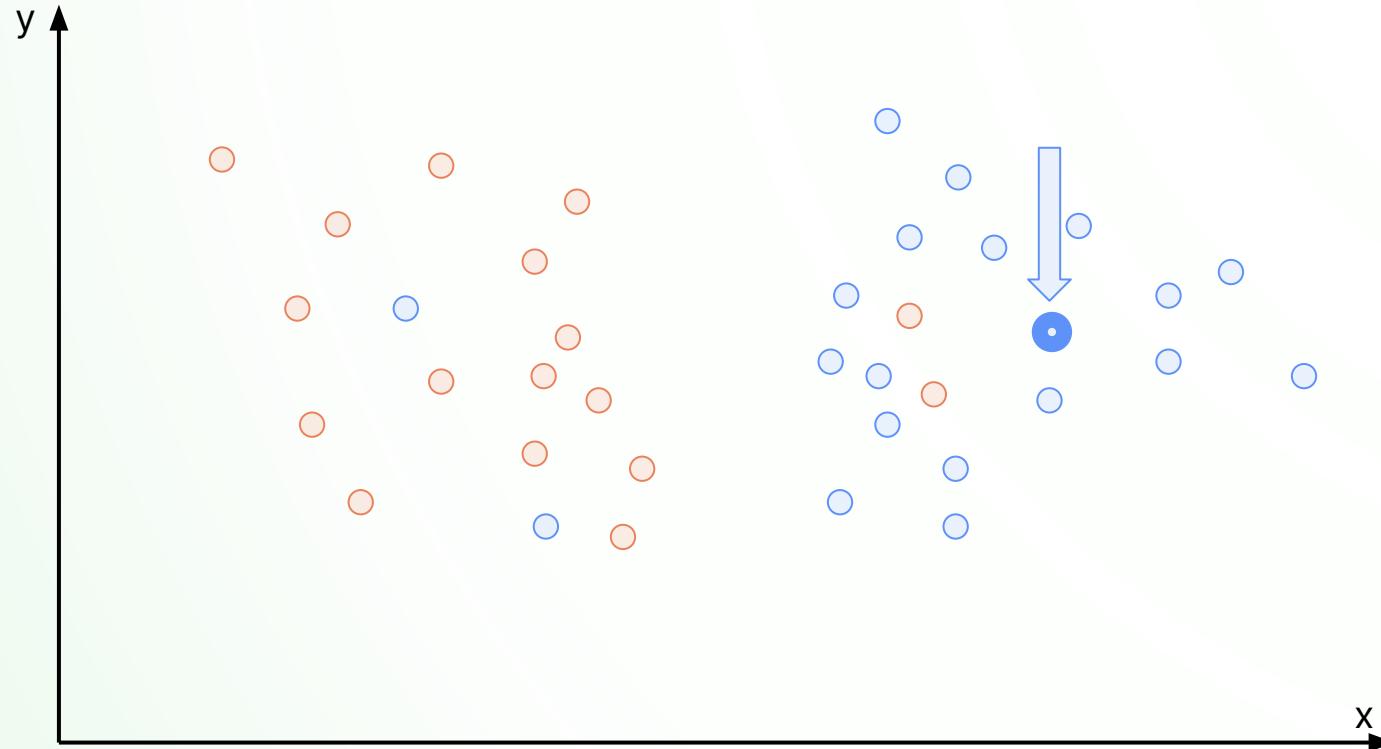


# Making recommendations

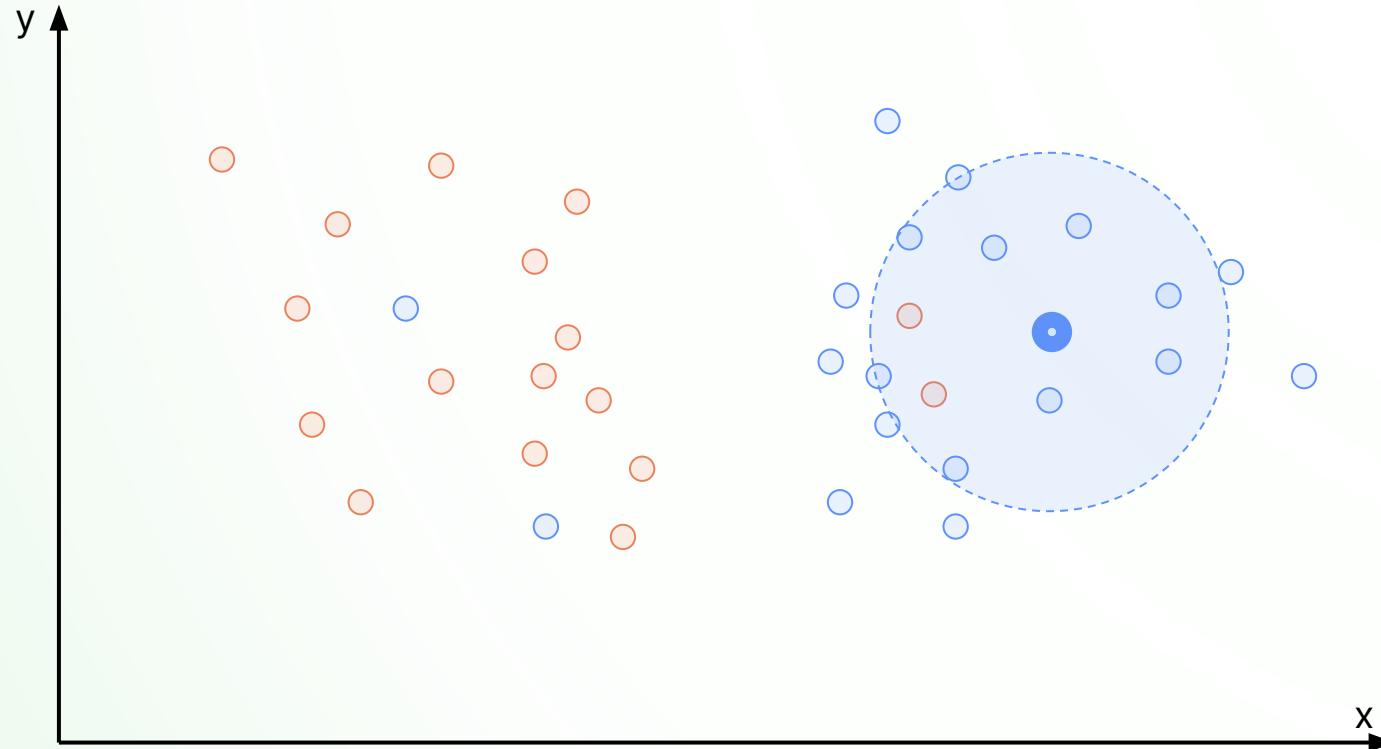
**very basic idea:**

- collect all interesting records and their embeddings
- compute the mean of this embedding as a representative embedding
- derive the k-nearest-neighbors of that representative embedding for suggestions

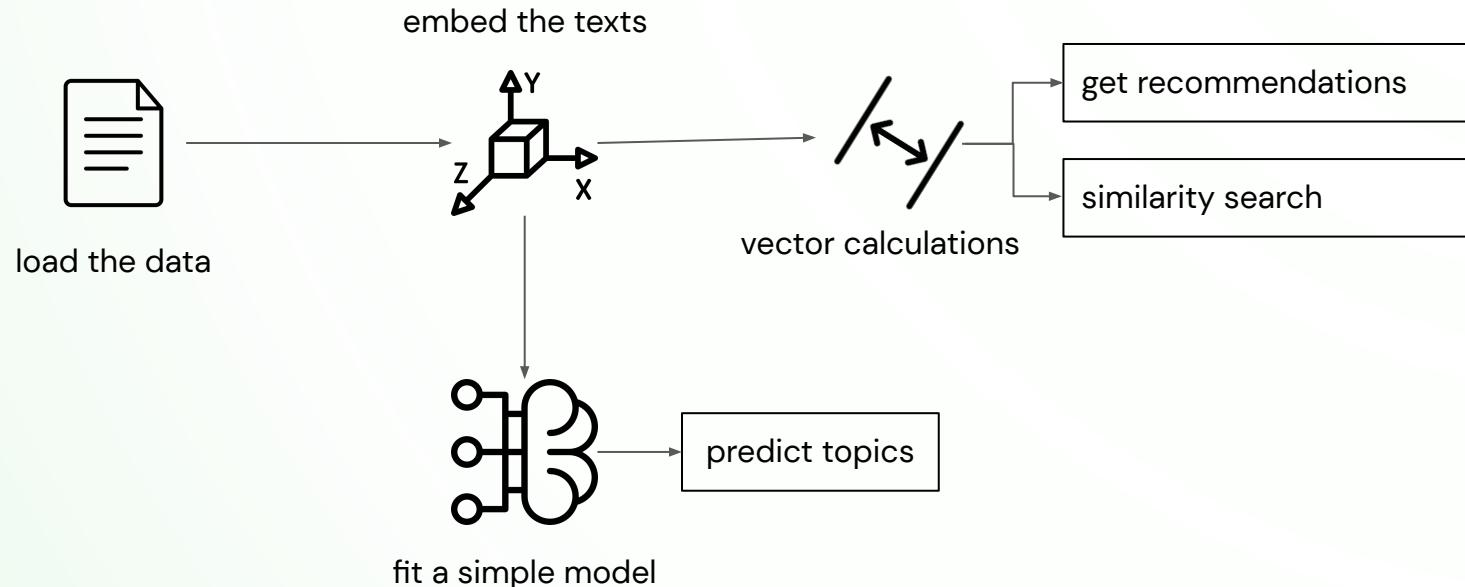
# Making recommendations



# Making recommendations



## Practical part 4



# Building the frontend

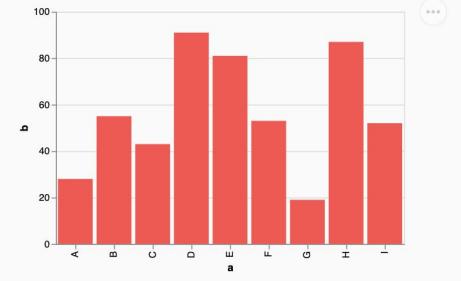
## small + easy-to-build UI

- communicate results
- improve usability
- all in Python!

Pick a number

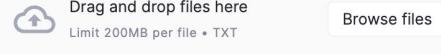


```
number = st.slider("Pick a number", 0, 100)
```



```
st.altair_chart(my_chart)
```

Pick a file



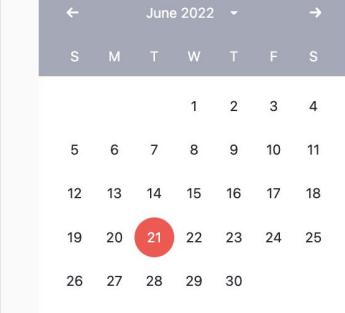
```
file = st.file_uploader("Pick a file")
```

Pick a pet

- Dog
- Cat
- Bird

```
pet = st.radio("Pic
```

Pick a date



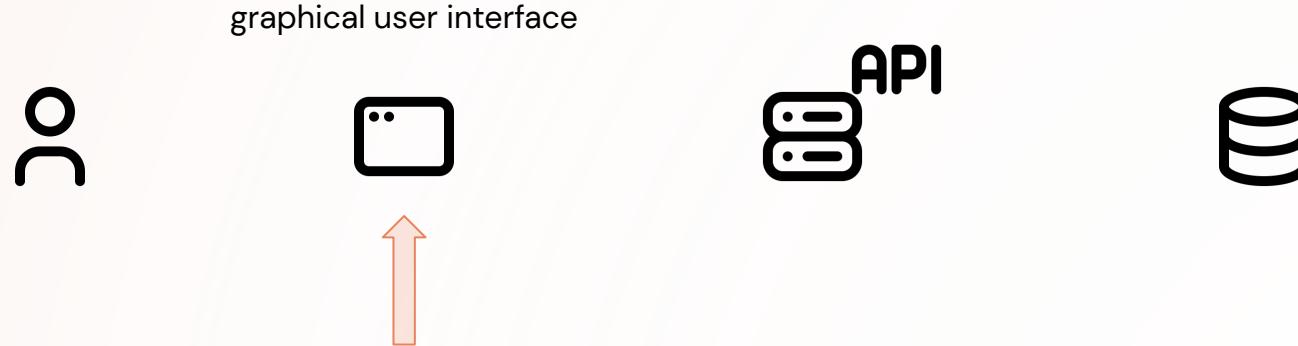
```
date = st.date_input("Pick a date")
```

Pick a color



```
color = st.color_picker()
```

# Building the frontend



# Building the frontend

```
import streamlit as st
import requests

HTML: <h1>      → st.title("Prediction UI")

styled <input>   → input_ = st.text_input(
                    "Try out your new machine learning model in this user interface. Just type some text below."
                )

call backend     → if st.button("Predict!"):
                    if input_ is not None:
                        # Get request output from the fastapi
                        response = requests.post(
                            "http://localhost:7531/predict", json={"text": [input_]}
                        )
                        if response.status_code == 200:
                            st.markdown(response.json(), unsafe_allow_html=True)
```

# Streamlit main concepts

## Data flow



## Caching

```
import streamlit as st

@st.cache # ⚡ This function will be cached
def my_slow_function(arg1, arg2):
    # Do something really slow in here!
    return the_output
```

## Session State

```
import streamlit as st

def get_session_state_value(key : str):
    # Initialization
    if key not in st.session_state:
        return None
    else:
        return st.session_state[key]
```

# Practical part 5

**Settings**

Topic selection: research and science

Newsletter selection: TLDR

Fetch recommendations

show session state

## Recommendations - 1/10

**Real World Recommendation System - Part 1**

Training a collaborative filtering based recommendation system on a toy dataset is a sophomore year project in colleges these days. But where the rubber meets the road is building such a system at scale, deploying in production, and serving live requests within a few hundred milliseconds while the user is waiting for the page to load. To build a system like this, engineers have to make decisions spanning multiple moving layers like...

[Next recommendation](#)

## Similar Stories - 1/10

**Ultra-light liquid hydrogen tanks promise to make jet fuel obsolete (3 minute read)**

Gloyer-Taylor Laboratories (GTL) has developed ultra-lightweight cryogenic tanks that have a 75% mass reduction compared with other aerospace cryotanks. A 12 kg tank from GTL is able to hold over 150 kg of hydrogen. The weight reduction means that hydrogen-fueled aircraft may be able to fly at least four times as far as comparable aircraft running on jet fuel while completely eliminating carbon emissions. It could also mean increased cargo or passenger capacity.

[Next similar](#)

## Story Browser

**A new and outlandish delivery drone concept can carry 100 pounds up to 80 miles (2 minute read)**

Austria-based Cyclotech and Japanese delivery firm Yamato have partnered to create a concept delivery drone. The CCY-01 uses a thrust vectoring propulsion system developed by Cyclotech that enables it to land in confined spaces and handle challenging wind conditions. The drone is able to produce horizontal sideways thrust without tilting. The CCY-01 has a payload capacity of 99 lbs and it can fly up to 25 miles at speeds of around 80 mph. A video of the CCY-01 performing its first free flight is available in the article.

[View full newsletter](#)

[Get similar stories](#)

## Full Newsletter HTML

If you don't want to receive future editions of TLDR, please [click here to unsubscribe](#).

**TLD**R****

Daily Update 2022-04-15

**SSH To Anywhere With Tailscale (Sponsor)**

No additional hardware to manage. No complicated firewalls. Tailscale keeps it simple & secure. [Learn more](#).

If you would like to sponsor TLDR, please let me know by replying to this email or check out our [sponsorship page](#).

**deploy the model we've built as a micro web-service**



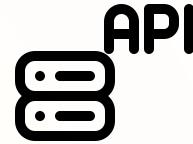
- enable fast development to production
- integrate with other services (e.g. UI)
- keep code simple



## REST API with HTTP methods

- **POST**: is used to submit an entity to the specified resource
- **GET**: requests a representation of the specified resource. Requests using GET should only retrieve data
- **PUT**: replaces all current representations of the target resource with the request payload
- **DELETE**: deletes the specified resource.
- ...

# Finishing with the backend

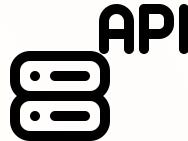


# Finishing with the backend

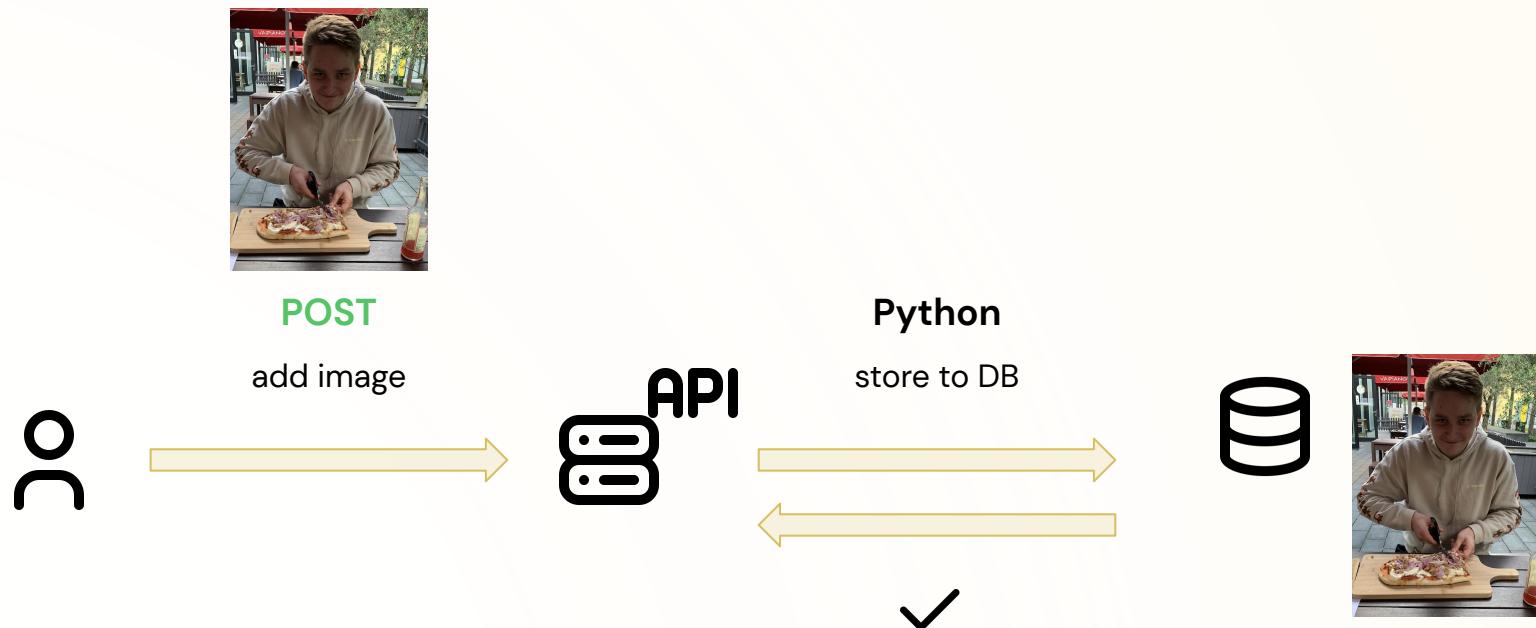


POST

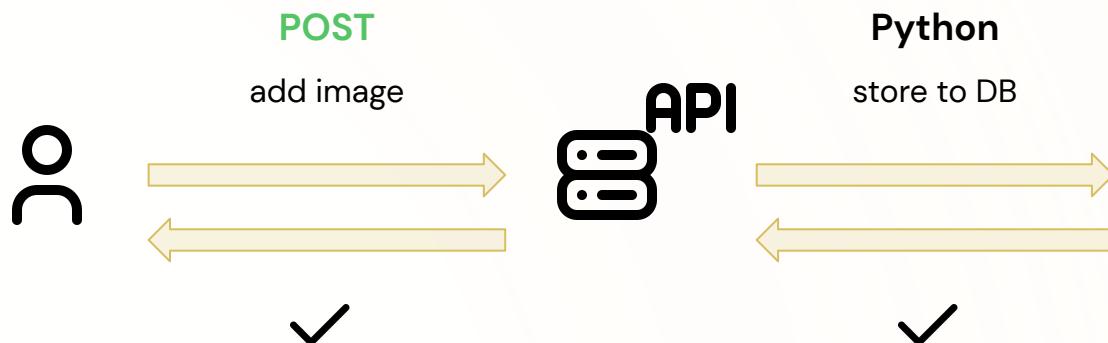
add image



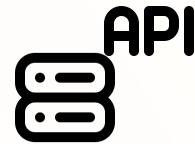
# Finishing with the backend



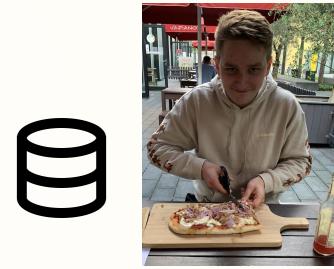
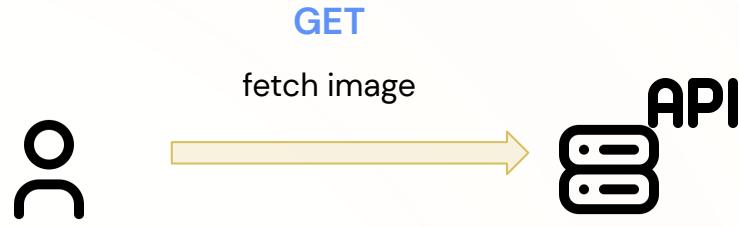
# Finishing with the backend



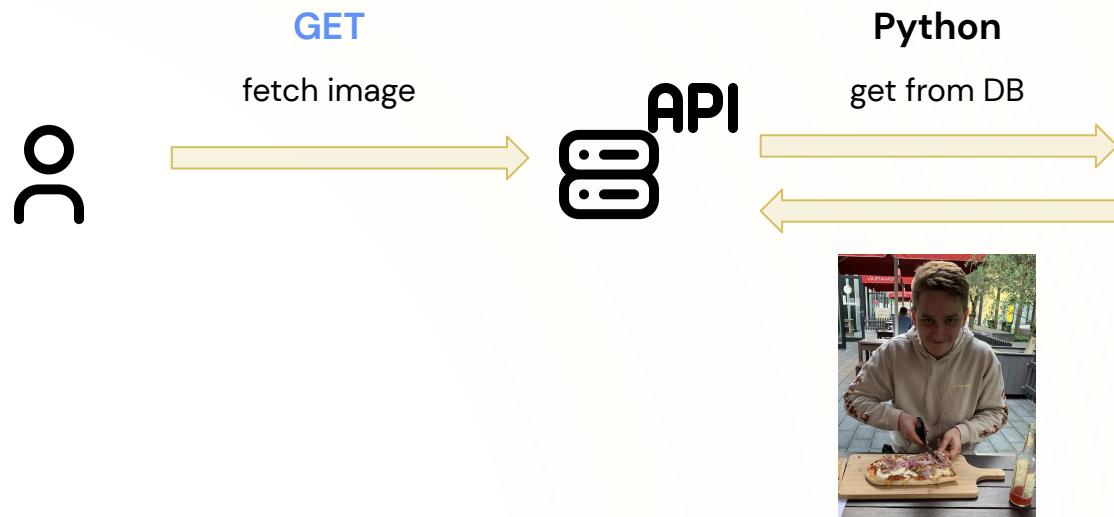
# Finishing with the backend



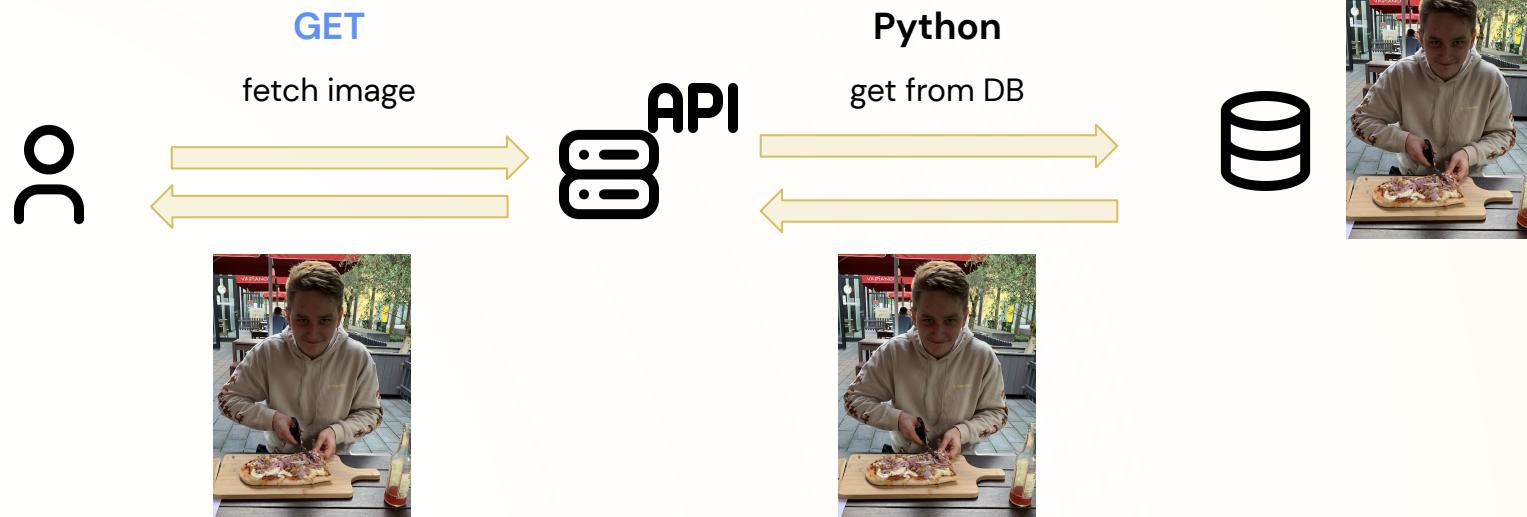
# Finishing with the backend



# Finishing with the backend



# Finishing with the backend



# Finishing with the backend

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
async def root():
    return {"message": "Hello World"}
```

That's a minimalistic but complete web API!

# Finishing with the backend

can also take as input  
parameters, e.g. as query  
params or request body

endpoint URI



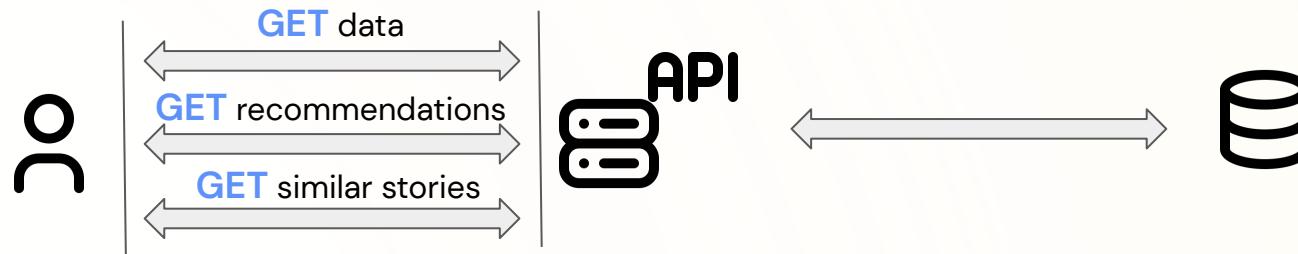
e.g. http://localhost:8000/

```
@app.get("/")
def root():
    return {"message": f"Hello World"}
```



GET value, e.g. as JSON

## Practical part 6



Increasing complexity

## Streamlit / fastAPI

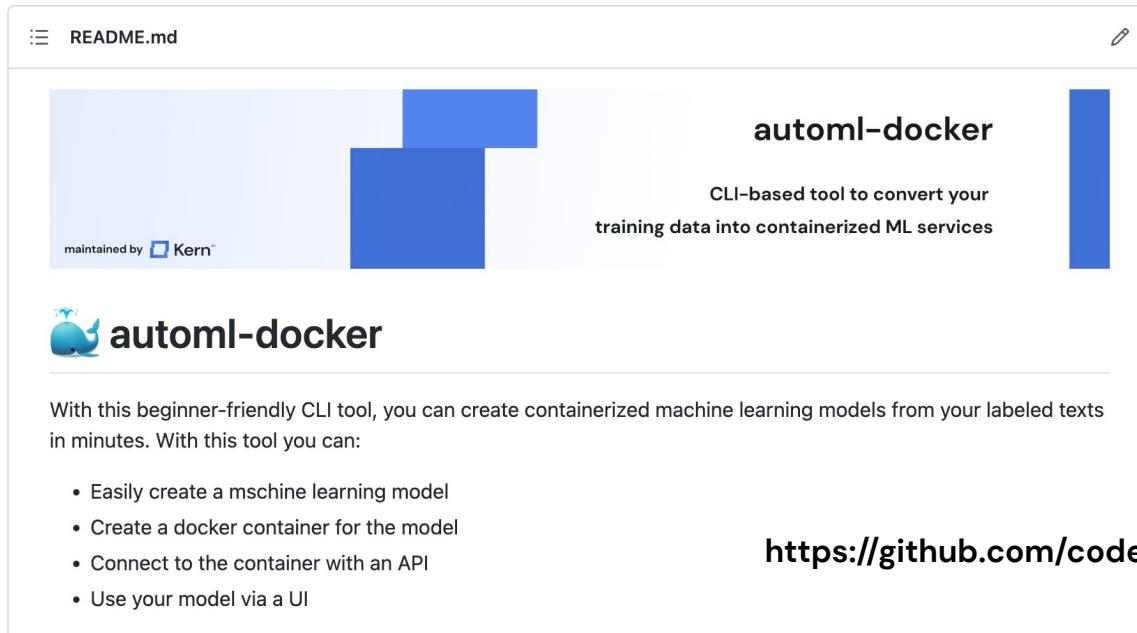
- add hyperlinks to the articles in the Story Browser
- add more filter settings (e.g. "last 7 days", "no sponsored posts", "skip empty entries", ...)
- make entries removable
- add more attributes to the cards, e.g. date, newsletter, topic
- provide label functionality for interesting articles to update recommendations
- add pagination in the front- and backend
- make it "online" with real incoming mails from your inbox

## Data Science

- reduce embedding dimensions to 2D for an explorable newsletter map
- play around with different recommendation methods and see what works best
- try more sophisticated topic modeling with BERTopic

# Check out our open-source automl-docker

**GitHub repo containing ML training, containerized FastAPI backend and Streamlit UI for custom Natural Language Classifiers.**



The screenshot shows the GitHub repository page for 'automl-docker'. The README.md file is displayed, featuring a large blue 'T' icon, the project name 'automl-docker', and a description: 'CLI-based tool to convert your training data into containerized ML services'. The repository is maintained by 'Kern'.

**automl-docker**

CLI-based tool to convert your training data into containerized ML services

maintained by  Kern<sup>AI</sup>

**automl-docker**

With this beginner-friendly CLI tool, you can create containerized machine learning models from your labeled texts in minutes. With this tool you can:

- Easily create a machine learning model
- Create a docker container for the model
- Connect to the container with an API
- Use your model via a UI

<https://github.com/code-kern-ai/automl-docker>

# Data-centric IDE for NLP tasks on 17.07.2022

The screenshot displays the Kern AI Data Enrichment interface. At the top, there are four main statistics boxes: 'Records uploaded' (71,897), 'Labeling Tasks' (3), 'Data Slices' (2), and 'Heuristics' (10). Below these are filtering options for 'ATTRIBUTE' (Online Feedback), 'LABELING TASK' (Sentiment), 'DATA SLICE' (Product-related), and a search bar with filters like 'FUNC is\_product\_related IS TRUE' and 'Category IN ['General', 'Function', 'Usability']'. The main workspace shows a 'REVIEW' table with three rows of user feedback, each with sentiment categories (General, Function, Usability) and a 'Negative' category. To the right, two detailed cards are shown: 'is\_product\_related' (86% Precision, 12% Coverage) and 'contains\_positive\_terms' (91% Precision, 8% Coverage), each with links to 'Labeling Function', 'API', 'Learning Classifier', and 'Learning Extractor'. At the bottom, three charts are displayed: 'Category Distribution' (GENERAL, FUNCTION, USABILITY), 'Sentiment Distribution' (NEGATIVE, NEUTRAL, POSITIVE), and 'Sentiment Confusion Matrix' (Manual vs. Weak Supervision).

Make sure to check out our  
**open-source** release! 😎

Register on our website  
[www.kern.ai/pages/open-source](http://www.kern.ai/pages/open-source)  
for our raffle  
(win a GeForce RTX 3090 Ti,  
Kern AI T-Shirts, ...)

# Thanks for having us!



Moritz Feuerpfeil

[moritz.feuerpfeil@kern.ai](mailto:moritz.feuerpfeil@kern.ai)

<https://www.linkedin.com/in/moritz-feuerpfeil/>



Johannes Hötter

[johannes.hoetter@kern.ai](mailto:johannes.hoetter@kern.ai)

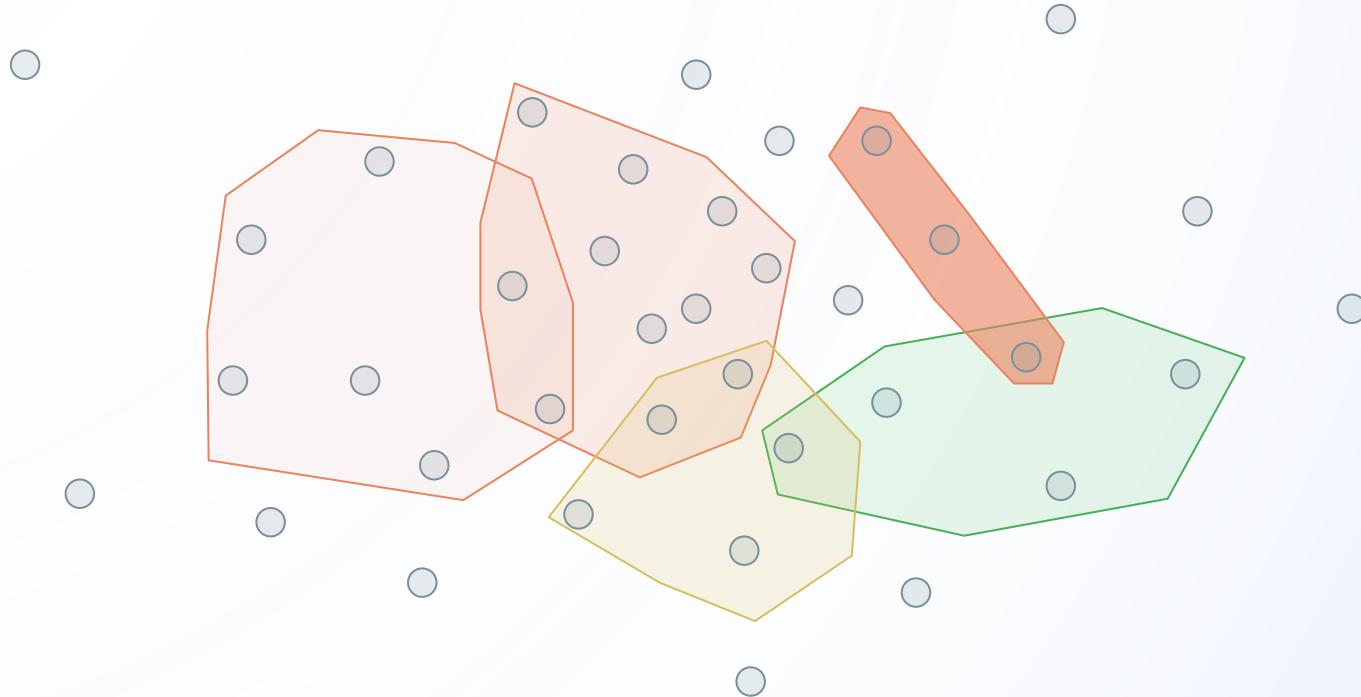
<https://www.linkedin.com/in/johanneshotter/>

# Backup Slides

# Building training data

|           | Heuristic #1 | Heuristic #2 | ... | Heuristic #N | <b>Weakly supervised</b> |
|-----------|--------------|--------------|-----|--------------|--------------------------|
| Record #1 | News         | none         |     | News         | 78.2%                    |
| Record #2 | none         | Politics     |     | Sports       | 48.1%                    |
| ...       |              |              |     |              |                          |
| Record #N | none         | none         |     | News         | 32.6%                    |

# Building training data



|  |   |
|--|---|
| <b>Majority Vote</b>                     | requires no labeled data, <i>baseline</i>   |
| <b>Stochastic Gradient Descent-based</b> | unsupervised learning,<br>labeling functions as feature matrix                      |
| <b>Closed form solutions</b>             | magnitude faster than SGD approach,<br>requires e.g. triplets as features           |
| <b>Hidden Markov Model</b>               | for sequence labeling (e.g. named entity recognition),<br>learns label dependencies |
| <b>Weighted Majority Vote</b>            | requires few labeled reference data,<br>fast and precise                            |

# Building training data

- Labeling functions

```
def starts_with_digit(record):
    if record["headline"].text[0].is_digit:
        return "Clickbait"
```

- Distant supervision (lookup values)
- Active (transfer) learning modules
- Zero-shot classifiers
- Unexperienced labelers (e.g. crowdlabeling)
- 3rd party systems, legacy systems, ...

Interface to collect noisy labels;  
Relevance of each heuristic can be  
derived from e.g. manually labeled  
reference data

# Building training data

```
def lkp_orgs(record):  
    for chunk in record["details"].noun_chunks:  
        if any([chunk.text in trie or trie in chunk.text for trie in ["M3", "..."]]):  
            yield "ORG", chunk.start, chunk.end
```

green

```
def window_cue_search(record):  
    for chunk in record["details"].noun_chunks:  
        left_bound = max(chunk.sent.start, chunk.start - (window // 2) + 1)  
        right_bound = min(chunk.sent.end, chunk.end + (window // 2) + 1)  
        window_doc = record["details"][left_bound: right_bound]  
        if any([term in window_doc.text for term in ["visits", "..."]]):  
            yield "PERSON", chunk.start, chunk.end
```

red

spaCy

```
def model_prediction(record):  
    for prediction in my_model(record["details"]):  
        label, start_idx, end_idx = prediction  
        yield label, start_idx, end_idx
```

yellow

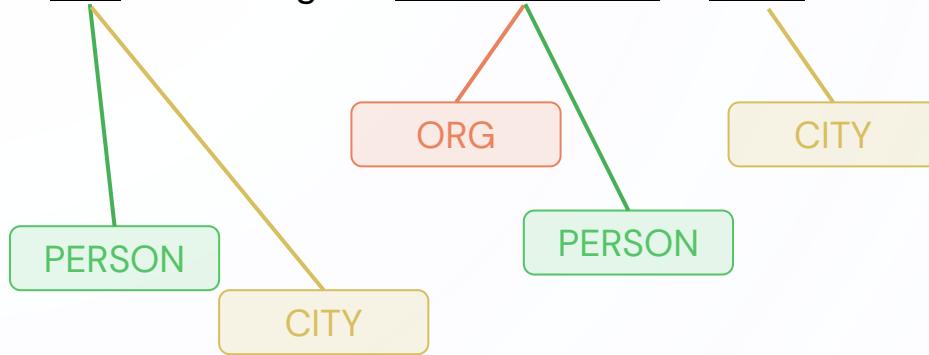


further heuristics could be:

- spaCy lookups
- requests / API
- zero-shot models
- token-constraints

# Building training data

"Max visits the great Datalift summit in Berlin. He loves to meet people in person again!"



```
{  
    "person": "Max",  
    "org": "Datalift summit",  
    "city": "Berlin"  
}
```

selection depends on algorithm!

# Building training data



# Building training data

1. pick a random sample
2. get the most diverse  
(distant) sample from  
labeled samples



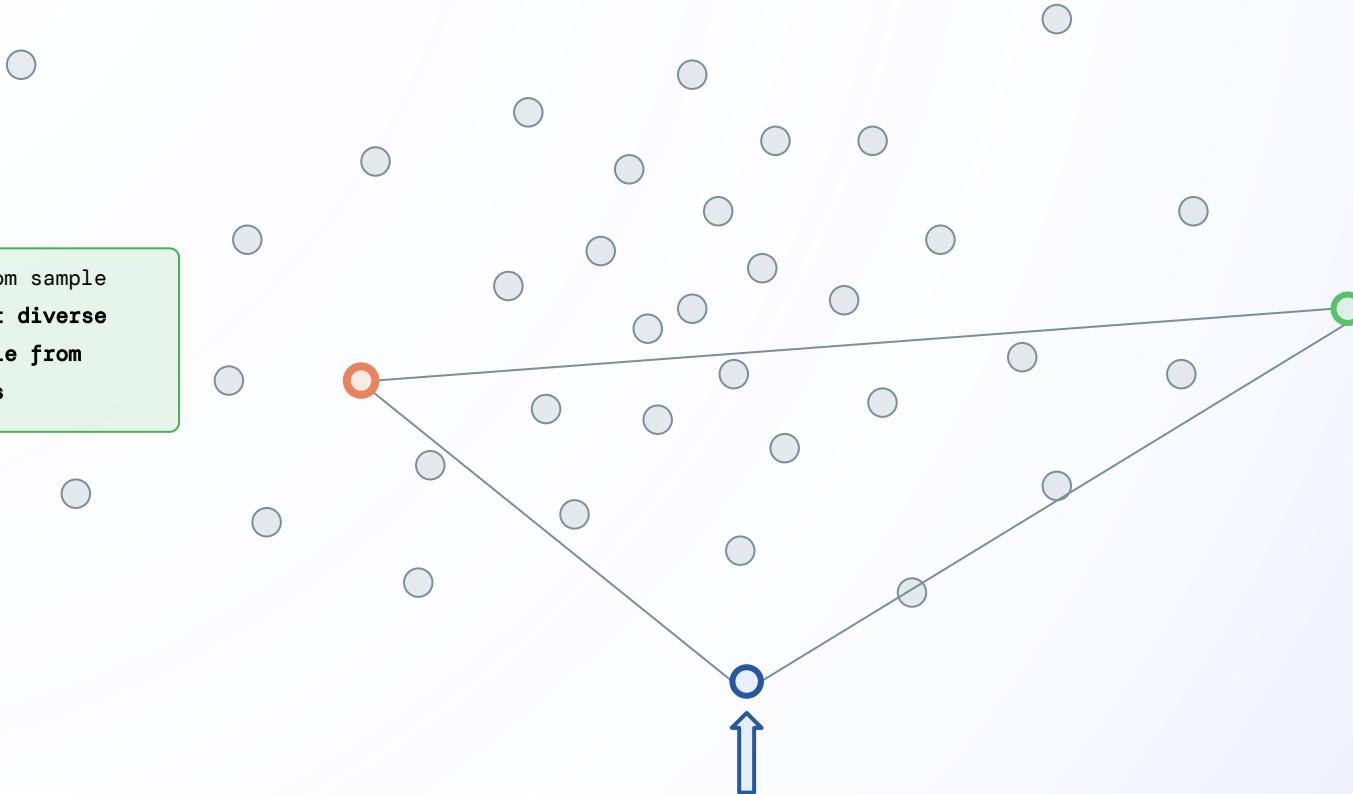
# Building training data



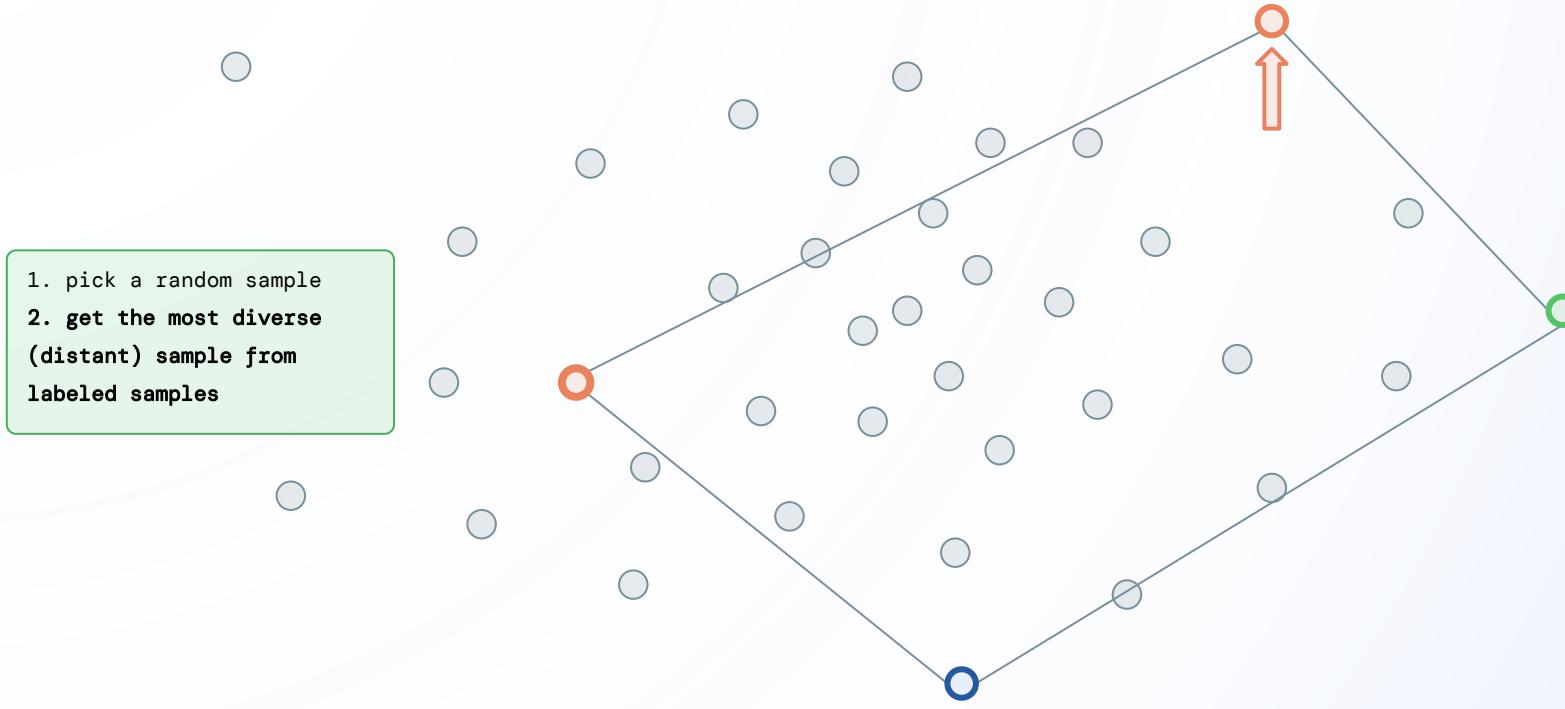
1. pick a random sample  
2. get the most diverse  
(distant) sample from  
labeled samples

# Building training data

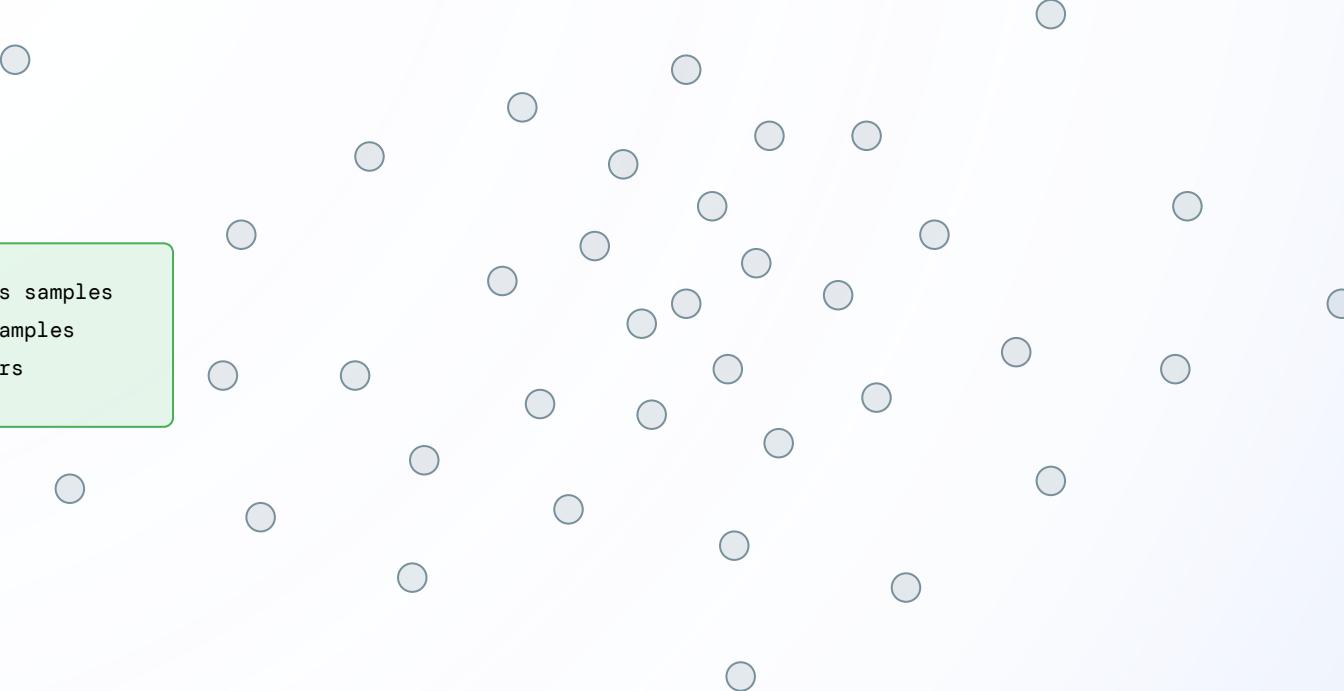
1. pick a random sample
2. get the most diverse  
(distant) sample from  
labeled samples



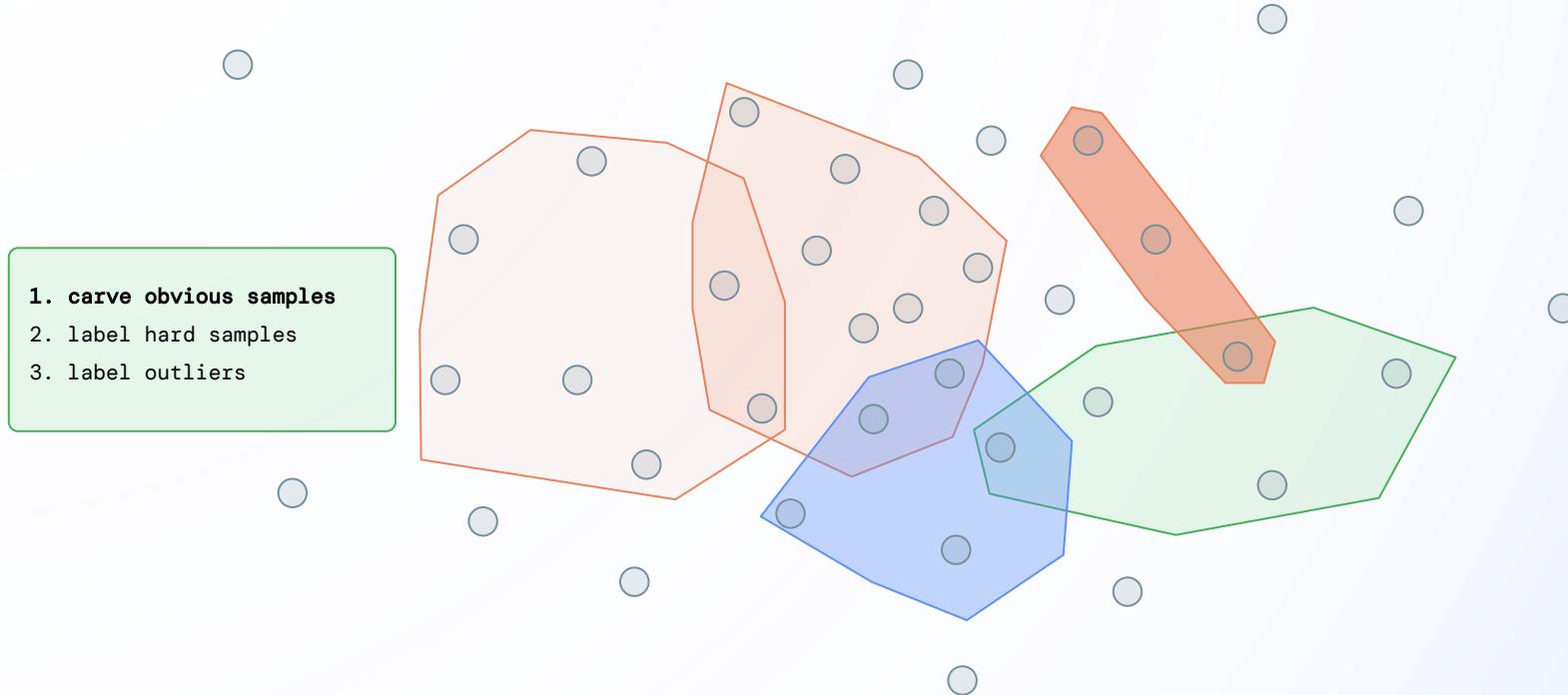
# Building training data



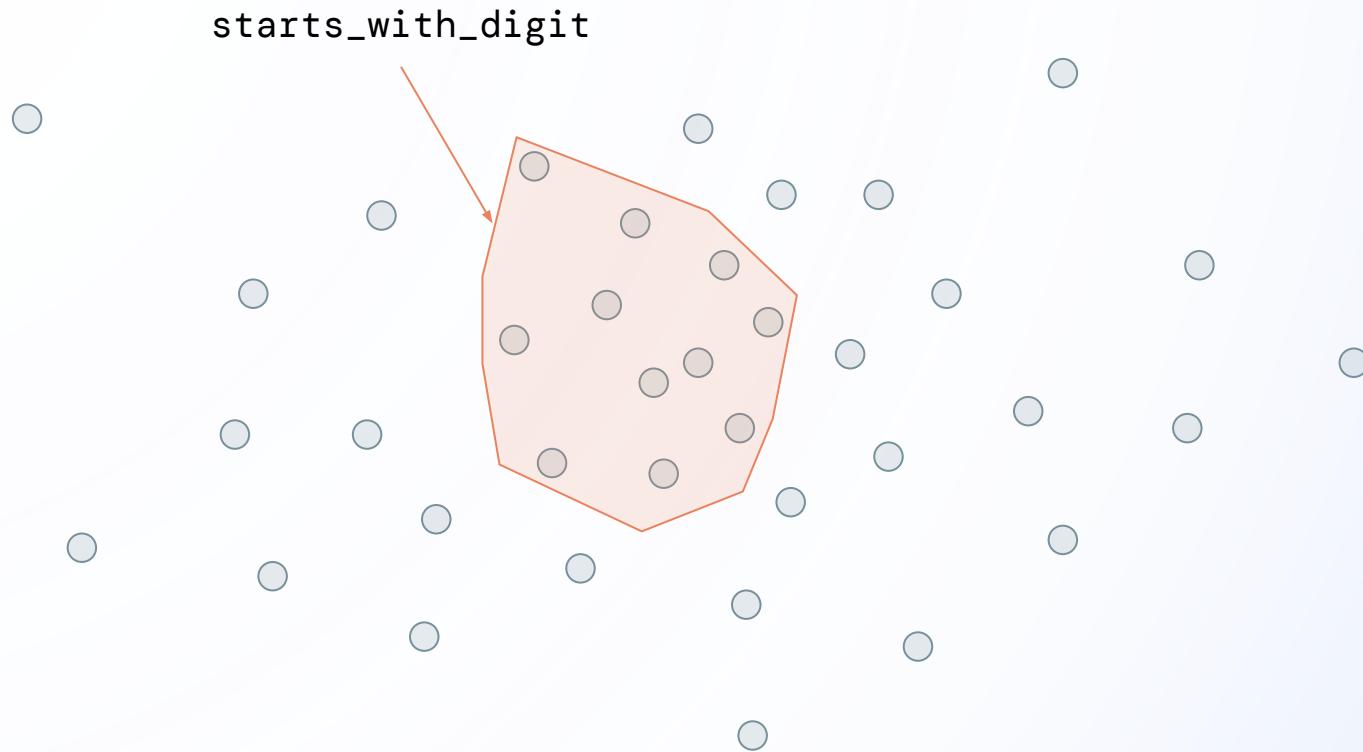
# Building training data

- 
- A scatter plot showing numerous light blue circular data points distributed across a white background. The points are scattered in a somewhat uniform pattern, with no clear linear or non-linear trend.
- 1. carve obvious samples
  - 2. label hard samples
  - 3. label outliers

# Building training data



# Building training data



# Building training data

```
def starts_with_digit(record):
    if record["headline"].text[0].is_digit:
        return "Clickbait"
```

Precision 83%  
Coverage 2.5%



analyze filter where `starts_with_digit == "Clickbait"`

```
def starts_with_digit(record):
    if record["headline"].text[0].is_digit
        and record["sentiment"] > 0.7:
            return "Clickbait"
```

Precision 92%  
Coverage 1.8%

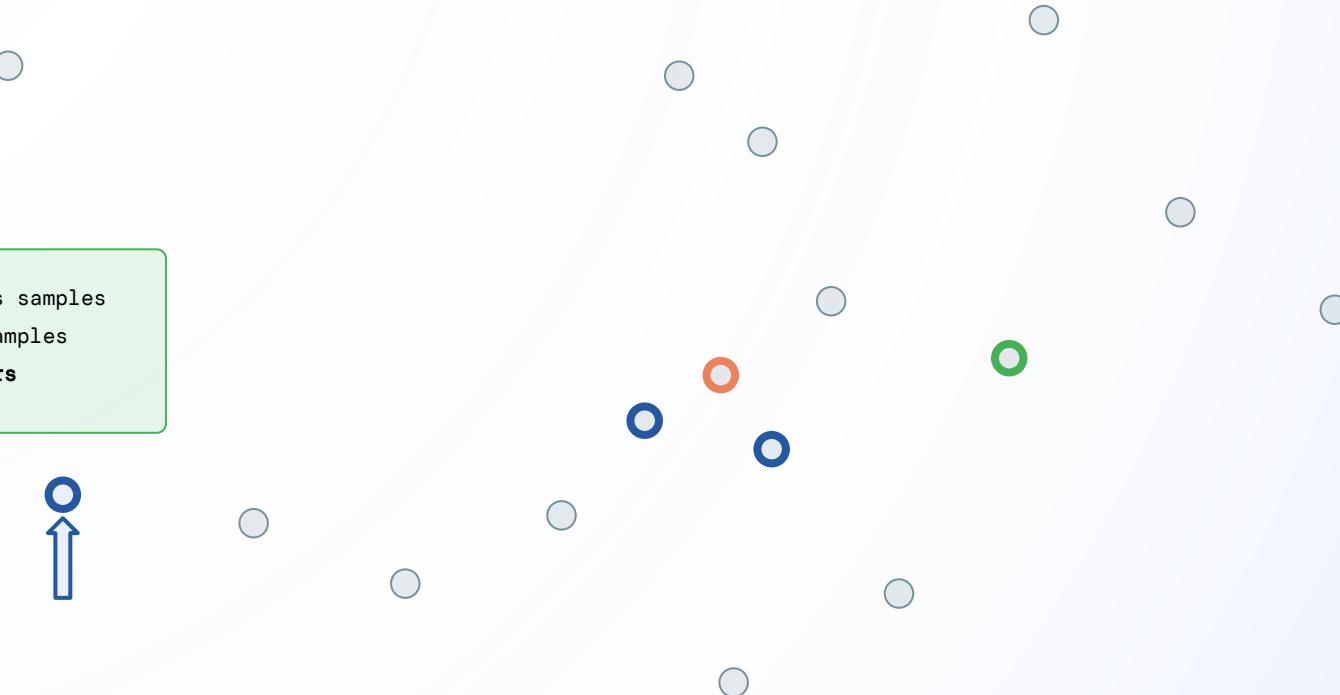
# Building training data



# Building training data



# Building training data



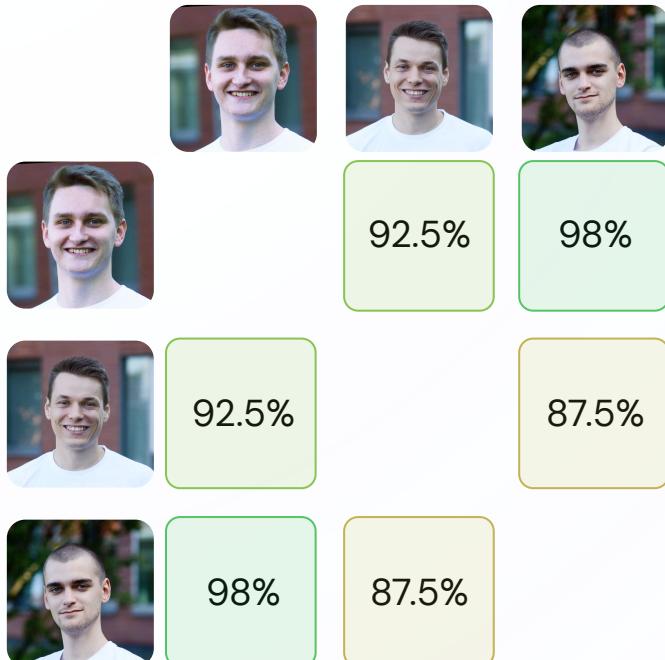
# Building training data



- 1. carve obvious samples
- 2. label hard samples
- 3. label outliers**



# Building training data



```
def starts_with_digit(record):  
    if record["headline"].text[0].is_digit:  
        return "Clickbait"
```



95% precision



75% precision

# Setting oauth2 up for Gmail

## requires App in Google Cloud Platform

1. set up a GCP account
2. set up a new project and go to [API page](#)
3. activate GMail API in GCP via the library
4. create an oauth2 consent screen
5. set up credentials (oauth2 client ids) and download them as json file
6. write a little python script to integrate Gmail using the newly created oauth2 consent screen

# Set up a new project in GCP

≡ Google Cloud

## Neues Projekt

Projektname \*  ?

Project ID: datalift-354017. Sie kann später nicht mehr geändert werden.

[BEARBEITEN](#)

Organisation \*  ▼ ?

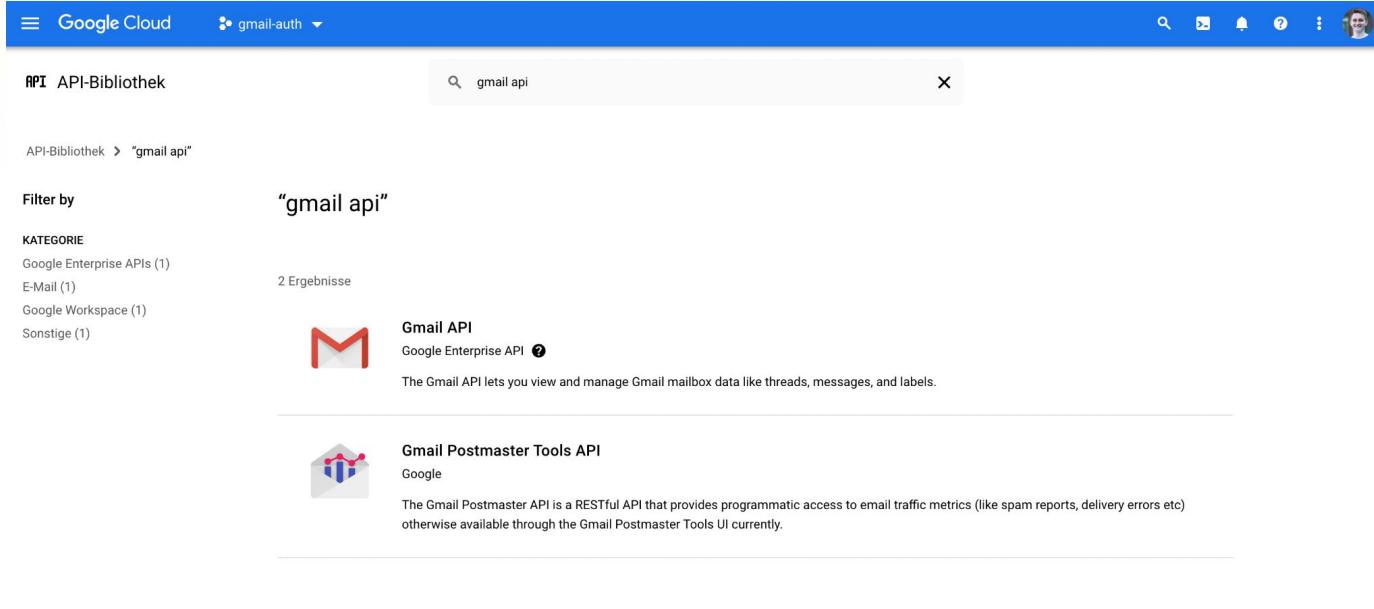
Wählen Sie eine Organisation aus, um sie mit einem Projekt zu verknüpfen. Diese Auswahl kann nicht rückgängig gemacht werden.

Speicherort \*  DURCHSUCHEN

Übergeordnete Organisation oder übergeordneter Ordner

[ERSTELLEN](#) [ABBRECHEN](#)

# Activate GMail via the library

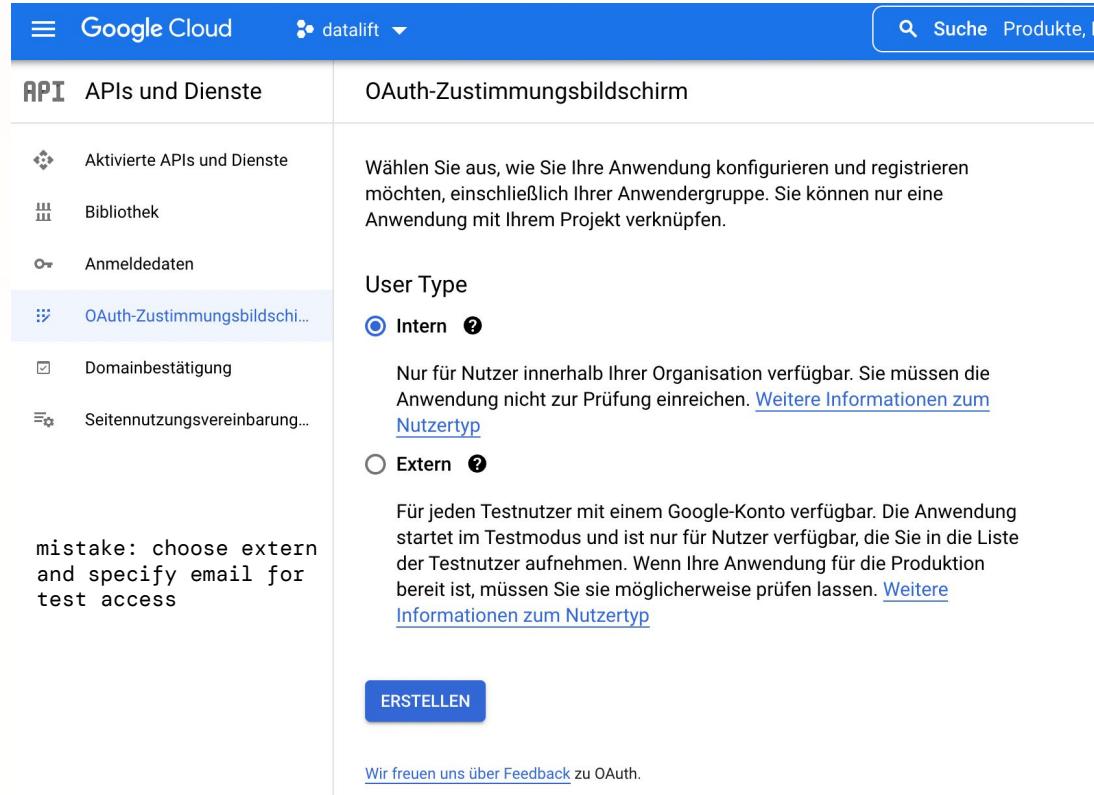


The screenshot shows the Google Cloud API Library interface. At the top, there's a blue header bar with the Google Cloud logo and a dropdown menu labeled "gmail-auth". A search bar contains the query "gmail api". Below the header, the title "API API-Bibliothek" is displayed, followed by a breadcrumb trail: "API-Bibliothek > "gmail api"".

A "Filter by" section is present, with "KATEGORIE" expanded, showing categories like "Google Enterprise APIs (1)", "E-Mail (1)", "Google Workspace (1)", and "Sonstige (1)". The search results show "2 Ergebnisse".

The first result is the "Gmail API", which is a "Google Enterprise API". Its description states: "The Gmail API lets you view and manage Gmail mailbox data like threads, messages, and labels." The second result is the "Gmail Postmaster Tools API", provided by Google, with a description stating: "The Gmail Postmaster API is a RESTful API that provides programmatic access to email traffic metrics (like spam reports, delivery errors etc) otherwise available through the Gmail Postmaster Tools UI currently."

# Create a consent screen



The screenshot shows the Google Cloud API & Services page with the sidebar expanded. The left sidebar has the following items:

- Aktivierte APIs und Dienste
- Bibliothek
- Anmelddaten
- OAuth-Zustimmungsbildschirm** (highlighted with a blue background)
- Domainbestätigung
- Seitennutzungsvereinbarung...

The main content area is titled "OAuth-Zustimmungsbildschirm". It contains the following text:

Wählen Sie aus, wie Sie Ihre Anwendung konfigurieren und registrieren möchten, einschließlich Ihrer Anwendergruppe. Sie können nur eine Anwendung mit Ihrem Projekt verknüpfen.

**User Type**

**Intern** ?

Nur für Nutzer innerhalb Ihrer Organisation verfügbar. Sie müssen die Anwendung nicht zur Prüfung einreichen. [Weitere Informationen zum Nutzertyp](#)

**Extern** ?

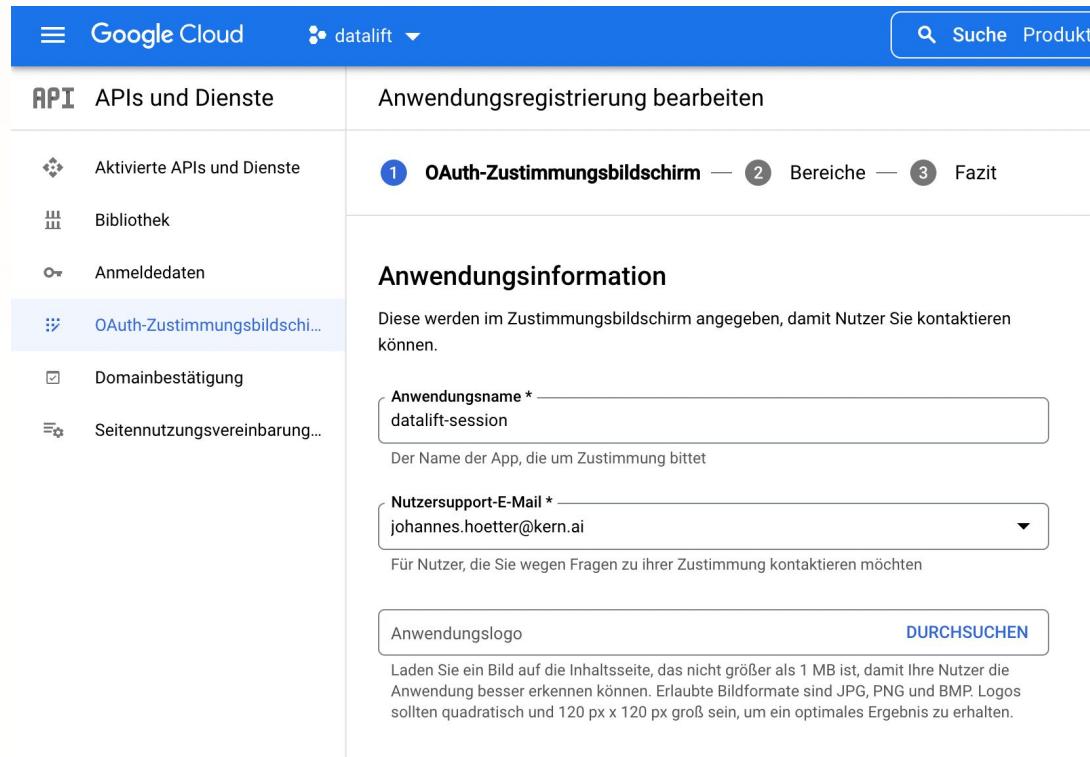
Für jeden Testnutzer mit einem Google-Konto verfügbar. Die Anwendung startet im Testmodus und ist nur für Nutzer verfügbar, die Sie in die Liste der Testnutzer aufnehmen. Wenn Ihre Anwendung für die Produktion bereit ist, müssen Sie sie möglicherweise prüfen lassen. [Weitere Informationen zum Nutzertyp](#)

**ERSTELLEN**

[Wir freuen uns über Feedback](#) zu OAuth.

mistake: choose extern and specify email for test access

# Create a consent screen



The screenshot shows the Google Cloud API registration interface. On the left, a sidebar lists options like "Aktivierte APIs und Dienste", "Bibliothek", "Anmelddaten", and "OAuth-Zustimmungsbildschirm". The "OAuth-Zustimmungsbildschirm" option is selected and highlighted in blue. The main content area is titled "Anwendungsregistrierung bearbeiten" and displays the "OAuth-Zustimmungsbildschirm" configuration step, which is the first of three steps. It includes fields for "Anwendungsnname" (dataflow-session) and "Nutzersupport-E-Mail" (johannes.hoetter@kern.ai). Below these fields, there's a section for "Anwendungslogo" with instructions about logo requirements. At the bottom right, there's a "DURCHSUCHEN" button.

Anwendungsregistrierung bearbeiten

1 OAuth-Zustimmungsbildschirm — 2 Bereiche — 3 Fazit

### Anwendungsinformation

Diese werden im Zustimmungsbildschirm angegeben, damit Nutzer Sie kontaktieren können.

Anwendungsnname \* dataflow-session

Nutzersupport-E-Mail \* johannes.hoetter@kern.ai

Anwendungslogo

Laden Sie ein Bild auf die Inhaltsseite, das nicht größer als 1 MB ist, damit Ihre Nutzer die Anwendung besser erkennen können. Erlaubte Bildformate sind JPG, PNG und BMP. Logos sollten quadratisch und 120 px x 120 px groß sein, um ein optimales Ergebnis zu erhalten.

DURCHSUCHEN

# Create a consent screen

Google Cloud    dataflow ▾

Suche Produkte, Ressourcen, Dokumente (/)

API APIs und Dienste Anwendungsregistrierung bearbeiten

Aktivierte APIs und Dienste OAuth-Zustimmungsbildschirm — 2 Bereiche — 3 Fazit

Bibliothek Anmelddaten OAuth-Zustimmungsbildsch... Domainbestätigung Seitennutzungsvereinbarung...

Bereiche stellen die Berechtigungen dar, die Sie bei Ihren Nutzern zur Autorisierung für Ihre Anwendung anfordern. Sie erlauben Ihrem Projekt den Zugriff auf bestimmte private Nutzerdaten aus ihrem Google-Konto. [Weitere Informationen](#)

**BEREICHE HINZUFÜGEN ODER ENTFERNEN**

**Meine nicht vertraulichen Bereiche**

| API ↑                     | Umfang | Für den Nutzer sichtbare Beschreibung |
|---------------------------|--------|---------------------------------------|
| Keine Zeilen zum Anzeigen |        |                                       |

**Meine vertraulichen Bereiche**

Vertrauliche Bereiche sind Bereiche, die Zugriff auf private Nutzerdaten anfordern.

| API ↑                     | Umfang | Für den Nutzer sichtbare Beschreibung |
|---------------------------|--------|---------------------------------------|
| Keine Zeilen zum Anzeigen |        |                                       |

**Meine eingeschränkten Bereiche**

Eingeschränkte Bereiche sind Bereiche, die Zugriff auf sehr vertrauliche Nutzerdaten anfordern.

| API ↑                     | Umfang | Für den Nutzer sichtbare Beschreibung |
|---------------------------|--------|---------------------------------------|
| Keine Zeilen zum Anzeigen |        |                                       |

1 Nur Bereiche für aktivierte APIs sind im Folgenden aufgelistet. Wenn Sie einen fehlenden Bereich hinzufügen möchten, suchen und aktivieren Sie die API in der [Google API-Bibliothek](#) oder verwenden Sie das Textfeld für eingefügte Bereiche unten. Aktualisieren Sie die Seite, damit alle neu aktivierten APIs aus der Bibliothek aufgeführt werden.

Filter Name oder Wert des Attributs eingeben

| API ↑                               | Umfang   | Für den Nutzer sichtbare Beschreibung   |
|-------------------------------------|--|---|
| <input type="checkbox"/>            | Cloud Trace API .../auth/trace.readonly                | Trace-Daten für ein Projekt oder eine Anwendung lesen                                       |
| <input type="checkbox"/>            | Cloud Trace API .../auth/trace.append                  | Trace-Daten für ein Projekt oder eine Anwendung schreiben                                   |
| <input type="checkbox"/>            | Gmail API https://mail.google.com/                     | Gmail-E-Mails lesen, schreiben, senden und endgültig löschen                                |
| <input type="checkbox"/>            | Gmail API .../auth/gmail.modify                        | E-Mails über Ihr Gmail-Konto aufrufen, verfassen und senden                                 |
| <input type="checkbox"/>            | Gmail API .../auth/gmail.compose                       | Entwürfe verwalten und E-Mails senden   |
| <input type="checkbox"/>            | Gmail API .../auth/gmail.addons.current.action.compose | Entwürfe verwalten und E-Mails senden, wenn mit dem Add-on interagiert wird                 |
| <input type="checkbox"/>            | Gmail API .../auth/gmail.addons.current.message.action | E-Mails abrufen, wenn Sie mit dem Add-on interagieren                                       |
| <input checked="" type="checkbox"/> | Gmail API .../auth/gmail.readonly                      | E-Mails und Einstellungen abrufen   |
| <input type="checkbox"/>            | Gmail API .../auth/gmail.metadata                      | Metadaten der E-Mail-Nachricht abrufen, z. B. Labels und Header, aber nicht den E-Mail-Text |
| <input type="checkbox"/>            | Gmail API .../auth/gmail.insert                        | E-Mails Ihrem Gmail-Postfach hinzufügen   |

Zeilen pro Seite: 10 ▾ 21 – 30 von 39 < >

**Bereiche manuell hinzufügen**

Wenn die Bereiche, die Sie hinzufügen möchten, nicht in der Tabelle oben aufgeführt werden, können Sie sie hier eingeben. Tragen Sie jeden Bereich in eine neue Zeile ein oder trennen Sie die Bereiche durch Kommas. Geben Sie den vollständigen Bereichsstring (beginnend mit „https://“) an. Klicken Sie dann auf „Zu Tabelle hinzufügen“.

ZU TABELLE HINZUFÜGEN

# Creating credentials

The screenshot shows the Google Cloud Platform interface for creating an OAuth Client ID. The top navigation bar includes the Google Cloud logo, a dropdown for 'datalift', a search bar, and a products menu.

The main content area is titled 'OAuth-Client-ID erstellen' (Create OAuth Client ID). On the left, a sidebar lists several sections: 'Aktivierte APIs und Dienste', 'Bibliothek', 'Anmelddaten' (selected), 'OAuth-Zustimmungsbildschirm...', 'Domainbestätigung', and 'Seitennutzungsvereinbarung...'. The 'Anmelddaten' section contains fields for 'Anwendungstyp \*' (set to 'Webanwendung') and 'Name \*' (set to 'Webclient 1'). A note below states that the name is used for client identification in the console and is not shown to end-users. Another note explains that authorized domains are automatically listed on the OAuth consent screen.

Below this, there is a section titled 'Autorisierte JavaScript-Quellen' with a note about its use for browser-based requests. It features a '+ URI HINZUFÜGEN' button.

Further down is a section titled 'Autorisierte Weiterleitungs-URIs' with a note about its use for server-based requests. It shows a single entry 'http://localhost:8000' in a field labeled 'URIs 1 \*' and a '+ URI HINZUFÜGEN' button.

# Creating credentials

## OAuth-Client erstellt

Auf die Client-ID und das Secret können Sie immer über "Zugangsdaten" unter "APIs & Dienste" zugreifen.

**i** Der OAuth-Zugriff ist auf Nutzer in Ihrer Organisation beschränkt, bis der [OAuth-Zustimmungsbildschirm](#) veröffentlicht und überprüft wurde

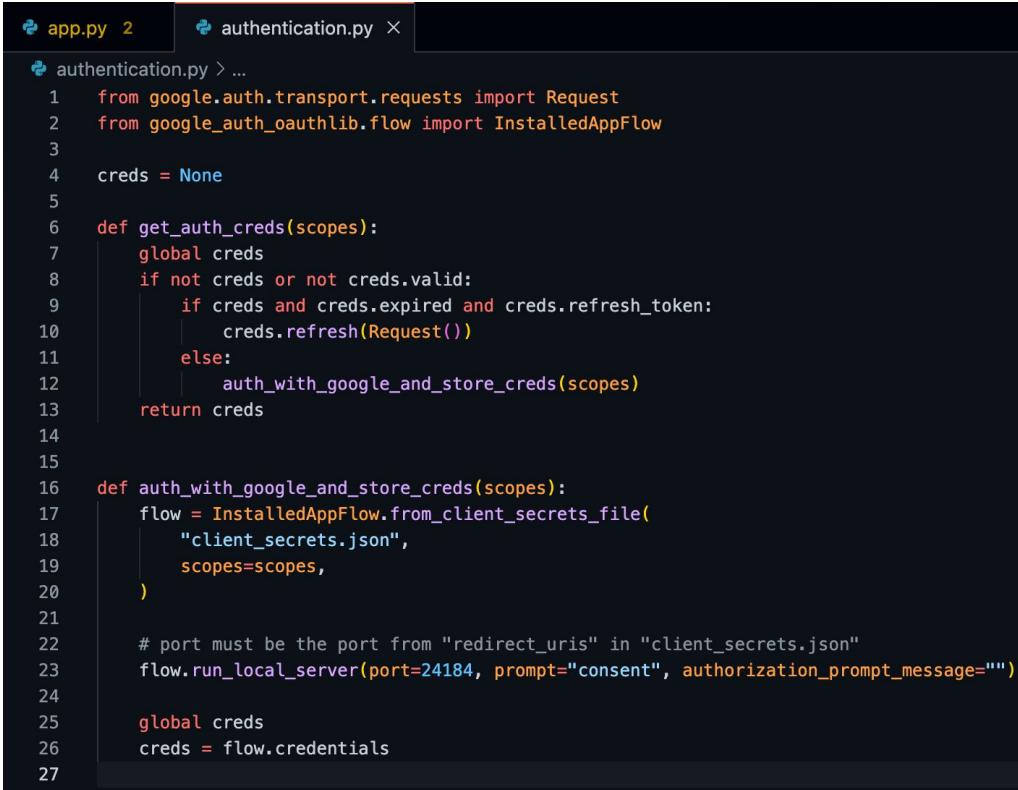
Ihre Client-ID — `662718159420-h9fhe2r7ilvunmon6ubgu6mi2ov4b1rs.apps.googleusercontent.com` 

Ihr Clientschlüssel — `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` 

 [JSON HERUNTERLADEN](#)

[OK](#)

# Integrating via Python



```
app.py 2 authentication.py X

authentication.py > ...
1  from google.auth.transport.requests import Request
2  from google_auth_oauthlib.flow import InstalledAppFlow
3
4  creds = None
5
6  def get_auth_creds(scopes):
7      global creds
8      if not creds or not creds.valid:
9          if creds and creds.expired and creds.refresh_token:
10              creds.refresh(Request())
11          else:
12              auth_with_google_and_store_creds(scopes)
13      return creds
14
15
16  def auth_with_google_and_store_creds(scopes):
17      flow = InstalledAppFlow.from_client_secrets_file(
18          "client_secrets.json",
19          scopes=scopes,
20      )
21
22      # port must be the port from "redirect_uris" in "client_secrets.json"
23      flow.run_local_server(port=24184, prompt="consent", authorization_prompt_message="")
24
25      global creds
26      creds = flow.credentials
27
```

check if new creds are needed

credentials from GCP

authorize on local server

# Integrating via Python

```
app.py 2 X authentication.py
app.py > ...
1  from googleapiclient.errors import HttpError
2  from googleapiclient.discovery import build
3  from authentication import get_auth_creds
4  from tqdm import tqdm
5
6
7 SCOPES = ["https://www.googleapis.com/auth/gmail.readonly"]
8
9 if __name__ == "__main__":
10    creds = get_auth_creds(SCOPES)
11
12    mail_data = []
13    try:
14        # find docs for service here:
15        # https://developers.google.com/resources/api-libraries/documentation/gmail/v1/python/latest/index.html
16        service = build("gmail", "v1", credentials=creds)
17
18        api_result = service.users().messages().list(userId="me", q=query).execute()
19        messages = api_result["messages"]
20        result_size_estimate = api_result["resultSizeEstimate"]
21        for idx, message_meta in enumerate(tqdm(messages, total=result_size_estimate)):
22
23
24            message_dict = {}
25            message_id = message_meta["id"]
26            message = (
27                service.users().messages().get(userId="me", id=message_id).execute()
28            )
29            payload = message["payload"]
30            headers = payload["headers"]
31            for header in headers:
32                key = header["name"]
33                if key in ["Delivered-To", "Cc", "From", "Date", "Subject"]:
34                    value = header["value"]
35                    message_dict[key] = value
36            message_dict["Snippet"] = message["snippet"]
37            mail_data.append(message_dict)
38
39
40    except HttpError as error:
41        print(f"An error occurred: {error}")
42
43
44    # once everything is collected, store it to the project id; directly via DB or via API?
45    print(mail_data)
```

get credentials

authorize and build API

fetch data and convert  
into target format