

Foundations of Trustworthy AI – Integrating Reasoning, Learning and Optimization TAILOR Grant Agreement Number 952215

Integration of OpenML with AI4EU v.2 Progress Report

Document type (nature)	Report			
Deliverable No	11.4			
Work package number(s)	11			
Date	11 October 2024			
Responsible Beneficiary	#12, TU/e			
Author(s)	Joaquin Vanschoren			
Publicity level	Public			
Short description	OpenML is an open source platform for sharing machine learning datasets, algorithms, and models. We plan to interface OpenML with the AI on-demand platform, so that OpenML resources are accessible via AI on-demand and can be used in all AI on-demand services. (See also Appendix 1)			

History						
Revision	Date	Modification	Author			
version 1	2024-10-11	-	Joaquin Vanschoren			

Document Review					
Reviewer	Partner ID / Acronym	Date of report approval			
Peter Flach	UNIVBRIS, #16	09/10/2024			
Fredrik Heintz	LIU, #1	10/10/2024			



Table of Contents

1. Summary of the report	3
OpenML - AloD integration	3
AloD chatbot (Talk2AloD)	4
OpenML chatbot	4
Research dissemination assistant (QuickRePost)	5
2. Approach	6
OpenML-AloD integration	6
AloD Connectors	6
Dataset access	7
Implementation	7
Using OpenML datasets	9
Chatbot development	9
Phase 1: Research and selection of NLP approaches	10
Phase 2: Implementation of the service and AloD integration	16
Research dissemination assistant	16
Phase 1: Research and selection of NLP approaches	17
Phase 2: Implementation of the service and AloD integration	20
3. Achieved results	21
OpenML-AloD integration	21
OpenML Chatbot	24
AloD Chatbot (Talk2AloD)	25
Research dissemination assistant	28
4. Future work	31
OpenML-AloD integration	31
OpenML Chatbot	31
AloD Chatbot	32
Research dissemination assistant	32
Appendix 1: Text of the original deliverable	33



1. Summary of the report

We have integrated OpenML with the Al on-demand (AloD)¹ platform, which makes all verified OpenML datasets (more than 5000) accessible in the AloD catalogue and services. Moreover, to ensure that these datasets are easy to use for a wide audience, we have created several LLM interfaces (chatbots) that allow people to discover and use these resources using natural language questions, and additional LLM dissemination tools that make it easy for researchers to disseminate their work (especially papers) to a wide audience, thus creating new incentives to share their work on the AloD platform.

We created two LLM chatbots (one for AloD and one for OpenML) that provide easy-to-use and powerful search capabilities. The reason for creating two chatbots is that AloD is still under active development, while OpenML already has a large user base (over 300k users) and extensive documentation. By creating a separate OpenML chatbot, we can explore more advanced use cases, such as training the chatbot on documentation to answer more advanced questions and generate code, and evaluate the chatbot implementation with a large user base. As such, the OpenML chatbot serves as a **technology pathfinder**: lessons learned can be easily transferred to the AloD chatbot as the platform matures.

The integration of OpenML with the AloD platform was part of the original TAILOR project. The Large Language Model (LLM) work was performed based on funding made available from the TAILOR Connectivity Fund and by collaborating with the HumaneAl project.

Below is an overview of the four main newly developed features and how they enhance the AloD platform.

OpenML - AloD integration

This integration was made possible by the AloD application programming interface (API), which was co-developed with TU/e engineers based on experience developing the OpenML platform. This API allows us to **exchange Al resources**, **especially datasets and models**, **between both platforms**. This is done through an *OpenML connector* inside the AloD platform that can automatically import OpenML datasets. This allows AloD users to:

- Discover and use 5750 OpenML datasets and 15862 OpenML ML models/pipelines in the AloD platform catalogue. New datasets and models added to OpenML will also be imported *automatically* to AloD
- Search these datasets and models in the AloD 'MyLibrary' service, and add them to their list of favourites
- Seamlessly use OpenML datasets in AloD services such as 'RAIL' and 'Al Builder' to train new models and run reproducible experiments.
- Build new services on top of the AloD platform that use these datasets and models.

¹ This deliverable originally mentioned the AI4EU platform, but this platform has been replaced by the AI on-demand (AloD) platform, developed in the AI4Europe project.



AloD chatbot (Talk2AloD)

The **AloD chatbot** (currently called Talk2AloD) will be available on the AloD website and will provide users an interface enabling them to get relevant information about the AloD platform, related services and resources available in the platform, through interaction in natural language. This addresses the fact that users may find it difficult to understand what the AloD platform and related services can provide them. The chatbot will be made available through the AloD Marketing portal and potentially through other services. It can answer questions based on the current information and resources in the AloD Marketing portal. It also provides advanced search capabilities that address some of the main challenges Al researchers face, such as finding the right Al resources (e.g. dataset, model) for their goals. The enhanced search utilises models specialised on semantic similarity modelling to retrieve relevant Al Assets (more specifically, their metadata) available in the AloD metadata catalogue based on user query provided in natural language.

The AloD chatbot currently supports:

- Guiding users through the website, and finding and aggregating website information
- Providing links to the AloD webpages that were used as sources
- A conversation memory that allows natural interactions with the user
- Help users find datasets, publications, machine learning models, experiments, educational resources available through mylibrary
- It supports function calling, e.g. fetching data from the AloD catalogue
- Answer complex queries requiring multiple function calls

Its functionality has been evaluated in two user studies, quantitatively (n=108) and qualitatively (n=12), and a technical evaluation comparing its functionality and energy efficiency to web agents (paper submitted to CHI2025). Talk2AloD was deployed on a server to conduct these studies.

OpenML chatbot

The **OpenML chatbot** offers the same capabilities as the AloD chatbot to OpenML users, but also goes further and includes capabilities that will only be available on AloD later. For instance, it can generate Python code for downloading datasets, training models, and running evaluations. This is possible because OpenML has extensive documentation. When AloD has similar documentation, we can bring the same capabilities to the AloD platform. The OpenML chatbot also serves as a pilot that allows testing on a wide user base (>300k users).

The OpenML chatbot additionally supports:

- Searching for useful datasets in natural language, with very specific requirements (e.g. 'datasets on biodiversity with more than 10.000 examples')
- Searching across multiple asset types at the same time (currently datasets and models)
- Adapting the search result interface depending on the user query. For instance, if a user asks for datasets of a certain size or with a certain number of features, this will be detected and used to filter the search results better and adapt the user interface (e.g. setting a filter in a table of results).



- Generating code examples on how to use OpenML datasets in machine learning libraries (e.g. 'how do I load the dataset X into PyTorch')
- Answering general questions about how to use and contribute to the OpenML platform

Research dissemination assistant (QuickRePost)

The LLM-based assistant for research dissemination (currently called QuickRePost) aims to assist researchers with disseminating their research (especially research papers) on various platforms like social media and posts. It's being implemented as a separate service that utilises language models, LLMs and text processing to generate a draft of a social media post that follows requirements set by the user (e.g. target audience, length and others). The service will be integrated with AloD authentication and authorization mechanism (Keycloak / EGI Checkin) - in this way, the service aims to contribute to onboarding of new users to the platform.

QuickRePost generates proposals for LinkedIn posts from the text of research papers. The service enables users to upload a PDF file (with a limit of maximum file size), select preferences, and generate a draft of the post. The service utilises two AI models - one model for text summarization, the second one for generating a social media post based on a summarised research paper and user requirements. The service is configurable and it can be deployed with a compatible HuggingFace model for both text summarization and generation. This improves the flexibility and reusability of the service as it can be deployed with different models based on available hardware.

We thoroughly compared the performance of different open-source and closed-source models (language models for text summarization and LLMs for generation) and their ability to generate social media posts that meet user requirements. We tested and fine-tuned the prompts used to instruct the LLMs to generate social media posts (human testing and automated testing) and created a mechanism to dynamically compose prompts based on user requirements.

We implemented and deployed the initial version of an application on a testing server, to be tested by the first users.



2. Approach

OpenML-AIoD integration

The OpenML-AloD integration is implemented² through an AloD *connector*. These connectors are a feature of the AloD architecture that provides access to Al resources that are physically hosted on existing infrastructure, without the need to move or duplicate these resources.

AloD Connectors

As illustrated in Fig.1, the AloD architecture in built on a set of 'root nodes' (in blue) which each have a copy the AloD metadata catalogue, a database with meta-data on all known Al resources, such as datasets, models, papers, educational resources, and many more. Most current AI resources are available on a rich ecosystem of AI platforms (in red), OpenML being one of them, and it doesn't make sense to duplicate these resources. The difficulty is that each platform makes its resources available in different ways, with different meta-data and with different levels of organisation. The AloD platform aims to offer a unified view on all these resources. The task of AloD connectors is to index AI resources available on existing platforms, provide uniform metadata for all of them, and store this in the AloD meta-data catalogue. They are continuously run on AloD root nodes and will poll platforms to learn about updates on the available resources. After helping to set up this architecture, we have implemented such a connector for OpenML. It will regularly poll OpenML to get all the latest datasets and make them available to the AloD platform and all its services, including third-party services anywhere that interface with the AioD API (leaf nodes).



Fig. 1: AloD physical architecture

² The code can be found on https://github.com/aiondemand/AIOD-rest-api/tree/develop/src/connectors



Dataset access

How OpenML datasets can be accessed from AloD is illustrated in more detail in Fig. 2. Central in this figure in the metadata catalogue. The OpenML connector will poll the OpenML platform via its API to get the latest updates on OpenML datasets, and post these updates to the metadata catalogue. Other similar connectors exist for HuggingFace and Zenodo. When AloD services need access to datasets, they consult the metadata catalogue. For any dataset in the catalogue, it will provide a reference to where the dataset is physically stored (i.e. on OpenML for OpenML datasets) and information on the dataset structure and format. The services can then download the datasets directly from OpenML and use them as they wish. For instance, they could evaluate models on hundreds of OpenML datasets at once, or they could analyse OpenML datasets for biases.



Fig. 2: AloD access to data and compute



Implementation

The integration occurs through the OpenML API (programming interface), which offers a Python client. This is illustrated in Fig.3. The client can be very easily installed (e.g. via PIP), and imported in any python code. Then, as illustrated, one can list all datasets, download the metadata by ID or name, and download the data itself directly into useful data structures.

```
import openml
# List all datasets and their properties
openml.datasets.list_datasets(output_format="dataframe")
# Get dataset by ID
dataset = openml.datasets.get_dataset(61)
# Get dataset by name
dataset = openml.datasets.get_dataset('Fashion-MNIST')
# Get the data itself as a dataframe (or otherwise)
X, y, _, _ = dataset.get_data(dataset_format="dataframe")
```

Fig. 3: OpenML Python interface

When the dataset meta-data is fetched from OpenML, the connector will store it in the metadata catalogue via the AloD REST API.³ The REST API is illustrated in Fig. 4a and 4b. One can list, download, and upload datasets, even directly from their original platforms. We zoom in on the 'PUT' method. Here, a fraction of the metadata is shown that must be provided to register a new dataset, such as the name, origin, version, data, author, licence, and application areas.⁴

datasets	^
GET /datasets/v1 List Datasets	\sim
POST /datasets/v1 Dataset	
GET /counts/datasets/v1 Count Of Datasets	\sim
GET /datasets/v1/{identifier} Dataset	\sim
PUT /datasets/v1/{identifier} Dataset	â V
DELETE /datasets/v1/{identifier} Dataset	i ~
GET /platforms/{platform}/datasets/v1 List Datasets	\sim
GET /platforms/{platform}/datasets/v1/{identifier} Dataset	\sim
GET /datasets/v1/{identifier}/content Dataset	\sim
GET /datasets/v1/{identifier}/content/{distribution_idx} Dataset	\sim

Fig. 4a: AloD REST API: endpoints for listing, retrieving, editing, and deleting datasets.

³ As of August 2024, AloD also has a Python interface. Hence, it would make sense to update the OpenML connector to use that instead of calling the REST API directly. This will simplify maintenance. ⁴ For the complete documentation, please see <u>https://api.aiod.eu/docs</u>



PUT	/datasets/v1/{identifier} Dataset
Update ar	n existing dataset.
Paramete	ers
Name	Description
identifie integer (path)	r * required identifier
Request I	body ^{required}
Example Va	alue Schema
<pre>{ "platf "platf "platf "name_ "same_ "same_ "same_ "sis_acd "versi "issn" "measuu "tempo "aiod_ "aid "sta }, "alter "ali "ali "fra "Voi "Dis], "citat </pre>	<pre>Form": "example", form": "example", form_resource_identifier": "1", form_resource_identifier": "1", form.resource_identifier": "1, published": "2022-01-01T15:15:00.000", ass": "https://www.example.com/resource/this_resource", foressible_for_free": true, form: "1.1.0", form: "1.1.0", form: "1.1.0", form: "1.1.0", form: "1.1.0", form: "1.1.0", form: "1.1.0", form: "1.1.0", form: "20493630", form: "1.1.0", form: "20493630", form: "1.1.0", form: "1.1.</pre>

Fig. 4: AloD REST API: meta-data returned for specific datasets.

Using OpenML datasets

The user interaction diagram of the OpenML integration is shown in Fig. 5. As shown, any user can ask the AloD API to list datasets, which are then looked up in the catalogue database and returned to the user. These datasets can be anywhere, on the AloD platform itself, but also on OpenML, HuggingFace, Zenodo, or any other connected platform. When the user requests a specific OpenML dataset, the AloD API will fetch the information from the database, detect where it is hosted, download the metadata (or actual data) from the resource provider (i.e. OpenML), potentially combine it with other known metadata, and return it to the user.

Users will typically be unaware of this complexity as it is entirely hidden. They can access any OpenML dataset with a single line of code, or through a graphical interface or any other AloD service, without having to worry about where the data is hosted and how to discover and access it. These higher-level interfaces are discussed in section 3.







Fig. 5: UML diagram of the OpenML connector

Chatbot development

Within project Al4Europe, finding the relevant Al assets (e.g. models, datasets) was identified as one of the crucial and most challenging activities Al researchers do. Currently, the AloD provides a fulltext search for selected Al Assets through its REST API and in the service MyLibrary. Although fulltext search provides value to the end users, it's often the case that the users are not able to precisely formulate the search queries so that relevant assets are retrieved as expected.

In this project, we aim to enable users to search for relevant assets in a more descriptive way by combining semantic search and Retrieval Augmented Generation (RAG). Semantic search uses LLMs to enable users to formulate their queries loosely (e.g. semantic search should be robust enough to retrieve an asset containing images of "trees" even if the user's query contains the word "forest"). RAG first retrieves relevant information from a database using a query generated by the LLM. This information is then integrated into the LLM's query input, enabling it to generate more accurate and contextually relevant text.

We split the work on AloD enhanced search capabilities service into two phases:

- Phase 1: Research of relevant NLP approaches, testing and selection of models;
- Phase 2: Implementation of the service and its integration with the AloD platform;

Phase 1: Research and selection of NLP approaches

Recent advances in NLP enable us to select from a wide range of available open- and closed-source models that can be used to implement semantic search on top of the AloD platform. The challenging part is to select the right model and text processing procedure that result in best performance on particular data, as the form, content and distribution of the data, as well as the type of user queries, have significant impact on the overall performance.



If we extend the aforementioned even further, not only for the AloD metadata catalogue itself we need to select the right model(s), but we could also potentially want to select a best performing model for every individual AI Asset type. An argument supporting this assumption is that different asset types are described by different properties (e.g. Datasets vs. Educational resources), resulting in a different textual representation. Although this is a valid point, having different models for different assets would introduce (potentially significantly) increased computational requirements on the run of the service. Therefore, we decided to use one model that can be assumed to work sufficiently well for different types of AI Assets. Due to the limited time frame of this activity and to speed up the experimentation, we decided to constrain the set of AI Asset types we experimented with to Datasets.



Fig. 6a: Enhanced search, experimentation setup. First, we downloaded all Datasets from the AloD metadata catalogue and transformed them into textual representation. Then, we computed vector embeddings with different models in order to thoroughly compare performance of different models. The goal was to select the right model to be used to implement a semantic search service.

The experimentation setup can be seen in Fig. 6a. First, we downloaded a snapshot of all Datasets from the AloD metadata catalogue, available through its REST API, to a local database. For experimentation we used ChromaDB.

Next, individual datasets, which are represented as documents (JSON structure) in the AloD metadata catalogue, were transformed to textual representation. We experimented with two different representations:



- 1. All relevant fields/properties of a datasets were joined to create the textual representation. This representation was longer and contained more information. This technique achieved in general better performance.
- 2. Only "basic" properties of the dataset were joined, namely its *name*, *description*, *tags* and the *platform* the datasets originated from. This resulted in shortened representation.

Later, we compared these two forms with the goal to test which form results in better vector embeddings.



Keywords: ecology | bird | apis mellifera | honeybee | parus major | animal behaviour | pica pica | unpulished

Fig. 6b: Comparison of two forms of textual representation of a Dataset.

As the next step, we computed vector embeddings for all documents in the database. The candidate models, GTE, E5 and BGE (different sizes of these) were selected based on literature overview, the current leaderboards, empirical experience and model size. With every model, we computed two different vector representations for each Dataset - one for "relevant fields" and one for "basic fields" textual representation. As many texts were longer than the maximal context length of the models under test, we needed to chunk such texts and work with embeddings of individual chunks. Here, we tested two techniques:

- 1. We combined the embeddings of individual chunks of a text into one aggregated vector representation of the Dataset.
- 2. In the retrieval phase, we retrieved individual chunks. Then, if one chunk from a Dataset was retrieved, we considered the whole Dataset to be a match. This technique achieved in general better performance.

For effective evaluation, the embeddings were stored in a specialised vector database. We compared two vector databases in terms of performance and ease of use. For the first, experimentation-oriented phase, we selected ChromaDB as it's easier to set up and use.



Despite being sufficient in this phase, for implementation of the service (Phase 2) we decided to use MilvusDB as it, based on experimental results, achieved better performance.

After computing and storing vector representations of all datasets, we conducted an extensive set of experiments where we compared all potential combinations of models, json-to-text transformations and chunking strategies in order to discover the combination that provides best overall results in the retrieval task based on semantic similarity between user guery and a document (Dataset). The overall evaluation setup can be seen in Fig.7:

- 1. Query generation.
- 2. Retrieval evaluation.

Current state of AloD enhanced search capabilities service:

- We thoroughly evaluated different embedding models (GTE, E5, BGE) to select the right model for computing vector embeddings enabling implementation of semantic search. Based on the evaluation, we selected the "<u>GTE large</u>" model to be used as the backbone of semantic search.
- We thoroughly evaluated different forms of data preparation and text preprocessing. As the models aimed to be used are only able to compute embeddings from text data, the AI Assets, represented as structured information in the AIoD platform, need to be transformed to a textual representation. We examined different ways how to transform structured data to text (the main difference being what properties of the AI Assets should be used to compute embeddings in order to achieve best retrieval results).
- We implemented complex evaluation pipelines, examining individual variables entering the search pipeline (models, text processing). The pipelines focused on a specific type of AI Assets available in AIoD Datasets and provided insights (different metrics) enabling us to select the preferred configuration.
- We computed embeddings of all currently available Datasets.
- We prepared an implementation plan for implementing a separate service, which is to be executed in the following months.





Figure 7: Evaluation setup. To evaluate the performance of semantic search (retrieval), first, we need to collect a sufficient number of relevant testing queries. The queries were generated by using LLMs, either directly or through the Ragas.io tool. In the evaluation phase, we evaluated the performance in terms of recall (whether documents we expected should be retrieved were retrieved by a query) and precision (how many of top K retrieved documents were considered relevant by an LLM).



There are two main approaches to evaluation of retrieval task performance: automated and manual evaluation. In the manual approach, a human expert would query the model and assess the quality of the presented results. This has two downsides; it is very time consuming, and it is hard to assess whether there are good matches in the database which are not presented to the user. So, in this activity we focused on automated evaluation as it enabled us to thoroughly and fairly compare different retrieval settings.

For automated retrieval evaluation, it's necessary to have a set of evaluation queries. Then, we test whether a particular query retrieved the documents it was expected to retrieve, or, in other words, whether the documents retrieved by the query can be considered relevant. For this, we need to have a large amount of labelled examples to assess the models. We considered both labelling manually, and using LLMs to assist in this task. We first did a small-scale experiment to assess the quality of LLM-generated labels. In our experiment, we selected a subset of 100 datasets, each labelled manually with topics based on the dataset description, title, features, and meta-features by multiple human experts. We then also generated labels with an LLM. We found that while the LLMs still seemed slightly less accurate and had slightly more disagreement with the humans than the humans did amongst themselves, the difference was small and likely not significant given the small sample size. Hence, in our evaluations we relied more heavily on LLM-generated data.

In order to have a high amount of sufficiently variable queries, **we generated the testing queries automatically** by using LLMs either with or without a Ragas tool. We generated two types of queries:

- Detailed queries tied to specific assets. To generate these queries, LLMs and Ragas (which internally uses LLMs) were fed a text representation of a Dataset and were instructed to generate a set of questions/queries that, if used for retrieval tasks, were expected to retrieve this particular dataset. In this way, we generated a large amount of (*query*, *asset*) pairs. In the evaluation phase, if the generated query was used, we tested if the asset from which the query was generated was retrieved.
- **Generic search queries**. These queries were generated automatically by LLMs. The LLMs were instructed to generate queries that users of the platform could use to retrieve datasets from the AloD metadata catalogue (i.e. pretend to be users of the platform). We generated three types of such queries: *least descriptive*, *moderately descriptive* and *most descriptive* (see Fig. 8).
 - *Least descriptive*: A concise user query, up to 70 characters, capturing only the essential and most significant properties of the dataset.
 - Moderately descriptive: A detailed user query, up to 200 characters, providing additional information and properties to offer a clearer description of the dataset.
 - Most descriptive: A comprehensive user query, up to 500 characters, encompassing a wide range of details and characteristics to thoroughly describe the dataset.



DESCRIPTIVENESS LEVEL: least_descriptive image classification dataset text summarization data speech recognition dataset

DESCRIPTIVENESS LEVEL: moderately_descriptive datasets for image classification tasks with labels text summarization datasets in English language speech recognition datasets with transcriptions

DESCRIPTIVENESS LEVEL: most_descriptive image classification datasets with high-resolution images, labeled categories, and balanced class distribution text summarization datasets in English, containing news articles, summaries, and metadata speech recognition datasets with transcriptions, audio recordings in various accents, and noise levels

Figure 8: Examples of LLM-generated generic search queries with various descriptiveness. In general, we evaluated the retrieval in terms of precision and recall.

To estimate **recall**, we utilised **detailed queries tied to specific assets**. For every query we retrieved top K Datasets from the vector database and we evaluated whether datasets expected to be retrieved were retrieved among these top K. It's important to note that this is only an estimate of how well a model (or, rather, a semantic search based on this model) recalls relevant datasets - as we don't know how many datasets in the whole Metadata catalogue are in fact relevant for a particular query, we were only able to measure if the datasets we explicitly expected (i.e. those from which the queries were generated) to be retrieved were retrieved.

To estimate **precision**, we utilised both **detailed queries tied to specific assets** and **generic search queries**. The difficulty of estimating precision lied in the fact that we had no ground-truth labels (annotated by humans) telling us whether a particular Dataset is relevant and therefore should have been retrieved by a query. To address this issue, we utilised an LLM to estimate whether every single Dataset from the top K Datasets retrieved for a query should be considered really relevant and therefore were retrieved correctly. For this, we used the current (presumably) strongest large language model - GPT-40. To increase the chance that the LLM provides correct estimate whether a Dataset is relevant to a particular query, we instructed the model not only to provide a prediction, but also:

- To provide Explanation (why is the dataset relevant to a query?);
- To provide Relevance score. The model was instructed to estimate relevance of an asset to the query on a scale 1-5 with following meaning:
 - 1 = Not relevant at all
 - \circ 2 = Slightly relevant
 - 3 = Moderately relevant
 - 4 = Very relevant
 - 5 = Extremely relevant
- To tell whether the conditions given by the user in the query were satisfied. The conditions (e.g., "the dataset must have at least 10000 samples") are automatically identified by the LLM.

The complete prompt is shown in Figure 9.



Project No 952215 October, 2024, D.11.4, Integration of OpenML with AI4EU, v.2 [PU]

system_prompt = """	
You are an expert evaluator tasked with assessing the relevance of machine learn	ing assets
(such as models or datasets) to specific user queries. Each query describes the	requirements
and desired properties of an asset. You will be given a query and a correspondir	g document
(a short asset description together with additional metadata) and are asked to p	rovide the following:
1) A detailed explanation structured into the following sections:	
 Condition Evaluations: For each key condition or constraint mentioned in t Condition: The specific condition being evaluated. 	he query, provide:
- Details: A brief description of how the condition was or was not met.	
 Mandatory: Whether the condition is mandatory (by default it is, unles Met: Whether the condition was met (true/false/cant tell). 	s specified otherwise)
- Overall Match: A summary statement of the overall relevance of the documer	t to the query.
2) A relevance rating on a scale from 1 to 5, where:	
- 1 = Not relevant at all	
- 2 = Slightly relevant	
- 3 = Moderately relevant	
- 4 = Very relevant	
- 5 = Extremely relevant	
prompt_template = """	
Query: {query}	
Document: {document}	

Figure 9: System prompt used to instruct the LLM how to evaluate the relevance of a document to an input user query.

Phase 2: Implementation of the service and AloD integration

In the second phase, we started the implementation of the search service. The service will provide a REST API through which users can search different types of AI Assets available in the AloD metadata catalogue by using queries in natural language.

The service will be periodically synchronised with the AloD platform to be able to find newly added Al Assets. See section <u>Future work</u> for more details.

Research dissemination assistant

Dissemination of research work and research results is an important part of researchers' job. It increases the overall awareness of new research findings in various communities, from experts in the field through fellow researchers from other fields to general public.Despite being important, researchers often neglect propagation of their own research as it may require a lot of effort to prepare a high-quality material that in a couple of paragraphs (or even sentences) summarises the main points of their work in a way that's not only understandable to experts, but also to lay people.

We address this challenge by creating a **service that helps researchers to draft a social media post** (later, it will be extended to a regular, more extensive blog post) from a **research paper**. In the background, the service utilises current NLP methods, namely summarization language models and large language models (LLMs).



Similarly to AloD enhanced search capabilities service, we followed a two-phase process:

- Phase 1: Research of relevant NLP approaches, testing, model selection and prompt tuning and parametrization;
- Phase 2: Implementation of QuickRePost application;

Phase 1: Research and selection of NLP approaches

The goal of this phase was to create an NLP pipeline that transforms a text of a research paper, extracted from a PDF file, to a draft of a post. In this initial phase, we focused on a social network for professionals, LinkedIn. We thoroughly compared the performance of different open-source and closed-source models (language models for text summarization and LLMs for generation) and their ability to generate social media posts that meet user requirements. We tested and fine-tuned the prompts used to instruct the LLMs to generate social media posts (human testing and automated testing) and created a mechanism to dynamically compose prompts based on user requirements. In summary, in this phase we needed to:

- Create a parameterizable prompt for the LLM.
- Define a set of parameters and their possible values that enable users to define how the desired post should look like.
- Compare performance of different open-source and closed-source LLMs;
- Design the final pipeline to be integrated into the application;

Create a parameterizable prompt for the LLM

The main purpose of the prompt is to instruct the LLM to generate a specific type of social media post from a textual representation of a research paper, and to specify desired properties of the post. The properties are selected by the user and injected into the parameterizable prompt.

We experimented with various forms of prompts and various language models. In the initial phase, we manually evaluated several posts generated from research papers by different LLMs (LLAMA3-8b, LLAMA3-70b, Mistral). We conducted several refinement rounds during which we both quantitatively and qualitatively evaluated the generated posts and iteratively refined the prompt, as illustrated in Fig 12 and Fig 13. The output of this step was a refined prompt that was later parameterized.

Define a set of parameters and their possible values that enable users to define how the desired post should look like

In order for the users to be able to define how the post should look like, we decided not to build the application with one predefined fixed prompt. On the other hand, to increase the chance that the LLM generates a high-quality post for a selected social network, we also decided not to leave the users complete freedom in how they instruct the LLM. Therefore, we decided to utilise parameterizable prompts.



Project No 952215 October, 2024, D.11.4, Integration of OpenML with AI4EU, v.2 [PU]

prompts = ["Can you create me a LinkedIn post blog from summarization of this text: " "Can you create me a LinkedIn post blog from summarization of this research article: ", "Summarize this research article into a brief LinkedIn post: " "Draft a LinkedIn post summarizing the key points of this article for a general audience: ", "Create a LinkedIn post summarizing this report, keeping the language simple: ", "Generate a LinkedIn post summarizing this study for professionals in the field: ", "Write a LinkedIn post summarizing this paper for a technical audience: ", "Create a LinkedIn post with a brief overview of this content: ", "Draft a LinkedIn post with a detailed analysis of this article: "Summarize this article in basic English for LinkedIn post: " "Create a LinkedIn post in simple English summarizing this text: ", "Summarize this scientific paper in basic English with a formal tone for researchers, including key statistics: ", "Create a LinkedIn post in simple English summarizing this scientific paper for university students, highlighting key f "Draft a concise LinkedIn post in advanced English for IT professionals, focusing on recent technological advancements "Write a LinkedIn post in intermediate English with an engaging tone for AI and ML professionals, incorporating an anec "Summarize this scientific paper in basic English with a professional tone aimed at software developers, emphasizing pr "Create a brief LinkedIn post in simple English for high school students, outlining the main benefits discussed in the "Summarize this scientific paper in simple English, suitable for university students, focusing on the core conclusions: "Summarize this scientific paper in simple English for college students, explaining the implications of the findings: "Create a 150-word summary of the text provided to me on LinkedIn: "Create a LinkedIn post in simple English summarizing this scientific paper for university students, highlighting key f "Write a LinkedIn post summarizing this scientific paper in 5 sentences: "Write a LinkedIn post summarizing this scientific paper in 10 sentences: "Create a LinkedIn post summarizing this scientific paper in 2 paragraphs: "Create a LinkedIn post summarizing this scientific paper in 3 paragraphs: " "Write a brief summary of this scientific paper in 1 paragraph and create me LinkedIn post: "Can you create me a LinkedIn post blog from summarization of this text with some hastags: "Can you create me a LinkedIn post blog from summarization of this text without hastags: ", "Can you create me a professional LinkedIn post with everything that belongs to it from the summary of the text: ", "You can create an elaborate LinkedIn post for me with everything that goes with it from the summary text:

Figure 12: Several examples of prompts used in the first phase of this step. The results were manually evaluated.

Paper	Prompt	Output_Llama3_8b_without_references	Eval	Note	Rate
https://scholar.g	<u>o</u> Can you create me a LinkedIn post blog fr	Here's a LinkedIn post summarizing the text:	Bad-Something else	The summarization was pretty go	3
https://scholar.g	o Can you create me a LinkedIn post blog fr	Here's a LinkedIn post based on the research artic	OK-minor issues 🔹	Based on the prompt, output looks	4
https://scholar.g	o Summarize this research article into a brie	Here's a brief LinkedIn post summarizing the rese	a Bad-Style 👻	Not the best form for LinkedIn pos	3
https://scholar.g	o Draft a LinkedIn post summarizing the key	Here's a LinkedIn post summarizing the key points	Bad-Style 🔹		3
https://scholar.g	o Create a LinkedIn post summarizing this re	Here's a LinkedIn post summarizing the report in s	oK-minor issues 🔹	The summarization was pretty go	4
https://scholar.g	o Generate a LinkedIn post summarizing thi:	Here's a summary LinkedIn post:	Bad-Style 👻	This is not a text for experts and p	2
https://scholar.g	o Write a LinkedIn post summarizing this pa	Here's a summary of the paper for a technical aud	Bad-Style	In a LinkedIn post, the author of the	2
https://scholar.g	o Create a LinkedIn post with a brief overvie	Here's a LinkedIn post based on the provided con	t Bad-Style 👻	The post also mentions the weak	3
https://scholar.g	o Draft a LinkedIn post with a detailed analy	Here's a draft LinkedIn post analyzing the article:	Bad-Style -	the author of the post should not v	2
https://scholar.g	o Summarize this article in basic English for	Here's a summary of the article in basic English fo	Bad-Style 👻	Wrote about weakness of the artic	3
https://scholar.g	o Create a LinkedIn post in simple English s	Here's a LinkedIn post summarizing the text in sin	OK-minor issues 🔹	Maybe it seems to be OK. Maybe	4
https://scholar.g	o Summarize this scientific paper in basic Er	Here is a summary of the scientific paper in basic	E Bad-Style 👻	Wrote about weakness of the artic	2
Paper	Prompt	Output Llama3 70b without references	val	Noto	Pato
Paper	Prompt	Output_Llama3_70b_without_references	Eval	Note	Rate
Paper https://scholar.g	Prompt o Can you create me a LinkedIn post blog fr	Output_Llama3_70b_without_references E Here's a possible LinkedIn post based on the pro	OK-minor issues	Note Based on the prompt, output looks	Rate 4
Paper https://scholar.g https://scholar.g	Prompt o Can you create me a LinkedIn post blog fr o Can you create me a LinkedIn post blog fr	Output_Llama3_70b_without_references Here's a possible LinkedIn post based on the pro Here's a LinkedIn post summarizing the researcl	OK-minor issues OK-minor issues	Note Based on the prompt, output looks Based on the prompt, output looks	Rate 4 4
Paper https://scholar.g https://scholar.g https://scholar.g	Prompt o Can you create me a LinkedIn post blog fr o Can you create me a LinkedIn post blog fr o Summarize this research article into a brie	Output_Llama3_70b_without_references Here's a possible LinkedIn post based on the pro Here's a LinkedIn post summarizing the research Here's a brief LinkedIn post summarizing the res	OK-minor issues OK-minor issues Bad-Style T <lit< li=""> <l< td=""><td>Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked</td><td>Rate 4 3</td></l<></lit<>	Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked	Rate 4 3
Paper https://scholar.g https://scholar.g https://scholar.g https://scholar.g	Prompt o Can you create me a LinkedIn post blog fr o Can you create me a LinkedIn post blog fr o Summarize this research article into a brie o Draft a LinkedIn post summarizing the key	Output_Llama3_70b_without_references Here's a possible LinkedIn post based on the pro Here's a LinkedIn post summarizing the research Here's a brief LinkedIn post summarizing the res Here's a draft LinkedIn post summarizing the key	OK-minor issues OK-minor issues Bad-Style Bad-Style Bad-Style Comparison Comparison<td>Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked I don't know whether to consider it</td><td>Rate 4 4 3 3</td>	Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked I don't know whether to consider it	Rate 4 4 3 3
Paper https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g	Prompt o Can you create me a LinkedIn post blog fr o Can you create me a LinkedIn post blog fr o Summarize this research article into a brie o Draft a LinkedIn post summarizing the key o Create a LinkedIn post summarizing this re	Output_Llama3_70b_without_references Here's a possible LinkedIn post based on the prot Here's a LinkedIn post summarizing the researcl Here's a brief LinkedIn post summarizing the res Here's a draft LinkedIn post summarizing the key Here's a LinkedIn post summarizing the report:	OK-minor issues OK-minor issues Bad-Style Bad-Style OK-minor issues Image: Style OK-minor issues Image: Style Image: Style	Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked I don't know whether to consider it It's look good. Maybe some minor	Rate 4 3 3 4
Paper https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g	Prompt 2 Can you create me a LinkedIn post blog fr 2 Can you create me a LinkedIn post blog fr 3 Summarize this research article into a brie 2 Draft a LinkedIn post summarizing the key 2 Create a LinkedIn post summarizing this re 2 Generate a LinkedIn post summarizing this	Output_Llama3_70b_without_references Here's a possible LinkedIn post based on the pro Here's a LinkedIn post summarizing the researcl Here's a brief LinkedIn post summarizing the res Here's a draft LinkedIn post summarizing the key Here's a LinkedIn post summarizing the report: Here's a LinkedIn post summarizing the study fo	OK-minor issues OK-minor issues Bad-Style OK-minor issues Bad-Style OK-minor issues Charling Cha	Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked I don't know whether to consider it It's look good. Maybe some minor This is not a text for experts and p	Rate 4 4 3 3 4 2
Paper https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g	Prompt Can you create me a LinkedIn post blog fr Can you create me a LinkedIn post blog fr Summarize this research article into a brie Draft a LinkedIn post summarizing the key Create a LinkedIn post summarizing this ra Generate a LinkedIn post summarizing this Write a LinkedIn post summarizing this pa	Output_Llama3_70b_without_references Here's a possible LinkedIn post based on the pro- Here's a LinkedIn post summarizing the researce Here's a brief LinkedIn post summarizing the key Here's a draft LinkedIn post summarizing the report: Here's a LinkedIn post summarizing the study fo Here's a LinkedIn post summarizing the paper for	OK-minor issues OK-minor issues Bad-Style Bad-Style OK-minor issues Bad-Style OK-minor issues State OK-minor issues State State	Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked I don't know whether to consider it It's look good. Maybe some minor This is not a text for experts and p Maybe a different name, tags defir	Rate 4 3 3 4 2 4
Paper https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g	Prompt Can you create me a LinkedIn post blog fr Can you create me a LinkedIn post blog fr Summarize this research article into a brie Draft a LinkedIn post summarizing the key Create a LinkedIn post summarizing this Write a LinkedIn post summarizing this Write a LinkedIn post summarizing this Create a LinkedIn post summarizing this paj Create a LinkedIn post with a brief overvie	Output_Llama3_70b_without_references Here's a possible LinkedIn post based on the pro Here's a LinkedIn post summarizing the research Here's a brief LinkedIn post summarizing the res Here's a draft LinkedIn post summarizing the report: Here's a LinkedIn post summarizing the study for Here's a LinkedIn post summarizing the study for Here's a LinkedIn post summarizing the paper for Here's a possible LinkedIn post based on the pro	Val OK-minor issues OK-minor issues Bad-Style OK-minor issues OK-minor issues OK-minor issues	Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked I don't know whether to consider it It's look good. Maybe some minor This is not a text for experts and p Maybe a different name, tags defir Based on the prompt, output looks	Rate 4 4 3 3 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4
Paper https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g	Prompt 2 Can you create me a LinkedIn post blog fr 2 Can you create me a LinkedIn post blog fr 3 Summarize this research article into a brie 4 Draft a LinkedIn post summarizing the key 5 Create a LinkedIn post summarizing this ra 6 Generate a LinkedIn post summarizing this pal 6 Create a LinkedIn post summarizing this pal 6 Create a LinkedIn post with a brief overvie 9 Draft a LinkedIn post with a detailed analy:	Output_Llama3_70b_without_references Here's a possible LinkedIn post based on the pro Here's a LinkedIn post summarizing the research Here's a brief LinkedIn post summarizing the res Here's a draft LinkedIn post summarizing the report: Here's a LinkedIn post summarizing the report: Here's a LinkedIn post summarizing the paper for Here's a possible LinkedIn post based on the pro Here's a draft LinkedIn post with a detailed analy	Wal OK-minor issues Bad-Style Bad-Style OK-minor issues OK-minor issues OK-minor issues OK-minor issues OK-minor issues	Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked I don't know whether to consider it It's look good. Maybe some minor This is not a text for experts and p Maybe a different name, tags defir Based on the prompt, output looks Based on the prompt, output looks	Rate 4 3 3 4 2 4 4 4 4 4
Paper https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g	Prompt 2 Can you create me a LinkedIn post blog fr 2 Can you create me a LinkedIn post blog fr 3 Summarize this research article into a brie 4 Draft a LinkedIn post summarizing the key 5 Create a LinkedIn post summarizing this re 6 Generate a LinkedIn post summarizing this 6 Write a LinkedIn post summarizing this pal 9 Create a LinkedIn post with a brief overvie 9 Draft a LinkedIn post with a detailed analy: 9 Summarize this article in basic English for	Output_Llama3_70b_without_references Here's a possible LinkedIn post based on the pro Here's a LinkedIn post summarizing the research Here's a brief LinkedIn post summarizing the res Here's a LinkedIn post summarizing the report: Here's a LinkedIn post summarizing the study fo Here's a LinkedIn post summarizing the paper for Here's a LinkedIn post summarizing the paper for Here's a draft LinkedIn post with a detailed analy Here's a summary of the article in basic English.	OK-minor issues OK-minor issues Bad-Style OK-minor issues Bad-Style OK-minor issues OK-minor issues OK-minor issues OK-minor issues OK-minor issues VoK-minor issues	Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked I don't know whether to consider it It's look good. Maybe some minor This is not a text for experts and p Maybe a different name, tags defir Based on the prompt, output looks In the future it will be necessary to	Rate 4 4 3 3 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4
Paper https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g https://scholar.g	Prompt 2 Can you create me a LinkedIn post blog fr 2 Can you create me a LinkedIn post blog fr 3 Summarize this research article into a brie 4 Draft a LinkedIn post summarizing the key 5 Create a LinkedIn post summarizing this ra 6 Generate a LinkedIn post summarizing this 9 Write a LinkedIn post summarizing this para 6 Create a LinkedIn post summarizing this para 9 Create a LinkedIn post with a brief overvie 9 Draft a LinkedIn post with a detailed analy: 9 Summarize this article in basic English for 9 Create a LinkedIn post in simple English s	Output_Llama3_70b_without_references Here's a possible LinkedIn post based on the pro- Here's a LinkedIn post summarizing the research Here's a brief LinkedIn post summarizing the rese Here's a draft LinkedIn post summarizing the key Here's a LinkedIn post summarizing the study fo Here's a LinkedIn post summarizing the study fo Here's a LinkedIn post summarizing the paper fo Here's a possible LinkedIn post based on the pro- Here's a draft LinkedIn post with a detailed analy Here's a summary of the article in basic English, Here's a LinkedIn post summarizing the text:	OK-minor issues OK-minor issues Bad-Style OK-minor issues Bad-Style OK-minor issues OK-minor	Note Based on the prompt, output looks Based on the prompt, output looks The form is not suitable for Linked I don't know whether to consider it It's look good. Maybe some minor This is not a text for experts and p Maybe a different name, tags defir Based on the prompt, output looks Based on the prompt, output looks In the future it will be necessary to Based on the input prompt the res	Rate 4 4 3 3 4 2 4 4 4 4 4 4 4 5

Figure 13: Prompt evaluation and refinement

We defined a scaffold of a prompt that contains all the basic instructions instructing the LLM to create a specific type of a post (that was fine-tuned in the previous step). Then, in the application, the user can define various properties of the post like target audience, length of the post or whether the post should include hashtags. All these choices influence the post generated by the LLM. The parameterizable properties were consulted with marketing experts with the goal to select a limited set of properties that may have the highest impact on the quality and potential reach of the post, with the goal to not overwhelm the user with too many properties to be set. See Fig. 14 for an example of these settings.



```
audience_input = gr.Dropdown(
        "High school student",
        "University teacher",
        "Researcher",
        "Business executive",
        "IT professional",
        "Non-profit organization member",
        "General public",
   ],
    label="Select your audience:",
english_level_input = gr.Dropdown(
    ["", "Beginner", "Intermediate"], label="Select the English level:"
length_input = gr.Dropdown(
    ["", "Long", "Short", "Very short"], label="Select the length of the post:"
hashtag_input = gr.Dropdown(
        ••.
        "No use hashtags",
        "Use hashtags",
        "Use 3 hashtags that are already existing, popular and relevant",
    ],
    label="Hashtag usage:",
)
perspective_input = gr.Dropdown(
    ["", "First person singular", "First person plural", "Second person"],
   label="Choose the narrative perspective:",
emoji_input = gr.Dropdown(
        "Use emoticons",
        "Use emoticons instead of bullet points",
        "Do not use emoticons",
    ],
    label="Emoji usage:",
```

Figure 14: Users can select various parameters/properties of the post. These are fed as an input to the LLM (together with the text of the paper and basic prompt), influencing the generated post.



Compare performance of different open-source and closed-source LLMs

We thoroughly compared the performance of different open-source and closed-source models (language models for text summarization and LLMs for generation) and their ability to generate social media posts that meet user requirements.

The generally observed trend was that bigger and more recent models, like LLAMA 3.1 - 70b, generated higher-quality posts and more precisely followed the instructions (i.e. properties of the post selected by the user) than smaller models. Although this advocates in favor of not considering smaller models to be deployed as the backbone of the service, there is one important consideration to be taken into account - computational requirements. For instance, LLAMA 3.1 - 70b requires a rather powerful and expensive GPU (or even multiple GPUs) to run smoothly. Therefore, we also tested smaller LLMs with the conclusion that in many cases the quality of generated posts is decent. As described in the following subsection <u>Phase 2: Implementation of QuickRePost application</u>, we implemented the service to be deployable with models of different sizes.

Design the final pipeline to be integrated into the application

Although the text of the paper can be directly fed as an input to the LLM, there are two serious limitations we needed to address:

- If the paper is long, its length may be longer than the maximal context length supported by the selected LLM (it also needs to be taken into account that the prompt itself requires to be fed to the model). This may result in fitting only a part of the paper to the model.
- Due to the nature of current LLMs, longer inputs take more time to be processed.

Therefore, we proposed a two-step approach. Instead of feeding the text representation of the paper directly into the LLM, we first summarise the paper with a smaller and computationally less expensive text summarization model with decoder-only transformer architecture <u>bart-large-cnn</u>. For testing purposes or when the service is deployed on a low-resource hardware, we used mode from <u>Falconsai</u>.

Phase 2: Implementation of the service and AloD integration

In the second phase, we implemented a frontend-backend web application that provides a convenient user interface. The service utilises two AI models - one model for text summarization, the second one for generating a social media post based on a summarised research paper and user requirements.

The service is configurable and it can be deployed with a compatible HuggingFace model for both text summarization and generation. This improves the flexibility and reusability of the service as it can be deployed with different models based on available hardware.

Technology-wise, the service was implemented in Python and in framework <u>Gradio</u> which provides sufficient functionalities for multi-user applications (e.g. queueing system) and makes development of simple front ends easier. The first version of the application was implemented in a different framework, Streamlit, which was abandoned due to issues with asynchronous operations.



3. Achieved results

OpenML-AIoD integration

OpenML datasets are easily discoverable in the AloD catalogue and directly usable within services built on the AloD API. We will highlight the most salient ones: MyLibrary, Al Builder, and RAIL, but any new services built on AloD will also have access to all OpenML datasets.

MyLibrary

All OpenML datasets can be easily discovered on the AloD platform via the Al resources search (called MyLibrary), as illustrated in Fig. 15. The interface also offers a dropdown that allows users to search OpenML datasets specifically. Selecting a dataset card will show all the details exported from the OpenML API, including the original authors, source, keywords, and full description. An example is shown in Fig. 16.



Fig 15. OpenML datasets can be found and accessed in the AloD catalogue (MyLibrary)



Fig 16. Dataset detail for one of the OpenML datasets on AloD.



Al Builder

OpenML datasets can be imported directly into the AI Builder, a no-code environment in which AI models can be created by connecting datasets and models on a canvas. It has its own Marketplace, where all OpenML dataset can be found under 'Data Sources', shown in Fig. 17. From there, they can be imported and pipelines on the AI Builder canvas, illustrated in Fig. 18. Further guides can be found in the AI Builder documentation.⁵



Fig 17. OpenML datasets can be found and accessed in the AI Builder



Fig. 18 Using datasets in pipelines on the AI Builder canvas.

RAIL

RAIL is an AloD service aiming to simplify large-scale experimentation and benchmarking. The user only needs to select datasets, models, and experiment variables. The user interface is shown in Fig. 19. Here, 'diabetes', is a dataset imported from OpenML, and a simple experiment is shown evaluating kNN classifiers on this dataset. In the background, docker containers and scripts are created that will execute the experiments in any computational environment (e.g. a HPC center), and return the results. The template for doing this is illustrated in Fig. 20. It includes the environment (here it will install and import OpenML and scikit-learn), and the script that is generated to run the experiment.

⁵ https://gitlab.eclipse.org/eclipse/graphene/tutorials/-/tree/main/openml_data_integration



Train KNN classifier for diabetes dataset 📎

Sep 12, 2024			
G Sep 12, 2024			
e diabetes			
🔁 The name of this r	resource		
L Experiment temple	ate: OpenML KNN Training Te	emplate	
Environment variabl			
	65		
Name			Value
к•			З
D			
Trains and source a KNN	l alaasifias fax diabataa dataaa	-	
	Classifier for diabetes datase	et	
Rupe			
Runs			
🕂 Create new experime	ant run		
Experiment ru	ins		
Experiment re			
ID	State 0	Created at Updated at Metrics Actions	
66e36e48ba5e292b	78356f0a CREATED S	Sep 12, 2024, 10:42:16 PM Sep 12, 2024, 10:42:16 PM O	
Fig. 19. Usir	ng OpenML da	atasets in RAIL experiments.	
	U .	•	
	Experiments Data	asets Docs Profile Admin	
Al on Demand	Experiments Data	asets Docs Profile Admin	
Al on Demand	Experiments Data	openML KNN Training Template 🛇	
Rul-Research and Innovation Al Lab Beta version	Experiments Data Public experiments Mulexperiments	asets Docs Profile Admin OpenML KNN Training Template 🛇	
Al on Demand Mult Research and Innovation Al Lab Bete version	Experiments Data Public experiments My experiments	asets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved.	
Al on Demand Michael Research and Incometion AL Lab Beta version	Experiments Data Public experiments My experiments + Create an experim	asets Docs Profile Admin OpenML KNN Training Template Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with DecoML datasets	
Al on Demand With Research and Innovation Al Lab Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates	asets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets.	
Al on Demand Will Research and biotoverion Al Lab Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates Mu templates	asets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime	
Al on Demand Will Research and Information Al lab Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates . Create a templates	asets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) ()	
Al on Demand Multi-Research and Innovation Al Lab Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	asets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) (*	
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	asets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) (* 1 FROM puthon:3.12 2 3 Torificate public	
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	assets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. Kenze is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) C 1 FROM python:3.12 2 VMRKDIR /app 3 CCPPY requirements.txt. 3 ICCPPY requirements.txt.	
Beta versian	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	asets Docs Profile Admin	
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	assets Docs Profile Admin	
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	asets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) (* 1 [FRCM pgthon:3.12] 2 VORKOIR /app 3 [CDFY requirements.txt. 5 RUN pip install -r requirements.txt Dependencies (libraries) (*	
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	esets Docs Profile Admin	
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	essets Docs Profile Admin	
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	assets Docs Profile Admin CpenML KNN Training Template Solution: Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. The template is only compatible with OpenML datasets. Docker image (experiment execution environment) 1 FROM python:3.12 2 WORKDIR / app 3 COPP requirements.txt: Copendencies (libraries) <th>×E5")</th>	×E5")
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	assets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) Docker image (experiment execution environment) 1 FROM puthon:3.12 2 WORKOR /app 3 CDPY requirements.txt. 5 RUN pip install -r requirements.txt Dependencies (libraries) 1 openmI==0.142 2 scikit-learn==15.1 Executable (script)	MES.)
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	assets Docs Profile Admin OpenML KNN Training Template Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) 1 1 1 1 2 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 <td>MES")</td>	MES")
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	assets Docs Profile Admin OpenML KNN Training Template Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) 1	MES")
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	essets Docs Profile Admin	мев")
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	assets Docs Profile Admin	мез")
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	Basets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K prameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) Colspan="2">Cerver image (experiment execution environment execution execution environment execution execu	MES.)
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	asets Docs Profile Admin OpenML KNN Training Template Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K profile Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a KKN classifier implemented to be selected. Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trains a Kine implemented to implemente the environment variable DATASET, NM return of Researce provide the number of neighbors (k) in the environment variable K return int(k) Trains ValueErron("Researce runame() Tret	мез") "]
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	asets Docs Profile Admin OpenML KNN Training Template S Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) Docker image (experiment execution environment) I [FRCM python:312 I (FRCM python:312 I (FRCM python:312 I (Dependencies (libraries)) I (FRCM python:312 I (Dipanti-G142) I (Dipanti-G142) I (Dependencies (libraries)) I (Dependencies)) I (Dependencies)) I (Dependencies)) I (Dependencies)) I (Dependencis libraries)	меся")
Beta version	Experiments Public experiments My experiments + Create an experim Public templates My templates + Create a template	assets Docs Profile Admin OpenML KNN Training Template Trains a KKN classifier implemented in scikit-learn on a selected OpenML dataset. Trained model is saved. K parameter of KNN needs to be selected. The template is only compatible with OpenML datasets. Template details and runtime Docker image (experiment execution environment) Image: [FRQM pgthon:3.12 Image: [FRQM pgthon:3.12 <td>MES")</td>	MES")
Beta version	Experiments Data Public experiments My experiments + Create an experim Public templates My templates + Create a template	assets Docs Profile Admin	YES')

Fig. 20. An experiment template based on OpenML datasets in RAIL.



OpenML Chatbot

We developed a simple query interface allowing users to answer any questions, shown in Fig. 21. It is now running on the OpenML test server and accessible to beta users.

Query									
find	find me a housing dataset								
Select	Ouerv Tv	pe							0
Data	iset	pc							√
Batt									
	Use Al p	owered filter	ing 🤉						
Sea	rch								
Jea									
	•								
	OpenMI	Agent: Hello	o! How c	can Thelp you today?					
	A Note	that results a	are pow	ered by local LLM models ar	nd may not b	be accur	ate. Plea	ise refer	r to
	theomo		website	for accurate mormation.					
etected	d Filter(s	s): null							
\bigcirc	find me	a housing da	itaset						
_									
		Unnamed: 0	did	name		versior	n uploa	der sta	atus
		1,240	1558	bank-marketing		2	2	64 ac	tive
		3,657	43465	Boston-house-price-data			L 30,:	126 ac	tive
		3,736	43547	House-Rent-in-Indian-Cities-a	nd-Localities		L 30,:	126 ac	tive
		5,445	45933	HousingPrices			L 39,9	999 ac	tive
		3,588	43393	USA-Housing-Listings		1	L 30,:	126 ac	tive
		5,446	45934	HousingPrices		2	2 39,9	999 ac	tive
		3,891	43705	California-Housing-Prices		1	L 30,:	123 ac	tive
		3,166	42635	house_sales_reduced		1	L 14,4	480 ac	tive
		4,075	43926	ames_housing		1	L 30,8	861 ac	tive
		3,556	43361	Condominium-Comparable-R	ental-Income	-i 1	L 30,:	125 ac	tive
0	find me	a mushroom	datase	t with multiple classes					
		Unnamed: 0	did	name	version u	oloader	status	format	Мајо
		4,359	44272	Meta_Album_FNG_Micro	1	30,980	active	arff	
		4,422	44335	Meta_Album_FNG_Extended	1	30,980	active	arff	
		10	24	mushroom	1	1	activo	AREE	

Fig 21. Example OpenML chatbot interaction: looking for relevant training datasets.



Project No 952215 October, 2024, <u>D.11.4</u>, Integration of OpenML with AI4EU, v.2 [PU]

The interface consists of a standard query textbox, with some additional options. It is possible to specify what kind of resource the user is looking for (e.g. a dataset or a model), as well as the option to filter results automatically based on the query. For instance, if the user says she is looking for 'large datasets', the chatbot will only return the largest datasets and order them from large to small. In the final interface, these options may be hidden.

As shown in Fig. 21, when the user indicates that she is looking for datasets, the result will be a table that lists the most relevant datasets together with relevant metadata that the user can further filter and sort on. The figure also shows how users can easily rephrase their query based on earlier results.

If the user is interested in better understanding how to use an OpenML dataset, the chatbot will provide detailed information on the various ways to do this. It will also generate and explain code that users can copy-paste to include OpenML datasets in their projects. This is illustrated in Fig. 22, where step-by-step instructions are given on how to download datasets in Python. In the same vein, the chatbot can explain how to import datasets into libraries like PyTorch and TensorFlow, and how to train and evaluate models. This allows novice users to easily train new AI models without advanced programming experience, and allows advanced users to include OpenML datasets in new services and more complex projects.

The current version of the chatbot is based on <u>LLAMA 3.1</u> and <u>LangChain</u>, running inference on a single GPU. We have set up an API to which queries can be submitted and the final responses are returned in real time. All code is available online.⁶

AloD Chatbot (Talk2AloD)

In the first phase of AloD chatbot development, we evaluated different semantic search configurations and final configuration is fully implemented. Based on thorough evaluation (see section <u>Approach</u>) and experimental results, the following configuration of semantic search will be implemented in the second phase:

- Text representation: Use all relevant properties of an AI Asset
- Model: GTE-large (<u>https://huggingface.co/Alibaba-NLP/gte-large-en-v1.5</u>)
- **Chunking**: Retrieve individual chunks (don't aggregate embeddings of individual chunks of a document). If one chunk from a document is retrieved, the whole document is retrieved.

The main technologies used for implementation of the first phase are:

- Language: Python 3
- Experimentation environment: Jupyter notebooks, Python scripts
- HuggingFace: <u>https://huggingface.co/</u>
- LangChain: <u>https://www.langchain.com/</u>
- OpenAI API: <u>https://learn.microsoft.com/en-us/azure/ai-services/openai/reference</u>
- OLLAMA: https://ollama.com

⁶ https://github.com/openml-labs/ai_search



Search	
 OpenML Agent: Hello! How can I help you today? A Note that results are powered by local LLM models and may not be ac the official OpenML website for accurate information. 	ccurate. Please refer to
low do I download a datast on openml?	
To download a dataset from OpenML using Python, you can use the ope function. Here's an example:	
<pre>import openml # Replace '12345' with the actual dataset ID dataset_id = 12345</pre>	
<pre># Download the dataset data, _, _ = openml.datasets.get_data(dataset_id) # Print the first few rows of the dataset print(data.head())</pre>	
This code downloads the dataset with ID 12345 and prints the first few r If you want to specify a target column or include row IDs in the download	rows using Pandas. ded dataset, you can pass
<pre>additional arguments to the get_data() function: data, _, _ = openml.datasets.get_data(dataset_id, target)</pre>	et='target_column', i
Make sure to replace <u>12345</u> with the actual ID of the dataset you want to	o download.
<pre>data, _, _ = openml.datasets.get_data(dataset_id, datas</pre>	<pre>set_format='array')</pre>
This will return the dataset as a NumPy array. If you prefer a Pandas Data dataset_format='dataframe'	aFrame, set
For more information on the available options and arguments, refer to the documentation: https://openml.github.io/openml-	he OpenML
Sind me a housing dataset	

Fig 22. Example OpenML chatbot interaction: getting code to download datasets in Python



The user interface for the AloD chatbot is a simple and easy-to-use query-reponse interface, shown in Fig. 23. Examples of more advanced user queries and the raw structured outputs returned by the LLM are shown in Fig. 24 and 25. These will be integrated into the graphical interface soon.

Talk2AIoD

This Chatbot was built to guide you through the AloD website and recommend you resources for your work.



Fig 23. Example AloD chatbot interaction.



Figure 24: An example of a user query (general) and a retrieved Dataset textual representation.





Figure 25: Structured output provided by an LLM. Besides relevance_rating, the output also provides an overall evaluation of the match and an explanation of how well the dataset meets individual conditions given in the query by the user. The conditions were automatically extracted from the query by the model. Such an evaluation approach is more transparent and provides better overall results.

The second phase, implementation of a service with REST API, providing semantic search functionality, is currently under active development (see section <u>Future work</u> for more details). The source codes implemented in both phases can be found on AloD GitHub: <u>https://github.com/aiondemand/aiod-enhanced-interaction/tree/main</u>

Research dissemination assistant

We implemented and deployed the initial version of an application on a testing server, to be tested by first users. Currently, the service enables generating proposals of LinkedIn posts from the text of research papers. The service enables users to upload a PDF file (with a limit on the maximum file size), select required properties of the social media post and generate a draft of the post.

The user interface and workflow are illustrated below. It follows these steps:

- 1. The user uploads their research paper as PDF. The file can be either selected by clicking on a button or through drag-and-drop functionality, as shown in Fig. 26.
- 2. The user clicks on the "Start summarization" button. This initiates the summarization with a dedicated (smaller) language model, shown in Fig. 27. The summarization is later used as an input to the LLM.
- 3. The user selects the desired properties of the post (e.g audience, length, use of emoji's,...). An example setting is shown in Fig. 28. The selection can be done while the summarization is in the process.
- 4. The user generates a draft of the post by clicking on the "Generate post" button, and can further edit it later.



Al on Demand QuickRePost
Document Summarization
C Upload your PDF Document
<u>ا</u> ۲.
Drop File Here
- or -
Click to Upload
Summary of the PDF
Start summarization
Reset
Social Media Post Panel
Select your audience:
•
Select the English level:
Select the length of the post:
•
Hashtag usage:
Choose the narrative perspective:
· · ·
Emoij usaeer
Paste the URL to your paper here:
Declarable Venerables Annuals
begin with nodejiterrowanig Question.
Enter your custom requirements here:
Generate post
Stop

Fig 26. User interface of QuickRePost. It offers simple and sequential guidance, only requiring users to upload a document and selecting the settings to generate the desired social media posts.



Project No 952215 October, 2024, <u>D.11.4</u>, Integration of OpenML with AI4EU, v.2 [PU]

Al on Demand QuickRePost

ocument Summarization		
1 Upload your PDF Document	×	
ecun2015.pdf	674.2 KB	
summary of the PDF		
HAL is a multi-disciplinary open access archive for the deposit and dissemination of sci-entific research documents. The documents may come from teaching and research institutions in France or abroad, or from public or private researcenters, /nDeep learning is a class of techniques at a allows a machine to be fed with raw data and to automatically discover the representation. To classification. Deep-learning techniques are representation. Incern modules that each transform the representation at one level (starting with the raw input) into a suitable internal representation. The input into a suitable internal representation or learning transition and data. Supervised learning transition technicate structure in large dust, what more the devised that more or abroad, and that more relabours that indicate what mount the error would increase or decrease if the weight were increased by a tiny amount. This weight vector is then adjusted in the opposite direct. tion to the gradient vector. Deep learning discovers intricate structure in large dust yusing the backpropagation adjusted in the orgonealization ability of the machine – its ability to produce sensible answers on new inputs it has anverse training. The current practical applications of machine learning use linear classifices on top of hand-engineered features. Attwo class linear classificatine class different were or derivative with the same leave the sense to the generalization ability of the machine – its ability to produce sensible answers on new inputs it has nevers set training. Many of the current practical applications of machine learning insplications used in neural networks include the restified linear unit (ReLU) [12) – ma(R)(22) – man(2)(2), commonly used in recent years, as well as the more conve	ch ng ctor ning will to build a es by lata sets niques18 een during eparable. atural ional Hidden buted, it is a a From the ights, one itute for nouts) in reed many s: 1D for	
Start summarization		
Reset		

Fig. 27. After the summarization process is finished, the user can see the result. The summarization process can be cancelled by clicking on the Reset button.

Social Media Post Panel		
Select your audience:		
General public	•	
Select the English level:		
Intermediate	•	
Select the length of the post:		
Short	•	
Hashtag usage:		
Use 3 hashtags that are already existing, popular and relevant	•	
Choose the narrative perspective:		
First person plural	•	
Emoji usage:		
Use emoticons instead of bullet points	•	
Paste the URL to your paper here:		
	1	
Begin with Thought-Provoking Question.		
Enter your custom requirements here:		
Generate nost		
Ston		
Stop		

Fig. 28. User interface for selecting the desired language properties of the post.



4. Future work

Even though the TAILOR project has come to an end, the Al4Europe project will run for several more years and OpenML has a very active open-source development community that will keep developing it to make sure it stays relevant and useful for machine learning research. Below are the main plans to further finalise the developed integrations and LLM interfaces.

OpenML-AloD integration

Datasets are now fully integrated, but OpenML also has a large repository of benchmarks (experiments) and machine learning pipelines. These can be quite easily integrated when the AloD API is developed further.

Another point of attention is the schema used to share datasets. In the context of facilitating the exchange of datasets between platforms, OpenML has joined forces with HuggingFace, Kaggle, Google, and others, to develop a metadata standard, called Croissant⁷, for sharing machine learning datasets. It's built on top of existing standards such as schema.org, but adds a lot of important metadata that's unique to machine learning datasets, including how to use them properly and responsively. This makes datasets interchangeable between these platforms. Extending the AloD API to support Croissant may help closer integration. Moreover, Croissant is being supported by a growing number of Al libraries, including PyTorch and TensorFlow, making it easier to load Croissant datasets into these libraries to build new models. We aim to extend Croissant to also describe models and evaluations as well.

The Croissant format was accepted as a spotlight paper at NeurIPS 2024:

M.Akhtar, O. Benjelloun, C, Conforti, P. Gijsbers, J. Giner-Miguelez, N. Jain, M. Kuchnik, Q. Lhoest, P. Marcenac, M. Maskey, P. Mattson, L. Oala, P. Ruyssen, R. Shinde, E. Simperl, G. Thomas, S.Tykhonov, J. Vanschoren, J. van der Velde, S. Vogler, C-J. Wu. (2024) Croissant: A Metadata Format for ML-Ready Datasets. NeurIPS 2024.

OpenML Chatbot

Initial experimentation and user testing has shown that while dataset retrieval works quite well, adding complex constraints (e.g. limits on size and number of classes) are not always interpreted correctly. Moreover, the generated code for downloading and using datasets can still contain small mistakes, even though the model was trained on correct examples on how to use it. This is inherent to the use of LLMs and RAG. At the moment we are working on improving the service to avoid these mistakes, before we deploy it on the main OpenML website.

⁷ https://research.google/blog/croissant-a-metadata-format-for-ml-ready-datasets/



AloD Chatbot

In the following months (Sep-Dec 2024) we're going to finish the implementation of the service providing enhanced semantic search capabilities. The service will be integrated with the AloD REST API and will provide its own REST API, extending the interface of the platform.

Afterwards, we will add support for other types of AI Assets available in the platform, like Models or Educational resources.

The service will regularly synchronise with the AloD database in order to be able to provide search capabilities also for newly added Al assets.

Regarding deployment, the service will be distributed as a set of Docker images and as a Docker Compose configuration file. The main design decision is that the service may but doesn't have to be installed when a new node of the AloD platform is installed - it's up to the users to decide. This has several practical reasons:

- There may be hardware constraints that prevent users from deploying a service that utilises resource-demanding AI models (although they're not as demanding as LLMs, they're still best to be run on a specialised hardware such as GPU, which is not always available).
- As the platform aims to be extensible, potentially in the future there'll be other services providing semantic search built by 3rd parties and the users may select them.
- The AloD platform is meant to be robust, but lean. Therefore, some instances may not want to provide semantic search at all.

The current state of the implementation of the semantic search service can be found on AIoD GitHub: <u>https://github.com/aiondemand/aiod-enhanced-interaction/tree/main/api</u>.

Research dissemination assistant

Regarding QuickRePost, following activities are to be performed in Sep-Dec 2024:

- Integrate with the AloD authentication mechanism Keycloak/EGI checkin (to be finished in Sep 2024);
- Create easily deployable service distribution as a set of Docker images and a Docker Compose configuration;
- Deploy the service to the production and make it available to the users;



Appendix 1: Text of the original deliverable

Rationale: OpenML is an established platform for sharing machine learning datasets, algorithms, and models. It contains repositories of over 20000 datasets, 14000 machine learning pipelines, and 10 million machine learning experiments. It guarantees that all shared results are reproducible, and allows easy sharing and downloading of all resources via APIs and integrations into the most commonly used machine learning libraries. OpenML is an open source platform primarily developed by AI researchers in the Netherlands, Germany, and France. It is used yearly by over 150.000 people.

Approach: We plan to interface OpenML with the AI4EU platform, so that AI4EU resources can be accessed via OpenML interfaces already used by many AI researchers, and OpenML resources can be viewed via AI4EU. We will also engage in discussions with AI4EU developers to create a more integrated, vibrant ecosystem. OpenML is leveraged in WP7 and Task 9.1, and this integration will therefore also make this work directly available to the AI4EU platform.