A RECOMMENDER SYSTEM FOR SCIENTIFIC DATASETS AND ANALYSIS PIPELINES

Mandana Mazaheri

A thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

August 2021

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Mandana Mazaheri**

Entitled: **A recommender system for scientific datasets and analysis pipelines**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

_____ Chair

_____ Examiner

_____ Examiner

_____ Examiner

_____ Supervisor

Approved _____
Chair of Department or Graduate Program Director

_____ 20 _____ _____

Mourad Debbabi
Faculty of Engineering and Computer Science

Abstract

A recommender system for scientific datasets and analysis pipelines

Mandana Mazaheri

Scientific datasets and analysis pipelines are increasingly being shared publicly in the interest of open science. However, mechanisms are lacking to reliably identify which pipelines and datasets can appropriately be used together. Given the increasing number of high-quality public datasets and pipelines, this lack of clear compatibility threatens the findability and reusability of these resources. We investigate the feasibility of a collaborative filtering system to recommend pipelines and datasets based on provenance records from previous executions. We evaluate our system using datasets and pipelines extracted from the Canadian Open Neuroscience Platform, a national initiative for open neuroscience. The recommendations provided by our system (AUC= 0.83) are significantly better than chance and outperform recommendations made by domain experts using their previous knowledge as well as pipeline and dataset descriptions (AUC= 0.63). In particular, domain experts often neglect low-level technical aspects of a pipeline-dataset interaction, such as the level of preprocessing, which are captured by a provenance-based system. We conclude that provenance-based pipeline and dataset recommenders are feasible and beneficial to the sharing and usage of open-science resources. Future work will focus on the collection of more comprehensive provenance traces, and on deploying the system in production.

acknowledgments

I am very grateful for my supervisor, Dr. Tristan Glatard, he has been so supportive and provided dedicated guidance from the beginning of this journey. I also appreciate the help and support of the Big Data lab members who kindly answered my questions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Traditionally, scientific research was conducted in the research or academic institutions and was kept there. That was closed science, where after the findings were published, the data were often inaccessible to others due to privacy issues and ownership of the resources. This made data difficult to find and access and made further research on that subject slower. For such reasons, Open Science emerged to prepare and release the data to the whole scientific community worldwide, so they can collaborate in research and discovery, which leads to faster growth of findings.

Open Science is an umbrella term covering open dissemination of data, manuscripts, software, materials, methodologies, and other outputs that scientists produce in their research. Open Science also aims to make the scientific process more transparent and accessible. Open Science research allows others to collaborate and contribute to the study using freely available research data and all the resources in the research process, which facilitates the reuse, redistribution and reproduction of the research outcome and its underlying data and methods [10].

In Open science, it is essential to share the resources correctly and make the research findings reproducible, since due to shortcomings in many current methods for sharing and capturing data, "approximately 50% of all research data and experiments are considered not reproducible, and the vast majority (likely over 80%) of data never makes it to a trusted and sustainable repository" [4].

Moreover, Open Science is increasingly important in the current science world, beneficial for scientists, patients, and the public and has emerged as a framework to improve the quality of scientific analyses. There are several reasons why Open

Science is necessary. The findings and output of publicly-funded scientific research would be more available and accessible. Therefore the scientific works would be more transparent and reproducible. Also, the public implication would be possible in conducting the research and might impact the results. Also, open peer-review would be possible, encouraging broader and more transparent review processes by extending knowledge exchange between researchers [75, 55].

There are different definitions for the concerns and principles required to follow in Open Science. As mentioned above, Open Science encompasses a variety of areas, including open access to publications, open research data, open-source software/tools, open workflows, open educational resources, and alternative methods for research evaluation, including open peer-review [49]. There is a set of five broad concerns as "schools of thought"; defined by Fecher & Friesike in 2014 [11], to be considered to implement these practices.

These concerns are represented in Figure 1: "Democratic school" believes scholarly knowledge (including publications and data) should be available freely for all. The second one is "Pragmatic school," which aims to make scholarly methods transparent and concerns with efficient knowledge creation through collaboration and critique. The third is "Infrastructure school", mentioning "efficient research requires readily available platforms, tools and services for dissemination and collaboration"[20]. The fourth is "Public school", claiming that the public should collaborate in research and that the scholarship should be more readily understandable through less formal communicative methods. The last one is "Measurement school" that believes it is required to define alternative metrics to track and measure the impact of scholarship.

School of thought	Central assumption	Involved groups	Central Aim	Tools & Methods
Democratic	The access to knowledge is unequally distributed.	Scientists, polititians, citizens	Making knowledge freely available for everyone.	Open access, intellectual property rights, Open data, Open code
Pragmatic	Knowledge-creation could be more efficient if scientists collaborated.	Scientists	Opening up the process of knowledge creation.	Wisdom of the crowds, network effects, Open Data, Open Code
Infrastructure	Efficient research depends on the available tools and applications.	Scientists & platform providers	Creating openly available platforms, tools and services for scientists.	Collaboration platforms and tools
Public	Science needs to be made accessible to the public.	Scientists & citizens	Making science accessible for citizens.	Citizen Science, Science PR, Science Blogging
Measurement	Scientific contributions today need alternative impact measurements.	Scientists & politicians	Developing an alternative metric system for scientific impact.	Altmetrics, peer review, citation, impact factors

Figure 1: Five Open Science Schools of Thought. Figure reproduced from [11]

Wilkinson defines the following widely referred definition for principles of Open

Science in 2016 as FAIR principles [73] which claims that the research should be Findable, Accessible, Interoperable, and Reusable for all users. Many of the currently available platforms for sharing resources are trying to guarantee the FAIR principles. Each one of the principles in FAIR is explained in the background chapter of this thesis.

Open Science affected almost all scientific research studies, particularly neuroscience [3], our main application domain of interest. Open science practices can help to address many challenges in neuroscience. For instance, the answer to many key neuroscience questions can be found when the research findings and outputs are openly shared. In this case, people can also collaborate and help discover innovative solutions for the treatment of brain-related illnesses.

The resources to be shared in neuroscience include the datasets consisting of raw data, pre-processed data or the result of analysis applied to raw or pre-processed data, which should be shared on data sharing platforms. The data-sharing platforms in neuroscience and other health-related areas are mostly following the principles defined by the National Institutes of Health (NIH) [45]. According to these principles, "Data should be made as widely and freely available as possible while safeguarding the privacy of participants, and protecting confidential and proprietary data". LORIS [66] is an open-source framework for storing, processing and sharing behavioural, clinical, neuroimaging and genetic data. Brain-CODE [70], also, is "A Secure Neuroinformatics Platform for Management, Federation, Sharing and Analysis of Multi-Dimensional Neuroscience Data".

The data itself is not enough in neuroscience research studies; there are a set of neuroimaging tools and pipelines for processing the data that should be shared as well. Therefore, to guarantee open science principles in neuroscience, it is required to use platforms for sharing neuroscience datasets and neuroimaging tools and pipelines and the possible outcome of conducted analysis. There are many neuroscience data and tool sharing platforms such as Canadian Brain Research Strategy (CBRS) [63], NeuroImaging Tools & Resources Collaboratory (NITRC) [67], OpenNeuro [40] and Canadian Open Neuroscience Platfrom [1]. Three of these platforms are explained in Chapter 2 of this thesis: NITRC, OpenNeuro, and CONP.

Although there are many platforms for data and tool sharing in neuroscience, it is still required to have a system to help users bridge the datasets and tools together

when they conduct an analysis. In the best scenario, the users can see the outcome of analysis applied on data in current platforms. However, providing a list of possible tools/pipelines for a data/dataset will facilitate the user's analysis process and prevent the confusing and time-consuming process of selecting an appropriate tool/pipeline for the candidate data/dataset. The same scenario would be helpful when the user needs to select data/dataset to execute a candidate tool/pipeline.

The aim of this thesis is to investigate the feasibility of implementing a recommender system to identify the compatible tools/pipelines and data/datasets based on the available records from previous executions. Meaning that, for a given dataset, we would provide a list of compatible pipelines which are supposed to end in successful execution process. Conversely, for a given pipeline a set of compatible datasets would be provided.

There is also a field of research called "dataset search" [7] which provides recommender systems or search engines to find the most related datasets based on the searched key-words. These systems are mostly based on semantic analysis, however, we focus of compatibility of datasets with the given neuroimaging pipelines which also depend on more low level syntactical and infrastructural details.

Consistently with our motivating use case, we focused on the available neuroscience tools/pipelines and data/datasets (from now on 'pipeline' and 'dataset') in the Canadian Open Neuroscience Platform (CONP) available at https://portal.conp.ca/index. The pipelines in CONP are described in Boutiques [14] which is a software library for sharing tools according to the FAIR principles. CONP has distributed datasets in neuroimaging, transcriptomics, genomics, and other related data modalities. More details about the CONP and its available pipelines and the dataset are provided in Chapter 2.

Recommender systems are widely used and increasingly successful in satisfying the users by suggesting the items they might like and helping them select the better choices faster. There have been many successful works on recommender systems such as Netflix [5] for movie recommender and Amazon [61] as product recommenders. There are two main strategies for recommender systems. The first one is Collaborative Filtering [50] which proposes recommendations based on the user-item interactions, recommending an item to a user if similar users like it. The other approach is Content-based Filtering [47] which recommends a user items that are similar to their previous

choices.

It is important to note that the goal of our recommender system is to identify the compatible datasets and pipelines, which is slightly different from the classical use of recommender systems which recommend the most relevant item to a user. This is a restriction of the general concept of recommender system that would adjust to the users' preferences. The ultimate goal of this project is to reach that broad recommender system and consider much more detailed information. However, in this first study we will focus on compatibility that would be a first step toward helping users in their neuroimaging analysis process.

We focused on collaborative filtering for our project to recommend neuroimaging data/datasets and tools/pipelines given the other. In this case, we have a database of the previously executed pipeline-dataset pairs and apply a collaborative filtering model to get the recommended items. We did not select a content-based approach for the recommender system since, in this case, there should be accurate and comprehensive descriptions for all datasets and pipelines.

The descriptions for datasets and analysis pipelines are supposed to be available through their metadata which in many cases are generated manually. Some automated approaches exists to generate metadata [36] for the documents and some types of resources in studies .However, to the best of our knowledge, in neuroscience, such automated metadata techniques have not been applied widely, although it would be an interesting topic to explore in another research.

This thesis is organized as follows. In Chapter 2, we explain the details about the required components and background knowledge. Chapter 3 explains the Canadian Open Neuroscience Platform (CONP). Chapter 4 presents the recommender system proposed in this thesis, which was submitted to the 16th Workshop on Workflows in Support of Large-Scale Science (https://works-workshop.org/). Finally, Chapter 5 expands the conclusions and explains more about the possible future works on this project.

Chapter 2

Background

In this chapter, we expand the explanations about the concepts and required knowledge for this thesis. First, we expand open-science and its principles since the resources and infrastructures we employed in the thesis are compatible with open-science criteria. In section one, we explain the FAIR principles [9] which are defined for open science, and how open science can be helpful for researchers and scientists. We will then explain that open science is highly noticeable in neuroscience in section two and will write about some of the existing data and tool sharing platforms that attempt to guarantee FAIR principles [73]. Recommender systems is a key concept for our project. Therefore, in section three, we will explain the recommender systems, the leading strategies and applications.

2.1 Open Science

Open science is a collection of actions designed to make scientific processes more transparent and results more accessible. Its goal is to build a more replicable and robust science [62]. Open science means that all steps in scientific research (including publications, data, physical samples, and software) should be publicly accessible to all levels of an inquiring society, amateur or professional [74]. In 2016, the 'FAIR Guiding Principles for scientific data management and stewardship' [73] were published and intended to provide guidelines to improve the Findability, Accessibility, Interoperability, and Reusability of digital assets to promote open science. Importantly, these principles apply to 'data' in the conventional sense and to the algorithms, tools,

and workflows that led to that data. All scholarly digital research objects (from data to analytical pipelines) benefit from applying these principles since all components of the research process must be available to ensure transparency, reproducibility, and reusability.

The following explanations about the definitions of FAIR principles are derived from the FAIR website [9].

Findability

Finding the data is the first step of reusing them so metadata and data should be easy to find for both humans and computers, to achieve that, the metadata should be machine-readable for automatic discovery of datasets and services. Findability includes four principles:

F1. (Meta)data are assigned a globally unique and persistent identifier

Data or metadata needs to be assigned a globally unique and persistent identifier to remove ambiguity, make findability more feasible and help others for the citation when they reuse the data. Some data repositories will automatically generate such identifiers for the deposited datasets

F2. Data are described with rich metadata

Rich metadata means that the description of the data should include information about the context, quality and condition, or characteristics of the data therefore, users will be able to find data based on the information provided by their metadata, even without the data's identifier.

F3. Metadata clearly and explicitly include the identifier of the data they describe

This principle is critically important since usually the metadata and the data itself are in separate files. Therefore the association between them should be made explicit by mentioning a dataset's globally unique and persistent identifier in the metadata.

F4. (Meta)data are registered or indexed in a searchable resource

To ensure 'findability,' the data and resources should be discoverable using indexing which will be achieved by F1-F3.

Accessibility

After finding the data, the user needs to know how to access or get the data. There are two required principles to make it happen :

A1. (Meta)data are retrievable by their identifier using a standardized communications protocol

To make the (meta)data retrievable, the protocol is required to guarantee the following principles:

A1.1. The protocol should be open, free, and universally implementable to facilitate data retrieval so that anyone with a computer and an internet connection can access at least the metadata.

A1.2. The protocol should allow for an authentication and authorization procedure, where necessary. Meaning that accessibility does not necessarily mean 'open' or 'free,' and it is required for the data repositories to provide the conditions or instructions under which the data will be accessible, creating an account in repositories for the data user, for instance.

A2. Metadata are accessible, even when the data are no longer available

Accessibility of metadata is related to the registration and indexing issues described in F4.

Interoperability

The data usually need to be integrated with other data interoperate with applications or workflows for analysis, storage, and processing. To make it happen, it is required that:

I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation

"Interoperability typically means that each computer system at least has knowledge of the other system's data exchange formats" [9]. To make this happen, it is required to use commonly used controlled vocabularies, ontologies, and a good data model to describe and structure (meta)data.

I2. (Meta)data use vocabularies that follow FAIR principles

These vocabularies are used to describe datasets and need to be documented using globally unique and persistent identifiers.

I3. (Meta)data include qualified references to other (meta)data

It should be specified in the metadata if one dataset builds on another dataset, additional datasets are required to complete the data, or complementary information stored in a different dataset. In particular, the scientific links between the datasets need to be described.

Reusability

To achieve the final principle in FAIR, it is required that:

R1. (Meta)data are richly described with a plurality of accurate and relevant attributes

It would be much easier to find and reuse data if metadata includes many labels. R1 is related to F2; however, R1 means if the user or machine can decide if the data is actually 'useful' in a particular context. Therefore, metadata should richly describe the context under which the data was generated.

R1.1. (Meta)data are released with a clear and accessible data usage license,

R1.2. (Meta)data are associated with detailed provenance

R1.3. (Meta)data meet domain-relevant community standards

2.2 Open neuroscience

In the world of neuroscience, many attempts have been made to simplify reproducible research, specifically in neuroimaging and functional Magnetic Resonance Imaging (fMRI) [59]. Also, such platforms need to satisfy FAIR principles. Among the existing platforms for open neuroscience, we introduce OpenNuero, NeuroImaging Tools & Resources Collaboratory (NITRIC) and Canadian Open Neuroscience Platform (CONP).

2.2.1 OpenNeuro

OpenNeuro [40, 15], formerly known as OpenfMRI [48], is a free online platform for storing, sharing and analyzing neuroimaging data [15]. Researchers can upload their data to share it with others and download others' shared data; also, they can run analysis pipelines on the data.

Uploading and sharing data

OpenNeuro only accepts datasets compatible with the Brain Imaging Data Structure (BIDS) [17], a standard for organizing and describing MRI datasets. Also, all datasets will be validated to be BIDS compatible before being uploaded to OpenNeuro. In OpenNeuro, users can specify whether their data will be accessible publicly or not; however, the user agrees that the uploaded data will be publicly accessible after 18 months. After uploading the data on OpenNeuro, the users can change the dataset, change metadata, apply versioning on the dataset or make copies of the dataset to guarantee the reproducibility of analysis.

The users can share their uploaded data with other colleagues or researchers and specify the level of access ranging from viewing to administration.

Running analysis on the data

The most exciting feature of OpenNeuro is that the users are able to run further analysis on the data and share the results of that. The user can select one of the containerized analysis pipelines among all available BIDS Apps to apply to the dataset. The only applicable tools are BIDS Apps [16] since, as mentioned above, OpenNeuro accepts only BIDS compatible datasets. After selecting a BIDS App, the

user can set parameters and specify if the analysis should run on the whole dataset or on specific subjects or sessions. Then the user can download all the generated results, use them for higher-level costume analysis and will have access to all logs for the executed analysis and will be able to debug the process if it has failed.

2.2.2 NeuroImaging Tools and Resources Collaboratory (NITRIC)

The Neuroimaging Informatics Tools and Resources Collaboratory (NITRIC) [26] provides a triad of services include a Resources Registry (NITRIC-R), Image Repository (NITRIC-IR) and a cloud Computational Environment (NITRIC-CE) to meet the needs of the neuroimaging researchers.

NITRIC-R

Resources, in this case, are broadly defined to include software, hardware, data, websites, community organizations, etc. NITRC-R gives researchers better and more efficient access to the tools and resources they need, better categorizing and organizing existing tools and resources, facilitating interactions between researchers and developers, and promoting better use through enhanced documentation and tutorials, forums, and updates.

Each NITRC project has a homepage that describes the resources, provides a standard set of resource characteristics (i.e. keywords, license, dependencies, etc.), and provides a standard set of links for the resources (i.e. download, documentation, support, etc.). Each resource page is maintained by the resource administrator, who is responsible for keeping it up to date. For every project, resource administrators are free to enable/disable any functionality and redirect any content to other sources pertinent to the resource developers' needs. Visitors to the NITRC site can search for resources based on keywords, free text and specific capabilities to find relevant resources. Currently, there are more than 1200 registered tools and 43000 registered users in NITRIC-R.

NITRIC-IR

NITRC Image Repository offers a cloud-based federated neuroimaging data storage system for sharing neuroimaging data in DlCOM and NIfTI formats. Currently, it includes thousands of subjects and imaging sessions searchable across over a dozen projects to promote the re-use and integration of valuable NIH-funded data.

The NITRC-IR is built on XNAT [39] and provides sharing infrastructure for images and related data that can be closely integrated with the NITRC-R resources in order to better support, promote, and manage data sharing functions for NITRC-hosted projects. NITRC-R projects can be associated with NITRC-IR (XNAT) 'projects,' which can be interlinked.

NITRIC-CE

The NITRC Computational Environment is a freely downloadable, virtual computing cloud-based platform built upon a NeuroDebian operating system. NITRC-CE preinstalls popular neuroimaging tools such as AFNI, ANTS, FreeSurfer, FSL, C-PAC, and MRIcron into a standardized computational environment to help users analyze their data quickly and easily, (for the complete list of tools, visit this page). This environment can be deployed in the cloud (using Amazon Web Services Elastic Compute Cloud (EC2) or the Microsoft Azure Cloud Computing Platform), or as a virtual machine for local use. To run analysis, NITRIC-CE can access data through NITRIC-IR, NITRIC-R, secure file transfer from outside sources, and file system mounting of AWS S3 resources.

2.2.3 The Canadian Open Neuroscience Platform (CONP)

The Canadian Open Neuroscience Platform includes all the datasets and analysis pipelines that we employed in our project for this thesis. I have contributed to the technical development of CONP web platform and implemented some features there. Also, I am a co-author in CONP paper which is under revision at Scientific Data [1]. Some of the figures or data in this section are derived from either the paper or the website of CONP.

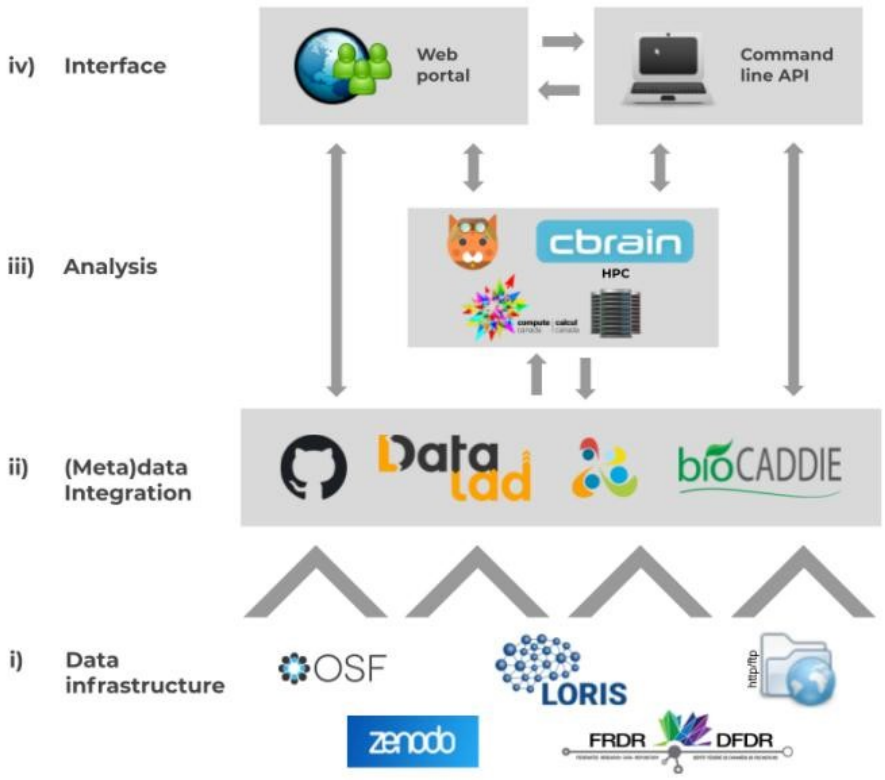


Figure 2: Architecture of the Canadian Open Neuroscience Platform. The platform is comprised of multiple tiers including: i) independent data infrastructure; ii) Metadata integration across tools and datasets via standard models (Biocaddie DATS, Boutiques descriptors); iii) Data analysis on High-Performance Computing and; iv) Web and command-line interfaces. The figure is reproduced from CONP paper.

2.3 Architecture of CONP

The Canadian Open Neuroscience Platform (CONP) provides an infrastructure for the promotion of open-science workflows and the sharing of neuroscience data. There are several open-source technologies integrated in CONP web portal to provide extensible distributed federation of datasets, unified search capabilities for data and software tools, the ability to run analyses either on High-Performance Computing (HPC) infrastructures or locally.

Figure 2 illustrates the architecture of CONP and since we focused on the available datasets and pipelines in CONP in our project, we explain the levels of this structure in following.

2.3.1 Data infrastructure

CONP employes different distributed data repositories with different infrastructures, access control requirements, APIs, and licensing such as domain-agnostic datastores (OSF, Zenodo, FRDR-DFDR), specific brain imaging repositories (LORIS, XNAT, Brain-CODE), and the commonly used HTTP and FTP web protocols. Also, CONP is extensible to any repository which allows access via programmatic web-compatible interfaces.

2.3.2 Data integration

In the (meta)data integration layer, CONP leverages DataLad [24] as a backend, GitHub to host the metadata, and enables uniform data search queries based on the Data Tags Suite model [54]. Datalad is responsible for integration between datasets, it is a software library for managing Git repositories referencing the data through storing metadata, file URLs and hashes of data managed by git-annex. Therefore, a DataLad dataset does not contain the data themselves, the actual datasets remain stored remotely.

Also there is a crawling framework developed in CONP which manages the life cycle of DataLad datasets on GitHub. Using this crawler as web platform, users are able to upload datasets to the CONP without knowledge of Datalad or the GitHub workflow used in CONP. This crawler searches for CONP-tagged datasets and whenever a new dataset is found creates a Datalad dataset for that, and updates the Datalad dataset whenever a modification is detected in a dataset, and then updates the CONP forked GitHub repository. Also generates DATS file with minimal information for a dataset whenever the dataset does not have one.

CONP uses CircleCI as a dataset testing suite to periodically and continuously test if datasets are available, installable by Datalad, and if data are accessible by testing the download of a few files from the datasets. To detect possible issues, CircleCI repeats such tests every four hours for all available datasets in CONP and provides a continuous monitoring for them. Also, since CONP Datalad datasets are hosted in GitHub, the integration with CircleCi would be transparent and more feasible.

2.3.3 Analysis and tools

The analysis layer not only allows finding and downloading of tools, also allows directly integrating tools into workflows and their execution on High-Performance Computing (HPC) systems such as CBRAIN [60, 70]. All tools or pipelines available in CONP are described in Boutiques, which is a software library for sharing tools based on the FAIR principles. Through Boutiques library, the tools and pipelines are described as JSON objects containing the specifications about input data, parameters, and output data. Boutiques descriptors are also linked to a Docker or Singularity container image "where the tool and all its dependencies are installed and configured for execution" [1]. Tools described by Boutiques can be published, archived, and retrieved in the Zenodo and then assigned a DOI, which makes their archives permanently findable.

2.3.4 Interface

All the technologies and methods used in CONP are described as a web portal [65] on which the users can search for, download and upload datasets, tools or pipelines, they are also able to lunch tools on their selected datasets using registered HPC systems such as CBRAIN without requiring advanced computing skills.

Through the analytics available on web interface of CONP, we can see a summary of available contents. As represented in Figure 3, the number of pipelines and datasets has been increased in the last three years. There are currently 57 datasets in CONP, as of July 2021. Each dataset is usually assigned a list of keywords and modalities which describes the category of compatibility of the dataset, Figure 4 illustrates that most of the datasets in CONP are 'neuroimaging'.

Also, according to Figure 3, the number of pipelines has increased steadily in that period of time. Currently there are 75 tools/pipelines which many of them come from neuroscience or genomics research institutes. There are some tags assigned to each tool/pipeline which describes its category of application, as illustrated in Figure 5, pipelines are mostly 'neuroinformatics' and 'mri'.

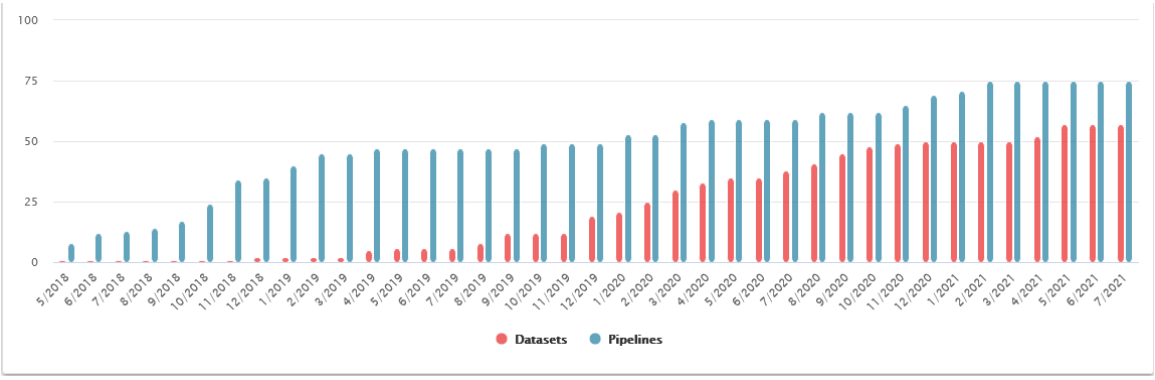


Figure 3: Cumulative number of datasets and pipelines.
(Figure is copied form CONP portal at https://portal.conp.ca/analytics)

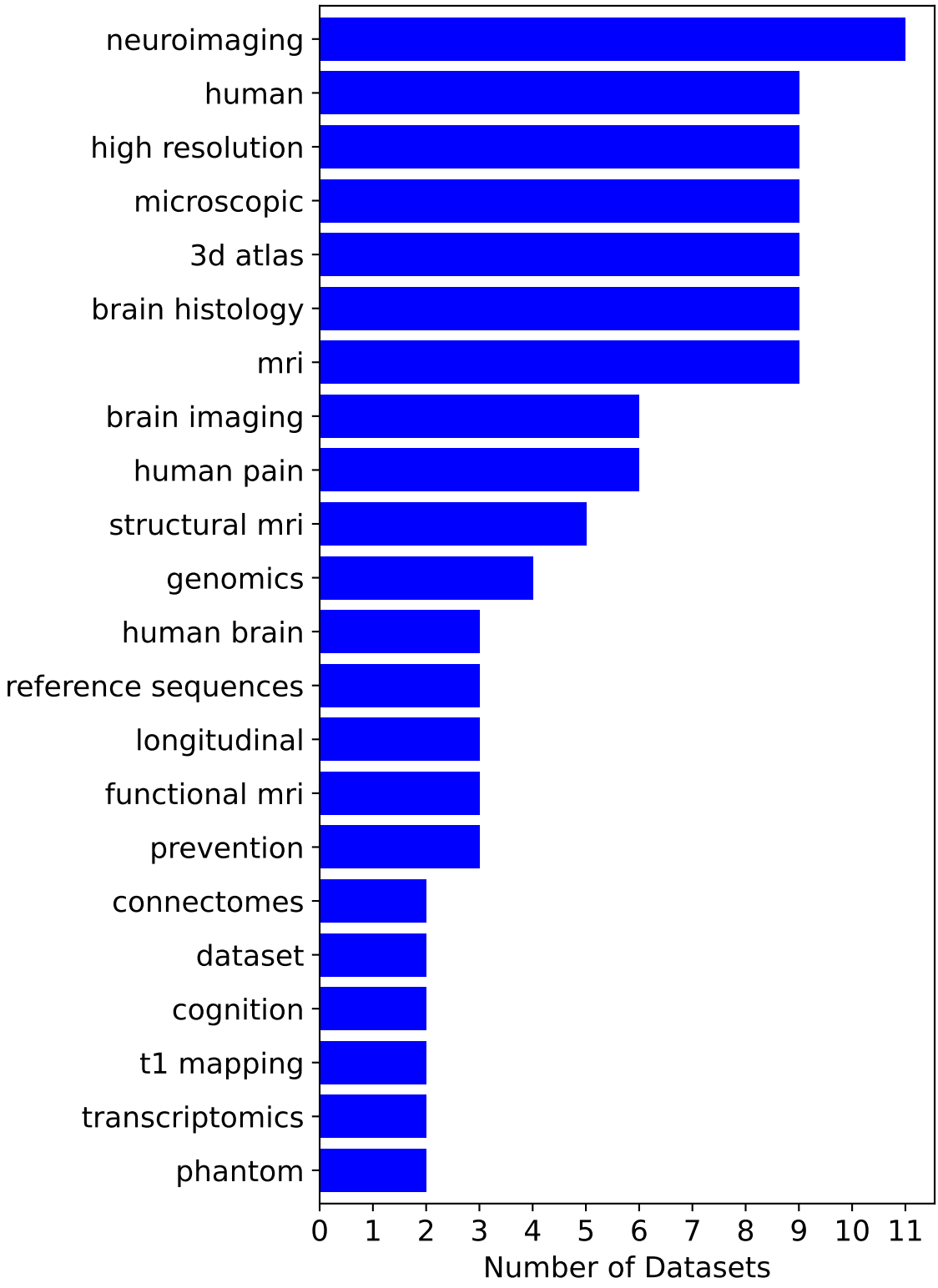


Figure 4: Datasets' keywords. There have been 74 other keywords that each one is assigned to one dataset only

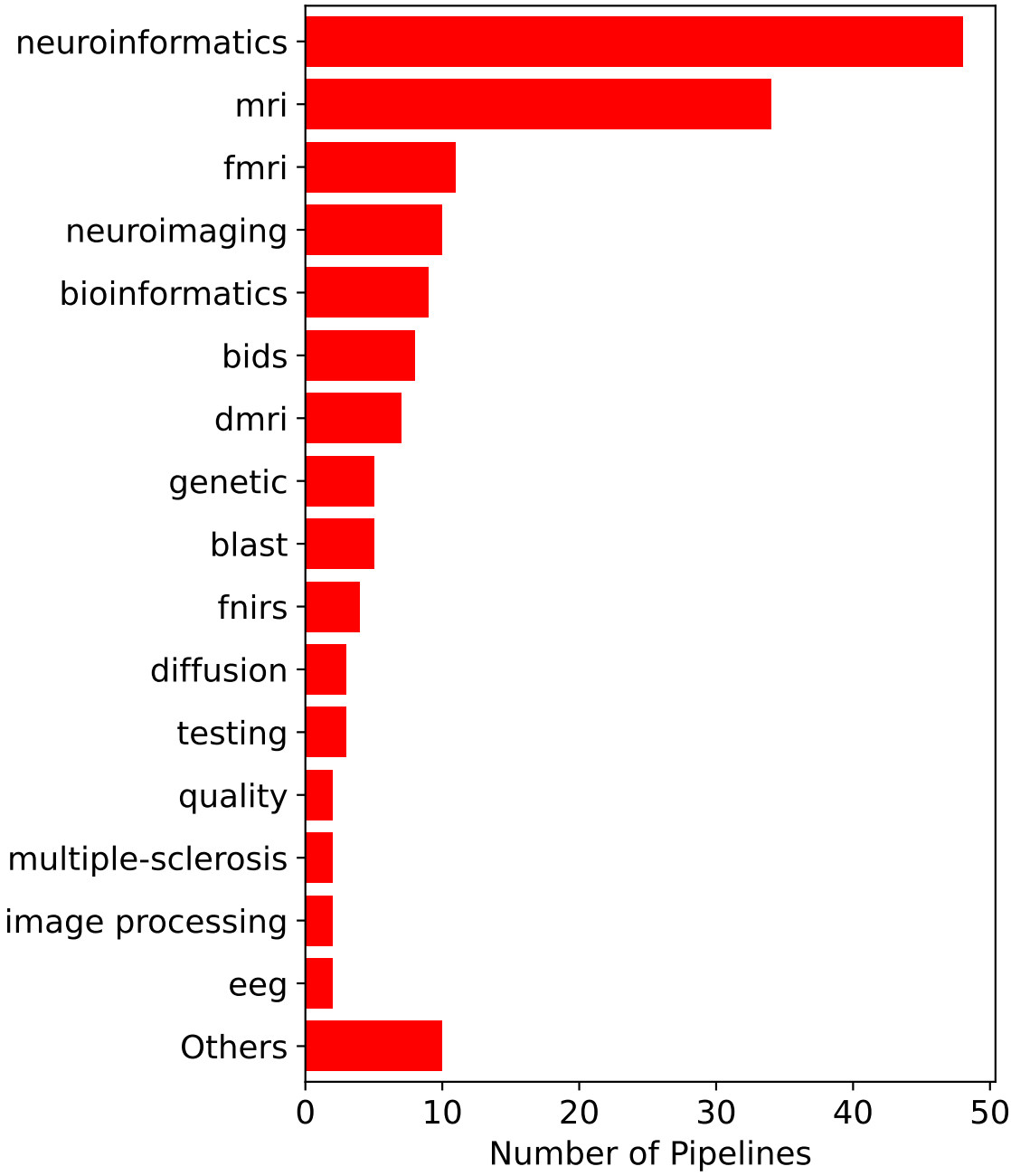


Figure 5: Pipelines' tags

2.4 Recommender Systems

Recommender systems or recommendation systems are included in the information filtering systems which aim at providing suggestions for items to a user by predicting the 'rating' or 'preference' a user would give to an item [23, 53]. These suggestions might come from various decision-making processes, whether or not to buy an item, listen to a piece of music or read an article. Also, 'item' is a general-purpose term referring to what the system will recommend to its users. For instance, it can be a software tool, news or magazine, and products to purchase. The definitions provided in the rest of this section are mostly a summary of Chapter 9 of [50].

Currently, the recommender systems can be seen anywhere that a possible user needs to choose an 'item.' From the most popular recommendation systems, we can point to product recommendations, movie recommendations and news and article recommendations. In product recommendations, online retailers such as Amazon are increasingly attempting to attract and keep their users by suggesting more relevant products that they might like to buy. This will lead to a win-win outcome, finding relevant products for the users and increasing the revenue for the business owners.

Another application of recommender systems is movie recommendations such as Netflix, which recommends to its users the movies or TV shows that they might like based on many factors such as previously watched movies, the ratings (currently 'like' or 'dislike') given or the similar users' choices. There are so many other factors that help Netflix to provide good suggestions for its users. Another application of recommender systems can be in news and articles in which the goal is to identify articles that users would like to read based on what they have read before.

There are several technologies in recommender systems to find the best 'item' for a 'user' that we can classify in two broad groups, Content-based and Collaborative filtering [50]. A content-based recommender system focuses on an item/user profile which describes a set of features and properties of that item/user. Then recommends an item to a user if it is similar to the previous choices of the user. However, a Collaborative filtering recommender system focuses on the relationship between users and items; items will be recommended to a user if similar users prefer it.

Before going through detailed explanations of Content-based and Collaborative Filtering recommendation approaches, we need to define the Utility Matrix and its role in recommendation approaches.

2.4.1 Utility Matrix

In the recommender systems, 'user' and 'item' are the terms used to represent the two main classes of entities. The Utility Matrix is indexed with users' ids on one axis and items' on the other and includes values assigned to each user-item pair, representing the user's preference for the item. In this case, values might represent the item's rating given by the user, 1-5 stars, for example. Usually, the utility matrix is sparse, meaning that the rating value for most user-item pairs is 'unknown,' which implies that there has not been explicit information about the user's preference for that item.

Figure 6 is an example in which the utility matrix represents the ratings (1-5) for movies (HP1, HP2, and HP3 for Harry Potter I, II, and III, TW for Twilight, and SW1, SW2 and SW3 for Star Wars episodes 1, 2, and 3) given by the users (A, B, C, D). The blank units represent unknowns, in which the user has not rated the movie. In practice, the utility matrices are mostly sparser than this matrix since a small fraction of real users gives explicit feedback on items. The goal of the recommender system is to predict the value for the blanks in the utility matrix.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Figure 6: A utility matrix representing ratings of movies on a 1–5 scale. Figure extracted from Chapter 9 of [50]

The utility matrix is the most important requirement to generate a recommendation system. However, it is usually difficult to acquire data from users for the utility matrix. There are two general approaches to gather data from users, first by asking them to rate the items. For instance, asking them to rate after watching a movie or buying a product. However, this this approach is not effective due to the fact that users usually are not willing to provide responses, so the collected information might be biased since it is provided by people who are willing to give ratings. Another

approach would be data acquisition from users interactions and behaviour. For instance, if a user buys a product, watches a movie, or reads an article, it represents that they 'liked' that item. For this sort of rating the only value is 1 (for getting user's response) and the unknown pairs would be 0; in this case 0 is not lower that 1, it is no rating at all.

2.4.2 Implicit and explicit feedback data

Recommender systems rely on different types of input data which are categorized as explicit feedback and implicit feedback. The explicit feedback which is more convenient one, includes the ratings, or any other types of inputs which the users have provided explicitly, such as the stars representing movie rating. However, explicit feedback is not always available and should be captured from investigating the behavior of user which indirectly reflect their preferences such as purchase history, browsing history, search patterns, or even mouse movements [22]. The majority of the literature related to recommender systems are focused on explicit feedback, however, in many practical situations it is required to consider the implicit feedback data.

There are some unique characteristics for implicit feedback data which prevent the direct use of algorithms that are designed for explicit feedback. First, there is no negative feedback, since the implicit data is based on user's behaviours, it is not possible to understand which items the user does not like. For example, not watching a movie does not necessarily mean that the user does not like it, it probably means that they did not know about that, or did not have access to that. In this case, the user-item pairs that there is no gathered data for, usually the vast majority of the data, are treated as "missing data" and will be omitted from the analysis.

Second characteristic is that implicit feedback data are noisy, meaning that the user-item interactions do necessarily mean the user's preference. For instance purchasing might be as a gift, or the user might be unhappy after using that. Third, numerical values in implicit feedback data describe the frequency of actions not user's preference, for instance, how many times a certain movie is watched by a user, or a certain product is ordered by a user. In this case a larger value does not mean a higher preference. "For example, the most loved show may be a movie that the user will watch only once, while there is a series that the user quite likes and thus is watching every week" [22].

And finally, evaluation of implicit-feedback recommenders is more sophisticated than for the explicit feedback. When the items are explicitly scored by the user, clear metrics such as mean square error can be used to measure the success rate of predictions. However,there are many factors to be considered for implicit feedback such as the availability of the item, the repeat feedback, and the competition for an item among others. For instance, the user might have to choose one program among two of his/her favourite shows that will be played at the same time.

2.4.3 Content based

Content-based recommender systems focus on the item's properties described in its profile and then recommends a user the similar items to the ones they 'liked' before. The profile for an item or user is a set of properties or descriptions assigned to item or user. Similarity between items will be obtained by measuring the similarity between their profiles.

The item profile consists of item's properties, there are different classes of items and for each class of them the approach for collecting the properties is different. The first and most convenient way of obtaining features is when the features are explicitly assigned to the item. For instance, the movies usually are assigned with the number of stars indicating the level of users' preference, and other properties such as year of production, name of director and actors and the genre. Another example can be the products which usually are described by the manufacturer with relevant features.

There are other classes of items which the values of features are not immediately apparent, such as documents. Documents, articles and web pages, consist of thousands of words but do not tend to have readily available assigned information to be used to find their topic. To characterize this class of items, it is required to remove the stop words (such as 'is','they') and then compute the TF.IDF score [51] for all remaining words, the top n words with highest scores will characterize the document.

In content-based recommendation, there should be item profile and user profile [50]. Item profile consists of feature-value pairs and user profile summarizes the preferences of the user. There would be a feature matrix with columns that are features, the values for features can be Boolean or numerical, each row represents an item's profile as a vector of its features. The user profile should represent the preference of the user about items, which can be achieved by aggregation of the profiles of

items which user has 'liked'. Then for estimating the degree to which a user would prefer an item, the cosine distance between the user's and item's vectors should be computed, the less distance represents the more chance of preference.

2.4.4 Collaborative filtering

Collaborative filtering is one of the most successful approaches for recommender systems which identifies new user-item associations by analyzing the relationships between users and interdependencies among items (products) [29]. The fundamental assumption of Collaborative Filtering is that users with similar preferences in the past are likely to have similar preferences in the future [64]. "A major appeal of collaborative filtering is that it is domain free, yet it can address data aspects that are often elusive and difficult to profile using content filtering" [29]. There are three main categories of Collaborative Filtering, Memory-based, Model-based and Hybrid collaborative filtering (which combines Collaborative Filtering with other techniques).

In Memory-based Collaborative Filtering algorithms the assumption is that every user is from a group of users with similar interests. Therefore prediction of preferences of a new user on a new items can be produced by identifying the neighbors of that user. Neighbor-based collaborative filtering (kNN) is the prevalent algorithm in this class. Although memory-based Collaborative Filtering approaches are identified as easy-to-implement and highly effective, they have many limitations including the fact that user-item ratings should be stored in memory, adding a new items is sophisticated, and the performance decreases when the data is sparse since the similarity values are based on common items.

To overcome the limitations of memory-based collaborative filtering and providing better performance, the model-based collaborative filtering was investigated [64, 31]. Model-based collaborative filtering techniques use the pure rating data to learn/train a model which can later be used for predictions. In this class the model can be data mining or machine learning algorithms. Also, since it is not required to have data stored in memory, the prediction process can be fast and even using less data than the original. Among the possible techniques, the Latent Factor based algorithms are proven to be effective to address the scalability and sparsity challenges of collaborative filtering tasks.

Among the techniques for latent factor models, some of the most successful ones

are based on matrix factorization [29] which aims to extract meaningful latent connections between users and items based on user-item rating patterns. Recommender models based on Matrix factorization "map both users and items to a joint latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space". Each item i is associated with a vector of factors (features) $q_i \in \mathbb{R}^f$ which represents how much (positive or negative) this item contains those factors; similarly, each user u is associated with a vector of factors(features) $p_u \in \mathbb{R}^f$ which measure the extent of interest the user has in items that are high on the corresponding factors.

To learn the latent factors (p_u and q_i), more recent works suggested modeling based on the known ratings, therefore, the regularized squared error should be minimized on the set of known ratings:

$$\min_{q_*,p_*} \sum_{(u,i)\epsilon\kappa} \left(r_{ui} - q_i^T p_u\right)^2 + \lambda \left(\|q_i\|^2 + \|p_u\|^2\right) \tag{1}$$

where κ is the set of user-item pairs for which r_{ui}, the rating of item i by user u, is available. Unknown elements of the utility matrix are predicted by the value of $q_i^T p_u$ which represents the user's overall interest in the item's characteristics and is an approximation of the rated value if item i by user u.

2.4.5 Evaluation of Recommender systems

To evaluate the result of the recommender engines, there are two classes of metrics, statistical and decision support accuracy metrics [57]. In statistical accuracy metrics, the predicted rating will be compared directly with the actual user ratings. Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Correlation are usually used as statistical accuracy metrics [23], also precision at k (prec@k) "which counts the number of relevant items in the top-k positions of a ranked list" is included in this class of evaluation [38, 25].

However, the decision support accuracy metrics distinguish good items from those items that are not good, like a binary operation. The popular measurements in decision support include Receiver Operating Characteristics (ROC) and Precision-Recall Curve (PRC), Precision, Recall and F-measure. We focussed on Receiver Operating Characteristics (ROC) since we had a utility matrix of binary values and needed to classify the predictions to fail or succeed.

Chapter 3

A Recommender System for Scientific Datasets and Analysis Pipelines

This chapter is under review at the 16th Workshop on Workflows in Support of Large-Scale Science: Mandana Mazaheri, Gregory Kiar, Tristan Glatard, "A recommender system for scientific datasets and analysis pipelines". My contribution was to implement the system, design and conduct all the experiments, analyse the data and write the initial draft of the paper. The contribution of Gregory Kiar was to provide input on the experiment design and editing the manuscript. And the contribution of Tristan Glatard was to provide input on the experiment design, editing the manuscript and overall guidance to the project .

3.1 Introduction

Open science has emerged as a framework to improve the quality of scientific analyses, ideally leading to Findable, Accessible, Interoperable, and Reusable (FAIR [73]) datasets and analysis pipelines. In neuroscience, our main application domain of interest, platforms have emerged to facilitate the sharing of datasets and pipelines, including OpenNeuro [15], NeuroImaging Tools and Resources Collaboratory (NITRC [26]), and the Canadian Open Neuroscience Platform [1]. However, while public datasets and pipelines proliferate, researchers remain unassisted in creating relevant

analyses from these resources that therefore remain largely underutilized. Identifying the set of analysis pipelines that may be relevantly applied to a given dataset, or conversely the list of datasets that may be processed by a given pipeline, remains challenging. In this paper we investigate the development and use of recommender systems to address this issue.

The past decades witnessed the adoption of recommender systems as the major technology to help customers navigate the abundant product offerings of online retail platforms. Two main recommending strategies emerged: content-based strategies, which match content-rich item profiles with user profiles [47], and collaborative filtering, which recommends items to a given user based on those selected by users with similar preferences [50]. Both strategies have been successfully applied and have their own strengths and weaknesses.

While these techniques have largely been employed in the space of retail content delivery, we evaluate the feasibility of matching scientific pipelines and datasets using existing recommender system techniques. We focus on collaborative filtering approaches, since content-based methods would require extensive annotations about pipelines and datasets which are not broadly available despite on-going efforts [19, 56, 54]. However, collaborative filtering requires an affinity measure between users and items. To define such a measure between pipelines and datasets, we rely on past execution outcomes (e.g. exit status) available through provenance records.

The compatibility of a pipeline with a dataset depends on semantic, syntactic, and infrastructural factors. Semantic factors refer to the content of datasets and analyses. For instance, a pipeline developed for the segmentation of brain Magnetic Resonance Images (MRIs) would not produce meaningful results on other image types. Syntactic factors refer to file formats and dataset organization. For instance, brain images are commonly stored using the NIfTI [34] or MINC [72] formats and pipelines developed to ingest one format may not apply to others. In addition, the multiple files and directories composing a dataset are increasingly structured using the Brain Imaging Data Structure (BIDS [18]) which is required by some pipelines while other ones use their own structure or neglect structure entirely and require explicit pointers to specific files. Finally, infrastructural factors refer to the availability of pipelines and datasets which can be functionally deployed. For instance, some pipelines may require the loading of an entire dataset in memory, which may not be feasible for large

datasets. All these levels must be considered in pipeline or dataset recommendations.

Provenance is a key concept in computational analyses, referring to the detailed description of data transformations. Provenance records typically include information about the input data, the analysis software and parameters, the execution context and finally, the execution outcomes. In neuroimaging, the NeuroImaging Data Model (NIDM [41]) project proposed domain-specific formats and tools based on standards from the W3C PROV working group [43]. In its current form, our recommender system merely relies on execution outcomes (summarized via "exit codes") extracted from these or similar provenance records.

To summarize, this paper makes the following contributions:

- Describes and presents a provenance-based recommender system for scientific pipelines and datasets;
- Evaluates the system for datasets and pipelines from the Canadian Open Neuroscience Platform;
- Compares the system against domain expert recommendations.

3.2 Related Work

Systems have been used to recommend software in various contexts such as workflow composition and algorithm selection. However, as explained below, our context is slightly different since we aim at recommending analyses that are applicable to a given dataset. In neuroimaging, existing platforms focus on finding and reusing pipelines and datasets but do not include any recommender engine.

3.2.1 Workflow composition

Recommender systems have been described to assist users with workflow composition, in particular to identify candidate software components for a given workflow. For example, the Galaxy tool recommender [30] recommends possible workflow components using a deep learning model trained on $18,000$ bioinformatics workflows from the European Galaxy server. Recommendations depend on the definition and organization of all the tools in the workflow (so-called "higher-order workflow dependencies") instead of focusing only on the most recently added workflow components.

Workflow dependencies are learned using Recurrent Neural Networks, resulting in a mean accuracy of 98%. The system is available for Galaxy users as an extension.

Previous approaches to assisted workflow composition included loose programming, initially proposed in the PROPHETS system [33, 44, 32]. Loose programming enables workflow developers to program using concepts rather than accurate procedural code. Loose programming exposes to the workflow developers semantic annotations describing the functionalities of workflow components. Workflow developers can then assemble such components without having to care about correct typing, interface compatibility, platform parameters or other technical details that are taken care of through subsequent validations. PROPHETS was applied to various bioinformatics use cases, including mass spectrometry-based proteomics [46].

The WINGS (Workflow INstance Generation and Specialization) system [13] uses AI planning and semantic reasoners to assist users in creating workflows while validating that the workflows comply with the requirements of the software components and datasets. WINGS can reason about the constraints of the components and the characteristics of the data and propagate them through the workflow structure.

These approaches assist users by matching workflow components together. Instead, in our context, the workflows (or pipelines) are already available and need to be matched to relevant datasets. Conversely, relevant analysis workflows need to be recommended for a given dataset.

3.2.2 Algorithm selection

Other recommender systems aim at selecting specific algorithms for a given problem. For instance, the Oracle machine-learning toolkit [42] selects machine-learning algorithms and models for classification and regression problems. Algorithm selection uses advanced machine learning techniques to automatically rank the best algorithms for a dataset. Model selection identifies the best hyperparameters to maximize a given prediction performance score.

Another recent example is PennAI [31], a platform that recommends suitable models given a supervised classification problem. The platform was evaluated on 165 classification problems. Results showed that matrix factorization-based recommendation systems outperform meta-learning methods.

In addition, Dyad ranking [68, 69] represents the algorithm and problem instance

by a feature vector, and selects the best feature vectors using machine learning. The training dataset is a set of dyads ranked according to a specific preference relation. The dyad ranker learns using the neural-network-based PLNet [58] algorithm. Results show that this approach outperforms many algorithm selectors while using less computation.

All these approaches assume the existence of an objective function such as F1 score, accuracy or mean average error to compare algorithms or models that are known to apply to the problem. Instead, our goal is not to compare pipelines with each other but to predict if a given pipeline will work on a given dataset. From a methodological point of view, algorithm selection techniques are mostly content-based while we will adopt a collaborative filtering approach.

3.2.3 Finding tools and datasets in neuroimaging

Several platforms have been developed to facilitate the sharing of tools and datasets in neuroimaging. For instance, NITRC [26] provides a richly-annotated catalog of tools and datasets that can be processed in the NITRC computing environment. In OpenNeuro [40], datasets complying to the BIDS data structure can be publicly shared and processed using BIDS apps. Finally, the Neuroscience Information Framework (NIF [12]) is a powerful search engine for neuroscience software and data. While all these platforms provide substantial services for data and tool sharing, they do not seem to include any system to assist users in the matching of pipelines and datasets. Matching relevant pipelines and datasets would also enable the automated triggering of data processing when new pipelines or datasets become available.

3.3 Materials and Methods

The goal of our recommender system is to predict if a data processing pipeline will successfully run on a given dataset. Predictions are obtained from provenance records created from previous pipeline executions. Figure 7 presents an overview of our system that is detailed in the remainder of this section with our experimental methodology.

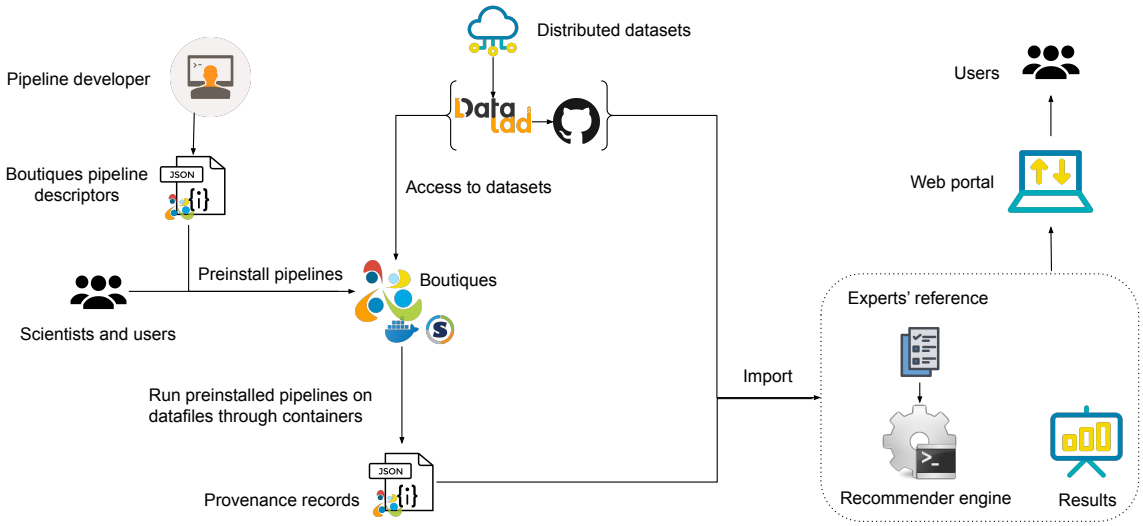


Figure 7: Overview of our recommender system

3.3.1 Data processing pipelines

Consistently with our motivating use case, we focused on the 64 neuroscience pipelines available in the CONP as of October 2020. The available pipelines have 23 different tags which shows an overview of the type of data and analyses supported by these pipelines. The most popular tags are "neuroinformatics", "mri", "fmri", "bioinformatics" and "dmri". These pipelines are command-line tools described by Boutiques descriptors [14] and containerized using Docker or Singularity, which makes them portable across a wide range of infrastructures, including local workstations, clusters, and clouds [27]. Each pipeline descriptor is stored on the Zenodo research archive [8] and identified by a permanent Digital Object Identifier (DOI). Through the Boutiques command line, the pipelines can be validated, installed, published, and executed. When the execution completes, Boutiques creates a JSON provenance record containing a summary of the execution process including the pipeline DOI, input and output file hashes, parameter values, and exit code. Our recommendation system uses such provenance records to track the outcome of pipeline executions on a given dataset.

3.3.2 Datasets

We used the 42 datasets available in the CONP as of October 2020, representing a total of 2,807,267 files, 3,078 GB, and 6,330 subjects. These datasets are coming from various sources distributed in Canada and abroad. The most frequent dataset keywords are "brain imaging", "human pain", "MRI", "neuroimaging" and "structural MRI". Datasets are made available through DataLad [24], a Git-based framework that provides a uniform and version-controlled view of distributed storage.

A DataLad dataset is a particular type of Git repository that stores data files using git-annex [21]. Git objects contain hashes and URLs of the data files but not the data itself. With the DataLad client, users can download specific files and track their versions. Using DataLad, our system matches provenance records to particular datasets through file hashes. In addition, specific files from a given dataset can be downloaded on demand without having to download the entire dataset.

In some cases, minor adjustments such as file renamings or exclusions (on 8/22 of tested datasets) to the organization of BIDS datasets were performed to make them fully compliant with the standard specification and reduce the processing time of our experiments by excluding some subjects. We did not apply any data type conversion since in neuroimaging they are known to create issues when not done properly [35].

3.3.3 Expert reference

To build an expert reference, we recruited 13 experts among graduate students, software developers and data engineers at the Canadian Open Neuroscience Platform. Since the number of pipeline-dataset pairs to evaluate was beyond the amount that could reasonably be evaluated by a human expert (2,688), we split our survey in two steps. In the first survey (confidence survey), experts were asked to rate their knowledge and confidence about each pipeline and dataset on a 4-level scale: no knowledge, some knowledge, good knowledge, and expert knowledge. In the second survey (prediction survey), experts were ask to predict the execution outcome (success or failure) for all pipeline-dataset pairs in which they had indicated good or expert knowledge in the first survey for both the pipeline and the dataset. Both surveys included links to dataset and pipeline description pages on the CONP portal, such that the experts were able to consult their detailed descriptions. Survey forms are available on GitHub for more information. Surveys happened between November

2020 and March 2021.

It should be noted that the experts were supposed to predict the compatibility of a given dataset and pipeline and not to what extent they are relevant which might lead to priming or evaluation bias [71].

3.3.4 Provenance records

Similar to pipeline descriptors, Boutiques provenance records can be published to Zenodo, which makes them publicly and permanently accessible. We created provenance records for each pipeline-dataset pair for which at least one expert predicted successful execution. To generate a provenance record we executed a pipeline using an invocation file including all required parameters for that pipeline. The instructions for generating the invocations are available through Boutiques for each pipeline. We mostly used default parameters and created minimal invocations, however, to execute some pipelines we requested domain experts' assistance to generate a working invocation file. The provenance records were entered in a database together with the file hashes of all datasets retrieved using DataLad. From this database, we generated (pipeline, dataset, execution outcome) triplets to use in our recommender system.

3.3.5 Recommender system

Collaborative filtering predicts the rating of a given item (dataset in our case) by a given user (pipeline in our case) from the "utility matrix" containing previous ratings [50]. Two approaches are commonly used: neighbor-based methods [28] and latent factor models [29, 6]. Neighbor-based collaborative filtering, also known as k-nearest neighbors, identifies like-minded users or similar items based on the similarity of entries in the utility matrix. In contrast, latent factor models, the method that we used, represent items and users in a latent space obtained from a factorization of the utility matrix r in a user matrix p and an item matrix q. The factorization is learned by minimizing the least square error between the available ratings and the ratings predicted by the factorization:

$$\min_{q_*,p_*} \sum_{(u,i)\epsilon\kappa} \left(r_{ui} - q_i^T p_u\right)^2 + \lambda \left(\|q_i\|^2 + \|p_u\|^2\right) \tag{1}$$

where κ is the set of user-item pairs for which r_{ui}, the rating of item i by user u, is available. Unknown elements of the utility matrix are predicted by the dot product of the corresponding vectors in q and p. In our case, pipelines represent users, datasets represent items, execution outcomes represent ratings, and κ is the set of pipeline-dataset pairs for which provenance records are available. Two minimization methods are commonly used: stochastic gradient descent and alternating least squares (ALS). We used ALS as implemented in the Apache Spark Machine Learning library (spark.ml) version 3.1.2.

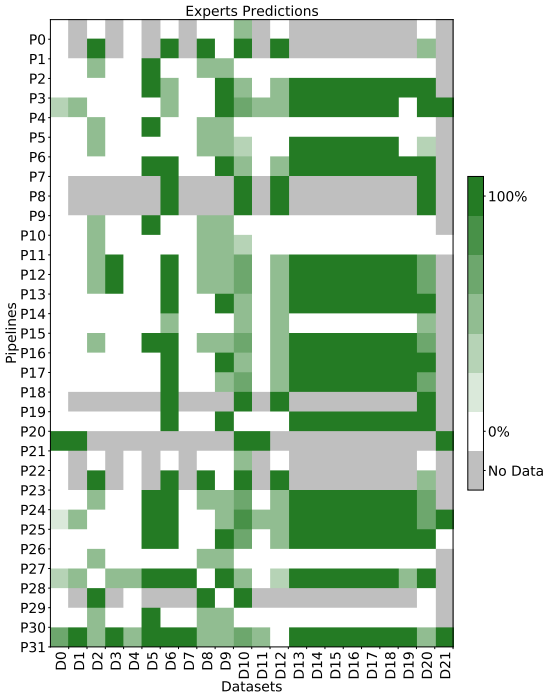
3.4 Results

We evaluated our approach in two different ways. First, we compared the expert predictions to real execution outcomes extracted from provenance records. Second, we evaluated the accuracy of our recommender system through 10-fold cross validation. The data and code required to reproduce our results are available at https://github.com/big-data-lab-team/paper-pipelines-datasets-recommender.

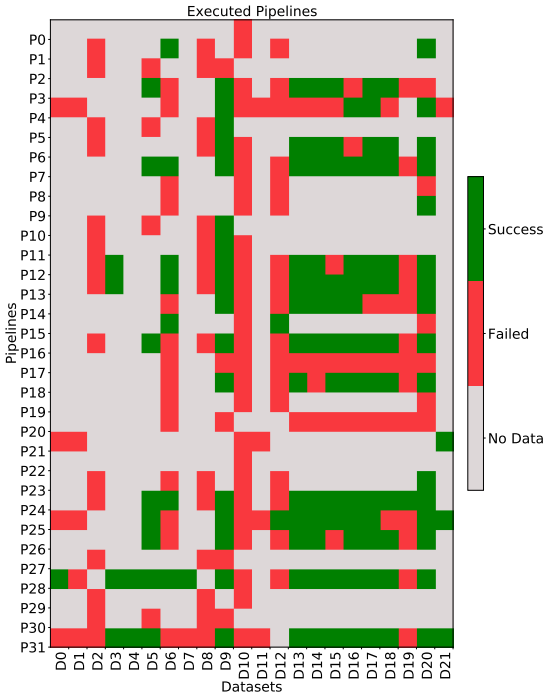
3.4.1 Expert predictions vs real executions

Figure 8a shows expert predictions of pipeline-dataset execution outcomes. The average number of expert predictions by pipeline-dataset pair was 1.39. Only the 32/64 pipelines for which at least one dataset was predicted to be successfully processed by at least one expert are represented. Similarly, only the 22/42 datasets for which at least one pipeline was predicted to be successfully executed by at least one expert are represented. Pipeline and dataset names are reported in Tables 1 and 2. Entries in these tables are clickable for more information. Out of a total of 704 pipeline-dataset pairs, the execution outcome of 37% was predicted as failed by all the experts (white cells), the outcome of 25% was predicted as successful by all the experts (dark green cells), 21% were not known with enough confidence by any expert (gray cells), and the outcome of the remaining ones was predicted as successful by some experts and as failed by other experts. The large fraction of pipeline-dataset pairs unknown to any expert reinforces the motivation for an automated recommender system.

Figure 8b shows the actual execution outcome for all the pipeline-dataset pairs



(a) Experts predictions



(b) Pipeline outcomes

Figure 8: Expert predictions vs real execution outcomes.

Index	Pipeline Name
P_0	ApplyTOPUP
P_1	ApplyWarp
P_2	BIDS App – FSL Diffusion Preprocessing
P_3	BIDS App - FreeSurfer 6.0
P_4	BIDS App - fmriprep
P_5	BIDS App - ndmg
P_6	BIDS app example
P_7	ANTS Brain Extraction
P_8	ANTS Concat Transfo
P_9	ANTS Cortical Thickness
P_{10}	DTIFit
P_{11}	Dipy Tracking and Connectome Generation
P_{12}	FLIRT
P_{13}	FNIRT
P_{14}	FreeSurfer-Recon-all
P_{15}	FreeSurferPipelineBatch-CentOS7
P_{16}	FslBet601
P_{17}	ICA_AROMA
P_{18}	MCFLIRT
P_{19}	MRIQC
P_{20}	PreFreeSurferPipelineBatch
P_{21}	SPARK (stage 1 of 3)
P_{22}	TOPUP
P_{23}	fsl_anat
P_{24}	fsl_bet
P_{25}	fsl_fast
P_{26}	fsl_first
P_{27}	fsl_probtrackx2
P_{28}	fslstats
P_{29}	mask2boundary
P_{30}	ndmg
P_{31}	oneVoxel

Table 1: Tested pipelines

Index	Dataset Name
D_0	BigBrain
D_1	BigBrain_3DClassifiedVolumes
D_2	Comparing_Perturbation_Modes _for_Evaluating_Instabilities _in_Neuroimaging__Processed_NKI _RS_Subset__08_2019_
D_3	BigBrainHippoUnfold
D_4	BigBrainMRICoreg
D_5	HCPUR100-Template
D_6	Learning_Naturalistic_Structure _Processed_fMRI_dataset
D_7	MRI_and_unbiased_averages _of_wild_muskrats_ _Ondatra_zibethicus_ _and_red_squirrels_ _Tamiasciurus_hudsonicus_
D_8	Numerically_Perturbed_Structural _Connectomes_from_100_individuals_ in_the_NKI_Rockland_Dataset
D_9	SIMON-dataset
D_{10}	cneuromod
D_{11}	mm_neo_atlas
D_{12}	multicenter-phantom
D_{13}	openpain/BrainNetworkChange_Mano
D_{14}	openpain/cbp_resting
D_{15}	openpain/placebo_1
D_{16}	penpain/placebo_predict_tetreault
D_{17}	openpain/subacute_longitudinal_study
D_{18}	openpain/thermal
D_{19}	preventad-open
D_{20}	preventad-open-bids
D_{21}	visual-working-memory

for which at least one expert predicted a successful execution outcome (green cells in Figure 8a). Out of 288 executed pairs, 134 were successful and 154 failed. Important discrepancies are observed between expert predictions and actual executions. Overall, 53% of the executions that were predicted successful by at least one expert failed in reality (red cells in Figure 8b). In addition, the average expert confidence was found to be significantly higher for failed executions than for successful ones ($p < 0.002$, Figure 9), which is unexpected. Therefore, expert predictions seem to be largely unreliable. Note that we used the experts' predictions as a baseline for prediction performance comparison rather than a ground truth on the execution outcome of a given pipeline on a given dataset.

Many practical reasons explain the observed discrepancy between expert predictions and pipeline executions (Table 3). First, some datasets did not match the format required by the tested pipeline. For instance, P_{26} (fsl-first) requires anatomical images in the NIfTI file format, however, some datasets such as D_{12} (multicenter-phantom) contain anatomical images in the MINC format.

Failure reason	**Fraction of failed executions**
File format not supported	31%
Type D pre-processing required	18%
Dataset not available	38.5%
Other	12.5%

Table 3: Execution failure causes

In addition, five pipelines (P_8, P_{17}, P_{19}, P_{20} and P_{27}) failed due to unresolved pre-processing requirements. We identified four types of dependencies between pipelines. Type A refers to pipelines such as P_{24} (fsl-bet) or P_{23} (fsl-anat) that can be executed directly on the tested dataset. Type B refers to pipelines that require the tested dataset as well as the results of the application of another pipeline on the tested dataset. For example, P_{31} (oneVoxel) requires a binary mask for its input image that is created by another pipeline. Type C refers to pipelines requiring inputs from more than one dataset. For instance, P_7 (ANTS Brain Extraction) and P_9 (ANTS Cortical Thickness) require external templates and segmentations obtained outside of the dataset. Type D refers to pipelines that process data derived from the tested

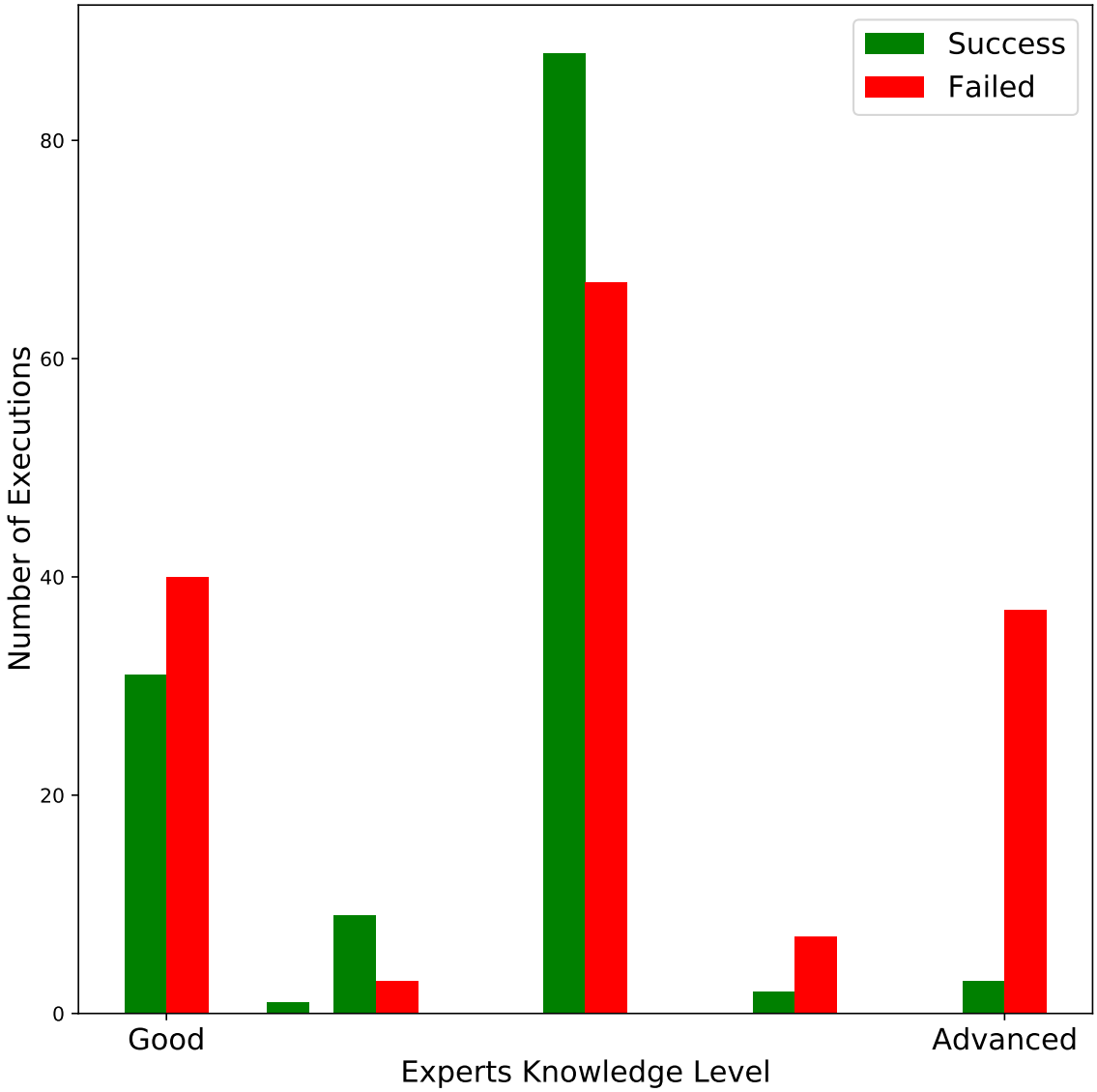


Figure 9: Expert confidence by actual execution outcome.

dataset but not the tested dataset directly. For instance, P_{27} (fsl-probtrackx2) performs probabilistic tractography on the output of bedpostx, a pipeline that no expert predicted to run successfully on any dataset. In a Type D configuration, the pipeline is considered to not successfully execute on the dataset.

In addition, 4 datasets were not available for download in CONP, due to various issues. For example, D_{10} (CNeuromod) is currently not downloadable in CONP due to technical issues. Finally, a set of executions failed for other reasons including issues in Boutiques pipeline descriptors or corrupted datasets.

Overall, experts seem to have neglected such practical failure reasons. In general, experts tend to rely on their semantic understanding of the interactions between pipelines and datasets (for instance, a given pipeline may operate on fMRI data), while in practice, pipeline executions depend on the lower-level syntactical and infrastructural details mentioned previously.

3.4.2 Recommender system evaluation

We evaluated the latent-factor model using 10-fold cross validation on the pipeline execution matrix in Figure 8b. We varied the threshold used to round predicted values to 1 (failed execution) or 2 (successful execution), resulting in the Receiver Operating Characteristic (ROC) curve in Figure 10. We obtained the ROC curve of experts predictions by predicting execution outcomes using various thresholds in the fraction of experts predicting successful execution.

The area under curve (AUC) of our recommender system was 0.83, showing that our model is significantly better than chance (AUC=0.5) and expert predictions (AUC=0.63). For instance, given a rounding threshold of 1.2 (black dot in the ROC curve), out of 10 pipelines recommended by our system for a particular dataset, 8 would be applicable to the dataset while only 2 would not. This good performance was expected to some degree given that the pipeline execution matrix in Figure 8b bears some sort of structure. A more random utility matrix would obviously be more difficult to predict.

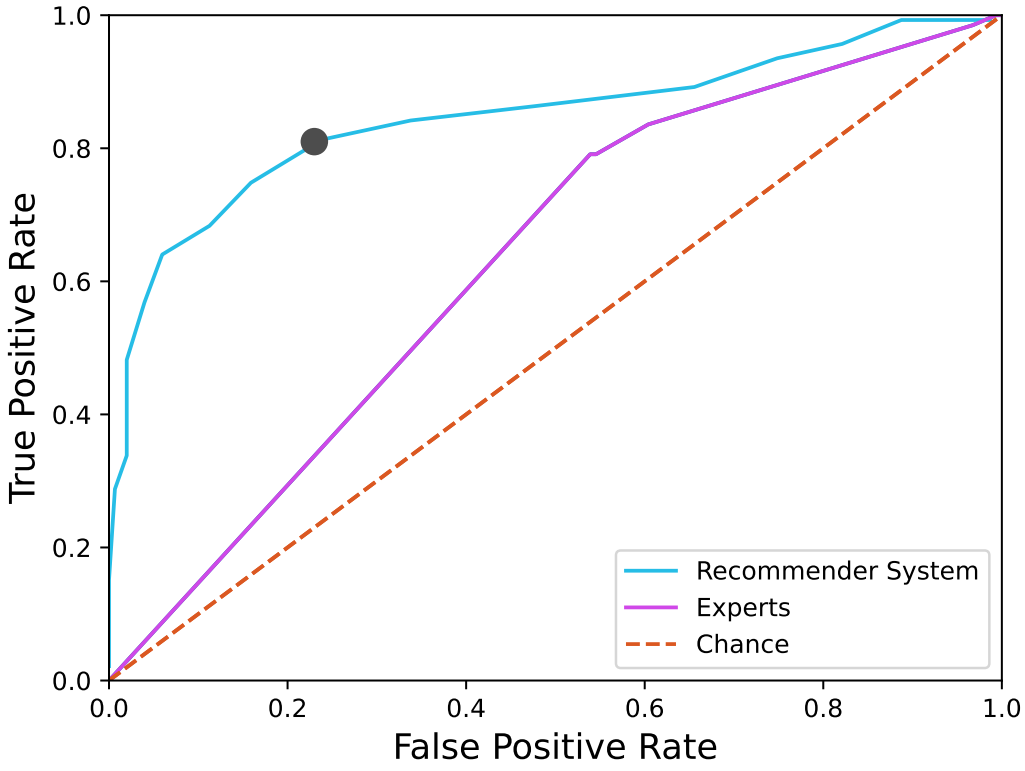


Figure 10: ROC curves of experts and recommender system predictions.

3.5 Discussion

The performance of the proposed recommender system is substantially higher than chance and expert predictions. Predicting the successful execution outcome of a pipeline on a given dataset is a difficult task for a human expert as it requires a comprehensive knowledge and understanding of the technical infrastructure, pipeline syntactical requirements, analysis types, data formats, and data semantic types. In practice, it is common for human experts to only master some parts of this environment. For example, the successful processing of BIDS datasets by BIDS applications requires datasets to pass BIDS validation, which can hardly be guessed from a high-level overview of the dataset. In addition, pipelines or data transfers may fail for technical reasons unknown to the experts. Therefore, automated recommender systems based on provenance records have a strong added value compared to human recommendations.

Our experiments were conducted using one of the largest data and pipeline sharing platform in neuroscience. Other platforms such as NITRC [26] contain larger collections of pipelines and datasets, but they are not available through a consistent interface such as Boutiques and DataLad, which would make such an experiment hardly feasible there.

The described system architecture could potentially scale widely beyond the specific context of CONP, as it only relies on file hashes and therefore does not require data sharing agreements or extensive data storage. In addition, the recommender system could leverage provenance records produced by multiple platforms provided that they are shared in some way. DataLad would detect possible duplication among datasets through file hashes. The framework is also expected to apply to other disciplines than neuroimaging although changes in the technical context — such as datasets being stored in databases instead of files — may require adaptations.

In production conditions, the recommender system would rely on provenance records of pipeline executions launched by arbitrary users. While this would increase the amount of data available, potentially resulting in more accurate recommendations, it would also come with challenges. For instance, while our framework models the execution outcome of a given pipeline-dataset pair as a binary variable (success or failure), different execution outcomes may be produced for a given pipeline-dataset pair, due to different parametrizations or analysis types. Besides, analyses launched

by less experimented users may produce misleading provenance records. We anticipate that the recommender could be configured to use implicit feedback to address this issue [22].

The successful execution of a pipeline on a given dataset does not necessarily imply that results are meaningful. Relying exclusively on execution exit statuses therefore requires that users producing execution records mostly execute meaningful experiments, which may not always be the case. Taking into account the popularity of the datasets derived from a given provenance record in the recommendations might help address this issue.

3.6 Conclusion

Collaborative filtering predicts the execution outcome of a given pipeline on a given dataset with usable accuracy (AUC=0.83) in the context of the Canadian Open Neuroscience Platform. The performance achieved by our system outperforms human expert recommendations, presumably due to syntactical and infrastructural factors neglected by human experts. Future work will focus on the deployment of such a system in production conditions, which will require dealing with less reliable provenance records.

The framework could be extended by considering pipelines and datasets at a finer granularity. Pipelines can often be used in different ways depending on their parametrization. Different parametrizations could be identified in the provenance records and recommended accordingly for specific datasets. Besides, datasets often consist of multiple sub-parts corresponding to different subjects or data types. A recommender system could be designed to recommend analyses for such sub-parts, resulting in more specific recommendations.

Chapter 4

Conclusion

In this thesis we investigated if it would be feasible to implement a provenance-based recommender system for recommending compatible scientific datasets and pipelines given the other. We focused on the neuroimaging datasets and pipelines available in Canadian Open Neuroscience Platform (CONP) as of October 2020. For our recommender approach, we needed the provenance records generated in pipeline execution processes, however, there have been 2688 pairs of pipeline-datasets for execution.

Therefore, we recruited 13 experts among graduate students, software developers and data engineers at the CONP and sent them a two phase survey to predict the execution outcome of all pipeline-dataset pairs which they have knowledge for both pipeline and dataset. Then we executed only the pipeline-dataset pairs for which at least one expert predicted a successful execution. All the generated provenance records are available at https://github.com/big-data-lab-team/paper-pipelines-datasets-recommender/tree/main/data/.

Moreover, we have contributed to the web interface of CONP portal and uploaded all the provenance records there to be accessible on a dashboard, Figure 11, so that users can search for any available combination of executed pipeline-dataset pairs and see the actual provenance records by clicking on their status, Figure 12 shows the provenance record for processing a file in 'SIMON' dataset by 'fsl_bet' pipeline whith successful execution outcome ('exit-code' = 0).

From the outcome of these executions we filled the pipeline execution matrix ,Figure 8b, as the utility matrix for our recommender approach. Then, we applied

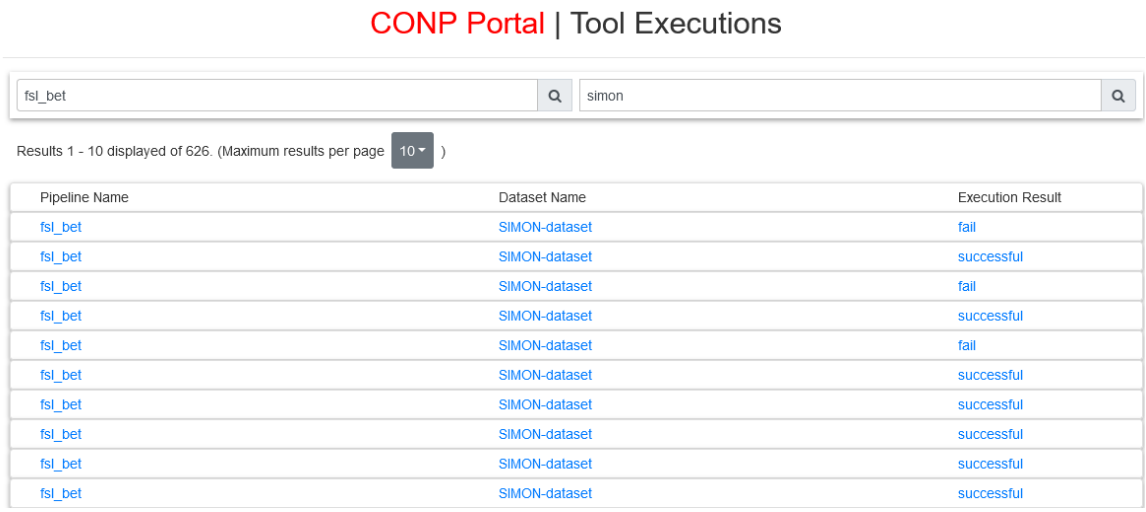


Figure 11: Dashboard of Provenance Records integrated in CONP portal

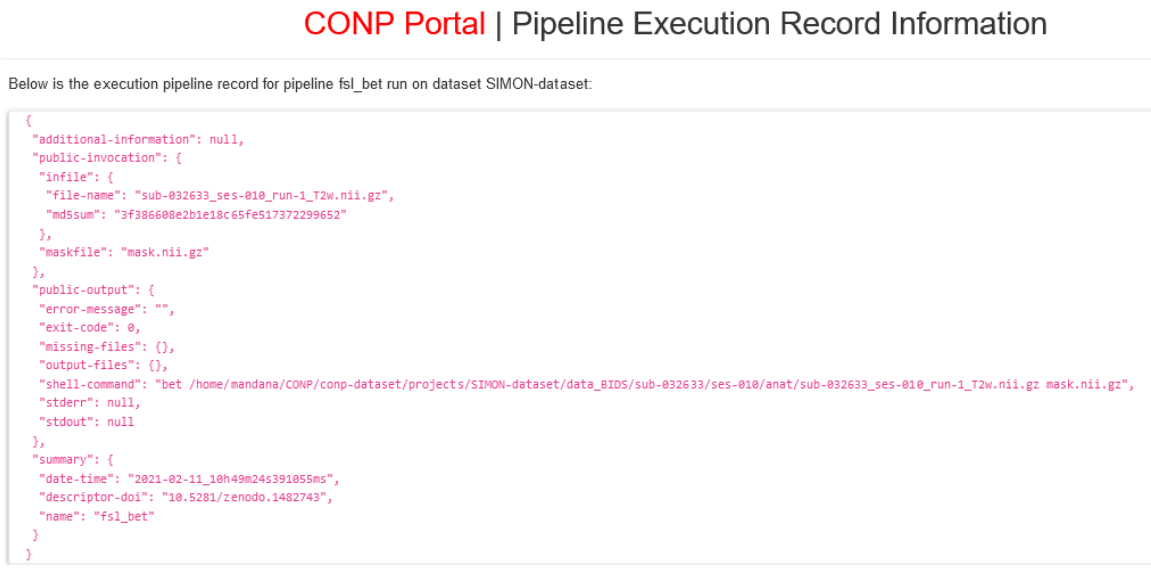


Figure 12: One of the generated provenance records (JSON object)

Alternative Least Square model, which implements latent-factor model in Collaborative Filtering recommendation approach, on this matrix. We evaluated this model by applying 10-fold cross validation on the pipeline execution matrix and varying the threshold for rounding the generated values to 1 (failed execution) or 2 (successful execution). Therefore, by creating the Receiver Operating Characteristic (ROC) curve we got the area under curve (AUC) as 0.83. We also evaluated the experts' predictions by applying varying thresholds on the fraction of experts predicting successful execution and got AUC = 0.63 for the corresponding ROC curve which indicates that our system outperforms human experts' recommendations.

There might be a several reasons why our system has a better performance than experts recommendations (predictions). There are many technical details that usually are ignored or neglected by the experts such as type compatibility of pipeline and dataset, pre-processing requirements for the pipelines, and availability and accessibility of the data at that time are some the reasons. Moreover, the good performance of our system could be expected to some extent since the pipeline execution matrix is not a random one and there are some sort structural patterns which helps the model having better accuracy.

In this study we contributed to three out of four of FAIR principles. We contributed to findability, since using our system people are assisted in finding the datasets that will work with a given pipeline or conversely the compatible pipelines for a given dataset. We also address the interoperability question, since our system highlights what datasets and pipelines will inter-operate and what will not. Moreover, we contribute to Reusability, since for a newly added dataset or pipeline for which there is no record, our system will recommend the popular ones which helps to promote the adoption of that new dataset or pipeline.

Among the possible future works for this project, the most highlighted one is increasing the number of provenance records. These provenance records would be either uploaded by public users to a repository for CONP usage only or to distributed repositories which CONP would be responsible to collect them. Therefore, by having a large set of provenance records, more than one outcome for each pipeline-dataset pair, it might be required to change the recommendation approach for implicit feedback data. Since there would not be any limitation or range for the number of provenance records per pair, if the approach is going to focus on the number of executions per

pair, the feedback data will not be explicit and it will be required to use collaborative filtering for implicit feedback data [22].

There would be some points to consider in this case, focusing only on the number of (successful) executions per pair will not provide strong and rational enough recommendations. For instance, some pipelines are simpler to use and mostly will be executed successfully, such as 'fsl-bet', also accessing to some datasets is more straightforward than others, this leads to higher number of provenance records for specific pipeline-dataset pairs while they will not necessarily provide the best recommendations. It is important to consider the different factors: How many (successful) executions per pair? What fraction of all executions per pair is successful? How frequently the pipeline or dataset is used among all pipelines and datasets? What is the subtraction of successful and failed executions per pair? For considering these different factors, one possible approach can be an ensemble of utility matrices (one per factor) resulting a set of recommendation lists for each pair, then the final list can be the union of all these recommendations for each pair.

We are integrating the current version of this recommender system to the CONP portal and it will provide a full operational recommender system based on available provenance records. This system is supposed to work with the publicly uploaded provenance records on the CONP portal and provide a list of highly recommended pipelines for a dataset in each dataset card and the highly recommended datasets for in each pipeline card. Figure 13 represents the sample dataset recommendations for 'fsl-bet' pipeline.

Another work to be done in future is handling cold start problem. Our system focuses on the available datasets and pipelines that we have provenance records for, therefore, whenever there is no provenance record for a dataset or pipeline we would face cold start problem [37]. In collaborative filtering, cold start happens when there is no record for the interactions between the cold-user and all items or between the cold items and all users. One of the most common approach to handle cold start in collaborative filtering is to recommend the most popular items to the cold user or the most popular users to the cold item [52]. Therefore, one possible approach in our case is to recommend the pipelines with most number of successful executions to a new dataset or the datasets with highest number of successful executions to a new pipeline.

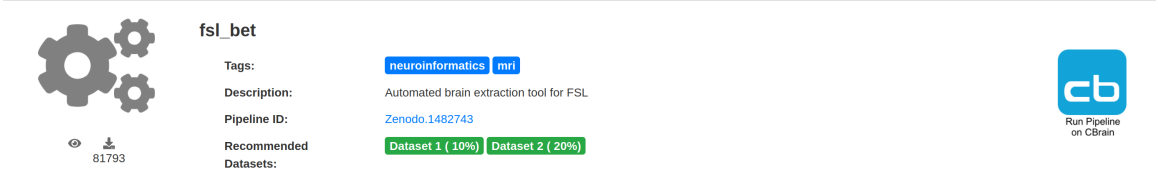


Figure 13: The mock-up for the recommender system to be integrated in CONP portal

Handling cold start with recommending the most popular datasets or users might raise the problem of popularity bias [2], meaning that the system will more and more recommend the datasets or pipelines which have the highest number of interactions, however, there would be many other datasets and pipelines with fewer interactions that would not be recommended. Therefore the unpopular datasets or pipelines would receive much less visibility than popular ones and will rarely receive interactions. To address this issue, it would be helpful to combine the content-based approach, meaning that find the similar datasets and recommend the pipelines which have more interactions with those datasets. However, currently, in our context, the available descriptions of datasets and pipelines in CONP are limited, which is a challenge.

Bibliography

[1] Canadian open neuroscience platform. http://portal.conp.ca, Accessed: 23 Jul, 2021.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Controlling popularity bias in learning-to-rank recommendation. In Proceedings of the eleventh ACM conference on recommender systems, pages 42–46, 2017.

[3] Mathew Birdsall Abrams, Jan G Bjaalie, Samir Das, Gary F Egan, Satrajit S Ghosh, Wojtek J Goscinski, Jeffrey S Grethe, Jeanette Hellgren Kotaleski, Eric Tatt Wei Ho, David N Kennedy, et al. A standards organization for open and fair neuroscience: the international neuroinformatics coordinating facility. Neuroinformatics, pages 1–12, 2021.

[4] Paul Ayris, Jean-Yves Berthou, Rachel Bruce, Stefanie Lindstaedt, Anna Monreale, Barend Mons, Yasuhiro Murayama, Caj Södergård, Klaus Tochtermann, and Ross Wilkinson. Realising the European open science cloud. European Union, Luxembourg, 2016.

[5] James Bennett, Stan Lanning, et al. The netflix prize. In Proceedings of KDD cup and workshop, volume 2007, page 35. New York, NY, USA., 2007.

[6] Dheeraj Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. Matrix factorization model in collaborative filtering algorithms: A survey. Procedia Computer Science, 49:136–146, 2015.

[7] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. Dataset search: a survey. The VLDB Journal, 29(1):251–272, 2020.

[8] European Organization For Nuclear Research and OpenAIRE. Zenodo, 2013.

[9] GO FAIR. Material was copied from this source, which is available under a creative commons attribution 4.0 international license, 2020.

[10] Benedikt Fecher, Sönke Bartling, and Sascha Friesike. Opening science: The evolving guide on how the internet is changing research, collaboration and scholarly publishing. Impact of Social Sciences Blog, 2014.

[11] Benedikt Fecher and Sascha Friesike. Open science: one term, five schools of thought. Opening science, pages 17–47, 2014.

[12] Daniel Gardner, Huda Akil, Giorgio A Ascoli, Douglas M Bowden, William Bug, Duncan E Donohue, David H Goldberg, Bernice Grafstein, Jeffrey S Grethe, Amarnath Gupta, et al. The neuroscience information framework: a data and knowledge environment for neuroscience. Neuroinformatics, 6(3):149–160, 2008.

[13] Yolanda Gil, Varun Ratnakar, Jihie Kim, Pedro Gonzalez-Calero, Paul Groth, Joshua Moody, and Ewa Deelman. Wings: Intelligent workflow-based design of computational experiments. IEEE Intelligent Systems, 26(1):62–72, 2010.

[14] Tristan Glatard, Gregory Kiar, Tristan Aumentado-Armstrong, Natacha Beck, Pierre Bellec, Rémi Bernard, Axel Bonnet, Shawn T Brown, Sorina Camarasu-Pop, Frédéric Cervenansky, et al. Boutiques: a flexible framework to integrate command-line applications in computing platforms. GigaScience, 7(5):giy016, 2018.

[15] Krzysztof Gorgolewski, Oscar Esteban, Gunnar Schaefer, Brian Wandell, and Russell Poldrack. Openneuro—a free online platform for sharing and analysis of neuroimaging data. Organization for human brain mapping. Vancouver, Canada, 1677(2), 2017.

[16] Krzysztof J Gorgolewski, Fidel Alfaro-Almagro, Tibor Auer, Pierre Bellec, Mihai Capotă, M Mallar Chakravarty, Nathan W Churchill, Alexander Li Cohen, R Cameron Craddock, Gabriel A Devenyi, et al. Bids apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods. PLoS computational biology, 13(3):e1005209, 2017.

[17] Krzysztof J Gorgolewski, Tibor Auer, Vince D Calhoun, R Cameron Craddock, Samir Das, Eugene P Duff, Guillaume Flandin, Satrajit S Ghosh, Tristan Glatard, Yaroslav O Halchenko, et al. The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. Scientific data, 3(1):1–9, 2016.

[18] Krzysztof J Gorgolewski, Tibor Auer, Vince D Calhoun, R Cameron Craddock, Samir Das, Eugene P Duff, Guillaume Flandin, Satrajit S Ghosh, Tristan Glatard, Yaroslav O Halchenko, et al. The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. Scientific data, 3(1):1–9, 2016.

[19] INCF Neuroimaging Data Sharing Group. Neuroimaging data model, 2016.

[20] A group of fourteen authors. Open concepts and principles, 2018.

[21] Joey Hess. git-annex, 19 October 2010.

[22] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In 2008 Eighth IEEE International Conference on Data Mining, pages 263–272. Ieee, 2008.

[23] Folasade Olubusola Isinkaye, YO Folajimi, and Bolande Adefowoke Ojokoh. Recommendation systems: Principles, methods and evaluation. Egyptian informatics journal, 16(3):261–273, 2015.

[24] Yaroslav Halchenko John T. Wodder II. datalad/datalad: 0.14.6 (version 0.14.6), 2021.

[25] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Surrogate functions for maximizing precision at the top. In International Conference on Machine Learning, pages 189–198. PMLR, 2015.

[26] David N Kennedy, Christian Haselgrove, Jon Riehl, Nina Preuss, and Robert Buccigrossi. The nitrc image repository. NeuroImage, 124:1069–1073, 2016.

[27] Gregory Kiar, Shawn T Brown, Tristan Glatard, and Alan C Evans. A serverless tool for platform agnostic computational experiment management. Frontiers in neuroinformatics, 13:12, 2019.

[28] Yehuda Koren and Robert Bell. Advances in collaborative filtering. Recommender systems handbook, pages 77–118, 2015.

[29] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, 2009.

[30] Anup Kumar, Helena Rasche, Björn Grüning, and Rolf Backofen. Tool recommender system in galaxy using deep learning. GigaScience, 10(1):giaa152, 2021.

[31] William La Cava, Heather Williams, Weixuan Fu, Steve Vitale, Durga Srivatsan, and Jason H Moore. Evaluating recommender systems for ai-driven biomedical informatics. Bioinformatics, 37(2):250–256, 2021.

[32] Anna-Lena Lamprecht. User-level workflow design. Lecture Notes in Computer Science, 8311, 2013.

[33] Anna-Lena Lamprecht, Stefan Naujokat, Tiziana Margaria, and Bernhard Steffen. Synthesis-based loose programming. In 2010 Seventh International Conference on the Quality of Information and Communications Technology, pages 262–267. IEEE, 2010.

[34] Michele Larobina and Loredana Murino. Medical image file formats. Journal of digital imaging, 27(2):200–206, 2014.

[35] Xiangrui Li, Paul S Morgan, John Ashburner, Jolinda Smith, and Christopher Rorden. The first step for neuroimaging data analysis: Dicom to nifti conversion. Journal of neuroscience methods, 264:47–56, 2016.

[36] Elizabeth D Liddy, Eileen Allen, Sarah Harwell, Susan Corieri, Ozgur Yilmazel, N Ercan Ozgencil, Anne Diekema, Nancy McCracken, Joanne Silverstein, and Stuart Sutton. Automatic metadata generation & evaluation. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pages 401–402, 2002.

[37] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. Facing the cold start problem in recommender systems. Expert Systems with Applications, 41(4):2065–2073, 2014.

[38] Li-Ping Liu, Thomas G Dietterich, Nan Li, and Zhi-Hua Zhou. Transductive optimization of top k precision. arXiv preprint arXiv:1510.05976, 2015.

[39] Daniel S Marcus, Timothy R Olsen, Mohana Ramaratnam, and Randy L Buckner. The extensible neuroimaging archive toolkit. Neuroinformatics, 5(1):11–33, 2007.

[40] Christopher J Markiewicz, Krzysztof Jacek Gorgolewski, Franklin Feingold, Ross Blair, Yaroslav O Halchenko, Eric Miller, Nell Hardcastle, Joe Wexler, Oscar Esteban, Mathias Goncalves, et al. Openneuro: An open resource for sharing of neuroimaging data. bioRxiv, 2021.

[41] Camille Maumet, Tibor Auer, Alexander Bowring, Gang Chen, Samir Das, Guillaume Flandin, Satrajit Ghosh, Tristan Glatard, Krzysztof J Gorgolewski, Karl G Helmer, et al. Sharing brain mapping statistical results with the neuroimaging data model. Scientific data, 3(1):1–15, 2016.

[42] David McDermid. Oracle machine learning for python user's guide, release 1.0.

[43] Paolo Missier, Khalid Belhajjame, and James Cheney. The w3c prov family of specifications for modelling provenance metadata. In Proceedings of the 16th International Conference on Extending Database Technology, pages 773–776, 2013.

[44] Stefan Naujokat, Anna-Lena Lamprecht, and Bernhard Steffen. Loose programming with prophets. In International Conference on Fundamental Approaches to Software Engineering, pages 94–98. Springer, 2012.

[45] National Institute of Mental Health. Nimh data archive.

[46] Magnus Palmblad, Anna-Lena Lamprecht, Jon Ison, and Veit Schwämmle. Automated workflow composition in mass spectrometry-based proteomics. Bioinformatics, 35(4):656–664, 2019.

[47] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In The adaptive web, pages 325–341. Springer, 2007.

[48] Russell A Poldrack and Krzysztof J Gorgolewski. Openfmri: Open sharing of task fmri data. Neuroimage, 144:259–261, 2017.

[49] Nancy Pontika, Petr Knoth, Matteo Cancellieri, and Samuel Pearce. Fostering open science to research using a taxonomy and an elearning portal. In Proceedings of the 15th international conference on knowledge technologies and data-driven business, pages 1–8, 2015.

[50] Anand Rajaraman and Jeffrey David Ullman. Mining of massive datasets. Cambridge University Press, 2011.

[51] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning, volume 242, pages 29–48. Citeseer, 2003.

[52] Suban Ravichandran. A state of the art survey on cold start problem in a collaborative filtering system.

[53] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In Recommender systems handbook, pages 1–35. Springer, 2011.

[54] Philippe Rocca-Serra and Alejandra Gonzalez-Beltran. biocaddie dats's documentation, 2016.

[55] Tony Ross-Hellauer and Edit Görögh. Guidelines for open peer review implementation. Research integrity and peer review, 4(1):1–12, 2019.

[56] Susanna-Assunta Sansone, Alejandra Gonzalez-Beltran, Philippe Rocca-Serra, George Alter, Jeffrey S Grethe, Hua Xu, Ian M Fore, Jared Lyle, Anupama E Gururaj, Xiaoling Chen, et al. Dats, the data tag suite to enable discoverability of datasets. Scientific data, 4(1):1–8, 2017.

[57] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web, pages 285–295, 2001.

[58] Dirk Schäfer and Eyke Hüllermeier. Dyad ranking using plackett–luce models based on joint feature representations. Machine Learning, 107(5):903–941, 2018.

[59] Mikkel Schöttner. Open science and neuroimaging - a practical guide, Jan 2020.

[60] Tarek Sherif, Pierre Rioux, Marc-Etienne Rousseau, Nicolas Kassis, Natacha Beck, Reza Adalat, Samir Das, Tristan Glatard, and Alan C Evans. Cbrain: a web-based, distributed computing platform for collaborative neuroimaging research. Frontiers in neuroinformatics, 8:54, 2014.

[61] Brent Smith and Greg Linden. Two decades of recommender systems at amazon.com. IEEE Internet Computing, 21(3):12–18, 2017.

[62] Bobbie Spellman, Elizabeth A Gilbert, and Katherine S Corker. Open science: What, why, and how, Apr 2017.

[63] Canadian Brain Research Strategy. Canadian brain research strategy (cbrs), 2015.

[64] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009, 2009.

[65] CONP team. The canadian open neuroscience platform (conp) web interface, 2019.

[66] LORIS team. Longitudinal online research and imaging system, 2006.

[67] NeuroImaging Tools and Resources Collaboratory. Nitrc.

[68] Alexander Tornede, Marcel Wever, and Eyke Hüllermeier. Algorithm selection as recommendation: From collaborative filtering to dyad ranking. In CI Workshop, Dortmund, 2019.

[69] Alexander Tornede, Marcel Wever, and Eyke Hüllermeier. Extreme algorithm selection with dyadic feature representation. In International Conference on Discovery Science, pages 309–324. Springer, 2020.

[70] Anthony L Vaccarino, Moyez Dharsee, Stephen Strother, Don Aldridge, Stephen R Arnott, Brendan Behan, Costas Dafnas, Fan Dong, Kenneth Edgecombe, Rachad El-Badrawi, et al. Brain-code: a secure neuroinformatics platform for management, federation, sharing and analysis of multi-dimensional neuroscience data. Frontiers in neuroinformatics, 12:28, 2018.

[71] Luiz Victorino, Ronaldo Pilati, and Alexandre Linhares. Priming and prejudice: The bias effect of origin information on peer review, judgment and evaluation. Avances en Psicología Latinoamericana, 37(1):169–178, 2019.

[72] Robert D Vincent, Peter Neelin, Najmeh Khalili-Mahani, Andrew L Janke, Vladimir S Fonov, Steven M Robbins, Leila Baghdadi, Jason Lerch, John G Sled, Reza Adalat, et al. Minc 2.0: a flexible format for multi-modal images. Frontiers in neuroinformatics, 10:35, 2016.

[73] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. Scientific data, 3(1):1–9, 2016.

[74] Michael Woelfle, Piero Olliaro, and Matthew H Todd. Open science is a research accelerator. Nature chemistry, 3(10):745–748, 2011.

[75] Dietmar Wolfram, Peiling Wang, Adam Hembree, and Hyoungjoo Park. Open peer review: promoting transparency in open science. Scientometrics, 125:1033–1051, 2020.