Lessons learned from Eurostat's Deduplication Challenge

Yves-Laurent Benichou, Insee
Antoine Palazzolo, Insee

Trusted Smart Statistics – Web Intelligence Network

Grant Agreement: 101035829

Web Intelligence Network

Funded by the European Union

Who are we?

- **Yves-Laurent Benichou**
  - Senior Data Scientist at *Insee*
- Team *Spub.Fr*
  - Collaboration between *Insee* and *Dares*
  - Ranked 5th on accuracy

- **Antoine Palazzolo**
  - Data Scientist in the innovation lab of *Insee*
  - Member of the WP4 of the WIN
- Team *Nins*
  - 2nd prize for reproducibility
  - Ranked 4th on accuracy

Plan of the webinar

- **I. The challenge**
  - *Context of the competition*
  - *Presentation of the task*
- **II. Deduplication techniques**
  - *Data processing*
  - *MinHash-based solution*
  - *Embeddings-based solutions*
  - *Putting it all together*
- **III. Examples of coding best practices**
  - *The Onyxia Datalab*
  - *The Kedro framework*

Funded by the European Union

I. Eurostat's Deduplication Challenge

Deduplication of online job advertisements (OJA)

Web Intelligence
Network

Funded by
the European Union

What is the context of this challenge?

- Series of competitions by the ***European Statistics Awards Program***
  - 2 rounds of nowcasting challenges
  - 1 round for web intelligence: the deduplication challenge

- Open to all, with the goal of "*unveiling innovative methodologies and valuable data resources that could improve the production of European statistics*"

- The deduplication challenge:
  - Started in December 2022
  - Ended in April 2023

Funded by the European Union

What is the goal of this challenge?

- 200 millions job advertisements scraped from the Web and classified since July 2018 for statistical purposes *(**OJA** project)*

- However, the same offer can be posted:
  - On different websites (LinkedIn, Indeed, own website, etc.)
  - In different languages or with different phrasings
  - By different companies (ex: recruitments groups)
  - …

- Need to **deduplicate** them in order to **publish unbiased statistics**!

What do the job offers look like?

- 112k anonymized online job advertisements (OJA), retrieved from around 400 websites, with:
  - A job title
  - A description of the job
  - A location, extracted automatically from the job description
  - A company name, extracted from the description as well
  - The advertisement retrieval date (by the bots of the WIH)

What kinds of duplicates were to identify?

- **Full duplicates**
  - Same job title and description
- **Semantic duplicates**
  - Same job position advertised
  - Same content in terms of the job characteristics (ex: required skills or education) but expressed differently
- **Temporal duplicates**
  - Semantic duplicates with varying advertisement retrieval dates
- **Partial duplicates**
  - Describe the same job position but do not contain the same characteristics

Now that you know all about the challenge, let's dive in!

- How to approach the issue? What methods exist?
  - Although studied through the prism of job advertisements, what follows **can be generalized** to broader deduplication problems

- Main difficulties:
  - **Multilingual** dataset
  - Distinction between **the different kinds of duplicates**
    - Ex: After what threshold is a semantic duplicate considered as partial?
  - Limited trials to check performances
    - No annotated test set is furnished
  - F1-score chosen as the challenge metric
    - Meaning that both precision and recall need to be optimized

Funded by the European Union

II. Deduplication techniques

How to identify duplicates in multilingual datasets?

Web Intelligence Network

Funded by the European Union

Data processing

Web Intelligence
Network

Funded by
the European Union

First off, let's do some cleaning!

- Web scraped data is sometimes messy:
  - Remaining HTML tags or special characters such as "\n"
  - Trailing or double spaces
  - …
- In NLP, some preprocessing operations are often necessary:
  - Standardization of texts
    - Removing accents and punctuation
    - Lowercase text
  - Lemmatization of texts : breaking down words to their root meanings
    - Went → Be / Better → Good
  - Removing stop words
    - Ex: the, but, and, …

Regex operations for a first data cleaning

Via regex we can remove:

- All html tags and \n, \r, …
- Special white spaces
- Punctuation
- Extra spaces (possibly created by previous operations)

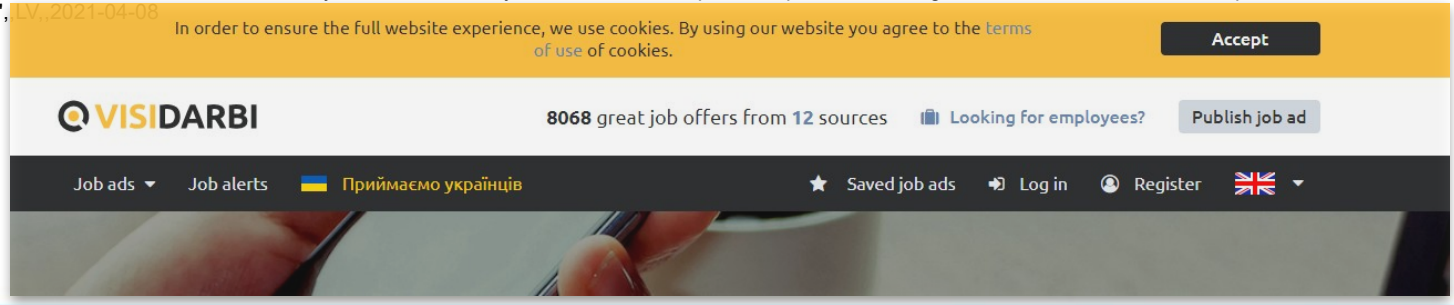
Depending on the approach, the preprocessing can stop here or include for instance lemmatization and stop words removal if we know the language of the offers.

```
eol_regex = re.compile(r'\r|\n')
multispace_regex = re.compile(r'\s\s+')
html_regex = re.compile(r'<[^<]+?>')
white_regex = re.compile(r'\xa0')
punctuation_regex = re.compile(r'[^\w\s]')
underscore_regex = re.compile(r'_')
```

More uses for Regex

- Identifying and removing scraped advertisements that actually do not refer to a job offer
  - In our dataset, 3198 instances like this one, in different languages:

111761,Technologist | VisiDarbi.lv,"We use cookies to ensure a full website experience. By using our website, you agree to the terms of use of cookies. Agree Looking for employees? Publish vacancy 1 Saved vacancies Log in Register Vacancies All vacancies Work in Riga Work in Vidzeme Work in Zemgale Work in Kurzeme Work in Latgale Work abroad Job ads with salary Vacancies by company Vacancies by e-mail Blog Advice for job seekers Remote work Great 11541 job opportunities from 15 sources Looking for employees? Publish a vacancy Vacancies All vacancies Work in Riga Work in Vidzeme Work in Zemgale Work in Kurzeme Work in Latgale Work abroad Job ads with salary Vacancies by companies Vacancies in e-mail Blog Tips for job seekers Remote work 1 Saved vacancies Enter Register LV RU EN Save Print Share: Send Send! Similar vacancies Save Print Share: Send Send! Up About us Advertising Terms of use for job seekers Contacts CV-Online Latvia Lithuania Estonia Contacts: E-mail: ********** Phone: ****** *** ***** Developed and maintained Log in to your profile Login to the system failed! Please check if the e-mail and password are correct. E-mail Password Forgot password | Register With social networks Registration for a job seeker Registration for an employer Register with social networks: First name, last name Email Phone Password Repeat password I would like to receive Visidarbi.lv news in my e-mail I have read and agree to the Terms of Use Register Thank you! Registration is successful. A confirmation link was sent to the specified e-mail address. Company name. Registration number. Address. First name, last name. Phone. Registration is successful. A confirmation link was sent to the specified e-mail address Benefits of a registered job seeker Ordering new relevant vacancies by e-mail Search history of advertisements Reviewing saved advertisements Adding a CV to a profile Managing applications Benefits of a registered employer Quick and convenient purchase of services Publishing and managing advertisements Processing of received applications in the Job system creating a donor profile Technical support and consultations Password renewal E-mail Thank you! Please check your e-mail and complete the password change Send e-mail Your e-mail Recipient's e-mail Message Thank you! Close Close",,LV,,2021-04-08



Data processing: computing more information

- **Detect language**, with *fastText* language detection library (here)
  - 2 columns added: “lang” and “score”, for the confidence of the language
  - Here low scores or errors do not matter much, as similar advertisements will be detected in the same language anyway, whether it is the right one or not

- Compute **lengths** of job titles and descriptions
  - Also compute their **MinHashes**, see later

- **Named Entity Recognition (NER)**
  - Using HuggingFace models specialized in multilingual datasets such as Davlan/distilbert-base-multilingual-cased-ner-hrl
  - Focus on ORG and LOC objects to try and detect company names and locations within the job titles and descriptions

Data processing: always possible to go one step further

- **Filtering out the most frequent words** that appear per language
  - They will likely not bring any meaningful information
  - Ex: job, application, profile, etc.

- **Filtering out *poorly described* offers**
  - Some descriptions are generic per company and do not necessarily refer to the same offers and thus may need to be treated differently
  - Some offers that contain too few non-empty fields may need special treatment

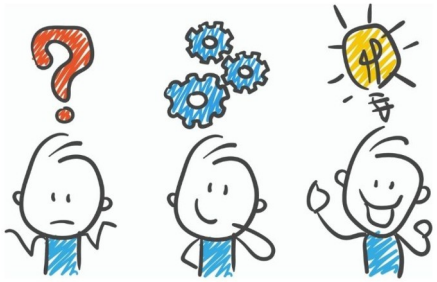
- **Creating splits** of the advertisements' contents for faster processing
  - **Multiprocessing** is already key in all of our operations because of the volume of the data

Identifying duplicates

Web Intelligence
Network

Funded by
the European Union

Finding duplicates: starting intuitively

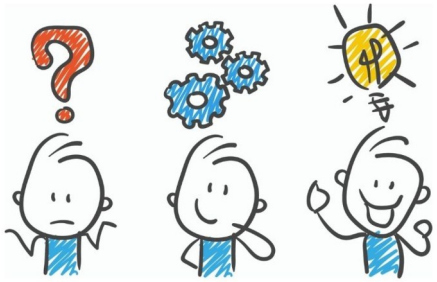


- Before jumping into big models, why not start the easy way?
  - Allows to at least spell the situation out without too much computing

- Try and find **matches based on subsets of fields**:
  - All columns for full duplicates
  - For other kinds of duplicates:
    - Title & description
    - Title, company name & location
    - Company name, location & retrieval date for companies known to be international from the available data (catches many multilingual duplicates)
    - ORG and LOC obtained from NER computation earlier

- The obtained pairs can then be re-checked later to avoid mistakes

Finding duplicates: starting intuitively

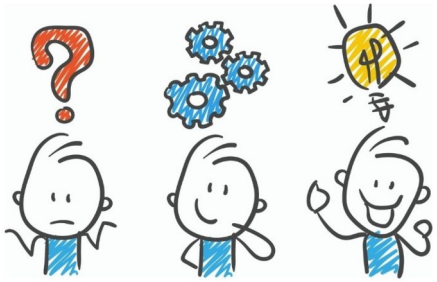


- Once exact matches are found on subsets of fields, can we look for close matches?
  - If two offers are exactly the same besides 1 character, can we pair them?

- In order to do that, we need to measure ***distances*** between offers
  - How to measure similarity between two texts?
  - Once distances are computed, we can also catch pairs of offers whose similarity is above a given threshold, with possible extra filters, such as limiting the difference of days between the retrieval dates of the offers

- We used two metrics:
  - The **Jaccard distance** & the **Jaro-Winkler similarity**

Finding duplicates: the Jaccard distance



- Out of all the words present in the union of two texts, how many are also included in their intersection?
  - How close are the vocabularies of the two texts we compare?

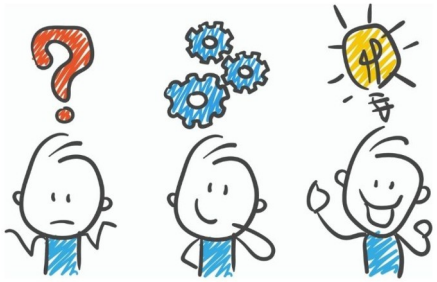
- Several limitations: word frequencies not included, very sensitive to synonyms or to multiple languages

$$J(doc_1, doc_2) = \frac{\{'data', 'is', 'the', 'new', 'oil', 'of', 'digital', 'economy'\} \cap \{'data', 'is', 'a', 'new', 'oil'\}}{\{'data', 'is', 'the', 'new', 'oil', 'of', 'digital', 'economy'\} \cup \{'data', 'is', 'a', 'new', 'oil'\}}$$

$$= \frac{\{'data', 'is', 'new', 'oil'\}}{\{'data', 'a', 'of', 'is', 'economy', 'the', 'new', 'digital', 'oil'\}}$$

$$= \frac{4}{9} = 0.444$$

Finding duplicates: bringing out the big guns



- Intuitive approaches are generally not enough, so we offer two other main categories of solutions.

- **MinHash-based solutions**
  - Fast to implement, very efficient on ‘big’ datasets, but will not help finding multilingual pairs
  - Global idea: check similarity on shingles of characters (words, ngrams of words & characters)
  - For this specific use case, we use shingles of 4 characters
  - **Record Linkage Toolkit** to put it all together

- **Embedding-based solutions**
  - Different models can be used to produce embeddings of the advertisements as vectors: **TF-IDF**, **Transformers** or other pre-trained models
  - We can then compare the embeddings through **cosine similarity**

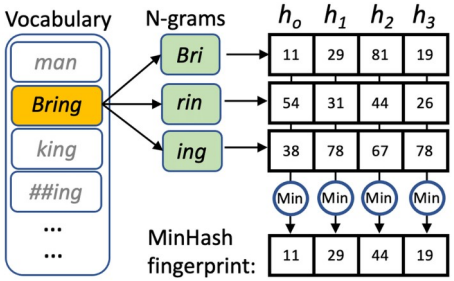
MinHash algorithm

More about MinHash here

Web Intelligence Network

Funded by the European Union

The MinHash algorithm

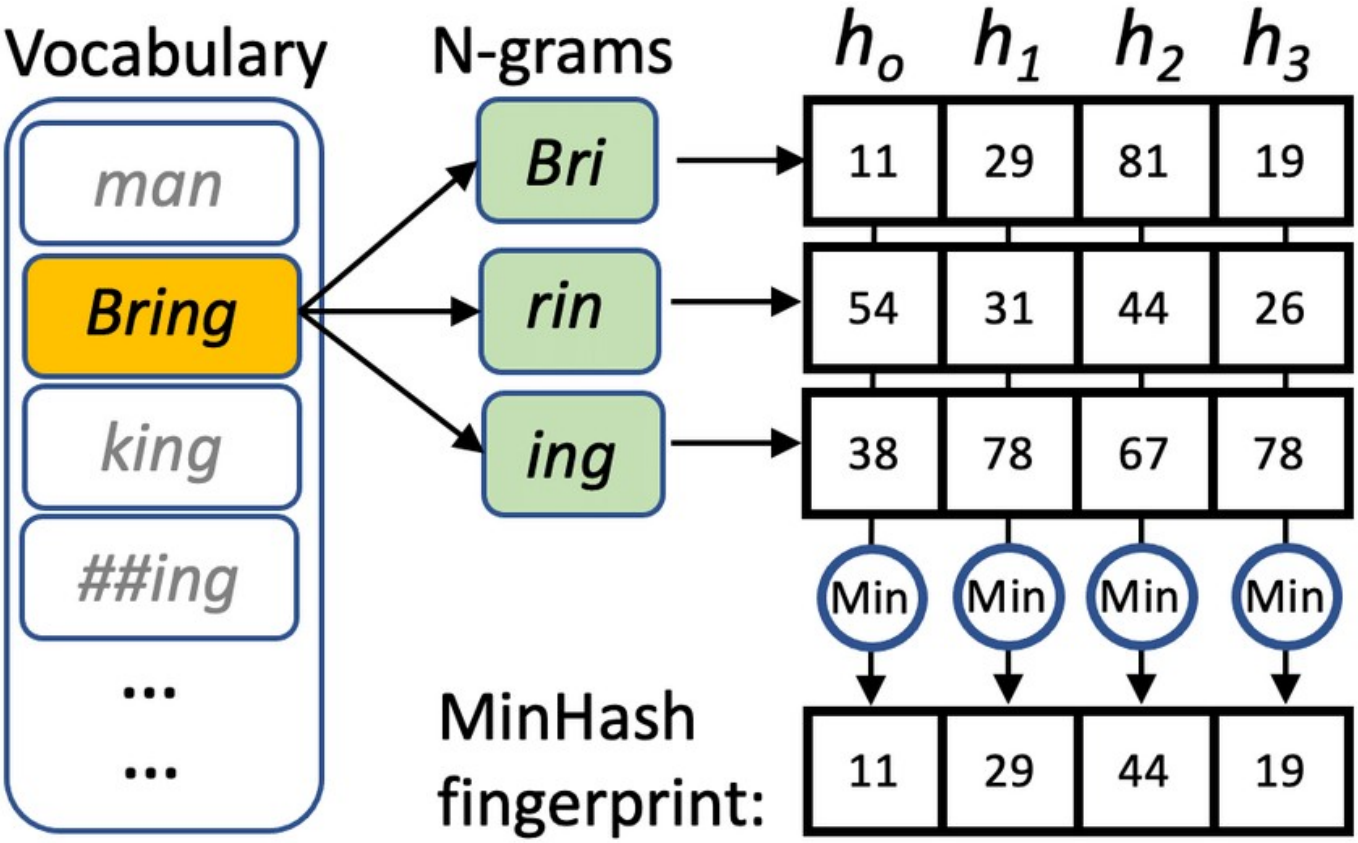


- **MinHash is a probabilistic algorithm used to estimate similarity between two sets**
  - Invented by Andrei Broder (1997) and initially used in the AltaVista search engine to detect duplicate web pages and eliminate them from search results (more here)

- It works by representing a set of data as a '*signature*'.
  - The signature is a hash value that capture the properties of the data set
- We apply several non-cryptographic hash functions and select the minimum hash value for that set
- **The probability of two sets having the same signature is proportional to their Jaccard similarity**

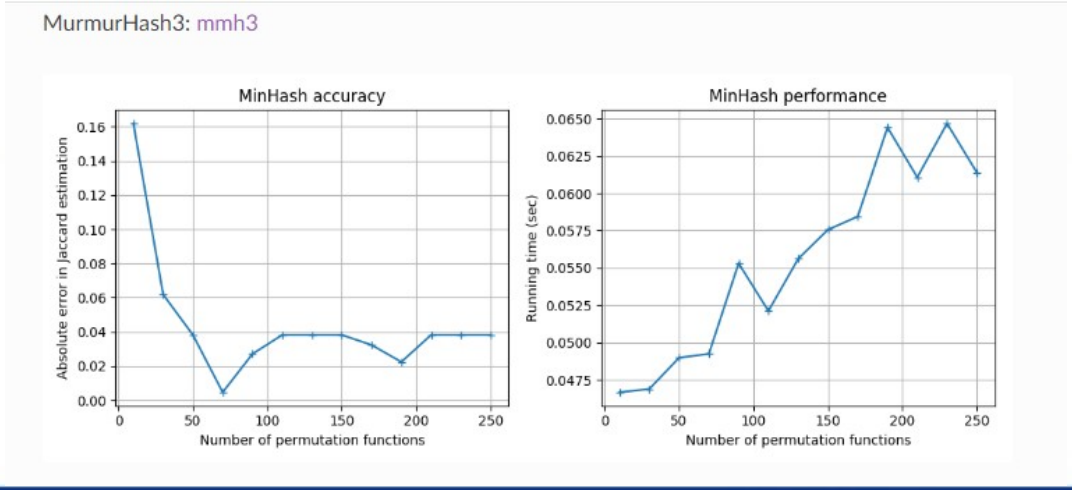
- MinHash is an efficient and robust technique that is well-suited to large and high dimensional data sets
  - Common Issue : high estimation error when set sizes differ by a lot

The MinHash algorithm: Zoom



The MinHash algorithm: in Python

- To compute MinHashes on titles and descriptions, we use the datasketch minhash library
- We choose *Murmur3 hash function* from the mmh3 lib



The Record Linkage toolkit

RecordLinkage

- An **open-source Python library for linking records in data sources**, available here
- Offers a range of methods for linking records, including deterministic matching, probabilistic matching, and machine learning-based approaches
- We also integrate a vectorized function to compute Jaccard distance on MinHashes

Funded by the European Union

The MinHash algorithm: putting it all together!

- For deduplication tasks such as ours, we use Record Linkage between the dataset and itself

- Record Linkage first computes ‘all’ candidate pairs, but without any filter on the features it would represent more than 12 000 000 000 pairs here!!
  - So we decide to filter on the **detected language**
  - We also cut the main dataset into 5 subsets, based on the description lengths ({]0, 700], [400,1300], [1000,2000], [1700, 3000], [2600, +[})
- Record Linkage will then compute:
  - The MinHash similarity for job titles and descriptions
  - Jaro-Winkler distance on company names and locations
  - Of course exact matches on all fields

Embedding-based approaches

Web Intelligence
Network

Funded by
the European Union

What makes the task difficult?

- Identifying job offers looking similarly is one thing, but how about:
  - Synonyms and different formulations
  - Advertisements available in several languages
  - Partial duplicates with very different description lengths
- We need to represent the offers based on the **meaning of their content**, not based on their form
- We want to pair duplicates based on the **content that discriminates them from other advertisements**, such as characteristic words
- A solution: **representing texts with vectors** instead!

TF-IDF

TF-IDF is a measure of originality of a word by comparing the number of times a word appears in a doc with the number of docs the word appears in.

$$\text{TF-IDF} = TF(t, d) \times IDF(t)$$

Term frequency

Number of times term t appears in a doc, d

Inverse document frequency

$$\log \frac{1 + n}{1 + df(d, t)} + 1$$

n ← # of documents

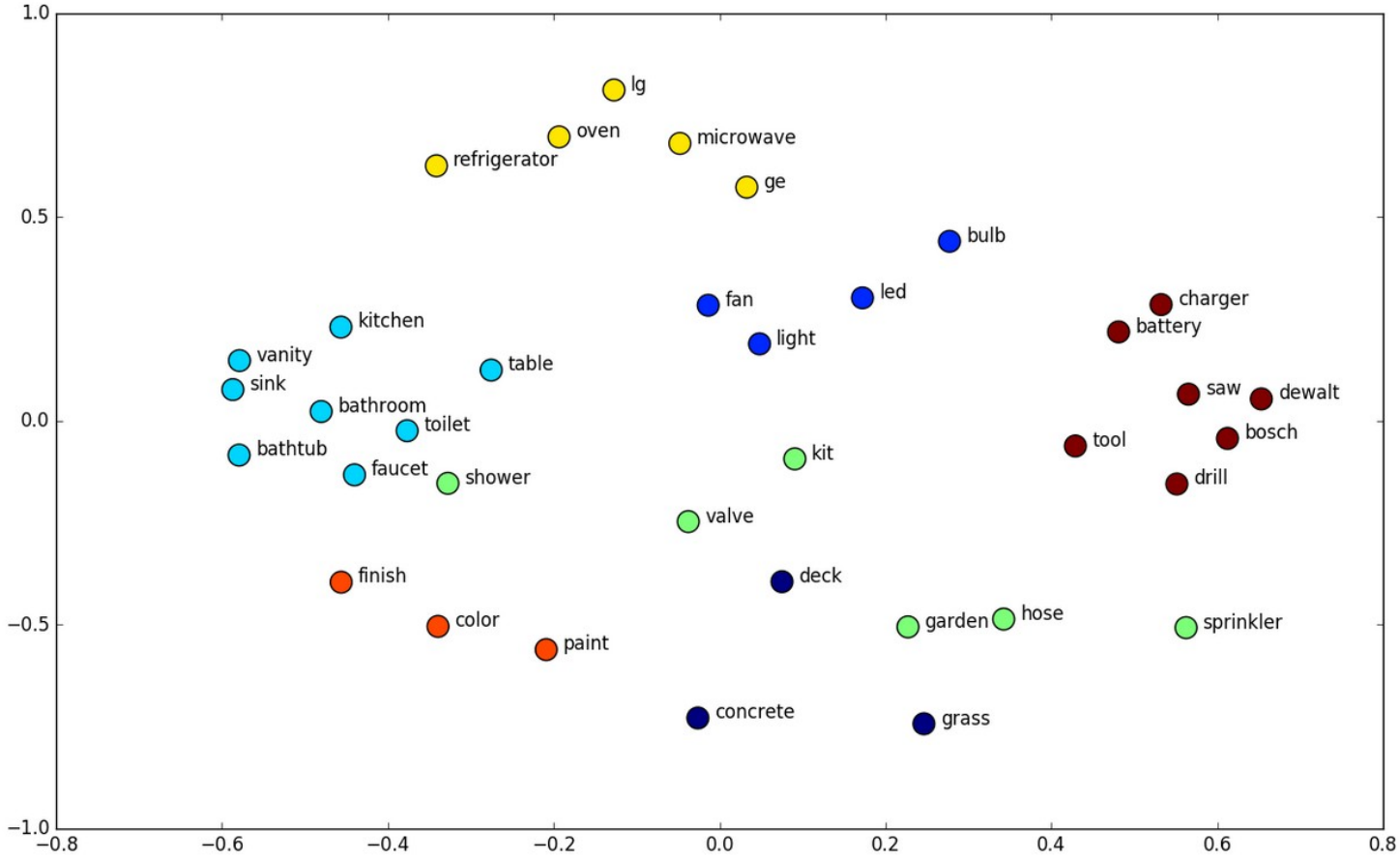
df(d, t) ↖ Document frequency of the term t

Source

TF-IDF as one way to tokenize the data

- Each dimension of the embedding corresponds to a word
  - For a given job offer, the value for dimension j will be the TF-IDF score of the word j in the advertisement
- Embeddings of very high (too high) dimension: we need to **manually reduce the dimension** without losing too much information, through:
  - Truncated singular value decomposition (**truncated SVD**)
  - Or Principal Components Analysis (**PCA**)
- Another (efficient) way to compare ads similarities numerically, but:
  - Still sensitive to synonyms or different languages
  - Does not rely on words meanings

Meaning-based embeddings



Meaning-based embeddings

- Using **pre-trained (on multilingual corpus) models** to embed our job advertisements
  - Retraining or finetuning big models on our own data was too costly in view of the possible gains, but it may be an idea
- The goal is to embed our offers in a high-dimensional space (ex: 100) where the vectors are as close as the contents' meanings are similar
  - The better the model is, the more relevant the embeddings will be
- Trade-off to find for the dimension of the embeddings

Funded by the European Union

What models are we talking about?

- **Transformers** pre-trained on multilingual data:
  - Multilingual BERT
  - XLM RoBERTa
  - Distiluse base multilingual (from sentence-transformers)

- The last one was the preferred option: **faster** and **lighter**
  - In our case it also happened to lead to better results

- Some even bigger models could be more relevant (ex: **LLMs**)

- Many resources on Transformers are available online
  - Example of a great explicative video here

Funded by the European Union

Pros and cons of transformers' embeddings

- **Advantages**:
  - The words are not analyzed independently, but the job advertisement is considered as a whole, context is taken into account
  - The embedding is resistant to paraphrasis as the meaning is caught

- **Disadvantages**:
  - If the texts are too long, all the information may not be picked up by the embedding
  - Not very sensitive to differences between named entities, while this kind of information is crucial for our use case
  - As our corpus is quite specific (to the corporate lexical field), a specialization of the models can be useful for more relevant embedding spaces

What do we do now with those embeddings?

- **Cosine similarity** as the similarity measure between two embeddings
  - It is the cosine of the angle between the two vectors
  - If the cosine similarity between two job offers exceeds a **given threshold** (ex: 0.99), we will consider than they are duplicates candidates

- 112k jobs offers leads to 112k*112k computations, which is A LOT
  - We thus **limit transformers approaches to *international companies***, i.e. known to publish offers in several countries or several languages
  - We do it by **chunks** to compute smaller **cosine similarity matrixes** and save storage space
  - We then **parallelize the comparisons** per chunk and save processing time

- This method leads to a **high recall**, but may lead to **poor precision**

Deduplication: putting it all together!

Web Intelligence
Network

Funded by
the European Union

Additional conditions to be considered as duplicates

- Previous approaches can lead to (too) **many eligible pairs**. Extra filters may be necessary to:
  - Limit ourselves to the actual duplicates
  - **Distinguish between the different types**: semantic, partial, temporal

- **Possible filters**:
  - Duration between the retrieval dates
  - Differences in lengths descriptions
  - Differences in the country ID
  - Levenshtein (or Jaro-Winkler) distance between company names or locations
  - Comparison between named entities extracted from NER

Funded by the European Union

Is the relation “A is duplicate of B” transitive?

- If A and B are semantic duplicates:
  - All semantic duplicates of A are supposed to be semantic duplicates of B
- If A and B are partial duplicates:
  - All semantic duplicates of A are supposed to be partial duplicates of B

- We can represent our job offers with a **non-oriented graph**:
  - Each advertisement is a node
  - An edge represents the “is semantic duplicate” relation
- **If our relation is transitive (as it should), all connected components obtained before can be converted into cliques**
  - This should improve our recall even more

Is the relation “A is duplicate of B” transitive?

- In practice,
  - Using the **semantic transitivity property** led to better results
  - Using the **transitivity on partial duplicates** led to worse results, as our precision on this kind of duplicates was already poor, and decreased even more when adding more edges within our connected components

- In conclusion, **relying on transitivity is efficient when the precision is high and the recall improvable**
  - We thus raised our similarity thresholds based on that for our previous approaches

Specificities and limits of the approach

- **Set of stackable approaches, more or less simple but overall fast to execute**, in order to identify eligible pairs
  - Followed by **additional layers to discard false positives** and distinguish the types of duplicates

- How could we improve the results?
  - If more time and resources available, **reduce the optimizations and proxies** in order to capture more duplicates
  - **Finetune the transformers** based on our corpus
  - Spend more time to **finetune the parameters** of the different models, for instance with a **grid search**

Funded by the European Union

III. Examples of coding best practices

Ideas to achieve reproducibility

Web Intelligence
Network

Funded by
the European Union

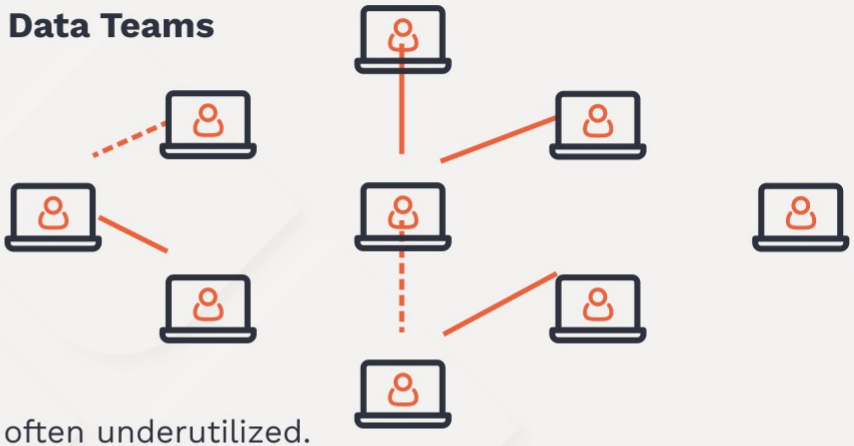
The *Onyxia Datalab*

Web Intelligence
Network

Funded by
the European Union

Onyxia

Overcoming the Constraints of Individual Computing in Data Teams

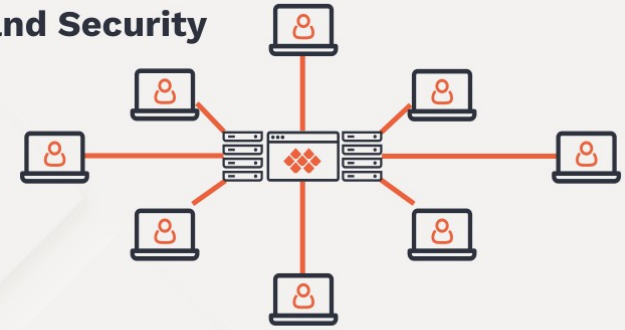


1. **Cost Efficiency**: Individual workstations are expensive and often underutilized.
2. **Computational Capacity**: Physical limitations on RAM/GPU in a single machine.
3. **Scalability**: Traditional computers lack the ability to scale on demand.
4. **Reproducibility**: Challenges in transitioning from experimentation to production due to difficulty in replicating unique software environments for accurate results.
5. **Administrative Restrictions**: Security measures restrict software installations, causing delays and additional administrative tasks.
6. **Data Security**: Downloading sensitive data on internet-exposed machines can spread and risk the data.

Funded by the European Union

Onyxia

The Power of Cloud-Based Data Platforms: Unmatched Flexibility and Security

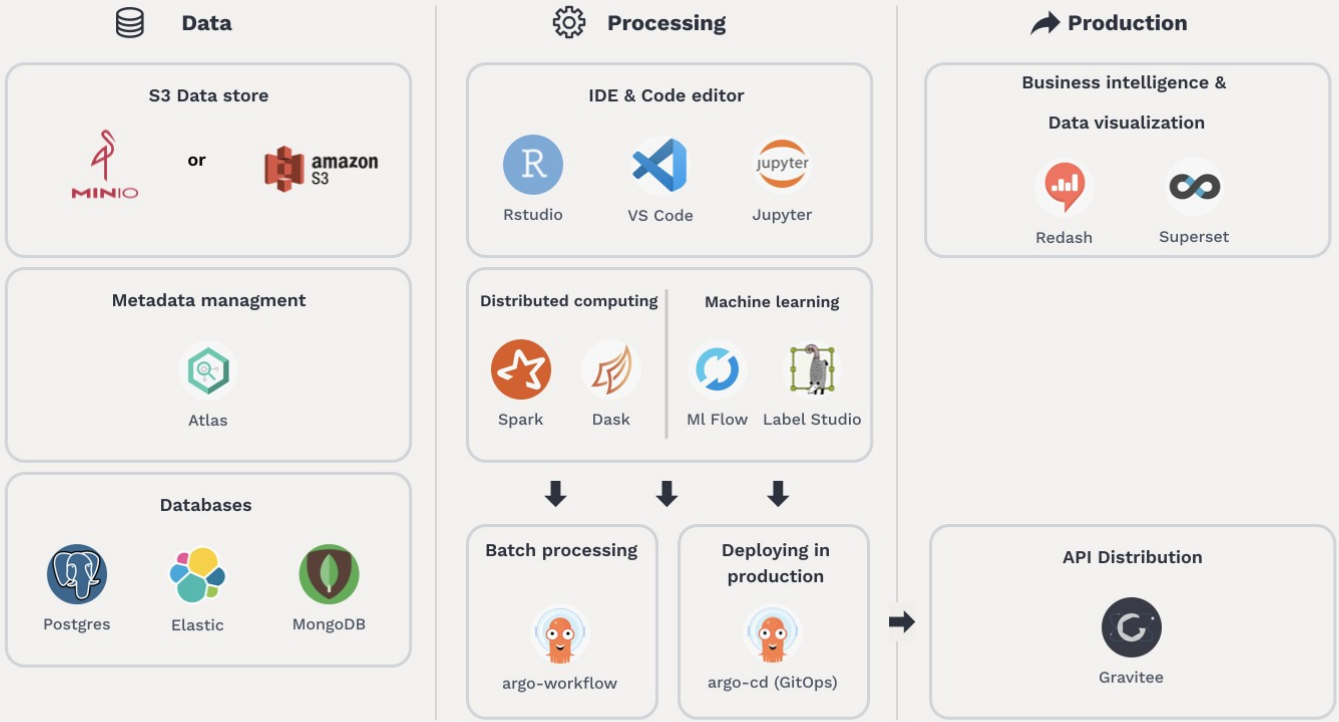


1. **Pooling Resources**: Cloud-based data platforms allow data scientists to access and release vast resources on-demand.
2. **Cost and Capacity Efficiency**: Shared resources lead to cost savings and access to levels of power unthinkable with personal workstations.
3. **Horizontal Scalability**: Additional processes can be initiated on-demand, without halting current operations.
4. **Containerization with Kubernetes**: Docker-based environments allow for quick, easy, and replicable setup and teardown of working environments.
5. **Enhanced Data Security**: Sensitive data can be processed in a secure environment, isolated from direct internet exposure.

Onyxia

The state of the art in data science technologies

The ideal technology stack for a data scientist



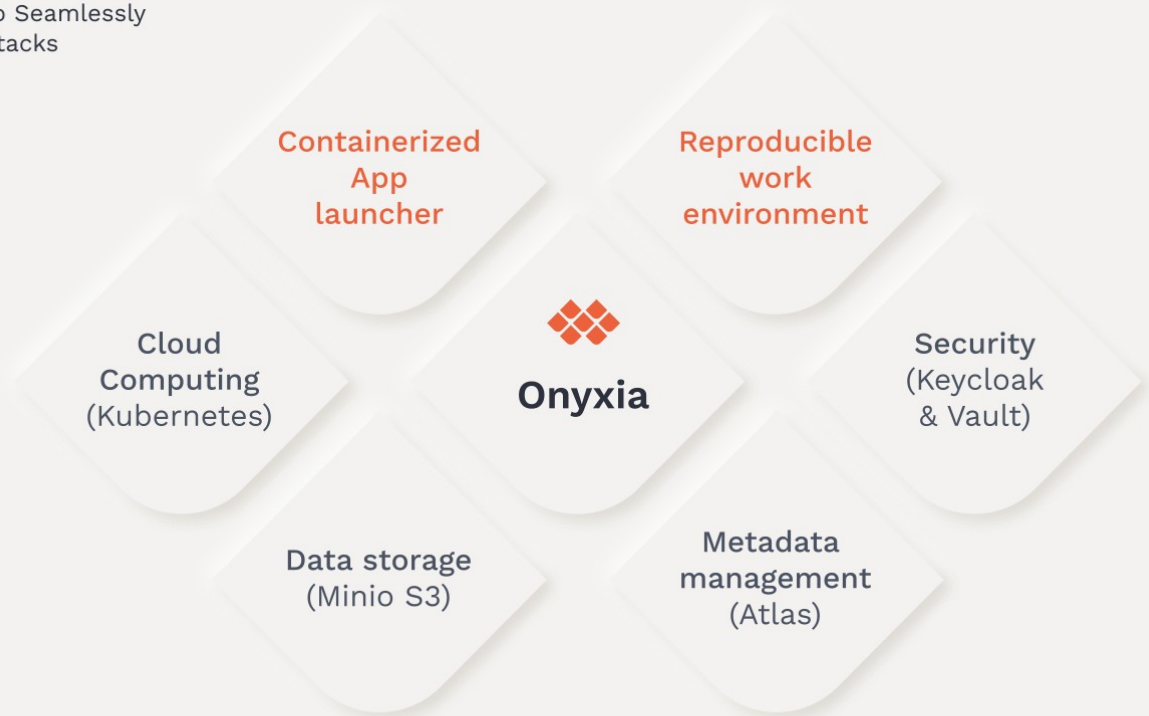
Web Intelligence Network

Funded by the European Union

Onyxia

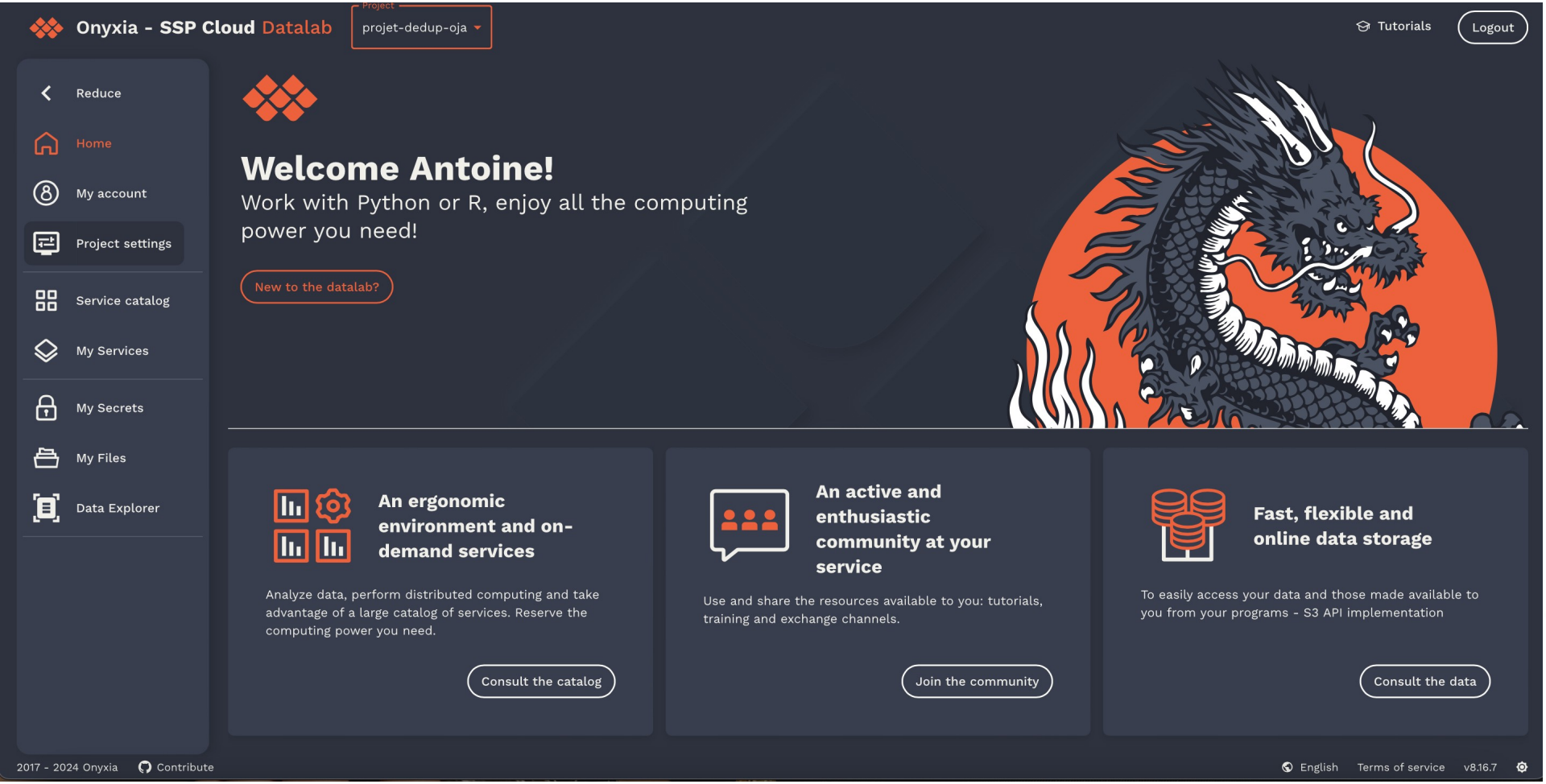
Bridging Technological Complexity: The Role of Onyxia

Democratizing Technology: How Onyxia Enables Non-IT Experts to Seamlessly Operate Advanced Tech Stacks



Web Intelligence Network

Funded by the European Union



Link here

Web Intelligence Network

Funded by the European Union

The ***Kedro*** framework

Web Intelligence
Network

Funded by
the European Union

Kedro

An open-source Python toolbox that applies software engineering principles to data science code, making it easier to transition from prototype to production.

FOUNDED IN

2017

STATUS

LF AI & DATA
INCUBATION PROJECT

Benefits

Reduces the time spent rewriting data science experiments so that they are fit for production.

Encourage **harmonious team collaboration** and improve productivity.

Upskills all collaborators on how to apply software engineering principles to data science code.

+8,200
GITHUB STARS

+467,000
MONTHLY DOWNLOADS

+15,000,000
PIPELINE RUNS IN 2022

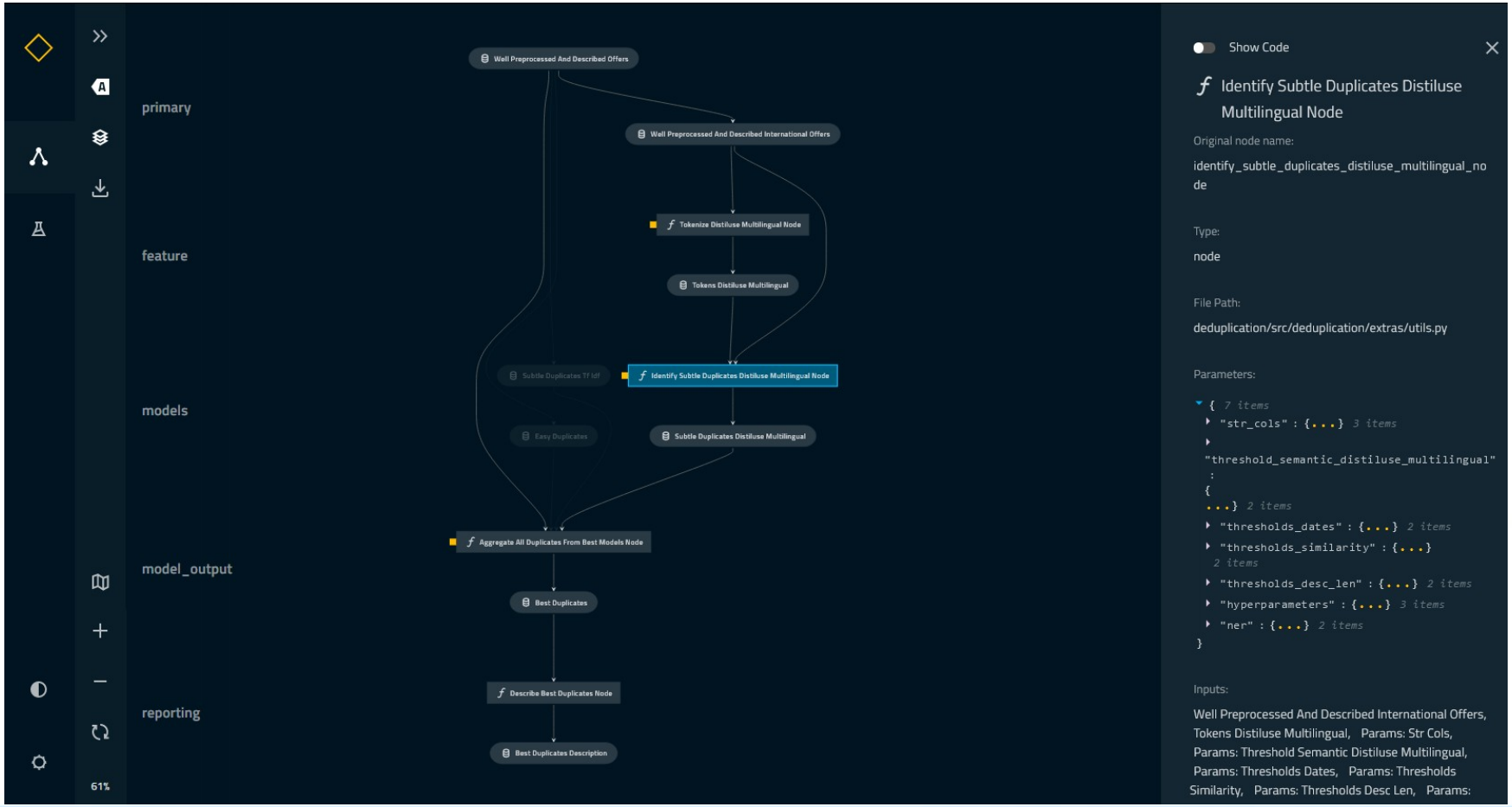
Web Intelligence Network

Funded by the European Union

What is Kedro?

- Essentially, a way to structure code to optimize data science projects
  - Each function is a **node**
  - Nodes and objects articulate themselves through **pipelines**
- Huge help to have **reproducible, maintainable and modular code**
- Several useful **features**
  - Pipeline visualization (demo here)
  - Data catalog
  - Various integrations

Pipeline visualization with Kedro



Other possible frameworks to use

- In Python:
  - Snakemake
  - Luigi
  - Metaflow
  - Ploomber
  - …
- In R:
  - Targets

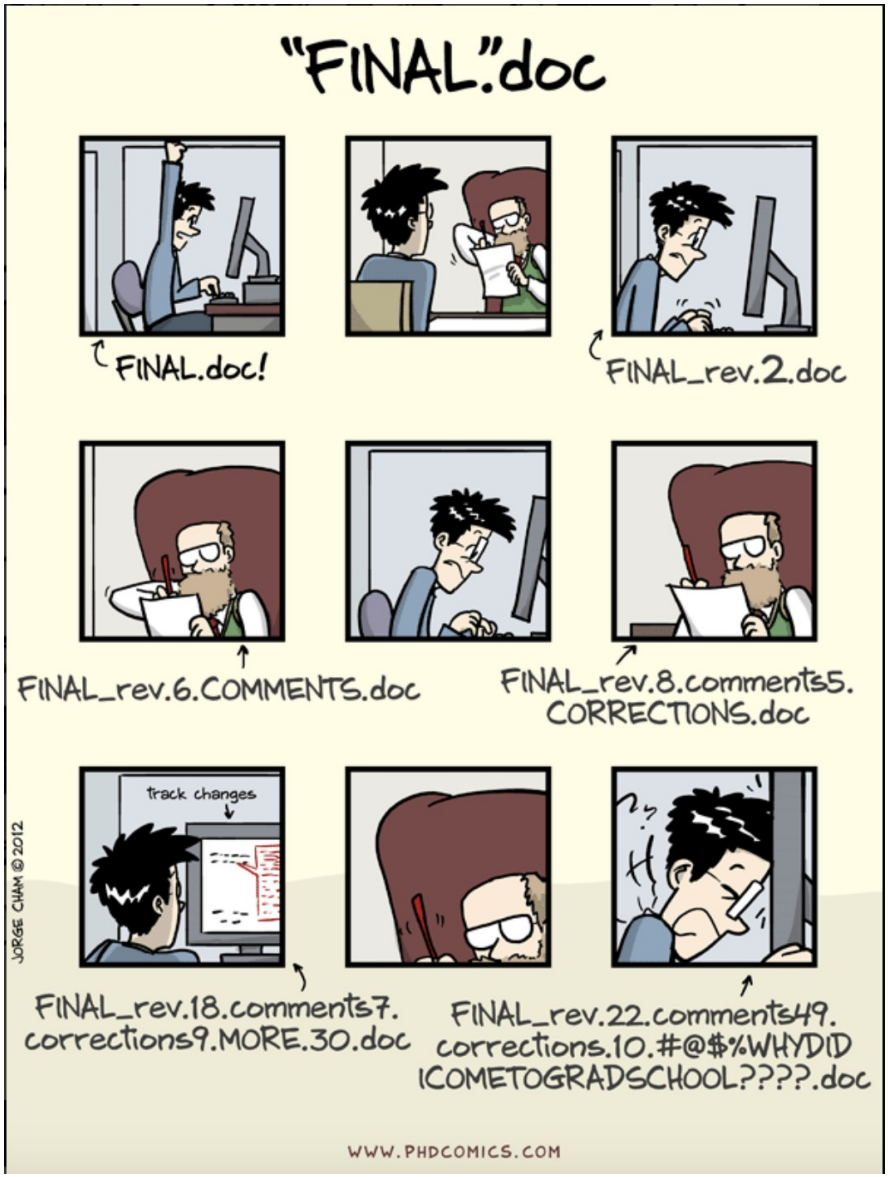
Funded by the European Union

Git

Web Intelligence
Network

Funded by
the European Union

Why using *git*?



Essential part for reproducibility: ***git***

- A distributed **version control** system
  - Store, archive and share code
  - Work more efficiently within teams or with your future self

- **Github** as a platform to share ***open source*** code
  - Team Spub.Fr repository available here
  - Team Nins' repository available here
  - Team Nins' reproducibility approach description available here
  - Other winners' repos and solutions available here (and congrats to them!)

Thank you for your attention!

Are there any questions?

Web Intelligence
Network

Funded by
the European Union