Feature Discovery for Data-Centric AI

Ionescu, A.

DOI
10.4233/uuid:433071e6-38d1-4442-b73b-be4e0c086a93

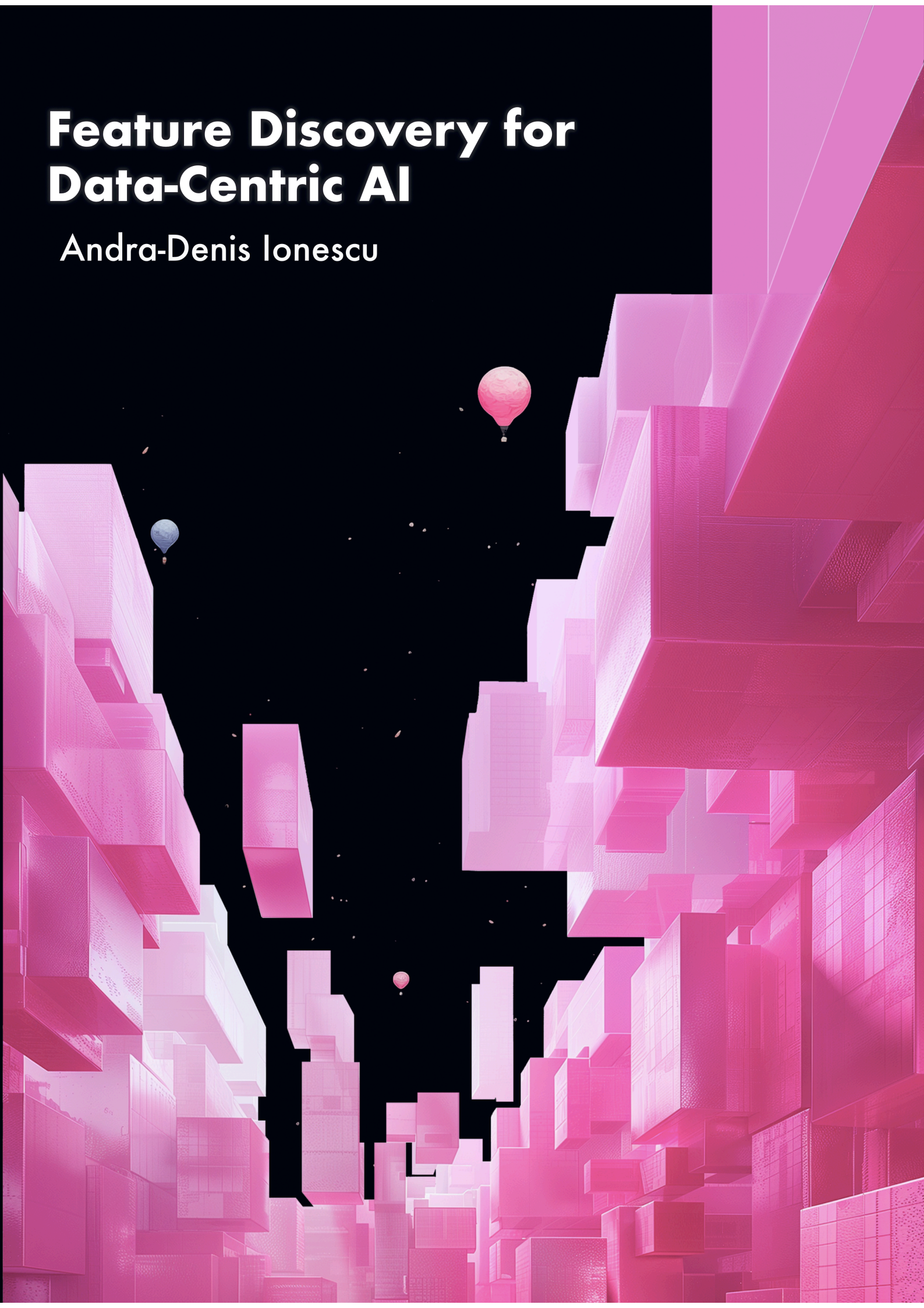
Publication date
2025

Document Version
Final published version

Citation (APA)
Ionescu, A. (2025). *Feature Discovery for Data-Centric AI*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:433071e6-38d1-4442-b73b-be4e0c086a93

Feature Discovery for Data-Centric AI

Andra-Denis Ionescu



Feature Discovery for Data-Centric AI

Feature Discovery for Data-Centric AI

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, Prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Monday, 13th of January at 17:30 o'clock

by

Andra-Denis IONESCU

Master of Science in Computer Science,
Delft University of Technology, the Netherlands,
born in Caracal, Romania

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. Dr. Ir. G.J.P.M Houben	Delft University of Technology, promotor
Dr. A. Katsifodimos	Delft University of Technology, copromotor
Dr. R. Hai	Delft University of Technology, copromotor
Independent members:	
Prof. Dr. Ir. A. Bozzon	Delft University of Technology
Prof. Dr. A. Bonifati	Lyon 1 University, France
Prof. Dr. P.T. Groth	University of Amsterdam
Prof. Dr. S. Schelter	Technische Universität Berlin, Germany
Prof. Dr. M.M. Specht	Delft University of Technology, reserve member

SIKS Dissertation Series No. 2025-03
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems. This research has been supported by the European Union's H2020 project OpertusMundi (870228).

Keywords: feature discovery, data-centric AI, data marketplaces, dataset discovery

Cover: The cover is an abstractization of the process of building or augmenting datasets, information, or knowledge. Designed with generative AI and edited by Alexandra Pituru.

Style: TU Delft House Style, with modifications by Moritz Beller
`https://github.com/Inventitech/phd-thesis-template`

Printed by: Gildeprint – www.gildeprint.nl

ISBN: 978-94-6366-991-7

Email address: andradenis.ionescu@gmail.com

An electronic version of this dissertation is available at `http://repository.tudelft.nl/`.

"In nature everything is connected, everything is interwoven, everything changes with everything, everything merges from one into another."

Gotthold Ephraim Lessing

Contents

Summary

We are witnessing a paradigm shift in machine learning (ML) and artificial intelligence (AI) from a focus primarily on innovating ML models, the model-centric paradigm, to prioritising high-quality, reliable data for AI/ML applications, the data-centric paradigm. This emphasis on data has led to the development of an economy around data, creating data marketplace platforms where data is traded as a commodity. However, trading data involves constraints that reflect the specific needs of users, such as enriching or augmenting their datasets or creating datasets with particular properties. These constraints pose challenges the data management community has already addressed independently of the marketplace platform context. As such, in this thesis, as a first act of research, we integrate approaches and practices from the data management community into the context of an open-source data marketplace platform, following a survey of industry professionals who produce, trade, and purchase data assets.

Aligned with the objectives of the data-centric AI paradigm to create high-quality training datasets, our research is focused on developing automated methods to identify relevant and related features (e.g., columns) that can be augmented to a given dataset. This effort has led to the research and design of feature discovery, which sits at the intersection of dataset discovery by discovering related datasets, data integration by joining datasets, and feature selection by selecting high-predictive features for ML models. We have developed an automated approach for feature discovery that improves upon existing automated data augmentation techniques, improving the effectiveness and efficiency of finding the most relevant features.

However, with the adoption of automatic approaches, we discovered that in moving towards data-centric AI, we risk detaching not only from model-centric but also from user-centric AI. To assess the extent to which users (e.g., data scientists, data engineers, ML engineers) rely on and trust automatic approaches and to determine their feature discovery pipeline, we conducted 19 interviews based on a use-case study. The results revealed that users doubt the automated methods and want to be involved in the process instead. Consequently, we decided to incorporate the users into the feature discovery process and to explore whether their involvement (e.g., by adding domain and business knowledge) improves the quality of the resulting dataset and the feature discovery process. Thus, we created a human-in-the-loop approach for feature discovery, which was evaluated by conducting interviews with a subset of our initial candidate pool. The results confirmed that a human-in-the-loop method is more approachable for users as it provides control over and insights into the process, as well as the opportunity to inject their knowledge, ensuring that the resulting dataset is relevant for their data tasks.

With this thesis, we make scientific contributions to the field of data management by offering novel insights into users' workflows and designing and developing resources that enhance feature discovery. We hope our contributions will serve as a valuable resource for future work in user-centric and data-centric feature discovery.

Samenvatting

We zijn getuige van een paradigmaverschuiving in machine learning (ML) en kunstmatige intelligentie (AI) van een focus die primair ligt op het innoveren van ML-modellen, het modelgerichte paradigma, naar het prioriteren van hoogwaardige, betrouwbare data voor AI/ML-toepassingen, het datagerichte paradigma. Deze nadruk op data heeft geleid tot de ontwikkeling van een economie rondom data, waarbij platforms voor datamarktplaatsen zijn ontstaan waar data als handelswaar wordt verhandeld. Het verhandelen van data gaat echter gepaard met beperkingen die de specifieke behoeften van gebruikers weerspiegelen, zoals het verrijken of uitbreiden van hun datasets of het creëren van datasets met specifieke eigenschappen. Deze beperkingen vormen uitdagingen die de databeheercommunity al heeft aangepakt, onafhankelijk van de context van platforms voor marktplaatsen. Daarom integreren we in deze thesis, als eerste onderzoeksactie, benaderingen en praktijken van de databeheercommunity in de context van een open-source platform voor datamarktplaatsen, na een enquête onder industrieprofessionals die data-assets produceren, verhandelen en kopen.

In overeenstemming met de doelstellingen van het datagerichte AI-paradigma om hoogwaardige trainingsdatasets te creëren, richt ons onderzoek zich op het ontwikkelen van geautomatiseerde methoden om relevante en gerelateerde kenmerken (bijv. kolommen) te identificeren die kunnen worden toegevoegd aan een gegeven dataset. Deze inspanning heeft geleid tot het onderzoek en ontwerp van kenmerkontdekking, dat zich bevindt op het snijvlak van datasetontdekking door het ontdekken van gerelateerde datasets, dataintegratie door het samenvoegen van datasets en kenmerkselectie door het selecteren van hoog voorspellende kenmerken voor ML-modellen. We hebben een geautomatiseerde benadering voor kenmerkontdekking ontwikkeld die bestaande geautomatiseerde data-uitbreidingstechnieken verbetert, waardoor de effectiviteit en efficiëntie van het vinden van de meest relevante kenmerken wordt verbeterd.

Echter, met de adoptie van automatische benaderingen, ontdekten we dat we bij het bewegen naar datagerichte AI het risico lopen niet alleen los te komen van modelgerichte maar ook van gebruikersgerichte AI. Om de mate waarin gebruikers (bijv. datawetenschappers, data-engineers, ML-engineers) vertrouwen op en afhankelijk zijn van automatische benaderingen te beoordelen en om hun pipeline van kenmerkontdekking te bepalen, hebben we 19 interviews afgenomen op basis van een use-case studie. De resultaten toonden aan dat gebruikers twijfels hebben over de geautomatiseerde methoden en in plaats daarvan betrokken willen zijn bij het proces. Daarom besloten we de gebruikers te betrekken bij het proces van kenmerkontdekking en te onderzoeken of hun betrokkenheid (bijv. door het toevoegen van domein- en bedrijfskennis) de kwaliteit van de resulterende dataset en het proces van kenmerkontdekking verbetert. Daarom hebben we een human-in-the-loop benadering voor kenmerkontdekking gecreëerd, die werd geëvalueerd door interviews af te nemen met een subset van onze oorspronkelijke kandidatenpool. De resultaten bevestigden dat een human-in-the-loop methode toegankelijker is voor gebruikers,

omdat het controle en inzicht in het proces biedt, evenals de mogelijkheid om hun kennis in te brengen, waardoor wordt gegarandeerd dat de resulterende dataset relevant is voor hun datataken.

Met deze thesis leveren we wetenschappelijke bijdragen aan het veld van databeheer door nieuwe inzichten te bieden in de workflows van gebruikers en middelen te ontwerpen en te ontwikkelen die kenmerkontdekking verbeteren. We hopen dat onze bijdragen een waardevolle bron zullen zijn voor toekomstig werk in gebruikersgerichte en datagerichte kenmerkontdekking.

Rezumat

Asistăm la o schimbare de paradigmă în domeniul învățării automate (ML) și al inteligenței artificiale (AI), de la un accent predominant pe inovarea modelelor de ML, paradigma centrată pe modele, la prioritizarea datelor de înaltă calitate și fiabile pentru aplicațiile AI-/ML, paradigma centrată pe date. Acest accent pe date a dus la dezvoltarea unei economii în jurul lor, creând platforme de piață de date, unde datele sunt tranzacționate ca o marfă. Cu toate acestea, tranzacționarea datelor implică restricții care reflectă nevoile specifice ale utilizatorilor, cum ar fi îmbogățirea sau augmentarea seturilor lor de date sau crearea de seturi de date cu anumite proprietăți. Aceste restricții ridică provocări pe care comunitatea de gestionare a datelor le-a abordat deja independent de contextul platformelor de piață online. Astfel, în această teză, ca prim act de cercetare, integrăm abordări și practici din comunitatea de gestionare a datelor în contextul unei platforme online de piață de date cu sursa deschisă, pe baza unui sondaj realizat cu profesioniști din industrie care produc, comercializează și achiziționează date.

Aliniată cu obiectivele paradigmei AI centrate pe date pentru a crea seturi de date de instruire (a modelelor ML) de înaltă calitate, cercetarea noastră se concentrează pe dezvoltarea de metode automate pentru a identifica proprietățile relevante (de exemplu, coloane din seturi de date) care pot fi adăugate la un anumit set de date. Acest efort a condus la cercetarea și proiectarea metodei de *descoperire a caracteristicilor* (unor coloane din seturile de date), care se află la intersecția descoperirii seturilor de date, efectuată prin găsirea seturilor de date cu relevanță, integrarea datelor prin alăturarea seturilor de date și selecția caracteristicilor prin selectarea celor cu predicție ridicată pentru modelele ML. Am dezvoltat o abordare automată pentru descoperirea caracteristicilor care îmbunătățește tehnicile existente de expansiune automată a datelor, îmbunătățind eficacitatea și eficiența găsirii celor mai relevante caracteristici.

Cu toate acestea, odată cu adoptarea abordărilor automate, am descoperit că în trecerea către AI centrată pe date, riscăm să ne detașăm nu numai de AI centrat pe model, ci și de AI centrat pe utilizator. Pentru a evalua măsura în care utilizatorii (de exemplu, oameni de știință de date, ingineri de date, ingineri ML) se bazează și au încredere în abordările automate și pentru a determina procesul lor de descoperire a caracteristicilor, am efectuat 19 interviuri pe baza unui studiu de caz. Rezultatele au arătat că utilizatorii sunt sceptici cu privire la metodele automate și doresc să fie implicați în proces. În consecință, am decis să integrăm utilizatorii în procesul de descoperire a caracteristicilor și să examinăm dacă implicarea lor (de exemplu, prin adăugarea propriilor cunoștințe asupra unui domeniu al unui set de date) îmbunătățește calitatea setului de date rezultat și totodată procesul de descoperire a caracteristicilor. Astfel, am creat o abordare pentru descoperirea caracteristicilor care implică utilizatorul și care a fost evaluată prin realizarea de interviuri cu un subset din grupul nostru inițial de candidați. Rezultatele au confirmat că o metodă care implică utilizatorul este mai accesibilă, deoarece oferă control si intuiție asupra procesului,

precum și oportunitatea de a adăuga cunoștințele proprii, asigurându-se că setul de date rezultat este relevant pentru procesele lor de date.

Prin această teză, aducem contribuții științifice în domeniul administrării datelor, oferind perspective noi asupra procesului de lucru al utilizatorilor și prin proiectarea și dezvoltarea resurselor care îmbunătățesc descoperirea caracteristicilor. Sperăm că aceste contribuții vor servi ca o resursă valoroasă pentru lucrările viitoare în descoperirea caracteristicilor centrate pe utilizator și pe date.

Acknowledgments

During my journey at TU Delft, it has become clear that I have been living the fairytale life. Despite the hardships of the 2020 pandemic and many other personal ones intertwined with high professional expectations, life in WIS has genuinely been a fairytale. This was made possible by the incredible people who surrounded me and cultivated a balanced environment filled with professional rigour, warmth and friendliness. To all of you, my heartfelt thanks and eternal gratitude.

Asterios, your unwavering positivity and optimism, paired with your pragmatic and realistic view of life, have been my guiding light. Thank you for being the best listener, my champion in the face of deadlines, and always standing by my side.

Rihan, your pursuit of perfection and belief in me pushed me to achieve more than I ever thought possible. Your ethics and kind spirit have been a constant source of inspiration. Thank you for lifting me up and showing me what true excellence looks like.

Geert-Jan, thank you for bringing balance into the supervision, for your infinite wisdom and invaluable feedback, and for guiding me towards a better version of myself professionally and personally.

Alex, my love, my rock, and my greatest supporter, thank you for everything. Your sharp mind, kind heart, and infectious cheerfulness have been the light guiding me through every challenge. Thank you for your unshakeable love and belief in me, for being the genie who has granted all my wishes and transformed my dreams into reality. I am eternally grateful for your presence in my life, for lifting me up with your love, and for standing by me through it all. My heart is forever yours.

Alisa and **Garrett**, thank you for the immense honour of being my paranymphs. You brought light to my darkest days with our cosy cake and coffee moments; you filled my heart with laughter during lively cocktail conversations and turned even the gloomiest times into joyful moments. Your support and boundless energy inspired me to be braver, more resilient, and ever-optimistic.

To the **committee members**, Alessandro, Angela, Paul, Sebastian, thank you for your thoughtful feedback, encouragement, and the stimulating and brilliant discussions that helped shape my research.

To the **Delta Team** (Marios, George S., Ziyu, Christos, Kyriakos, George C., Marcus, Wenbo, Danning, Aditya, Oto, and Tim), thank you for your kindness and support. From shared office spaces to cheerful trips, from indulging in sweets and candies from our home countries to discovering the best pubs and restaurants in Delft, you turned every moment into a cherished memory. Thank you for celebrating both our triumphs and our setbacks with equal enthusiasm and for making this journey so much more vibrant and meaningful.

To **Fenia, Sole, Christoph, Avishek, Ujwal, Jie**, thank you for your invaluable mentorship, encouragement, inspiration, and heartfelt compassion. Your inspiring words and uplifting spirit pushed me to think deeper, explore further, and find light even in the dark-

est moments when hope seemed lost. Your guidance has shaped my work and left a lasting mark on who I am today.

To the **WIS Team** (Agathe, Anne, Arthur, David, Esra, Felipe, Gaole, Gustavo, Jurek, Lijun, Lorenzo, Nirmal, Peide, Petros, Philip, Robin, Sara, Sarah, Sepideh, Shabnam, Shahin, Shreyan, Sihang, Tim, Venktesh), thank you for your understanding, your ambition, for the spontaneous conversations, for having coffee together and long lunches in the sunshine, and for creating the space to share both frustrations and triumphs. You shaped an environment where I could be my true self.

To **Daphne** and **Nadia**, your thoughtfulness and care have been beyond measure. Thank you for ensuring everything ran smoothly, for always checking in on me, and for encouraging my Dutch practice!

To **Shaad, Florena, Kiril, Zeger**, thank you for enduring the chaotic moments of paper submissions, for your creativity and collaboration, and for standing by my side when it mattered most.

To the **Topio Consortium**, your gentle guidance introduced me to a whole new world, expanding my creativity and teaching me patience and organization. Thank you for this invaluable experience.

To **Alexandra (Shu)**, thank you for bringing your creative spirit into my work, for dedicating your time to helping me design posters and interfaces, and for creating the beautiful cover of this dissertation. Beyond your creative contributions, I am deeply grateful for your support, attentive ear, and cheering me up.

To **Raluca, Matei, Ivan, Ruxandra, Andrei, Lyrica, Liam, Julia, Nino**, thank you for being such wonderful friends, for your support and for filling my life with countless cherished memories. From exciting trips to impromptu sleepovers, from joyous birthday celebrations to unforgettable New Year's Eve parties, you created endless reasons to celebrate and share moments of joy. You brought vibrant colour and comforting warmth to my world, making it all the more beautiful.

To **Ana, Stefanie, Carolin**, our friendship stems from this long and arduous journey. I could not be happier to have met you! Thank you for the walks, the endless laughter, and the deep conversations. You brought so much richness and joy to life beyond work, and for that, I am forever thankful.

To **my parents**, no words can truly capture the depth of my gratitude. Thank you for providing me with the tools, values, and support that have allowed me to reach this far in my career and become the person I am today. Your love, sacrifices, and belief in me have been the foundation of everything I have achieved. *Către **părinții mei**, nu există cuvinte care să poată exprima cu adevărat profunzimea recunoștinței mele. Vă mulțumesc că mi-ați oferit instrumentele, valorile și sprijinul care mi-au permis să ajung atât de departe în carieră și să devin persoana care sunt astăzi. Dragostea, sacrificiile și încrederea voastră în mine au fost fundamentale pentru tot ceea ce am realizat.*

To **Dexter**, my beloved furry "brother", for warming my days with your playful energy and gentle presence and for bringing happiness to my heart.

Thank you to the reviewers who steered my work to better versions, the interview participants who made my research possible, and everyone whose paths crossed with mine and brought light and joy to my journey. Thank you to the internet for always having the

answers and to the universe for bringing everyone and everything together at the right moment and time beyond perfection.

There is so much more to say about how each and every one of you contributed immensely to creating my fairytale world. Forever, my gratitude and appreciation. From the bottom of my heart, ***thank you!***

Andra-Denis Ionescu
Den Haag, 2024

1

Introduction

Machine Learning (ML) has emerged as a critical component across many domains, including but not limited to enhancing search capabilities [47], generating images and videos [13], as well as assisting healthcare [95, 136] and industrial professionals [197]. The widespread adoption of ML underscores the fundamental role of data, positioning it as an essential element for the development and effectiveness of Artificial Intelligence (AI) and ML applications [132]. This focus on data has led to a paradigm shift, from primarily concentrating on innovations in ML models to prioritising the quality and reliability of data [54, 198]. This transition has given rise to **Data-Centric AI** (DCAI), a framework that emphasises building and maintaining high-quality datasets to advance AI technologies and applications [159, 198, 199].

Within DCAI, the primary objective is creating or improving datasets that will be used for training ML models [159, 198, 199]. This approach stands in contrast to the traditional model-centric AI paradigm, which prioritises the development and refinement (i.e., fine-tuning) of models to align with a predetermined benchmark dataset [132]. Model-centric AI often placed excessive confidence in the accuracy of the datasets, leading to the problematic scenario of "garbage in, garbage out", a situation where poor quality input data results in poor quality output, regardless of numerous model iterations [132, 198]. This excessive reliance on the dataset raised critical questions about the representativeness and reliability of models. Specifically, it brought to light concerns regarding whether a model truly reflected the nuances of the benchmark dataset or if it was merely overfitting to that dataset [132]. As a result, the emphasis has shifted towards recognising data as a "first-class citizen" within the ML development process [187]. Now, data is a pivotal element in ML pipelines, playing a crucial role in determining the overall quality and effectiveness of the models [54]. This shift highlights a growing consensus in the field: the success of ML applications hinges not just on the algorithmic ingenuity of the models but equally on the **quality, relevance, and reliability** of the data they are trained on [54, 132].

Traditional ML models typically require tabular datasets for training. Therefore, the quality, relevance, and reliability of tabular data are pivotal factors that have been extensively studied within the data management community. A significant focus has been on dataset integration, especially schema matching, due to its broad applicability to bring to-

1

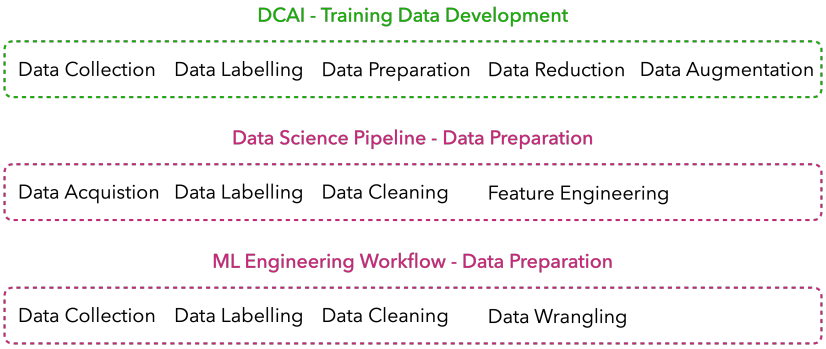


Figure 1.1: The data preparation layers within DCAI [198], Data Science Pipeline [183] and ML Engineering Workflows [173]. All three layers describe similar steps to create high-quality training datasets.

gether disparate data sources through join or union operations [17, 23, 58, 106, 146, 191, 201]. Furthermore, data quality is a pivotal characteristic that must be considered when developing ML models, a notion also supported by the DCAI paradigm [39, 94]. To maintain high-quality data, various strategies and techniques have been developed [65, 120, 155]. These methods address common issues such as missing data [169], duplicate data [189], and data heterogeneity [42], ensuring that the data used in ML models is of the highest integrity and value.

The taxonomies within DCAI describe various tasks of preparing the datasets (e.g., training set selection, data cleaning and debugging, or data acquisition and generative model prompting) [132] or outline specific goals for which the datasets are created or enhanced (e.g., training data development, inference data development, and data maintenance) [198]. These high-level goals encapsulate numerous sub-goals, paralleling the aforementioned tasks. For instance, under training data development, one encounters sub-goals such as data collection, labelling, preparation, reduction, and augmentation, illustrated in Figure 1.1 [198, 199]. When considering the simplified abstraction of an AI/ML workflow, as illustrated in Figure 1.2, particularly in data science or machine learning engineering, two primary layers emerge: the data and model layers[1] [54]. These layers form the backbone of AI/ML workflows, guiding the transition from raw data to operational AI solutions. Notably, the subtasks in the data layer, including data acquisition, wrangling, cleaning, labelling, and engineering [15, 41, 173, 183, 207], align closely with the sub-goals and tasks identified in DCAI taxonomies (Figure 1.1).

Consequently, we observe a convergence between the objectives of DCAI and the first layer of data science and ML workflows. On the one hand, DCAI emphasises the development of curated training data for ML models [199], and on the other, the data layer of the data science workflow concentrates on processing datasets to produce quality training data for modelling [15, 41, 173, 183, 207]. Both aspects underscore the critical importance of acquiring and generating datasets as a fundamental step in developing ML models [6, 199]. This interplay between DCAI tasks and AI/ML workflow layers reflects

[1]We acknowledge the model management (i.e., deployment) as one of the layers in AI/ML workflows. However, this thesis falls outside the scope of the model management layer.

Figure 1.2: The steps of the data science pipeline (top row) [41], and the steps of the ML engineering workflow (bottom row) [173]. We observe similar steps in the pipeline, which describe the high-level layers: data, model, deployment, and maintenance.

a broader trend where the emphasis on data quality has become paramount. Moreover, it highlights the dynamic evolution of the AI/ML field, where advanced data management practices have become integral to the development process. Incorporating methods and approaches for data management, from initial data collection to detailed augmentation, is pivotal for developing and refining AI and ML models. We are witnessing a shift towards a more data-centric approach, which reflects the growing influence of DCAI within the AI/ML field [54].

We introduce the concept of **feature discovery for tabular datasets**, which, for brevity, will be referred to simply as *feature discovery* throughout this dissertation. Feature discovery is an approach at the intersection between the data and model layers within the data science pipeline. This approach is intrinsically linked with the core objective of DCAI to generate high-quality training datasets for ML models. In the context of feature discovery, we prioritise the collection of datasets by employing dataset discovery techniques to identify valuable data sources [144]. We also explore dataset augmentation by integrating disparate data sources, thereby enriching the informational value of the dataset [60]. To reduce dimensionality, the process involves refining the dataset through feature selection methods, focusing on the most impactful features for ML modelling [198].

By leveraging the strengths of dataset discovery, integration, and feature selection, we aim to curate datasets that are optimally configured for ML model training. The synergy between identifying relevant features and integrating diverse data sources enhances the potential for creating high-quality datasets and more effective and efficient ML models. This underscores the critical role thorough data preparation plays in AI/ML development, which drives the DCAI initiative. By exploring feature discovery for tabular data, this work contributes to the ongoing efforts to improve AI/ML practices, emphasising the significant relationship between data quality and model performance in both DCAI and ML pipelines.

This thesis is divided into three main parts. *Part I: Tabular Data Acquisition with Data Marketplaces* concentrates on identifying datasets for acquisition, such that they can be further used to enhance a training dataset with the goal of improving the performance of ML models. We explore acquiring these datasets within the data marketplace platforms. In *Part II: Automated Feature Discovery for Tabular Data*, we use the acquired datasets and explore the development of automated methods for feature discovery and augmentation. Finally, in *Part III: Human-in-the-Loop Feature Discovery*, we begin with an examination of the feature discovery and augmentation workflows as performed by data specialists in real-life scenarios and conclude with the development of a human-in-the-loop approach, which incorporates the user's expertise and input into the feature discovery process. In the following sections, we elaborate on prior research, describe each part in detail, define the corresponding research question and outline our contributions.

1

1.1 Tabular Data Acquisition with Data Marketplaces

The dataset collection process entails gathering information from multiple sources [198]. It also involves a series of steps such as acquisition, labelling, and improving existing datasets [166, 187]. Some sources treat dataset collection and dataset acquisition interchangeably when referring to the process of searching and discovering (new) datasets [198] or even generating data through crowdsourcing or synthetic methods [166].

In data-centric AI, the term dataset collection refers to data acquisition and labelling processes, which are essential for creating high-quality training datasets for ML models. These models require substantial data volume (i.e., numerous rows) or detailed descriptive features (i.e., various columns) to effectively capture and describe specific problems. This requirement becomes even more critical in the context of deep learning, where the data demands exceed those of traditional ML models, requiring extensive training datasets [166, 187]. Thus, there is a significant need for relevant training datasets. When we take into account that the quality of data collected fundamentally influences the overall data quality [187, 198], searching for and discovering related datasets from various sources become even more critical [166]. This thesis focuses explicitly on **dataset acquisition** within the broader dataset collection process. The literature presents a variety of dataset collection strategies, each tailored to different or specific search repositories and platforms, highlighting the diverse approaches to gathering appropriate and high-quality data for ML applications.

The proliferation of dataset search and acquisition platforms or repositories has led to diversifying sources and methods for obtaining datasets. Notably, open data portals offer easy access to data acquisition. Prominent examples include the open government data portals of the UK and the USA[2], which facilitate dataset search using keywords over the metadata or the dataset content itself [76, 101]. Furthermore, these open data repositories, alongside numerous other web-based datasets, are accessible through the Google dataset search platform [21], underscoring the platform's utility in data discovery. Additionally, domain-specific data portals such as DataMed[3] emerge as invaluable resources, providing targeted datasets that align with distinct research objectives and use cases [30].

Data lakes are another effective solution for storing and managing large volumes of unprocessed datasets. However, dataset search and collection within data lakes present significant challenges. The sheer volume of data often makes it difficult for users to comprehend and navigate the information effectively, hindering their ability to explore the data and achieve their specific goals [68, 151, 206]. This complexity highlights the need for advanced tools and methods to simplify dataset discovery and acquisition in such expansive data environments. These solutions and systems include dataset integration, dataset discovery, metadata management, and data versioning, designed to maintain the data lake and facilitate dataset acquisition [70, 146].

Given that open data platforms and data lakes have been extensively studied, with numerous methods proposed to enhance dataset acquisition within these environments, our focus shifts towards data marketplace platforms. The emergence of data marketplace platforms has opened new avenues for monetising data. Platforms such as Dawex,

[2] `https://www.data.gov.uk`, `https://data.gov`
[3] `https://datamed.org/`

Worldquant, and Snowflake Marketplace[4] have been established for the exchange of datasets, thereby creating a marketplace [59]. These platforms adhere to specific rules and guidelines for the transaction of datasets. However, the integration of data management practices such as dataset discovery and integration can significantly enhance the benefits for both data consumers and producers within these marketplace platforms [59]. As such, in this thesis, we explore the integration of **dataset discovery approaches within data marketplace platforms**, providing a new dimension to data acquisition in DCAI.

Often formulated as a search problem, dataset discovery is the process of finding relevant data sources among a vast collection of datasets [17, 58, 106, 144, 203, 206]. Similar to conventional information retrieval techniques, dataset discovery leverages keywords as queries to find datasets [21, 30]. Additionally, it employs tables as queries, explicitly aiming to identify tables that can be joined or unioned, enhancing the process's utility and relevance [17, 24, 146, 206]. Furthermore, some approaches are specifically tailored to meet diverse user needs, acknowledging both the variety of user queries and the heterogeneous nature of the data involved [58].

The variety of approaches proposed for dataset discovery, their effectiveness in identifying related and relevant datasets, and the noticeable gap in data marketplace platforms concerning data management lead to our first research question:

> ***RQ1**: How can dataset discovery approaches enable and facilitate data acquisition in data marketplace platforms?*

Findings & Contributions. To answer our first research question **RQ1**, we start by conducting a survey with 122 participants to identify and understand the needs and requirements of data providers and consumers for efficient and effective dataset acquisition within data marketplace platforms, as well as evaluate the qualities and features of a web-based platform. Based on this information, we adapt our approaches to align with the user's perspective. Thus, we aim to enhance the effectiveness of the data acquisition process, ensuring that users can efficiently identify and access the relevant data they need. As such, in Chapter 3, we introduce `Topio`, an instance of an open-source data marketplace platform. With `Topio`, we research diverse strategies to aid users in the search and acquisition process within a marketplace setting and develop a suite of scalable, low-cost, value-added services that facilitate dataset exploration and discovery.

1.2 Automated Feature Discovery

The data augmentation process involves developing methods and techniques to increase the amount and quality of training data, particularly in ML/AI. For tabular textual data, data augmentation is more challenging due to the intricate syntactic and semantic structures of text [32]. Here, augmentation represents a variety of strategies to modify or generate new textual content while preserving the original semantic context. This could include techniques such as synonym replacement, sentence shuffling, or employing advanced methods such as leveraging language models for text generation or transformation [111, 186, 190].

[4] `https://www.dawex.com/en/`, `https://www.worldquant.com/data-exchange/`, `https://www.snowflake.com/en/data-cloud/marketplace/`

1

The concept expanded to data management, where "data augmentation" takes on a broader interpretation, referring to the enrichment of a dataset through the integration of additional information sourced from related datasets within vast repositories [134]. This approach not only augments the quantity but significantly enhances the quality and relevance of the dataset. Tabular dataset augmentation in data management specifically targets identifying and integrating relevant features (e.g., columns) from related tables to increase the performance of ML models [36]. By integrating relevant external features, the models gain access to a broader context, which can significantly increase their accuracy and efficacy.

Tabular dataset augmentation is an important initiative to advance the field of automated machine learning (AutoML) [36, 93]. By automating the dataset discovery process and the feature selection phase, we move a step closer to completing the AutoML cycle, aiming for a fully automated ML pipeline. This advancement not only simplifies the model development process but also amplifies the potential to develop more sophisticated and nuanced ML models, thus pushing the boundaries of AI research and applications. This approach emphasises a data-centric perspective in AI, where the focus is on the quality and contextual richness of the data. Refining the datasets through augmentation increases the potential of ML models to deliver more accurate and reliable outcomes substantially.

Current tabular dataset augmentation techniques integrate the ML model into the process [36, 129]. These methods are often paired with existing feature selection techniques, relying on the ML model to identify the most relevant features [129] or devising their feature selection mechanisms specifically for data augmentation [36]. From the data management perspective, advanced techniques for fast indexing have been developed, thereby accelerating the computation of correlation coefficients, an essential step for feature selection [53]. Additionally, methods for developing embeddings that effectively capture the semantic context of tables have been introduced [205]. These methods enhance the effectiveness of data augmentation by ensuring that the most semantically relevant and statistically significant features are identified and integrated into ML datasets. Tabular dataset augmentation approaches gain advantages from previous data integration efforts. For example, identifying related datasets, specifically those that are joinable or unionable, is a foundational step in any tabular dataset augmentation strategy within data management. Considerable research is dedicated to finding unionable tables [146], as well as joinable tables [26, 50, 58, 206], with some studies addressing both types of relatedness [17, 202].

Following the initial phase of dataset acquisition, our subsequent task for creating high-quality datasets for ML models involves integrating and augmenting the collected datasets. With dataset integration, we begin a selection process to distinguish between datasets that, by joining, are directly relevant to our research objectives (i.e., related datasets) and those that add unnecessary noise (i.e., irrelevant or noisy datasets). This phase is followed by dataset augmentation and feature selection, both aimed at enhancing a table with additional relevant features (i.e., columns) that increase the information value of a dataset, thereby increasing its utility for ML model training. This workflow describes **feature discovery**, a novel method to augment a dataset with more features. Feature discovery is a more nuanced and targeted form of dataset discovery, as we focus on identifying datasets that contain features suitable for augmenting a table with more relevant information. The complexity of this workflow guides us towards our second research question:

***RQ2**: How can we enhance the automation of the feature discovery process to create high-quality datasets for machine learning applications?*

Findings & Contribution. To answer our second research question **RQ2**, in Chapter 4, we develop a novel automated feature discovery method, which reduces the need for manual data engineering efforts and improves the performance (i.e., accuracy) of subsequent ML models. We contribute with `AutoFeat`, the library for automated feature discovery over tabular datasets. `AutoFeat` retrieves relevant features for augmentation effectively, exploring beyond directly connected tables and efficiently, based on relevance and redundancy metrics, reducing the need to train the ML model in the process.

1.3 Human-in-the-Loop Feature Discovery

Within the automated and complex feature discovery pipeline, we often overlook the users and their influence on the process. User-centric research in data management has increasingly expanded its scope, exploring the complexities of interactive user interfaces [140] and leveraging data obtained through crowdsourcing methods [116, 117, 175]. These efforts indicate the critical role of the user as an integral component within the data management process. Traditionally, the emphasis has been on technological advancements, with user engagement often playing a secondary supportive role in the development and refinement of these technologies [130]. However, recent trends in the field are shifting towards placing greater importance on the users, recognising them as vital drivers of research progress.

For example, to effectively collect datasets, a deep understanding of the business domain and the specific application is indispensable. Such domain-specific knowledge, essential for tailoring the data collection process to meet the precise needs of the application, can predominantly be sourced by involving the user in the data collection process [198]. Furthermore, user-generated information plays a pivotal role in enriching data lakes, contributing significantly to the overall data quality and utility. By providing tags, linking information, and creating structured vocabularies or ontologies, users add invaluable context and metadata that enhance data discoverability and usability [70]. Moreover, data integration studies have been increasingly including user studies in their evaluation frameworks to ensure that the tools and methods developed are technologically advanced, user-centric, and responsive to user needs [58, 112].

The user's role is paramount in data science and ML pipelines. The significant impact of the user on these pipelines is exemplified by the adoption of user-centric applications, such as the collaborative environments Jupyter Notebook, JupyterLab[5], Google Colab[6], and tools designed to assist users in the data layer, such as Data Wrangler [96], Voyager [18], and Data Civilizer [46]. These tools and environments have been developed with a strong focus on enhancing user interaction and productivity, underscoring the critical role of the user in the successful implementation and optimisation of data science and ML workflows. By providing intuitive interfaces, comprehensive functionalities, and collaborative features, these applications empower users to effectively manage, explore, and derive insights from their data, showcasing the user's central role in DCAI.

[5] `https://jupyter.org`
[6] `https://colab.research.google.com/`

1

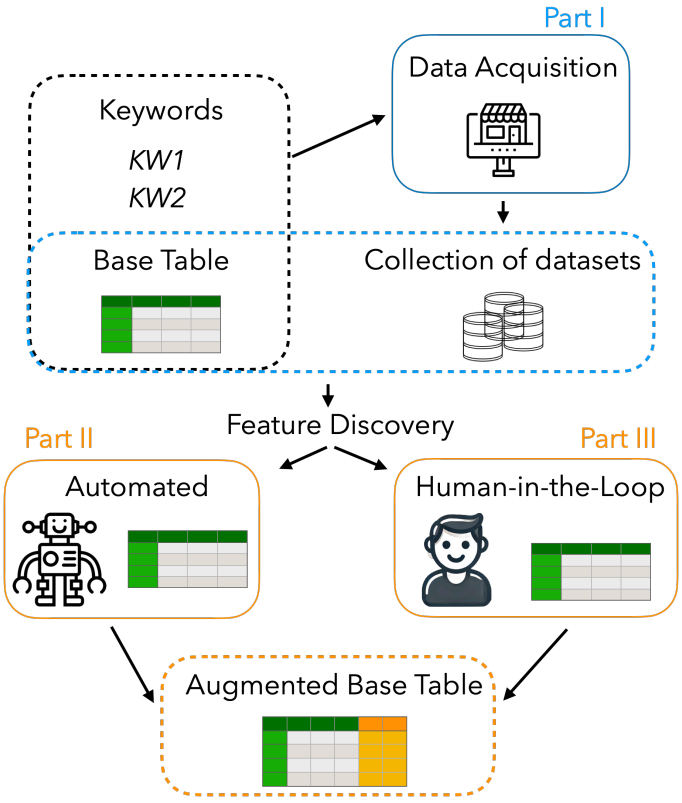


Figure 1.3: The figure summarises the process of enhancing a table with additional features, detailing the journey from the initial acquisition of datasets – which constitutes the first part of this thesis – to the final augmented table, which can be accomplished through two distinct methods for feature discovery: an automated approach – which is described in part two of this thesis – or a human-in-the-loop approach – which constitutes part three of this thesis.

The pivotal role of the user in creating and enhancing data, particularly as the domain expert, raises questions about the efficacy of fully automated approaches in ensuring high-quality data in DCAI. Thus, the following research question arises:

> ***RQ3**: Can human expertise and domain knowledge enhance the automatic feature discovery process?*

Findings & Contributions. To answer our third research question **RQ3**, in Chapter 5, we research the user's role within the feature discovery pipeline. Thus, we conduct a user study structured as a think-aloud use-case scenario with 19 participants to understand how data practitioners perform feature discovery in a real-life scenario. It is imperative first to understand how users interact with the feature discovery process, evaluating whether their involvement enhances the effectiveness of this process so that we can effectively adapt the automated process to meet the user's needs. Furthermore, by closely examining

the dynamics of user interaction within the feature discovery process, we aim to understand the advantages a human-in-the-loop approach could offer. In Chapter 6, we refine our automatic method, designing and developing a human-in-the-loop approach to feature discovery in DCAI. Tailoring the automated process to accommodate user requirements involves simplifying and ensuring that the system becomes flexible and intuitive for users, thereby facilitating them to produce high-quality training datasets for ML applications. We contribute with `HILAutoFeat`, a library for human-in-the-loop feature discovery over tabular datasets. We conclude the chapter by evaluating our approach with a subset of the participants from our first study. This evaluation helps us determine the balance between automation and human expertise, aiming to optimise the feature discovery process further.

By addressing these three research questions, we advance the objectives of data-centric AI to produce high-quality datasets. The process we envision, illustrated in Figure 1.3, starts with an initial dataset, subsequently referred to as *the base table*. Next, we offer an approach for the acquisition of related datasets that can be further augmented to improve the base table. Additionally, we introduce an automated method for tabular dataset augmentation, the feature discovery process which aids in selecting the most suitable features for the subsequent ML modelling efforts. Furthermore, our research explores the potential benefits of incorporating a human-in-the-loop strategy within feature discovery. By doing so, we aim to integrate the users' expertise and insights into the data-centric AI framework, paving the way for user-centric AI/ML workflows.

1.4 Thesis Origins

In this section, we enumerate the publications that form the research basis for each chapter and guide our analysis and discussions throughout the thesis.

Part I: Tabular Data Acquisition with Data Marketplaces

Chapter 2 is based on the following research paper:

- Andra Ionescu, Kostas Patroumpas, Kyriakos Psarakis, Georgios Chatzigeorgakidis, Diego Collarana, Kai Barenscher, Dimitrios Skoutas, Asterios Katsifodimos, and Spiros Athanasiou. 2023. Topio: An Open-Source Web Platform for Trading Geospatial Data. In: *Lecture Notes in Computer Science vol. 13893 - 23rd International Conference on Web Engineering (ICWE)*, (pp. 336-351).

Chapter 3 is based on the following research paper and demonstration paper:

- Andra Ionescu, Kostas Patroumpas, Kyriakos Psarakis, Georgios Chatzigeorgakidis, Diego Collarana, Kai Barenscher, Dimitrios Skoutas, Asterios Katsifodimos, and Spiros Athanasiou. 2023. Topio: An Open-Source Web Platform for Trading Geospatial Data. In: *Lecture Notes in Computer Science vol. 13893 - 23rd International Conference on Web Engineering (ICWE).*
- Andra Ionescu, Alexandra Alexandridou, Leonidas Ikonomou, Kyriakos Psarakis, Kostas Patroumpas, Georgios Chatzigeorgakidis, Dimitrios Skoutas, Spiros Athanasiou, Rihan Hai, and Asterios Katsifodimos. 2023.

Topio Marketplace: Search and Discovery of Geospatial Data. In: *Advances in Database Technology - Proceedings of the 26th International Conference on Extending Database Technology (EDBT)*.

🏆 This paper won the Best Demonstration Award at the 26th International Conference on Extending Database Technology (EDBT 2023).

Part II: Automated Feature Discovery for Tabular Data

Chapter 4 is based on the following workshop paper and research paper:

- Andra Ionescu, Rihan Hai, Marios Fragkoulis, and Asterios Katsifodimos. 2022. Join path-based data augmentation for decision trees. In: *Proceedings of the IEEE 38th International Conference on Data Engineering Workshops (ICDEW)*.
- Andra Ionescu, Kiril Vasilev, Florena Buse, Rihan Hai, and Asterios Katsifodimos. 2024. AutoFeat: Transitive Feature Discovery over Join Paths. In: *Proceedings of the IEEE 40th International Conference on Data Engineering (ICDE)*.

Part III: Human-in-the-Loop Feature Discovery

Chapter 5 is based on the following workshop paper and research paper, which is currently under review:

- Andra Ionescu, Zeger Mouw, Efthimia Aivaloglou, and Asterios Katsifodimos. Key Insights from a Feature Discovery User Study. 2024. In: *Proceedings of the Workshop on Human-In-the-Loop Data Analytics (HILDA)*.
- Andra Ionescu, Zeger Mouw, Efthimia Aivaloglou, and Asterios Katsifodimos. Feature Discovery for Machine Learning: a User Study, *under review*

Chapter 6 is based on the following demonstration paper:

- Andra Ionescu, Zeger Mouw, Efthimia Aivaloglou, Rihan Hai, and Asterios Katsifodimos. 2024. Human-in-the-Loop Feature Discovery for Tabular Data. In: *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*.

Additionally, this dissertation benefits from the following research papers:

- Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, and Asterios Katsifodimos. "Valentine: Evaluating Matching Techniques for Dataset Discovery". 2021. In: *Proceedings of the IEEE 37th International Conference on Data Engineering (ICDE)*.
- Rihan Hai, Christos Koutras, Andra Ionescu, Ziyu Li, Wenbo Sun, Jessie van Schijndel, Yan Kang, and Asterios Katsifodimos. "Amalur: Data Integration Meets Machine Learning". 2023. In: *Proceedings of the IEEE 39th International Conference on Data Engineering (ICDE)*.

I

Tabular Data Acquisition with Data Marketplaces

2

Introducing Data Market Platforms

In this chapter, we explore the evolving landscape of data marketplace platforms in data-centric AI, emphasising their role in facilitating the acquisition of high-quality datasets for ML applications. We begin by introducing data marketplace platforms and their increasing popularity as a source of valuable data. Subsequently, we present findings from our survey involving 122 data asset providers and consumers. This survey provides detailed insights into their needs and requirements, underscoring the importance of aligning platform functionalities with user expectations. Such alignment enhances usability and boosts participation, which is essential for the sustained growth and effectiveness of these platforms.

This chapter is based on the following research paper:

- *Andra Ionescu, Kostas Patroumpas, Kyriakos Psarakis, Georgios Chatzigeorgakidis, Diego Collarana, Kai Barenscher, Dimitrios Skoutas, Asterios Katsifodimos, and Spiros Athanasiou. "Topio: An Open-Source Web Platform for Trading Geospatial Data". ICWE 2023 [92].*

2.1 Introduction

In data-centric AI, the emphasis shifts towards creating high-quality training datasets rather than excessively engineering the ML model to overfit a given dataset [198]. As vast amounts of data continue to be produced, data is increasingly treated as a commodity, leading to the emergence of various businesses that derive value from trading data as an asset. This includes the development of data marketplaces and platforms, which are essential in the data economy but differ significantly in structure.

It is necessary to understand the distinction between a marketplace and a platform. Economically, a marketplace involves a first-party vendor relationship, where the producer sells the product to a retailer, who then owns the product. In contrast, a marketplace platform operates on a model where the producer sells directly to the consumer via the platform, establishing a third-party relationship [193]. In the data management community, data marketplace platforms are called shortly data marketplaces [2]. These platforms enhance the availability of quality datasets through value-added services, thereby supporting the development of more effective and efficient ML models and AI systems.

Despite the growing popularity of data marketplace platforms, consumers continue to face numerous challenges related to the purchasing process, such as contract negotiations and the legal complexities involved. Additionally, technical challenges, such as data format compatibility, transformations, and cleaning, further complicate the effective use of purchased data [33].

To address these challenges and foster a thriving environment for acquiring high-quality data through data marketplace platforms, the research community has established benchmarks and specific challenges. Notably, as part of the MLCommons DataPerf initiative [132], the Data Acquisition for ML (DAM) challenge has been developed to tackle key issues. These include ensuring transparent pricing, establishing unified data formats, and devising better acquisition strategies, particularly focusing on ML tasks [33]. This initiative aims to streamline the process and enhance the efficiency and efficacy of data transactions in market platforms. Moreover, it seeks to alleviate the consumer's burden by automating and optimising data acquisition strategies.

To better match supply and demand and encourage participation in data market platforms, several new designs for these platforms have been proposed [72, 75]. These proposals share a common focus on enhancing the efficiency of data marketplace platforms by facilitating the discovery of relevant assets and ensuring their acquisition at fair prices. This focus is critical in addressing the needs of both data providers and consumers, aiming to create a more transparent and equitable marketplace. By improving the mechanisms for asset discovery, these designs help users more easily find data that meets their specific requirements, which is vital for tasks such as training ML models or conducting advanced analytics. At the same time, the emphasis on fair pricing strategies aims to establish a balanced economic environment that encourages more entities to participate in the data marketplace, fostering a robust and dynamic ecosystem [33].

In this chapter, we first lay the foundational knowledge needed to understand the landscape of data marketplace platforms such that we can answer our first research question:

> ***RQ1**: How can dataset discovery approaches enable and facilitate data acquisition in data marketplace platforms?*

Moreover, to understand the key factors driving user engagement and satisfaction within these platforms, we conducted 122 surveys involving data asset providers and consumers, gathering insights into their needs and requirements. This analysis offers a deeper understanding of the marketplace dynamics and the challenges the participants face.

2.2 Data Marketplace Platforms

Data marketplace platforms have emerged in response to the need for effective data trading mechanisms, driven by the exponential growth of data and the widespread adoption of big data and cloud computing. These developments have contributed to the rise of the data-driven economy, where data is a critical asset for decision-making and innovation [45, 126]. The primary function of a marketplace platform is to facilitate the matching of providers (i.e., sellers) and consumers (i.e., buyers) [2]. Within the context of a *data* marketplace platform model, data itself becomes the asset being bought and sold, while the platform serves as the medium for these exchanges [122]. Formally, the data marketplace is defined as follows:

A data marketplace is a platform where users can upload and maintain datasets while various licensing models regulate access and usage [170].

Despite the proliferation of open data marketplace platforms, which offer open access to data, such as governmental data (e.g., U.S. Government's Open Data[1], EU Open Data Portal[2]), data marketplace platforms emerged as a business opportunity, allowing enterprises and individuals to monetise their assets [160]. Currently, the most prominent data marketplaces include Dawex[3], a global platform where companies can securely exchange data, Snowflake Data Marketplace[4], which allows users to share and access live, governed data and data services, AWS Data Exchange[5], which enables customers to find, subscribe to, and use third-party data in the cloud, Data Marketplace by Oracle[6], which offers a platform for accessing and sharing data across various industries, and Knoema[7], a platform where individuals and organizations discover, visualise, model, and present their data.

From an economic perspective, data marketplace platforms face significant challenges in achieving widespread adoption. The novelty of trading data introduces numerous challenges, particularly regarding pricing [45]. Consequently, data pricing has become one of the most extensively researched topics within the domain of data marketplace platforms across multiple research communities [1, 2, 9, 10, 121, 126, 157]. Besides pricing, data marketplace platforms also face technical challenges, such as developing trading market platforms [59], as well as ethical and economic challenges, such as arbitrage (i.e., protecting the data from being resold) [126]. The complexities in determining the value of data and concerns over data privacy and security present substantial obstacles that need to be addressed to foster broader acceptance and usage of data marketplaces [45, 126].

[1] https://data.gov/
[2] https://data.europa.eu
[3] https://www.dawex.com/en/
[4] https://other-docs.snowflake.com/en/collaboration/collaboration-marketplace-about
[5] https://aws.amazon.com/data-exchange/
[6] https://www.oracle.com/cloud/marketplace/
[7] https://knoema.com

Despite these challenges, specialised data marketplaces tailored to specific sectors have emerged and are subjects of ongoing research. Notably, platforms dedicated to facilitating data sharing for smart cities have gained attention [153, 160]. These marketplaces aim to enhance urban management and efficiency by leveraging the collective data generated within city environments. Data marketplaces designed for the Internet of Things (IoT) have also been developed [137]. These platforms focus on harnessing the vast amounts of data produced by interconnected devices to enable new forms of data commerce and insights. Moreover, there has been significant interest in data market platforms based on blockchain technology [108, 128, 152]. Blockchain-based data marketplaces offer enhanced security and transparency by using decentralised ledgers to facilitate data transactions. This technology ensures the integrity of data exchanges and builds trust among participants by providing a tamper-proof record of all transactions.

When data is traded between companies, the data exchange process becomes increasingly intricate due to the precise management required, such as schema definitions, data alignment, data standardisation and integration from disparate sources [55]. These complex processes ensure usability and value to the recipient [49]. Moreover, thorough data management approaches are required to support the variations in data structure and format and to ensure that the exchanged data retains its accuracy, relevance, and value.

2.2.1 Data Acquisition with Data Market Platforms

Using data platforms to acquire data for various tasks, such as training ML models or deriving statistics through data analysis, has become widely adopted [24, 28, 142]. However, the usability challenges associated with open data platforms have highlighted their limitations in acquiring valuable datasets [149]. As a result, data marketplace platforms have emerged as a preferable alternative for sourcing reliable data. In particular, collecting data to enhance (i.e., augment) a dataset with better predictive features or more data points to improve the performance of ML models has gained increasing attention [123, 125, 184]. To ensure the quality and relevance of data for such analytical tasks, extensive research has been devoted to addressing the challenges associated with data marketplace platforms. Key focus areas include identifying relevant assets [7] and fairly pricing these assets [10, 59].

The predominant strategy for addressing the challenges associated with data marketplace platforms is inspired by the multi-armed bandit model. This model provides a framework for balancing exploration and exploitation. Suppose the consumer has a predefined budget for acquiring data according to the pricing strategy determined by the platform. One approach within the multi-armed bandit framework starts by focusing on exploration. During this phase, data records are requested within the allocated budget to gain insights into the distribution of the provider's data. This knowledge allows for the design of more effective predicates for subsequent queries. Following the exploration phase, the consumer enters the exploitation phase, where the remaining budget is allocated based on the estimated utility of the data records [123].

The same research proposes a second, more integrated solution that continuously balances exploration and exploitation. This method involves iteratively posing queries that request a small number of records. The balance is achieved by simultaneously aiming to obtain more records with high expected utility and monitoring the diminishing returns of

each predicate as additional records are acquired [123]. This research indicates that these strategies are effective in acquiring records with higher relevance for ML models, which in turn is more likely to enhance accuracy. By adopting these strategies, consumers can optimise their data acquisition processes and ensure they derive maximum value from their investments in data marketplace platforms.

Another algorithm, known as the incremental estimation of adaptive score, effectively balances the trade-off between exploration and exploitation and is applicable across various data pools, including open data portals, data lakes, and data marketplace platforms [184]. The issue can be addressed statistically by simplifying the challenge of optimising the worst-case variance into finding an equilibrium in a zero-sum game between the provider and consumer [35]. To address the limitations concerning availability and efficiency, a different approach proposes using a learned model to estimate the potential for confidence improvement in samples. This facilitates efficient data acquisition without requiring full access to the data pool. The model owner trains a predictive model based on available data, which the data provider then uses to identify and provide the most valuable samples within a budget [125].

While data acquisition in data marketplaces presents significant challenges, the research and development of sophisticated algorithms and models have shown promising results in overcoming these obstacles. However, more efforts are needed to create an open data marketplace platform. Such a platform would aggregate these efforts, allowing for empirical evaluation with real users and real datasets to facilitate effortless data trading.

2.3 User Surveys

Many data marketplaces or data sharing platforms have been oriented towards the needs of data providers, developing and supporting features that cater primarily to their requirements [75]. Recognising the distinct perspectives and needs of providers and consumers, match-making platforms have begun to surface to bridge this gap [8, 75]. Thus, there is a need to develop a platform that addresses and harmonises the requirements and preferences of both data consumers and providers. Towards this goal, we conducted user surveys to identify and evaluate the desired qualities and features of a web-based data marketplace platform from the viewpoints of both providers (27 responses) and consumers (95 responses). This approach ensures a balanced consideration of the needs of both sides, aiming to foster a more inclusive and efficient data trading ecosystem.

2.3.1 Providers

The survey aims to understand the needs and preferences of data providers in the context of data marketplaces. The survey targeted stakeholders from various backgrounds (e.g., geography, information technologies, marketing), roles (e.g., legal experts, analysts, managers, developers), and business sectors (e.g., asset production, digitisation, geo-marketing). It consists of 44 questions and is designed to last approximately 15 minutes, beginning with a short introduction explaining the survey's primary goals and target audience. The questions are then divided into five categories: market activity, data assets, contractual life cycle, digital single market, and value-added services, ensuring a comprehensive understanding of user expectations and requirements from a web-based data market platform.

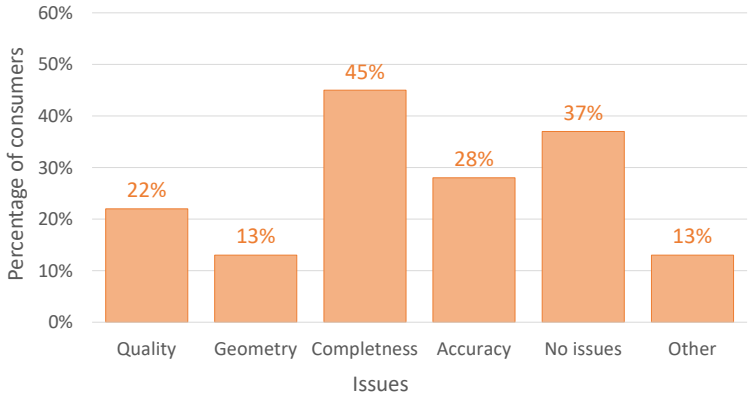


Figure 2.1: The figure illustrates the range of issues providers have identified as common complaints from consumers, according to survey responses.

Market activity. The majority of data providers in the geospatial domain currently list fewer than ten data assets for sale, with a common practice being to sell two to ten geospatial data assets to the same customer. Additionally, a significant portion of these providers have not yet embraced the concept of selling their assets through a digital marketplace. Furthermore, nearly half of the data providers do not offer their assets as a service. This observation suggests an opportunity for growth and innovation in the marketing and delivery of geospatial data assets to consumers.

Data assets. The landscape of geospatial data provision reveals that most providers, who are also the producers of their assets, do not distribute their offerings through a catalogue or utilise an asset management system. Additionally, a significant portion of these providers does not facilitate access to their assets via web services. Among those that do provide web service access, there is a preference for services based on standards by the Open Geospatial Consortium (OGC), such as Web Map Service (WMS) and Web Feature Service (WFS), or those using RESTful APIs. The predominant format for these assets is shapefiles (SHP), followed by CSV (Comma Separated Values) as the second most preferred format. Finally, the providers reported that most consumers raised concerns primarily about the completeness of the data. There are also notable complaints regarding the quality, accuracy, and geometry of the assets, as depicted in Figure 2.1. This feedback highlights critical areas for improvement in the provision of geospatial data, suggesting that providers need to enhance the accuracy, quality, and management of their geospatial assets to better meet consumer expectations and requirements.

Contractual life cycle. Over 60% of the survey respondents indicate that they include their terms and restrictions directly within a contract via license embedding, whereas the necessity for a contract to be signed is reported by only 57% of the participating data owners and producers. Notably, a significant proportion of these agreements accept a *digital* signature, underscoring the growing acceptance of digital methods in formal transactions. Interestingly, many providers report that a signed contract is not required for the transaction to proceed. Regarding the delivery of purchased geospatial data assets, the most

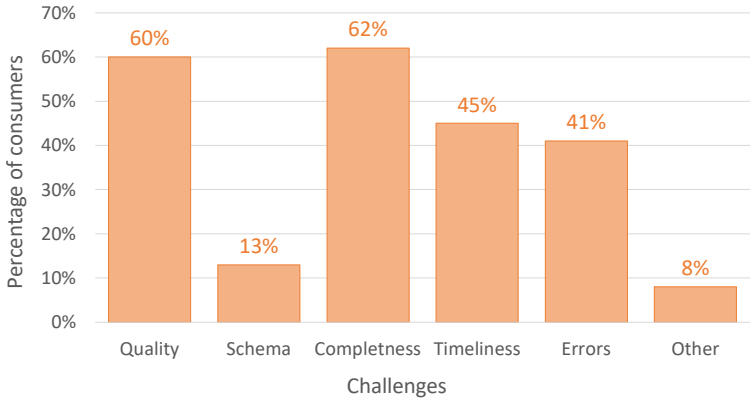


Figure 2.2: The figure summarises the key challenges consumers encounter when purchasing data assets, as identified in the survey.

common method reported by data owners and producers is through their websites, with email and physical media distribution following behind.

Digital single market. An overwhelming majority of the surveyed data owners and producers, exceeding 95%, expressed interest in participating in a digital marketplace. Yet, the primary obstacles identified for joining such a platform include the standardisation of pricing and contracts, along with the payment process. When considering the financial aspects of joining a digital market platform, providers preferred a model that involves a fixed commission on the sale price of each asset without a participation fee, with 42% favouring this approach. The second most preferred option, favoured by 23% of the respondents, involves a zero-fee structure.

Value-added services. Lastly, when asked about their readiness to adopt and use the services offered by a digital marketplace, over 85% of data owners believed that such a marketplace would augment their sales and revenue.

2.3.2 Consumers

To understand the dynamics involved in geospatial asset searching and purchasing, we conducted surveys with consumers to collect detailed insights into market activity, data assets, and the digital single market. The survey comprises 25 questions organised into three categories, facilitating a structured exploration of these areas.

Market activity. Similar to the data providers, most consumer respondents are from the Information and Communications Technology (ICT) sector, followed by notable representation from the environmental and software development sectors. Geospatial data consumers typically acquire geospatial data assets infrequently, with most making purchases only once or annually. Additionally, a significant portion of geospatial data consumers predominantly utilise *open* geospatial data assets, highlighting a preference for openly available resources within this community.

Data assets. Consumers typically engage with diverse georeferenced data types, promi-

nently including census information, place names, and socio-demographic data. Regarding vector data assets, administrative boundaries, Points of Interest (PoI), and road networks stand out as the top three choices for usage. When it comes to raster data assets, thematic maps, along with aerial and satellite imagery, are the preferred options.

Interestingly, most geospatial data consumers do not produce their own data assets, be it through direct generation or indirect contributions. Regarding the preferred format for receiving purchased geospatial data, shapefiles emerge as the clear favourite. Additionally, services that offer functionalities similar to those of Google Maps are widely used among consumers, with services based on OGC standards, RESTful APIs, and Geospatial Analytics also enjoying substantial popularity.

The challenges consumers face in the geospatial data market are notably concentrated around several key issues. Firstly, data availability poses a significant hurdle for 77% of consumers, indicating a pressing need for broader access to geospatial datasets. This is closely followed by concerns over the lack of transparent information regarding data quality, which is mentioned by 62% of consumers, underscoring the importance of clear and accessible quality metrics. Moreover, the licenses and contract terms remain a significant area of uncertainty for 52% of respondents, pointing to a demand for more straightforward and user-friendly licensing agreements.

The purchasing process further reveals critical areas for improvement, with the completeness of data assets, quality, and timeliness being the foremost concerns, highlighted by 61%, 60%, and 44% of consumers, respectively. Additionally, general errors within the datasets were noted as a concern by 41% of respondents. These insights, as illustrated in Figure 2.2, underscore the need for enhanced measures to ensure data integrity, timeliness, and transparency to address consumer challenges in the geospatial data market.

Digital single market. An overwhelming majority of data consumers surveyed, exceeding 95%, expressed keen interest in joining the marketplace. This solid positive response highlights the perceived benefits of the platform. Through their participation in the marketplace, consumers have clear expectations critical for the platform's success and relevance.

Primarily, 85% of consumers anticipate the ability to discover and acquire assets effortlessly, underlining the importance of a user-friendly and efficient search and purchase process. Additionally, 74% of respondents stress the need for transparent terms and restrictions available before the acquisition of assets, pointing towards a demand for clarity and predictability in transactions. Providing high-quality data is a priority for 65% of the consumers, highlighting the value placed on the reliability and accuracy of the data assets available through the platform. Cost transparency is another critical expectation for 63% of consumers, indicating a desire for clear, upfront pricing structures. Lastly, uniform formats are sought after by 50% of the respondents, underscoring the need for standardisation to facilitate ease of use and integration of data assets.

2.3.3 Summary

Surveying both data providers and consumers has revealed significant market interest and demand for a comprehensive portfolio of services. This insight underscores a vibrant potential for a marketplace platform to revolutionise how geospatial data assets are exchanged. A notable observation from the survey is that most data owners have not yet

embraced digital marketplaces for offering their assets, signalling an opportunity to fill a crucial gap in the geospatial data asset market.

The survey further reveals a convergence between consumers and producers regarding the preferred format for assets, with both groups favouring SHP and services such as OGC and REST APIs. This variety in delivery methods and services mirrors the broad spectrum of needs and preferences across the geospatial data marketplace.

Moreover, the challenges highlighted by consumers encapsulate the core issues, such as simplifying the process of publishing and discovering assets and providing industry-focused, relevant metadata. A critical insight from the responses is the prevailing uncertainty surrounding the quality and suitability of a geospatial asset before purchase, which hinders both initial usage and repeat transactions.

2.4 Conclusion

In this chapter, we report on the significant progress made in the data acquisition processes within data marketplace platforms. Advances in this area have led to developing sophisticated pricing strategies and creating price-aware algorithms. These innovations enable more precise and efficient financial constraint management, securing valuable data and enhancing the overall effectiveness and accessibility of data marketplace platforms. Moreover, our surveys reveal that aligning with the preferences and expectations of data providers and consumers significantly enhances the usability and appeal of these platforms, which can contribute to a more dynamic and trusted ecosystem for exchanging data assets, fostering a more robust marketplace. We introduced the foundational knowledge which paves the way towards answering our first research question:

> ***RQ1**: How can dataset discovery approaches enable and facilitate data acquisition in data marketplace platforms?*

In the next chapter, we will focus on the technical challenges involved in developing an open data marketplace platform specifically for data acquisition. We will advance past pricing mechanisms and concentrate solely on asset discovery and relevance, exploring how these factors influence the effectiveness of data marketplace platforms in meeting the users' needs and answering **RQ1**.

3

Facilitating Dataset Acquisition with Topio Market Platform

In this chapter, we report on the effort to design and develop an open-source modular data marketplace platform designed to empower entrepreneurs and researchers to establish and experiment with data marketplaces. We have researched and developed methods for data profiling, dataset search and discovery, and data recommendation, providing access to them as open-source libraries. We discuss the integration of these libraries to create Topio*, a real-world web platform dedicated to trading geospatial data, showcasing the practical application and impact of our work in facilitating data acquisition. The libraries presented in this chapter are openly available at* https://github.com/opertusmundi/

This chapter is based on the following research paper, demonstration paper, and open-source resources:

- *Andra Ionescu, Kostas Patroumpas, Kyriakos Psarakis, Georgios Chatzigeorgakidis, Diego Collarana, Kai Barenscher, Dimitrios Skoutas, Asterios Katsifodimos, and Spiros Athanasiou. "Topio: An Open-Source Web Platform for Trading Geospatial Data". ICWE 2023* [92]
- *Andra Ionescu, Alexandra Alexandridou, Leonidas Ikonomou, Kyriakos Psarakis, Kostas Patroumpas, Georgios Chatzigeorgakidis, Dimitrios Skoutas, Spiros Athanasiou, Rihan Hai, and Asterios Katsifodimos. "Topio Marketplace: Search and Discovery of Geospatial Data". EDBT 2023* [86]
- *Source-code* [82], *data* [85], *and the data marketplace platform* Topio*:* https://topio.market/

3.1 Introduction

The growing interest in exchanging datasets and creating value from them has led to the development of data marketplaces (DMs). As such, DMs treat data as a commodity and aim to facilitate and streamline data trading between data providers and consumers. Data may be exchanged directly by offering a dataset itself or indirectly by providing services on top of it [8].

Many DMs have been developed over the last years with highly diverse characteristics. As a result, the landscape is quite fragmented, lacking any interoperability standards [8]. DMs can be used to find and acquire specialised and high-quality data that are needed to train ML models, which are, in turn, crucial for many industrial or societal applications [124]. They can be general-purpose, such as AWS Data Exchange[1] or Datarade,[2] or focused on a specific industry or type of data. For instance, big geospatial data providers (e.g., Carto[3], Here[4]) have recently integrated private marketplaces into their platforms. A DM is typically expected to deal with commercial data assets; nevertheless, as pointed out in [8], there also exist some DMs that generate revenue by monetising the effort to collect and link open data, making them more easily and readily exploitable.

Moreover, research of DMs mainly focuses on investigating pricing policies and models for data [59, 157]. However, DMs struggle with many traditional data management challenges, such as data profiling and integration, metadata curation and enrichment, dataset search and recommendation. Such problems have been studied in the context of data catalogues and data lakes [29, 135, 145, 150]. Data lakes, however, typically deal with open datasets or data exchanged among users of the same organisation, whereas data in a marketplace is an asset to be traded. This underscores the urgent need for mechanisms that enable buyers to quickly and effortlessly discover relevant datasets and assess their suitability for a specific task before committing to a purchase. Developing a data marketplace is challenging, as companies ranging from large multinationals to young start-ups prefer products that bundle data with services in technically and business-streamlined offerings. Furthermore, the inherent flexibility, scalability, simplicity, and low-cost nature of such services overcome the comparative higher data quality offered.

In this chapter, we answer our first research question:

> ***RQ1**: How can dataset discovery approaches enable and facilitate data acquisition in data marketplace platforms?*

To answer this question, we present Topio marketplace, an instance of our open-source marketplace platform for geospatial data, which facilitates data exploration, discovery, and augmentation. We introduce the main design decisions and the challenges we had to overcome when developing the platform. Topio is designed with **openness and reusability** in mind: all of the components are packaged as reusable libraries (e.g., for data discovery, data pipelines, data profiling, etc.). We believe these reusable libraries can provide value to researchers and practitioners alike. We also offer descriptions of the different li-

[1] https://aws.amazon.com/data-exchange/
[2] https://datarade.ai/
[3] https://carto.com/spatial-data-catalog/
[4] https://www.here.com/platform/marketplace

braries we have developed alongside links to their respective repositories. These libraries can be used together to form a platform on which various data marketplaces can be built.

The goal of Topio is to develop a digital single market for proprietary geospatial data, addressing the heterogeneity, disparity, and fragmentation of geospatial data products in a cross-border and inclusive manner. Our goal is inspired by and grounded on the real-world landscape and industry-led challenges of the fragmented geospatial data value chain. Topio marketplace is a central hub and a one-stop shop for the streamlined and trusted discovery, sharing, trading, and use of proprietary and commercial geospatial assets. It offers high-quality value-added services and addresses geospatial data product heterogeneity, disparity, and fragmentation. The platform is simple, fast, cost-effective and safe for data providers and consumers. Topio facilitates the decision-making process by offering a descriptive suite of metadata and mechanisms to discover related assets and to augment one or more assets from the purchased asset collection. In this chapter, we make the following contributions:

- We present the underpinnings of Topio, the first open-source marketplace platform for geospatial data developed for publishing and purchasing assets (Section 3.3).
- We illustrate the asset lifecycle process throughout the platform and provide a pragmatic approach towards pricing (Section 3.4).
- We outline a suite of scalable, low-cost value-added services that we built on top of industrial geospatial assets published in the platform (Section 3.5).

3.2 Related Work

In this section, we explore existing research on data marketplace platforms and open data platforms. This overview provides a foundation for our study, highlighting significant contributions and challenges in developing and operating these platforms. We aim to position our work within this context, showcasing how it addresses gaps and builds upon the current understanding of data platform ecosystems.

3.2.1 Data Marketplace Platforms

Although many DMs have emerged over the last few years, they are highly diverse with respect to their characteristics, and the landscape is quite fragmented, lacking any interoperability standards [8]. Moreover, DMs have recently become an active area of research, with many works focusing on investigating pricing policies and models for data [2, 31, 34, 59, 127]. Still, DMs face many traditional data management challenges, such as data profiling and integration, metadata curation and enrichment, and dataset search and recommendation. Such problems have also been studied in the context of data catalogues and data lakes [29, 135, 145]. These, however, typically deal with open datasets or data exchanged among users of the same organization, whereas data in a marketplace is an asset to be traded. This makes even more imperative the need for mechanisms to facilitate buyers to quickly and easily discover relevant datasets and to be able to assess the suitability of a candidate dataset for a given task before proceeding to its purchase. Our assessment identified the lack of comprehensive and precise metadata as a significant deficiency of the current market landscape.

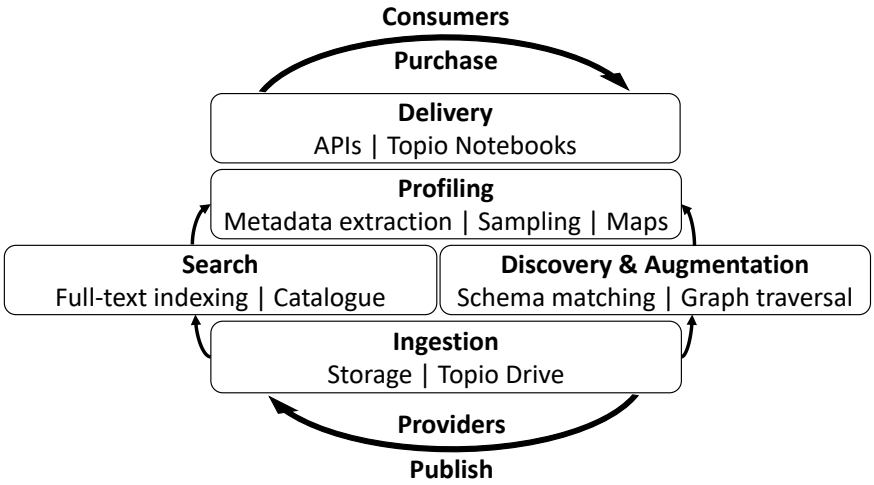


Figure 3.1: The figure shows an overview of Topio highlighting the search and discovery components and data asset lifecycle.

3.2.2 Open Data Platforms

Despite the extensive efforts of the research community towards data platforms openness and their added benefits (e.g. developing data-driven insights and analytics modules) [43, 141, 148], to the best of our knowledge, there is no existing open-source platform that facilitates building and running data marketplaces. Topio is the first open-source set of tools that can be used to build a data marketplace. Currently, Topio focuses on spatial data assets and can be easily extended to other data models and types.

3.3 Platform Overview

The design of Topio marketplace is inspired by the insights gathered through our surveys with data market users, the data providers, and data consumers, which we present in Section 2.3. Therefore, we focus on the following three objectives: (*i*) providing as much information about the assets as possible before the acquisition; (*ii*) supporting multiple asset formats and delivering them via web services; and (*iii*) providing means to discover and integrate various assets to improve the completeness, and quality. Through Topio design, we offer the absolute control of owners over their assets and our flexible support for real-world value chain instances along the entire lifecycle of geospatial data.

Figure 3.1 provides an overview of the components of the Topio marketplace platform. First, the geospatial assets are ingested and stored in Topio Drive. The data asset lifecycle includes publishing, purchasing, delivery, and pricing based on the selected asset delivery option (Section 3.4). We developed value-added services (VAS) to increase consumer benefits, including dataset discovery, a recommender system, and a profiler. These benefits are twofold: (*i*) better understanding of the value of the assets based on the metadata computed by the profiling service, and (*ii*) easier search and discovery and personalised recommendations of related or complementary data assets (Section 3.5).

3.4 Data Asset Trading

In this section, we describe the lifecycle of a data asset, tracing its progression from provision to the various delivery options available (Section 3.4.1). Following this, we review existing research on data pricing, discussing which concepts have been integrated into the Topio platform and how they influence our approach to asset valuation (Section 3.4.2).

3.4.1 Data Asset Lifecycle

This section presents the path a data asset takes from its initial provision to its final delivery. This journey consists of several key steps: the provision of the asset, its ingestion into a system, acquisition by consumers, and, ultimately, its delivery. This sequence ensures that data assets are prepared, managed, and distributed effectively, serving the needs of both providers and consumers in the data ecosystem.

Asset Provision. The provider of an asset has complete and highly granular control over the asset and can define if, when, and how an asset will be available at any point in the asset's lifecycle. An asset (e.g., file, database, service) is provided in a stand-alone manner, as a file with small or ad-hoc transformations, or derived/integrated with other assets. An asset is published in the platform along with its license, price policy, and contract terms. Publishing can be limited to metadata publishing alone or the metadata and the data asset itself.

Asset Ingestion. The entry point of data assets is the storage or Topio Drive. A data asset is uploaded, versioned, curated, and stored in the underlying storage. From there, it is delivered to consumers by transforming it into their preferred format. The data suppliers can provide descriptive metadata about an uploaded asset (e.g., format, price, coverage, topic, etc.). The ingestion service[5] encapsulates three main features: (*i*) reading, parsing, and extracting data types, (*ii*) storing the asset into a PostgreSQL[6]/PostGIS[7] database, and (*iii*) registering the asset as a service (e.g. publishing a layer associated with a PostGIS datastore to GeoServer). After publication, the asset can be available as a Web Map or Feature Service (WMS/WFS) from a GeoServer instance.

Asset Acquisition. Once an asset is uploaded in Topio Drive, the asset is immediately available throughout the application and all the services. The consumers can browse the asset catalogue and discover the desired assets based on the available metadata (Section 3.5.3). The consumer retains the right to access and use the assets within the Topio platform through notebooks or maps.

Asset Delivery. Topio delivers the assets and services in three main ways: (*i*) via Jupyter Notebooks after establishing an appropriate contractual agreement with the interested party (e.g., the platform or another asset owner) governing how joint value is created and shared, (*ii*) a service in one of the available APIs, and frameworks or (*iii*) integrated/derived and provided as a file. Following, we outline the asset delivery approaches.

> Topio **Notebooks** – Topio enables the consumers to directly use all geospatial assets purchased and uploaded and perform operations such as data cleaning and enrich-

[5] https://github.com/OpertusMundi/ingest
[6] https://www.postgresql.org/
[7] https://postgis.net/

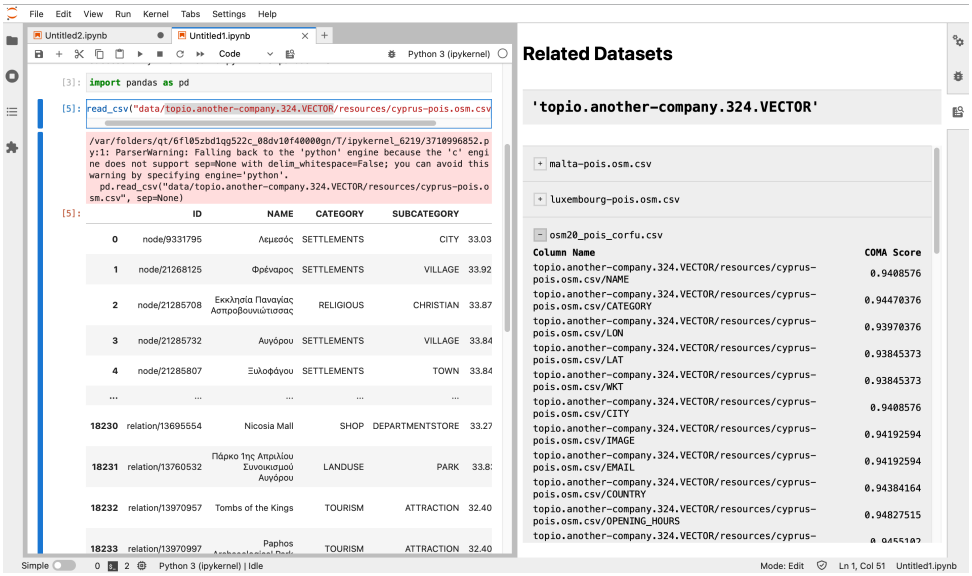


Figure 3.2: View of the discovery service in the notebook environment. On the right-hand side, we show how the user can inspect the list of related assets.

ment, geocoding and trend detection, and analysing satellite imagery in an online notebook. The notebook is backed by resources provided by Topio marketplace, which are charged to the data consumer in a separate agreement. This way, data analysis and transformation can be done without downloading the assets, enabling the use of high-value/size and complex assets with minimal effort. While working in the notebooks environment, Topio can automatically recommend new data sources for enrichment and integration based on the currently used data, as illustrated in Figure 3.2.

Moreover, with Topio Notebooks, the users can assess the quality and fitness of an asset before purchasing. Inside this environment, the users benefit from the discovery service (Section 3.5.2) and can directly inspect the list of related assets. This functionality helps them discover more assets without actually checking the asset page in the marketplace. Therefore, the users can only focus on processing, analysing, and understanding an asset's usefulness without interruptions.

Topio **Maps** – Topio Maps is a comprehensive framework for creating, using, sharing, and integrating interactive maps in web and mobile applications. The consumer can create custom maps using not only the data and services provided by the platform but also proprietary data. Some metadata extracted from spatial attributes is represented using maps. We represent the *spatial extent* using the Minimum Bounding Rectangle (MBR), the *convex hull* of the geometries, and the *spatial data distribution* using heatmaps.

Physical Delivery – Finally, the purchase and delivery of the asset are performed

within or outside the platform, according to owner preferences and asset type. When the files are very large or other constraints become an issue (e.g., company policies), data assets can be physically shipped to consumers.

3.4.2 Pricing Models

Much research has been done concerning pricing models for data [31, 59, 107, 127]. Early works primarily focus on pricing views of data assets such that they are arbitrage- and discount-free [107]. These pricing schemes help ensure that: (*i*) a buyer will not buy "cheaper" views of a dataset whose union costs less than the original dataset, and (*ii*) the use of these concepts in practice requires both training of the data providers but also a complete pricing market architecture to support such pricing schemes.

Topio's Pragmatic Approach to Pricing. During our research for pricing schemes, we investigated the possibility of deriving the prices from selling either a subset of the datasets or views of those datasets, but this was a very challenging task. When talking to data providers during our surveys (Section 2.3), the most common request was that the providers set a price for their dataset and a separate price for each of their derivatives (e.g., a subset of the businesses in France) set by the suppliers.

At this stage, `Topio` prices datasets in two main ways: (*i*) pay per dataset and (*ii*) pay per API call on a value-added service. The former is the simplest form of pricing: a provider offers a dataset to consumers for a fixed price and can provide discounts on bundles of datasets. For the latter, when consumers read data from value-added service APIs, as described in Section 3.5, providers can set a price per API call. API calls are logged, and the consumers are charged per call. We also offer consumers the possibility of buying API-call credits, e.g., buying 1M calls for a fixed price.

3.5 Value-Added Services

In Section 2.3, the surveys indicate that both providers and consumers face challenges coming from the assets themselves, such as quality, geometry, schema and, most importantly, completeness and accuracy. The value-added services provide a step forward towards facilitating asset completeness and accuracy, as they help discover new assets suitable for integration. Moreover, VAS helps us circumvent the deadlock where consumers are unsure about the data quality while providers are reluctant to disclose detailed information before securing the payment.

Figure 3.3 visually presents the functionality of the following open-source value-added services that we have developed and integrated into `Topio`: (*i*) data asset search which offers metadata-, faceted-, keyword-based search functionalities throughout `Topio`'s catalog (Section 3.5.1), (*ii*) data asset discovery and augmentation, which helps the user find related assets and provides multiple ways of augmenting them (Section 3.5.2), and (*iii*) data asset profiling to automatically extract various kinds of information from the content of a given asset and enrich its description (Section 3.5.3).

3.5.1 Data Asset Search

`Topio` offers rich search capabilities with a wide range of optional filtering criteria so prospective data consumers can quickly identify assets of their interest. All search opera-

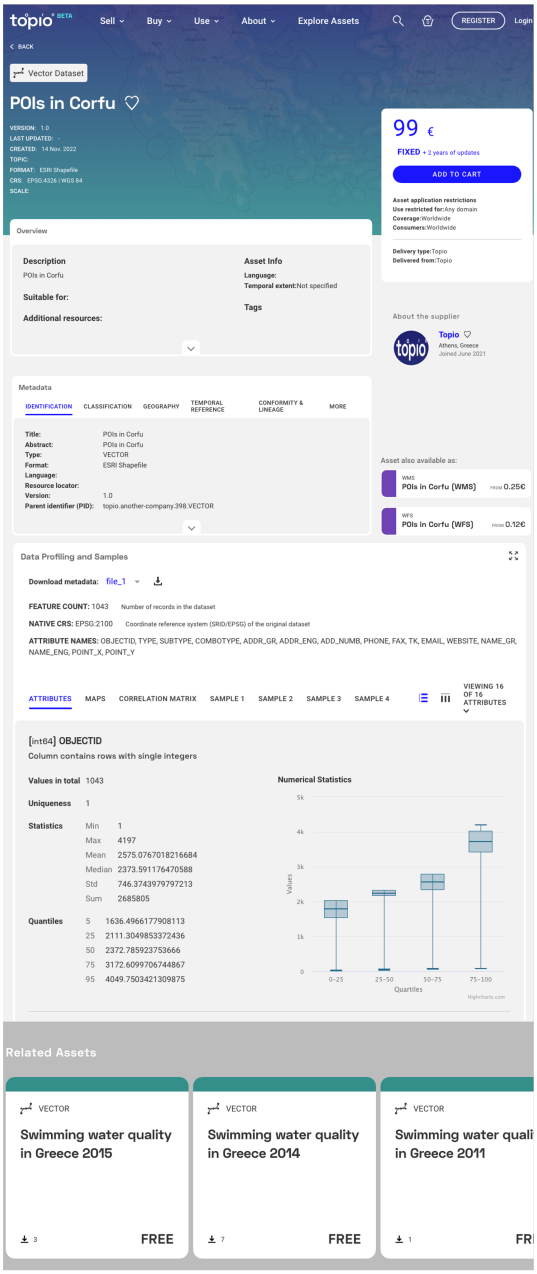


Figure 3.3: View of an asset page showcasing detailed metadata, in-depth profiling information, and related assets to the one being viewed.

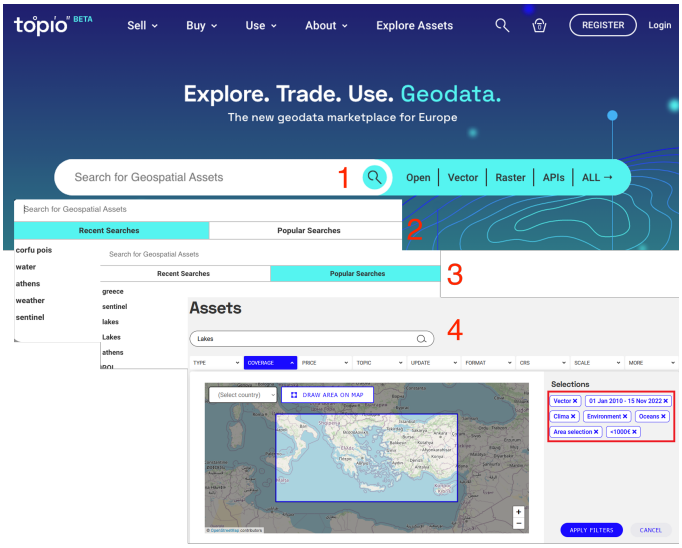


Figure 3.4: The figure illustrates the four types of input queries available for searching data assets on the platform: (1) keyword search, allowing users to input specific terms directly; (2) exploration of recent searches, offering insights into the user's past queries; (3) browsing through popular searches, highlighting trending or frequently sought-after data; and (4) utilising advanced search functionality, which enables more refined and targeted queries based on multiple criteria.

tions are powered by indexing all assets and their metadata (provided by the supplier), thus supporting various search conditions (e.g., textual, numerical, spatial, temporal). Some filtering conditions may come from pre-defined choices (e.g., asset types and file formats). In contrast, others can be user-specified (e.g., price range), enabling potential consumers to narrow their selection to assets that mostly match their preferences based on multiple filtering criteria. The search options are illustrated in Figure 3.4. The platform uses tools, such as PostgreSQL full-text indexing and Elasticsearch[8].

Catalogue. The catalogue assumes the role of geospatial-aware catalogue software, responsible for managing and maintaining metadata integrity and providing consistent, well-defined geospatial information to users and components. The catalogue services support publishing and searching collections of descriptive information (e.g., metadata) for data, services, and related information objects. The catalogue is based on a two-tier architecture. The first component is the *Metadata Store*, which comprises the Relational Database Management System (RDBMS) that stores and manages asset metadata. The metadata store is developed on top of the PostgreSQL with PostGIS (spatial) extension. The second component is the *Geospatial Catalogue API*, which publishes geospatial asset metadata to the other sub-systems of `Topio` architecture. The API is based on the OpenAPI 3.0 specification and is implemented in Python[9].

[8] `https://www.elastic.co/`
[9] `https://github.com/OpertusMundi/catalogue-service`

3.5.2 Data Asset Discovery & Augmentation

Dataset discovery is the process of navigating numerous datasets to find relevant ones and the relationships between them [106]. Its output represents the initial step in a data management pipeline and the input for schema matching, mapping and the subsequent processes [17]. The data asset discovery process is primarily used with tabular data, such as CSV, web tables, and spreadsheets [106].

Schema Matching. Dataset discovery is a difficult task as it relies on schema-matching techniques, which capture the semantic or syntactic relationships between datasets. These techniques are not entirely precise, as the datasets might share similar information but in a different context, which is difficult to capture by the schema matching methods alone [106]. The process tries to find a subset of relevant datasets that are similar or complementary in a certain way, e.g., with similar attribute names or overlapping instance values [17].

Our discovery process for geospatial data assets adopts methods from the Semantic Web, primarily using RDF and ontologies [11, 113]. Data mining and knowledge discovery approaches for geospatial data assets emphasise searching for co-location patterns given location points [79]. Dataset discovery is integral to Topio's asset discovery, enabling the platform to proactively or reactively recommend assets suited to their workflow context. The discovery service[10] allows end users to explore the collection of datasets by examining and understanding the relations between them and how they interconnect. During the discovery process, the users get more knowledgeable and understand the different layers of relatedness between the datasets, enabling them to make informed purchasing decisions.

Filtering. In the context of data marketplace platforms, where different types of datasets can be published and transformed for purchase, we employ the methodologies of structured tabular data for geospatial data. As such, we reduce system complexity by utilising the metadata extracted using the profiler component of the platform, which we will describe in the next section (Section 3.5.3). Such metadata is used as a filtering step to reduce the number of datasets to process for discovery.

Transformation. We use open-source software to transform geodata into CSV, such as mapshaper[11] [74]. The tool addresses the challenges posed by geodata formats, such as Shapefiles and GeoJSON, which are non-topological data formats. These formats do not store topological relationships between adjacent polygons. Instead, the shapefile represents polygons by rings, which are closed loops. With the transformed files, we then apply the discovery service based on [87]. This service relies on tools such as Metanome [154] for capturing dependencies between datasets, and also Valentine [106] for applying schema matching algorithms and capturing semantic or syntactic relationships between datasets.

Augmentation. The core of our discovery service is the *Dataset Relations Graph* (DRG) [87, 88]. The role of DRG is to encode information from different sources in a simplified and principled manner. We define a DRG as a directed graph with nodes representing columns with properties derived from data profiles and other automatically extracted metadata. The edges portray diverse relationships, such as syntactic similarity, as well as subsumption relations and joinability conditions. The discovery service leverages Valentine [106], our

[10] https://github.com/OpertusMundi/discovery-service
[11] https://mapshaper.org/

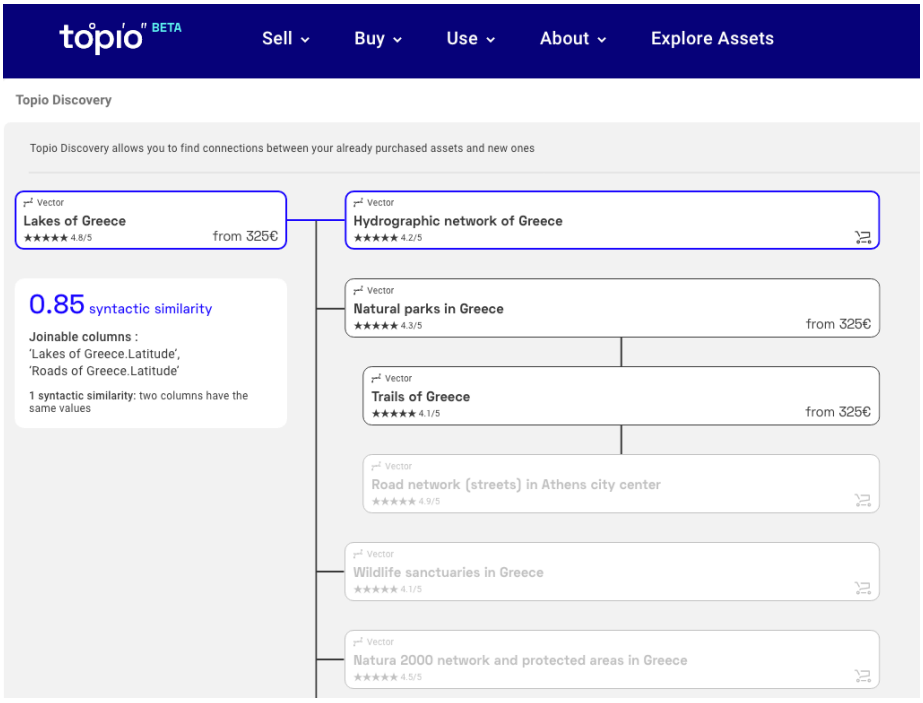


Figure 3.5: Depiction of the asset discovery and augmentation process, highlighting the current asset on the left side and displaying all potential assets for augmentation on the right side.

open-source schema matching tool providing various methods for computing similarities. We store the DRG in Neo4J and leverage graph traversal algorithms to create paths between the assets. The discovery service uses transitive relations and presents the user with multiple alternatives to link two assets.

The goal of augmentation is to recommend top-ranked assets which can be augmented to a given asset, named base asset. The approach consists of two steps. The first step is the enumeration of all the possible join paths to discover the assets that are not directly joinable with the base asset and could connect to the base via a series of transitive joins. The second step is ranking join paths using a ranking function integrated with feature importance measures to reduce the set of joined tables returned to the user. We use this feature to augment the user's purchased assets. Figure 3.5 shows an example of the data asset discovery process used for augmentation.

Recommendation. `Topio` provides contextualised asset recommendations to marketplace users, allowing them to discover a wide range of related geospatial assets. `Topio`'s recommender service[12] combines several data sources from the marketplace into a consolidated knowledge graph following the DCAT[13] ontology. This knowledge graph is an expressive and powerful data structure that naturally models the user-item marketplace

[12] `https://github.com/OpertusMundi/recommender-system`
[13] `https://www.w3.org/TR/vocab-dcat-2/`

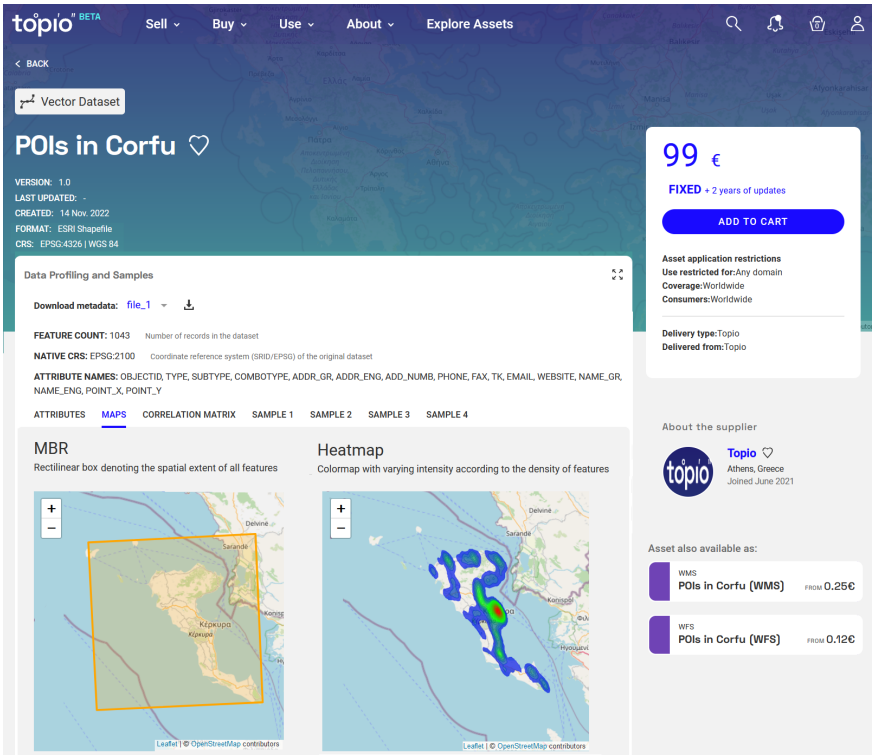


Figure 3.6: Detailed view of asset specifics, including version, format, and metadata obtained through the profiling service, featuring information on columns (i.e., features) and visual representation of the asset on maps.

interactions. Then, the recommender service applies knowledge graph embedding (KGE) models [37] to embed assets from the graph into a vector representation. Finally, the cosine function calculates the similarity among data assets in the graph.

The recommender service provides an REST API for integration into the marketplace. The core service operates by taking an asset identifier as input to generate recommendations. It uses an embedding model for this purpose, with current support extending to TransH [185], RotatE [177], and ComplExLiteral [110]. Additionally, the service is compatible with numerous other models found within the PyKEEN framework [4]. Users can specify the desired number of recommendations, with the default setting providing three. Based on these inputs, the recommender system initiates its processes and returns a JSON response containing the identifiers of the suggested assets.

As Topio accumulates more user feedback within the marketplace, including search history, views, and purchases, the recommender service will integrate this information into the knowledge graph. Including additional metadata will result in more robust embeddings of user interactions, leading to improved recommendation quality. Consequently, we aim to transition to a collaborative filtering algorithm, leveraging the enriched dataset to enhance the accuracy and relevance of the recommendations provided to users.

Table 3.1: Overview of the metadata computed by the data profiling value-added service, based on the asset type and application level (i.e., the asset as a whole, or specific features).

Type	Level	Metadata
Vector & Tabular	Dataset	Feature count
	Thematic attributes	Names, data types, cardinality, distribution, N-tiles, unique values, frequency, value pattern type, special data types, keywords per column numerical value patterns, numerical statistics, correlation among numerical attributes, equi-width histogram, date/time value distribution, geometry type distribution, attribute uniqueness, compliance to well-known schema
	Geometry	Native CRS, Spatial extent, convex/concave hull, heatmap, clusters, thumbnail generation
Raster	Dataset	Native CRS; Spatial extent
	Raster specific	Resolution; Width, height; COG
	Band related	Number of bands; Band statistics; Value distribution; Pixel (bit) depth; NoData Value(s)
Multi-dimensional	Dataset	Native CRS; Dimension count/info; Variable count/info
	Variable related	Spatial extent; Temporal range; Value distribution; NoData Value(s)

3.5.3 Data Asset Profiling

Providing comprehensive and precise metadata to prospective buyers for a given asset before a purchase has been one of the original aspirations to increase transparency and trust for `Topio`. The more actionable information prospective users have for a given asset, the more informed their decision can be and the more satisfied they will be with their purchase. Our assessment of the market landscape and competitors identified the lack of such automated metadata as a significant deficiency, validated by prospective users of our marketplace, as established from the user requirements elicitation (Section 2.3). These observations led us towards prioritising and strengthening automated metadata generation as a differentiator and unique selling point for the `Topio` marketplace platform.

Data profiling[14] comprises a collection of operations and processes for extracting metadata from a given dataset [147]. Such metadata may involve schema information, statistics, samples, or other informative data summaries, thus offering extensive and objective indicators for assessing datasets. This component can be internally invoked as part of the data publishing workflow or on demand when searching and browsing for datasets, as illustrated in Figure 3.6.

Geospatial datasets can be organised into various types, most commonly vector and tabular, raster, and multidimensional. Although some of the profiling metadata is common

[14] `https://github.com/OpertusMundi/profile`

among various data types (e.g., native CRS and spatial extent for spatial data), a different set of metadata is used in principle for each data type. Some metadata characterises the dataset as a whole (e.g., feature count for vector and tabular assets), while other metadata applies only to a specific data type feature. A summary of the metadata computed based on the asset type is listed in Table 3.1.

Metadata. To compute the data profiles and metadata, we created BigDataVoyant [138], which repurposes and extends various existing open-source software bundled together in a streamlined and scalable manner. Data profiling for each type of supported data type (i.e., vector, tabular, raster, multidimensional) is handled by a separate software component in the profiler, and precisely: (*i*) GeoVaex[15] (an extension of Vaex [20]) developed for out-of-memory processing of vector assets, (*ii*) GDAL/OGR for raster assets, and (*iii*) the netCDF Python module for multi-dimensional assets.

Sampling. In Topio, the profiling information describes a dataset by adding information that helps the users understand an asset better. Together with the profiling information, we show different data samples. For tabular data, we use various techniques such as random, stratified and cluster sampling. For geospatial data, we created a sampling algorithm that selects random samples within a given bounded box, functionality implemented within *BigDataVoyant*.

3.6 Preliminary Usability Evaluation

Topio Marketplace[16] is currently in beta version. Topio provides open access for unregistered users and offers a wide range of search possibilities to explore the asset catalogue. Unregistered users can search for assets using keywords, access the most popular searches, and reuse them. The platform supports functionalities such as creating an account, logging in, visualising the dashboard, etc.

Furthermore, we used the beta version to assess the data lifecycle in the platform, measure time spent on the publishing and purchasing processes, and evaluate our design decisions. At the moment, we are assessing the performance of each component (e.g., discovery service, recommender, profiler, etc.) using data gathered from the interactions of suppliers and consumers on our platform. However, this evaluation requires more users and specific experiments to be conducted. Until those experiments are completed, we have preliminarily investigated how much time is needed for publishing and purchasing assets. More specifically:

> **Data Asset Publishing** – Publishing an asset by a novice supplier (i.e., supplier with less than two assets published) takes, on average, three minutes from process start to submission for review. We do not account for the time to upload an asset, which depends on the asset size. Publishing an asset by an experienced supplier (i.e., a supplier with more than five assets published) takes an average of 25 seconds. Most suppliers opted to add optional metadata in the publishing wizard, which is a positive outcome as suppliers understand that the more metadata is available, the easier it is for users to discover and purchase their assets.

[15] https://github.com/OpertusMundi/geovaex
[16] https://beta.topio.market/

Service Publishing – A supplier with an existing published asset spends, on average, five minutes to create an OGC service operationalised by Topio. Most of this time is allocated to deciding the pricing of the created asset rather than completing the wizard. This is a surprising insight as we did not observe it for data publishing; suppliers generally know the price they want to set well in advance. However, operationalising their data represents a new market activity, and more consideration is needed to allocate the price point.

Data Asset Purchasing – The average time required for a prospective client to complete an asset purchase from visiting the cart until asset delivery is 12 seconds. This is an expected duration based on the assumption that purchasing data assets does not differ from a standard e-shop.

We have successfully showcased the platform across three distinct scenarios to a diverse audience, demonstrating its versatility and applicability: (*i*) how to use Topio marketplace platform to search for a wide variety of geospatial assets and discover related and unexplored assets, (*ii*) how to link and augment an asset to the purchased asset collection, and (*iii*) how to inspect an asset, understand its fit and purpose, and discover more related assets while working in Topio Notebook.

Scenario #1: Asset search and discovery – Suppose that we want to explore the assets currently available in Topio Marketplace. As users, we can explore all the assets from the catalogue and perform an advanced search based on the metadata provided during ingestion. With the advanced search, the users can filter the assets based on the type (e.g. vector, raster, tabular), coverage, which allows the user to draw an area of interest on a map, price range, topic (e.g. farming, health, oceans), last update date, format (e.g. CSV, KML, WMS), scale and more. Once the users find an asset during the search and exploration phase, they can inspect its page as illustrated in Figure 3.3. Here, Topio helps the users discover even more assets related to the current one. The discovery service finds connections for every asset from the marketplace to find other connected and related assets. This process helps the user expand the search towards unknown areas.

Scenario #2: Augmenting purchased assets – Suppose that the user found an adequate asset, purchased it and then explored the asset catalogue again. Topio supports the users in the new exploratory journey by showing how an asset from the catalogue can be connected to the purchased assets. Moreover, two presumably disjoint assets can be joined together through transitive joins. The users can visualise the join paths and how the disjoint assets can be connected and augmented, as illustrated in Figure 3.5.

Scenario #3: Analyse and discover assets with Topio Notebooks – Suppose that the users found an attractive asset but are still uncertain about its value. Once the users log in, Topio enables more advanced metadata computed by the profiling component (Section 3.5.3). This information helps the users inspect statistics about the assets, look at different samples of the data, and ultimately make an informed decision towards purchasing. To support the users even further towards assessing

an asset value, Topio provides a notebook environment for analytics and discovery (Figure 3.2). With Topio Notebooks, the users can assess the quality and fitness of an asset before purchasing. Inside this environment, the users benefit from the discovery service (Section 3.5.2) and can directly inspect the list of related assets. This functionality helps them discover more assets without checking the marketplace's asset page. Therefore, the users can only focus on processing, analysing and understanding an asset's usefulness without interruptions.

3.7 Conclusion

With Topio, our goal is to establish a solid groundwork for future open data marketplaces. The various components of the platform are freely accessible, marking a significant step towards open web engineering and development. We have crafted flexible and automated systems for managing the entire lifecycle of geospatial asset trading. However, it is worth noting that these components have the potential to be adapted for use beyond merely spatial data, offering a versatile foundation for the expansion of data trading capabilities.

In this chapter, we have presented Topio, an open-source data market platform instance, and we answer our first research question:

***RQ1**: How can dataset discovery approaches enable and facilitate data acquisition in data marketplace platforms?*

Our research has revealed the multifaceted role that dataset discovery plays in enhancing the dataset acquisition experience in a data marketplace such as Topio. Dataset discovery methods are a cornerstone for enabling users to navigate vast amounts of data. They offer a systematic approach to uncovering relevant datasets that meet specific requirements. This process simplifies the search for suitable data and fosters a deeper engagement with the data available on the platform. The implementation of dataset discovery in Topio has shown significant benefits for data acquisition, providing users with several pathways to interact with and assess the potential utility of datasets. We accomplished this in two ways: (*i*) by discovering related assets on the asset view page and (*ii*) by facilitating an active exploration of data in an interactive setting, such as Topio Notebooks. Within these notebooks, users can experiment with datasets directly, applying them to real-world scenarios to understand their relevance and quality.

By incorporating dataset discovery across both passive user experiences (i.e., where users receive suggestions without active querying) and interactive user experiences (i.e., allowing users to engage directly with data through analysis and visualization in Topio Notebooks), Topio significantly improves data accessibility. This approach ensures that the users can effortlessly discover relevant datasets, whether casually browsing or deeply analysing data, therefore **enabling** data acquisition. Moreover, introducing dataset discovery in Topio not only aids users in finding the data they need more efficiently but also encourages the exploration of new datasets that might not have been initially considered, thus **facilitating** the data acquisition process.

In the next chapter, we extend the discovery and augmentation service to automatically discover relevant features (i.e., columns) for a base table, increasing the accuracy of an ML model. We place our approach in a data lake scenario, where various datasets have

been previously acquired via multiple sources, including but not limited to data market platforms.

3.7.1 Future work

Future work for the market platform includes and is not limited to (*i*) exploring diverse pricing algorithms to assist suppliers who are uncertain about pricing their assets, (*ii*) improving the user interface within `Topio` Notebooks by facilitating seamless access to data samples from the platform, and (*iii*) enabling consumers to uncover related assets and opportunities for augmenting existing marketplace assets with their proprietary data, which can be uploaded as needed.

Furthermore, engagements with data providers have led to the understanding that commercial geodata products are regularly updated and offered at set intervals, enabling new avenues for research and development, such as metadata versioning and provenance tracking. Additionally, providers have expressed the utility of using `Topio` for automating the offering and sale of small regional data segments (e.g., socio-demographic data for a selection of municipalities in Germany). Traditionally, managing small regional datasets involves a considerable manual effort in preparation, delivery, and billing, yielding minimal returns for suppliers. Hence, `Topio` stands out as particularly advantageous for vendors in these scenarios, simultaneously offering cost benefits to consumers due to reduced expenses associated with data extracts.

II

Automated Feature Discovery for Tabular Data

4

AutoFeat: Transitive Feature Discovery over Join Paths

Increasing the accuracy of ML models by automatically discovering relevant features within a vast data lake poses a significant challenge. This chapter introduces a novel feature discovery approach, `AutoFeat`*, designed to address this challenge.* `AutoFeat` *meticulously navigates through multi-hop, transitive join paths to discover relevant features, which are then augmented to the training dataset, amplifying the accuracy of the ML model. This novel method streamlines the feature discovery process and significantly advances the potential for superior model performance. The code to reproduce this chapter is available at* `https://github.com/delftdata/autofeat`

This chapter is based on the following workshop paper, research paper, and open-source resources:

- *Andra Ionescu, Rihan Hai, Marios Fragkoulis, and Asterios Katsifodimos. "Join Path-Based Data Augmentation for Decision Trees". ICDEW 2022* [87]
- *Andra Ionescu, Kiril Vasilev, Florena Buse, Rihan Hai, and Asterios Katsifodimos. "AutoFeat: Transitive Feature Discovery over Join Paths". ICDE 2024* [93]
- *Source-code* [83] *and data* [84]

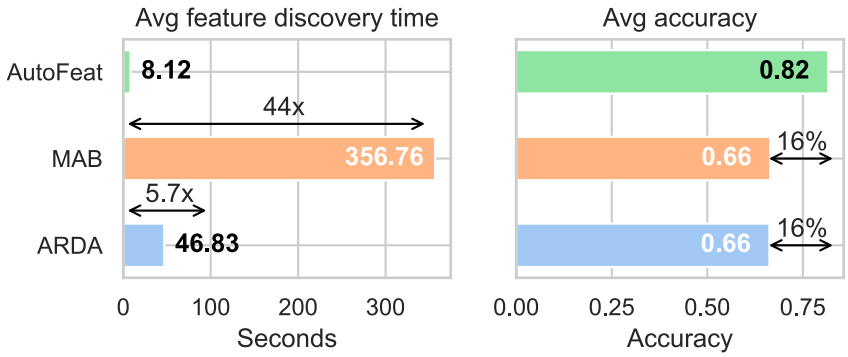


Figure 4.1: AutoFeat outperforms the state-of-the-art data augmentation frameworks regarding feature discovery/augmentation time, as it is faster than any approach, and the resulting augmented table shows an increase in accuracy when used for ML tasks.

4.1 Introduction

Machine learning is widely used in various domains, such as retail, medical diagnosis, and transportation. An ML model's performance (e.g., accuracy) heavily depends on its training data [56]. Although it is a common assumption that the input data of a model is a single table, in practice, the situation can be more complex. Features with high predictive power may reside in different database tables or multiple files in an open data repository, data lake, or even separate files acquired from a data market platform.

Recent works [36, 129] focus on augmenting a base table with features, using join paths to drive their search. However, they first search for data, then join, and then apply feature selection to prune out noisy or irrelevant features based on ML model performance. Given data lakes where primary-key/foreign-key (PK-FK) constraints are missing, it is necessary to use dataset discovery algorithms as a first step of data augmentation to find relationships between tables. However, this process is known to output spurious connections, where what might be considered a join produces an irrelevant table with unrelated data. This becomes an even bigger problem when two datasets can be joined via multiple join columns. In this context, state-of-the-art augmentation processes fail. This leads us to our second research question:

> ***RQ2**: How can we enhance the automation of the feature discovery process to create high-quality datasets for machine learning applications?*

To answer our research question, in this chapter, we introduce *transitive feature discovery*, which aims to discover and augment relevant features over join paths without using ML models in the process. The main idea behind our approach is to explore the space of joinable tables and to prune out the low-quality join paths based on data quality measures, as well as relevance and redundancy metrics. As shown in Figure 4.1, our approach (AutoFeat) outperforms the competition both in effectiveness and efficiency. AutoFeat's effectiveness benefits come from its ability to explore join paths beyond the star schemata supported by [36], managing to find join paths that contain features of high predictive

power. In addition, by simply ranking join paths using relevance and redundancy metrics, instead of training expensive ML models during search [36, 129], AutoFeat is not only able to explore a larger space of transitive joins paths, but also be 5x-44x more efficient than the competition.

Goal. The goal of feature discovery is to enrich an original base table with new features with high predictive power for a target ML model on a classification task. AutoFeat automates the process of identifying and joining relevant tables from a dataset collection to a base table, thus creating an augmented table. AutoFeat applies heuristic-based feature selection techniques on the augmented table to prune out any noisy or irrelevant features. It performs these tasks effectively and efficiently, reducing the need for manual data engineering efforts and improving the performance (i.e., accuracy) of the subsequent ML models. In this chapter, we contribute a novel feature discovery method which has the following properties:

> ***Versatility***: AutoFeat can explore join paths beyond star schemata to augment a given base table. In addition, AutoFeat's join path ranking mechanism does not depend on training the target ML model.
>
> ***Efficiency***: instead of repeatedly training the underlying model to assess the accuracy benefits of a given join path, AutoFeat: *i*) prunes out non-promising join paths using data quality metrics, and *ii*) proposes a ranking function that chooses the top-k most promising join paths according to information theoretical metrics that encode feature relevance and redundancy. The result is 5x-44x faster feature augmentation.
>
> ***Effectiveness***: while being faster, AutoFeat's feature augmentation strategy achieves on average 16% higher accuracy compared to the state-of-the-art methods on real-world, open data repositories.

4.2 Related Work

In this section, we examine the existing approaches spanning key domains intersecting feature discovery, including dataset discovery, dataset augmentation, transitive joins, and feature selection and underline the unique features of our approach.

Dataset Discovery. Dataset discovery helps users explore a vast collection of heterogeneous datasets and find related tables to perform a data-driven task [56, 70]. A large corpus of dataset discovery works focus on finding unionable tables [146], joinable tables [26, 50, 58, 206], while some tackle both relatedness scenarios [17, 202]. We consider feature discovery a more tailored and specific dataset discovery process, focusing exclusively on identifying tables containing features suitable for augmenting a base table with more *relevant* information.

Dataset Augmentation. Dataset augmentation has been studied from two angles: when KFK constraints are known and when these are discovered using dataset discovery approaches [24, 69, 112, 129, 191]. Dataset discovery approaches use joinability graphs to model the datasets and the relations between them, limiting the length of the join paths

Table 4.1: Comparative analysis of AutoFeat against state-of-the-art approaches across three key dimensions: join path length, path/feature selection, and joinability graph.

	Join path Length	Path / Feature Selection	Joinability Graph
ARDA	Single-hop	Model-execution based	Simple Graph
MAB	Multi-hop	Model-execution based	Simple Graph
AutoFeat	Multi-hop	Ranking-based	Multigraph

[36, 58]. These works mainly focus on augmenting directly joinable tables and rely on machine learning models for feature selection [36, 62]. In short, as seen in Table 4.1, ARDA supports single-hop paths (star schemata), while MAB and AutoFeat support multi-hop join paths. Both ARDA and MAB require an expensive model execution step to evaluate the quality of a join path, while AutoFeat ranks paths according to cheaper metrics (Section 4.5). Finally, for each pair of tables, instead of supporting a single join possibility, AutoFeat supports multiple ones (joinability *multigraph*).

Transitive Joins. Transitive joins have been proven to be effective for augmentation in the context of notebooks [202, 203], or as an augmentation strategy for tuples or missing values [191]. AutoFeat uses transitive joins to navigate the join space and explore candidates for *feature* augmentation using multi-hop join paths.

Feature Selection. Feature selection methods are categorised based on selection strategies into filter, wrapper, and embedded types [78, 119]. Filter methods are independent of the ML model, wrappers assess feature quality based on learner performance, and embedded methods integrate feature selection into the training process. While traditional feature selection assumes training data is stored in a single table, our feature discovery approach addresses challenges from data spread across multiple tables.

4.3 Transitive feature discovery

In this section, we provide an overview of the foundational concepts that underpin this paper, such as relevance and redundancy in the context of machine learning; then, we define the problem of transitive feature discovery and provide an overview of our approach.

4.3.1 Preliminaries

With transitive feature discovery, we aim to discriminate between relevant and redundant features and reduce useless information [139]. Given a dataset T_i which comprises a collection of features $X_1, X_2, ..., X_n$, and a feature containing the labels Y, and given $S_i = \{X_1, ..., X_{i-i}, X_{i+1}, ..., X_m\}$ a collection of features without X_i, we define the concepts of relevance and redundancy as follows.

Relevance. The relevance of a feature has multiple definitions based on the objective, and it has been shown that a general definition of relevance is not universally applicable [105]. Therefore, in feature selection, we must distinguish between *strong relevance* and *weak relevance* [78, 105, 196].

Strong relevance of a feature X_i means that removing the feature results in the degra-

dation of the optimal feature subset. Contrary to strong relevance, weak relevance of a feature X_i implies that the feature is not always necessary (i.e., not relevant), but the performance of a learner on a subset of features S_i is worse than on a subset of features $S_i \cup \{X_i\}$. Finally, the relevant features influence the output, and their role is unique (e.g., no redundancy), while irrelevant features do not affect the output [139, 196].

Redundancy. Feature redundancy is tightly coupled to feature dependency or feature correlation, meaning that perfectly correlated features are redundant to each other because they do not introduce any new information [78]. The redundant features are those which can take the role of another feature [139]. The most straightforward intuition is that a redundant feature is a duplicate of a relevant feature. It is worth mentioning that feature redundancy is not absolute, as it is conditioned on a feature subset. Changing the feature subset leads to changing the decision of whether a feature is redundant or not. Finally, feature redundancy aims to identify the redundant features and *remove* them [78].

Summary & Goal. The goal of this work is to maximise relevance and minimise redundancy to find the features which are highly correlated with the label (i.e., relevant) and are not yet represented by any other feature from the selected features subset (i.e., non-redundant).

4.3.2 Problem Definition

We introduce *transitive feature discovery*, a process at the intersection of dataset discovery, dataset augmentation and feature selection. We leverage the exploration step and joinability scores from dataset discovery to augment a given table. The (partially) augmented result undergoes a feature selection step, where we select only the features that increase the information value (i.e., relevant and non-redundant). Formally, we define transitive feature discovery as follows:

Definition 4.3.1 (*Transitive feature discovery*) *Given a base table T_i with a label column Y and a collection of datasets D with or without Key-Foreign Key (KFK) relations, transitive feature discovery extends the base table with more relevant features X_i with the aim of solving a task M.*

Input. We take as an input (i) a table T_0 comprising of n features $\{X_1, X_2, ..., X_n\}$, and a feature Y with the labels, which we further name the *base table*, and (ii) a collection of datasets $\mathbf{T} = \langle T_1, \ldots, T_n \rangle$.

Setting. The base table is located in a data lake, surrounded by a collection of datasets, where we distinguish two scenarios: (i) the relationships between the base table and the other datasets are undiscovered and unknown, or (ii) the base table has known KFK relationships with other datasets.

Output. Our approach outputs a ranked list of top-k join paths. Each join path contains the datasets for augmentation with their respective join keys and a list of selected features, representing the optimal subset of features in the join path, leading to increased performance of an ML classification task.

Example. Take as an example the collection of datasets from Figure 4.2. Transitive feature discovery aims to enrich the base table *Applicants* with new features that have

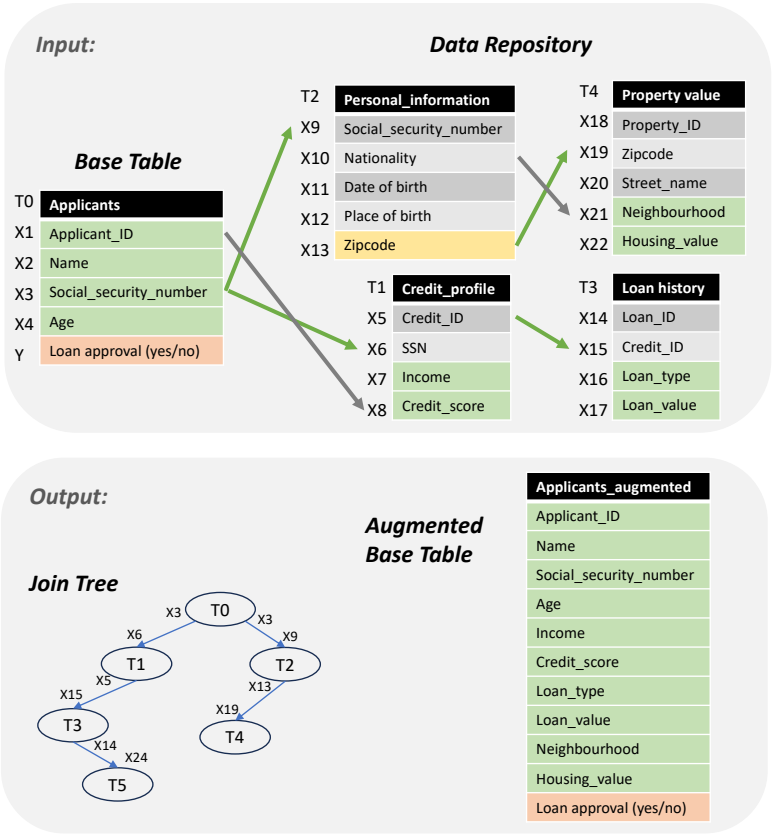


Figure 4.2: Running example highlighting the input and output of AutoFeat. The input consists of (1) the base table *Applicants*, which contains the label *Loan approval*, and (2) the data repository. The green-coloured features have high predictive power, while the yellow-coloured feature is the join column used to reach the transitive table *Property value*. The green arrows represent the paths which contain the relevant features. The output is abstracted on the bottom side of the figure and consists of the join tree and the augmented table.

high predictive power for the target task of predicting bank loan eligibility (*Loan approval*). The candidates for augmentation are the following datasets: *Personal_information*, *Credit_profile*, *Property value*, and *Loan history*. This example contains relationships discovered by dataset discovery algorithms, which can be spurious, such as the relation $Applicants.Applicants\_ID \longrightarrow Credit\_profile.Credit\_score$. The output shows the result of the best join path, which is depicted as a join tree with its corresponding join keys. It also shows the selected subset of features and the final augmented table.

4.3.3 Approach Overview

Our approach consists of an offline component: the joinability graph construction. This phase transforms the base table and the dataset collection into a graph structure, a process we detail in Section 4.4.

The online component is the entire augmentation pipeline: we traverse the graph, begin the discovery process, and the feature selection phase. During graph traversal, outlined in Section 4.4.1, we execute the following processes: (*i*) the identification of joinable tables, (*ii*) and pruning paths, (*iii*) the identification of relevant features within a join path, and finally (*iv*) the decision on which features should be pruned based on redundancy. We further elaborate on our design choices concerning the join type in Section 4.4.2, and the strategies for table pruning in Section 4.4.3. Our decision-making process for feature selection follows an empirical approach, presented in detail in Section 4.5.

`AutoFeat` traverses the joinability graph in search of relevant features, then ranks the join paths and produces a list of the top-k paths along with their corresponding features. The mechanisms and design decisions integral to the ranking process are provided in Section 4.6. Finally, the top-k join paths are used to train k ML models, and the best join path (i.e., which produces the most accurate result) is returned.

4.4 Dataset relation graph

This section delves into the core concepts and techniques behind the offline component. We make the general assumption that we have a collection of various input datasets coming from relational databases or data lakes. In a relational database, the metadata, such as the schemata and PK-FK constraints, are often defined. In data lakes, we assume the relationships between the datasets can be detected using dataset discovery tools [14, 17, 24, 26, 50, 58, 106, 202, 206].

We create the joinability graph *Dataset Relation Graph* (DRG). DRG is a weighted graph where the nodes represent the datasets, while the edges represent the relations between these datasets. We use DRG (*i*) to capture the relationships between datasets, (*ii*) to traverse the graph following transitive joins, and (*iii*) to be able to enumerate multi-hop join paths, which we assume they contain valuable and relevant information for the base table. Before we explain how the DRG is constructed, we introduce a few concepts which will be used throughout the rest of the paper.

Definition 4.4.1 (*Joinability and join column*) *Given two datasets T_i and T_j, and their attributes X_i^i and X_j^i, where the superscript denotes the attribute i, while the subscript denotes the originating dataset, T_i is* joinable *with T_j if* (1) *there is a primary-key/foreign-key relationship between T_i and T_j, i.e., $T_i.X_i^i \longrightarrow T_j.X_j^i$ where X_i^i is the foreign key and X_j^i is the corresponding primary key, or* (2) *X_i^i and X_j^i are related attributes (i.e., their intersection is non-empty). We refer to X_i^i and X_j^i as join columns.*

Definition 4.4.2 (*Join graph and path*) *Consider a set of datasets $\mathbf{T} = \langle T_1, \ldots, T_n \rangle$. Its join graph $G_\mathbf{T} = (V, E)$ is an undirected graph with nodes V and edges E. Each dataset $T_i \in \mathbf{T}$ is represented as a node $v_i \in V$. If two datasets T_i and T_j are joinable, there is an edge $e_{ij} \in E$ between the nodes v_i and v_j. In a join graph, a join path is a finite sequence of edges that connects a sequence of distinct nodes.*

Definition 4.4.3 (*Dataset Relation Graph*) *Consider a set of datasets $\mathbf{T} = \langle T_1, \ldots, T_n \rangle$. Its* dataset relation graph *is an undirected multi-graph $G_\mathbf{T} = (V, E)$, where each node $v_i \in V$ represents a dataset $T_i \in \mathbf{T}$, and the set of edges between two nodes v_i and v_j is a multiset*

E_{ij}. The multiset E_{ij} contains all the edges between v_i and v_j, representing multiple join opportunities given different join columns.

The DRG construction follows the next steps. First, we iterate through all the datasets and create the respective nodes in the graph. Second, if the datasets contain information about the integrity constraints, we ingest these constraints as edges with $weight = 1$. Finally, when the relations between tables are unknown, we employ a dataset discovery method: our current prototype uses COMA [48] for schema matching, according to Valentine [106]. The new relationships are modelled as edges with $weight = similarity\_score$, where the $similarity\_score$ is the score returned by COMA. DRG construction is independent of the dataset discovery algorithm; thus, any algorithm that outputs a similarity score can be used to model the relationships across datasets.

Join Path Enumeration. We model the DRG as a multi-graph to capture the set of possible join columns between two tables. Given the dataset relation graph $G_{\mathbf{T}}$ and the base table T_0, it is straightforward to enumerate all possible paths starting from the node representing T_0. A path in DRG is a directed join path of minimum $length = 1$.

Definition 4.4.4 (*Join path search space*) *The join path search space J_i is all the acyclic paths in $G_{\mathbf{T}}$ that start from T_i and have $length \geq 1$.*

In our approach, we navigate the join path space following transitive joins, thus creating longer join paths. We consider a different join path for every edge in the multi-graph and every $n-hop$ traversal, which is a concatenation of paths. In Figure 4.2, the following is a join path with $length = 1$:

$$Applicants.Applicant\_ID \longrightarrow Credit\_profile.Credit\_score,$$

while the following is a different join path of $length = 2$:

$$Applicants.Applicant\_ID \longrightarrow Credit\_profile.Credit\_score \longrightarrow Loan\_history.Credit\_ID$$

The DRG plays a crucial role in our methodology, serving as the backbone for all subsequent steps, including graph traversal, join operations, and path pruning, which we will elaborate on next.

4.4.1 Graph Traversal

Given a base table T_0, and its corresponding node v_0 in DRG, we set the stage for graph traversal. Among various graph traversal techniques, Depth First Search (DFS) and Breadth First Search (BFS) become particularly pertinent to our context [22, 178]. These methods require only the source node to traverse the graph, an approach that aligns with our setting: the feature discovery process starts with the base table. Exploring a join graph using either BFS or DFS can lead to a greedy approach, as both algorithms aim to find all join paths. However, AutoFeat uses BFS to traverse the join graph for feature augmentation for the following reasons.

BFS explores the join graph one level at a time, allowing us to evaluate the data quality after each level of join operations. This early determination of data quality enables us to optimise the data augmentation process and potentially avoid wasting computational

resources on irrelevant join paths. While DFS might find a feasible join path quickly, it may not necessarily consider the most relevant datasets early in the process. This can lead to a decrease in data quality as it explores longer, less relevant paths before discovering the more informative ones. Moreover, errors from one join can propagate deeper into the join path, affecting the quality of the results even more. BFS makes errors more straightforward to manage and contain, as the exploration is performed level by level.

4.4.2 Join

Continuing this process, `AutoFeat` only considers *left* joins: we perform a *left* join between the base table T_0 and any other tables T_j under the assumption that $\forall T_j \in \mathbf{T}, Y \notin T_j$, i.e., any other table from the dataset does not contain a feature column with the class labels. For transitive joins, we treat the intermediate join result as a base table and perform a *left* join with the following table along the path, etc.

The *left* join is chosen primarily to maintain the number of tuples and, more critically, the number and distribution of classes in the label Y, which aligns with prior data augmentation approaches [36]. Using a different type of join could either remove or duplicate rows, both of which skew the class distribution and introduce class imbalance [71]. If not handled properly, such imbalance could alter the ML task or degrade performance [39, 65, 120].

Join Cardinality. To ensure the base table size remains constant (i.e., neither shrinks due to row removal nor expands due to row duplication) even when a *left* join is used, we transform one-to-many and many-to-many joins, thereby preventing data duplication and inconsistencies in labels. We group by the join column and randomly select a row [36]. Given that the DRG is a multi-graph, we apply this strategy for every possible join column due to its direct impact on the subsequent joins.

4.4.3 Pruning Paths

To further refine our approach and improve efficiency, we must consider the complexities involved with the join path search space. Working directly with the raw output of a dataset discovery method results in a significantly expanded search space, demanding the application of robust pruning strategies to manage this increased complexity effectively.

Similarity Score-Based Pruning. There may be instances where multiple possible join columns exist between two nodes as a result of a dataset discovery algorithm. In such cases, we explore each potential join using every respective join column. Preliminary experiments, however, suggest that a significant portion of these join results contain *null* values across their entire right-hand side. This result is far from ideal as an input for any ML algorithm. Thus, we implement our first pruning strategy at the join column level. Using the similarity score, we can prune weaker join columns. Given a base table, a joinable table, and a set of join columns, `AutoFeat` selects the join column with the highest similarity score. When multiple join columns share the same top score, each join from T_i to T_j using the join column X_j^i is an individual join path.

Data Quality-Based Pruning. The resulting augmented table may still suffer from poor quality, even when using the join column with the highest similarity score. A critical dimension of data quality is *completeness* [169], which can be gauged by the amount of

null values present in a table. Several strategies can enhance table completeness: deletion, which is unsuitable in our context as it involves removing tuples; and imputation, which involves replacing *null* values with mean value, median value, most recurrent value, or a default value [65]. However, these artificially imputed values may skew the data distribution and introduce bias [65]. In light of the drawbacks associated with imputation or deletion, our goal is to augment the base table with datasets that result in a table with the highest possible completeness. As such, with our second pruning strategy based on completeness, we measure the *null* value ratio in the resulting join and prune the joins where this ratio falls below a predefined threshold τ. This threshold is incorporated into our approach as a hyper-parameter, and we demonstrate the effects of tuning it in our experiments (Section 4.7).

4

4.5 Feature Selection Strategies

In this section, we introduce streaming feature selection, explore a variety of relevance and redundancy metrics, and assess their performance. We use this empirical evaluation to drive our design decisions towards the best-performing methods and ensure our effectiveness and efficiency.

4.5.1 Streaming Feature Selection

Streaming feature selection assumes a constant number of rows. In contrast, the features arrive in a streaming fashion, one at a time, or in groups, with the goal to determine the subset of relevant features at a given time [5, 77, 118, 119, 194]. Each new feature batch is derived from a join operation in relational data. Moreover, transitive join paths involve an implicit dependency. Each transitive join relies on the intermediate join and the features representing the join columns. Ensuring the persistence of these join column features is essential, and the feature selection algorithms must not eliminate them. Therefore, we must refrain from pruning intermediate joins, even if they contain irrelevant and redundant features, as they establish the pathway towards multi-hop join paths. `AutoFeat` builds upon the pipeline of streaming feature selection using high-performance strategies for relevance and redundancy.

4.5.2 Empirical Evaluation Setting

Before we delve deeper into the empirical analysis of the methods for relevance and redundancy, we describe the experimental setup.

Datasets. To conduct a robust empirical analysis of the feature selection methods further considered in our study, we select six binary classification datasets that vary in terms of (*i*) domain (medicine, web data, pattern recognition), (*ii*) ratio of rows to columns, and (*iii*) number of numerical and categorical columns. Regarding (*iii*), the chosen datasets encompass discrete, continuous, nominal and ordinal features, ensuring that the selected data represents the full spectrum of feature types encountered in ML scenarios. A general overview of the datasets, ordered ascendingly by the number of features, can be visualised in Table 4.2. The selected datasets are open-source, originating from three widely employed ML repositories: OpenML, Kaggle and UC Irvine.

Machine Learning Models. Preliminary results have shown that the behaviour of fea-

Table 4.2: Overview of datasets used for the empirical analysis of the methods for relevance and redundancy.

Dataset	Source	# rows (excluding target)	# features ↓
breast cancer	Kaggle	569	31
spam e-mail	OpenML	4601	57
musk	OpenML	6598	169
arrhythmia	OpenML	452	279
internet advertisements	UC Irvine	3279	1558
gisette	UC Irvine	6000	5000

ture selection techniques is invariant to the choice of ML algorithm. Hence, we perform our analysis on LightGBM due to its ability to handle high-dimensional data, robustness against overfitting and ability to capture complex data relationships [102]. This is deployed using AutoGluon [52], an AutoML framework designed for tabular data that automatically handles data encoding and hyperparameter tuning.

Metrics. Our assessment of ML models for classification tasks is twofold (*i*) effectiveness-based, where we measure the *accuracy*, and (*ii*) efficiency-based, where we measure the aggregated amount of *feature selection time* and *algorithm training time*.

Methodology. We devise an ML pipeline that consists of three stages: (*i*) *data preprocessing*, (*ii*) *feature selection* and *algorithm training*, and (*iii*) *model evaluation*. In stage (*i*), we address the issue of missing values in the features by imputation of the most common value corresponding to that column. We note that, from our selected datasets, only Arrhythmia contains missing values, and this imputation method has been found to yield the best performance for the LightGBM algorithm.

We split the dataset into an 80% training set and a 20% test set, employed in stages (*ii*) and (*iii*), respectively. Part of stage (*ii*), feature selection retains $top - \kappa$ best-performing features from the total set of features [67], where the choice of values for κ varies based on the dimensionality of the dataset. After feature selection, we train the ML algorithm on the selected feature subset. Lastly, in stage (*iii*), the model is evaluated on the test set regarding the mentioned metrics.

4.5.3 Relevance Metrics

In feature selection, relevance is measured using heuristics. One of the most popular heuristics is correlation, which is based on the hypothesis that good features correlate with the label. We analyse two information-theoretic based methods: information gain, symmetrical uncertainty [5], two widely used correlation coefficients: Pearson, Spearman [44], and Relief, primarily used to remove irrelevant features [51].

Information Gain (IG). Information gain helps select features highly correlated with the label. The method assumes that if a feature has a strong correlation with the label, that feature will positively impact the performance of an ML model [5]. Information gain is symmetric (e.g., $I(X;Y) = I(Y;X)$), and it equals zero if two variables X and Y are independent [119].

Symmetrical Uncertainty (SU). Symmetrical uncertainty is a correlation metric which

measures the linear or non-linear association between two features [188, 195]. SU is based on information gain, and it is able to compensate for the bias towards features with multiple values, one of the shortcomings of IG. SU returns a normalised score in the $[0, 1]$ interval. A score close to zero indicates that the features are independent and, therefore, not relevant for the classification, while a score close to one indicates dependency and, thus, relevancy.

Pearson. Pearson correlation is widely used to assess if two variables are linearly related, independent of any non-linearity that exists in the distribution of the variables. To compute the Pearson correlation between two variables X_i and Y, we must know the covariance and variance of the variables [67].

Spearman. Spearman correlation is a non-parametric measure of rank correlation, which is the statistical dependence between the rankings of two variables. While Pearson's correlation assesses linear relationships, Spearman's correlation assesses monotonic relationships. The distinction between Pearson and Spearman is that Spearman involves transforming the sample values to ranks in the range $[1, N]$. If no repeated data values exist, a perfect correlation of +1 or −1 occurs when each of the variables is a perfect monotone function of the other. The higher the value, the higher the correlation [44].

Relief. Relief feature scoring is based on identifying feature value differences between nearest neighbour instance pairs. In other words, the method focuses on separating the data instances from different classes [179].

Choosing a Relevance Metric. Figure 4.3 shows that Pearson and Spearman are approximately 3x faster than the SU and IG from information theory due to the calculation simplicity of the first two methods. They also outperform SU and IG in accuracy by around 0.5. While Relief shows comparable efficiency to Pearson and Spearman, its effectiveness is notably lower. Spearman consistently performs best for all datasets, considering both effectiveness and efficiency. Consequently, Spearman is our recommended choice for handling high-dimensional datasets due to its efficiency and effectiveness across varying dataset characteristics.

4.5.4 Redundancy Metrics

In feature selection, redundancy is used to assess how much adding a new feature can benefit the performance of a model. Redundant features may have a high degree of overlap in the information they convey: adding a feature X_i, similar to feature X_j, is redundant, as it does not add any new information. We will assess information-theoretical methods based on Shannon's information terms [174].

Definition 4.5.1 *(Shannon's information terms) Given S a set of currently selected features, Y the feature with the class labels and $X_j \in S$ a feature from the current set S, the unified conditional likelihood maximisation feature selection framework is defined as a linear combination of Shannon's information terms [119]:*

$$J(X_k) = I(X_k; Y) - \beta \sum_{X_j \in S} I(X_j; X_k) + \lambda \sum_{X_j \in S} I(X_j; X_k | Y), \tag{4.1}$$

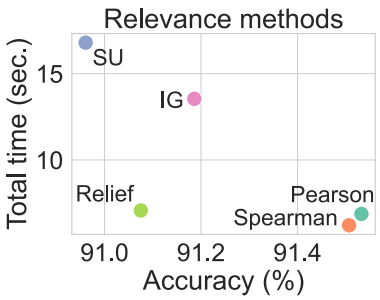


Figure 4.3: Comparison of relevance analysis methods in terms of aggregated accuracy and computational efficiency.

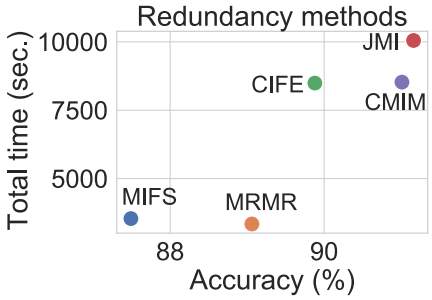


Figure 4.4: Comparison of redundancy analysis methods in terms of aggregated accuracy and computational efficiency.

4

where $I(X_k;Y)$ represents the information gain, while $I(X_j;X_k|Y)$ represents the conditional information gain.

We will use five methods to measure redundancy, all based on Equation (4.1).

Mutual Information Feature Selection (MIFS). MIFS helps select the features which are highly correlated with the label and un-correlated with each other [12]. As the name implies, the method uses mutual information to compute the scores and penalises the features correlated with already selected features. The method considers both feature relevancy and redundancy, and it sets the $\lambda = 0$ in Equation (4.1) [119]. In our analysis, we relied on $\beta = 0.5$.

Minimum Redundancy Maximum Relevance (MRMR). MRMR assumes that with more selected features, the effect of redundancy is gradually reduced. The method sets $\lambda = 0$ and $\beta = \frac{1}{|S|}$ in Equation (4.1), where $|S|$ is the size of the set of selected features [119].

Conditional Infomax Feature Extraction (CIFE). CIFE is a method that computes the feature-label and feature-feature correlations and the redundancy between the un-selected and already selected features. The method sets $\beta = 1$ and $\lambda = 1$ in Equation (4.1) [119].

Joint Mutual Information (JMI). JMI aims to increase the complementarity between un-selected and selected features and sets $\lambda = \frac{1}{|S|}$ and $\beta = \frac{1}{|S|}$ in Equation (4.1), where $|S|$ is the size of the set of selected features [119].

Conditional Mutual Information Maximization (CMIM). CMIM aims to select features with a strong predictive power while reducing the redundancy between the new and selected features. The method is a special case of Equation (4.1) as follows [119]:

$$J(X_k) = I(X_k;Y) - \max_{X_j \in S}[I(X_j;X_k) - I(X_j;X_k|Y)] \tag{4.2}$$

Choosing a Redundancy Metric. In Figure 4.4, we compare the five information theory methods for feature selection, and we find that MIFS and MRMR are nearly 3x faster than CIFE, JMI, and CMIM. This is due to their computation-saving characteristic of not needing to estimate conditional information gain. However, considering accuracy, CIFE,

JMI, and CMIM outperform as they effectively capture inter-column relationships and select less redundant features. Specifically, JMI achieves the highest accuracy but at the cost of the longest runtime, whereas MIFS offers 4x faster runtime but is 6% less accurate. Considering both runtime and accuracy, MRMR emerges as a balanced choice, offering considerably faster runtime while achieving high model accuracy.

Based on these findings, we devise our algorithm for transitive feature discovery. Spearman rank correlation is employed to measure the relevance, while Maximum Relevance Minimum Redundancy (MRMR) is utilised to quantify redundancy.

4.6 AutoFeat's: Ranking-based Feature Discovery

We introduce our ranking algorithm, built upon the foundations of a streaming feature selection pipeline [5], and adapted to suit our specific needs.

Streaming Feature Selection Pipeline. We follow the typical streaming feature selection pipeline, starting with a feature subset's relevance analysis. Subsequently, the resulting subset of relevant features undergoes the redundancy analysis. The final subset contains relevant and non-redundant features, while all others are discarded [5, 119]. As such, we populate the set of selected features with those from the base table T_0. Every join candidate T_i that was not pruned proceeds through the relevance analysis, followed by the redundancy analysis of the subset of relevant features. Algorithm 1 presents our strategy for transitive feature discovery over join paths. The process initiates with a queue containing the base table T_0 and its feature set, which forms the selected feature set R_{sel}.

DRG Traversal. We start the DRG traversal (Lines 7-10) using the strategy described in Section 4.4.1. Recall that we use dataset discovery algorithms in a data lake context to discover the relationships between datasets, leading to multiple join possibilities between two datasets. We treat each of these join possibilities as a valid join in our approach, allowing our pruning strategy and the feature selection algorithm to determine the significance of the join result. Therefore, in a data lake setting, the $join\_keys$ set results in multiple join possibilities (Line 11). We maintain an additional queue P to preserve all the paths containing relevant information, given a specific join column jk. When the dataset exclusively contains KFK connections, the $join\_keys$ set will always yield a single join column.

Join & Prune. Once the join keys are known, we compute the join (Line 15), as detailed in Section 4.4.2. If the join is unfeasible due to mismatched join columns (i.e., in a data lake scenario), the join path is pruned according to our first pruning strategy described in Section 4.4.3. The subsequent step involves pruning based on data quality (Line 16). More precisely, if the proportion of non-null values in the join column falls below the predefined hyper-parameter threshold, τ, the join path is pruned.

Feature Selection. All the remaining join paths undergo the feature selection process. We introduce a heuristic approach, named *select κ best*, where we sort the features based on the correlation score, then select the top-κ performers [67]. Consequently, we carry out the relevance analysis (Line 17) using the high-performing method from Section 4.5.3 and retain only the top-κ features. If the resulting subset of features is empty, all features are irrelevant. However, we do not prune the join path, as it is an intermediary step towards multi-hop join paths. The final step of the feature selection pipeline is the redundancy

Algorithm 1 AutoFeat: Automatic Feature Discovery

Input: queue Q containing the base table T_0, R_{sel} selected features from T_0, τ null value ratio threshold, κ maximum number of features to select from a table
Output: dictionary of paths with scores *rank*

```
1:  function AUTOFEAT(Q, P = None)
2:      ranking ⟵ {}
3:      P ⟵ Q // Previous queue initialisation
4:      if Q is empty then
5:          return ranking
6:      end if
7:      while Q do
8:          T_0 = pop(Q)
9:          neighbours = get_neighbours(T_0)
10:         for node T_n ∈ neighbours do
11:             join_keys = get_join_keys(T_0, T_n)
12:             C = empty queue
13:             while P do
14:                 for jk ∈ join_keys do
15:                     R = join(T_0, T_n, jk)
                        // If the join is not possible, prune
16:                     D = get_data_quality(R)
                        // If the data quality < threshold τ, prune
17:                     R_rel, score_rel = measure_relevance(R, κ)
                        // If R_rel is empty, all features are irrelevant, continue
18:                     R_red, score_red = measure_redundancy(R[R_rel], R_sel)
                        // If R_red is empty, all features are redundant, continue
19:                     R_sel = R_sel ∪ R_red
20:                     score = COMPUTE_SCORE(score_rel, score_red)
21:                     ranking = (score, R) // Save ranking
22:                     update(C, R) // Update current queue C with the join R
23:                 end for
24:                 update(P, C)
25:             end while
26:         end for
27:     end while
28:     return AUTOFEAT(neighbours, P)
29: end function
```

analysis (Line 18). Based on the finding from Section 4.5.4, we apply MRMR to the subset of features resulting from the relevance analysis. If the result is an empty set, all the features are redundant. Otherwise, we update the list of selected features R_{sel} with the new subset (Line 19).

Ranking. Upon completion of the feature selection process, we use the scores from the relevance and redundancy analysis to compute a rank for the corresponding join path

Algorithm 2 Compute the score for ranking

Input: $score_{rel}$ the scores from the relevance analysis, $score_{red}$ the scores from the redundancy analysis
Output: one score number $ranking\_score$

```
1: function COMPUTE_SCORE(score_rel, score_red)
2:     N_rel = length(score_rel)
3:     N_red = length(score_red)
4:     sum_rel = sum(score_rel)/N_rel
5:     sum_red = sum(score_red)/N_red
6:     ranking_score = (N_red * sum_rel + N_rel * sum_red)/(N_rel * N_red)
7:     return ranking_score
8: end function
```


(Line 20), as explained in Algorithm 2. We compute the sum of the relevance analysis scores, weighted by the cardinality of the selected subset (Line 4). We repeat the process for the redundancy analysis (Line 5). The final ranking score is the sum of sum_{rel} and sum_{red}, weighted by their common divisor (Line 6). Finally, we store the join path and its corresponding ranking score in an ordered list and update the queues. AutoFeat is a recursive algorithm. Once all neighbours are visited, the algorithm starts with the neighbours as base nodes until the entire graph is explored (i.e., all nodes are visited).

From Ranked Paths to Training ML Models. We use the top-k join paths from the list of ranked join paths (Line 21) to systematically train ML models and evaluate the paths. To improve the efficiency of our approach, we use stratified sampling to sample the base table at the beginning of the process. This sampling strategy, however, only impacts the feature selection process and does not limit the scope of data considered during model training. After training, we select the best join path based on the accuracy of the ML model.

4.7 Evaluation

In this section, we provide quantifiable evidence showing that AutoFeat outperforms existing works by being 5x-44x faster on average. Moreover, AutoFeat exhibits an average of 16% increase in the predictive power of the augmented features compared to the competition. With our experiments, we aim to show the following:

- *Effectiveness* - AutoFeat is able to augment a base table with relevant information, which is used to train an ML algorithm, leading to an increase in accuracy.
- *Efficiency* - Relative to SOTA approaches, AutoFeat exhibits superior efficiency. This is primarily due to its non-reliance on ML models for augmentation.

4.7.1 Experimental Setup

Datasets. We evaluate AutoFeat on real-world, open datasets collected from OpenML.org, and summarised in Table 4.3. The *school* dataset originates from the evaluation of ARDA [36]. The datasets are used for binary classification, and we present their best accuracy based on the corresponding tasks found in OpenML.org. For *school* dataset, we report the

Table 4.3: Overview of datasets used to evaluate AutoFeat.

Dataset	# Rows	# Joinable tables ↓	Total # features	Best accuracy (OpenML.org)
credit	1001	5	21	0.99
eyemove	7609	6	24	0.894
covertype	423682	12	21	0.99
jannis	57581	12	55	0.875
miniboone	73000	15	51	0.9465
steel	1943	15	34	1.0
school	1775	16	731	0.831
bioresponse	3435	40	420	0.885

highest accuracy from the source paper ARDA. We use a collection of eight datasets varying in size and number of joinable tables, which we split into multiple tables using two different settings.

Benchmark Setting. We design a technique to divide a dataset into multiple small tables with known KFK constraints. We extend the setting from ARDA, where they use star schemata with known KFK connections. We call this approach the *benchmark setting*. Here, the DRG contains only KFK relations, which resemble snowflake schemata. With this setting, we aim to reproduce the results of the baselines [36, 129] and to show that AutoFeat can explore longer join paths in search for relevant features.

Data Lake Setting. With the *data lake setting*, we aim to simulate a data lake scenario where the connections between datasets are unknown. Using the same tables from the *benchmark setting*, we remove the edges representing KFK relationships, and we apply dataset discovery algorithms to find the relations. We use COMA [48] as implemented in Valentine [106], with a default schema matching strategy and a similarity threshold of 0.55 to encourage spurious, but not irrelevant, connections. The DRG is now a dense multi-graph. This scenario aims to showcase the predictive power of our approach in a data lake setting and the fact that we can discriminate between join columns.

Machine Learning Models. To validate that our augmentation approach can add relevant information, we use machine learning tasks. We use four decision tree algorithms (e.g., LightGBM, Extreme Randomised Trees, Random Forest and XGBoost), all available in AutoGluon [52], the AutoML framework for tabular data, which automatically handles data encoding and hyper-parameter tuning. We choose to use decision trees, as they are one of the most popular ML models used in practice; they are explainable and can outperform neural networks [64, 161] in tabular training data.

Metrics. We evaluate the performance of our approach by measuring (*i*) *effectiveness* using accuracy, and (*ii*) *efficiency* by measuring the *feature selection time* - the total time it takes to assess the fitness of features for augmentation.

4.7.2 Baselines

AutoFeat. AutoFeat respresents our feature discovery approach. We use $\tau = 0.65$ as the null value ratio and $\kappa = 15$ as the maximum selected features from a table.

BASE. The BASE approach represents the base table, which is assumed to perform poorly on any ML model. This represents the table we want to augment with more predictive and relevant features.

ARDA. ARDA is one of the SOTA approaches and is the feature augmentation system which uses a random-injection approach to select the features for augmentation [36]. Since the source code was unavailable, we implemented the feature selection part of the system following the algorithms and details provided in the original paper [36].

MAB. The second SOTA technique we consider is the Multi-Armed Bandit (MAB) method [129]. The study presents two methods for feature augmentation: one leveraging the multi-armed bandit reinforcement learning approach and another utilising deep Q networks, a neural network approach. Our results are consistent with the original study that reported comparable effectiveness (i.e., 0.02 – 0.04 difference in AUC) but major differences in efficiency between the two methods. Our experiments showed that the deep Q network method required a prohibitive amount of computational time (i.e., extending to days for one experiment), exceeding the limits of our available resources. In contrast, the MAB approach enabled us to conduct the experiments within a reasonable time frame (i.e., hours).

JoinAll. The *JoinAll* baseline is the approach where all the tables which connect with the base table are joined in all possible ways. Given a perfect scenario where (i) the join cardinality is 1:1, and (ii) the joins are on KFK, *JoinAll* results in a *single* path. However, when the connections are not KFK, regardless of the type of join, the number of possible paths increases tremendously, as the order in which the tables are joined affects the resulting augmented table. We denote D, the maximum depth of the join tree, $N(d)$ the number of nodes at depth $d \in D$, and $k(v)$ the number of un-visited neighbours of node v. The total number of possible *JoinAll* paths P is the product of all choices at all levels:

$$P = \prod_{d=0}^{D} \prod_{v \in N(d)} k(v)!, \tag{4.3}$$

JoinAll+F. The *JoinAll+F* baseline is the *JoinAll* approach, where we apply filter feature selection on the resulting dataset before training the ML model.

The experimental results are split based on the schemata configuration: the *benchmark* setting and the *data lake* setting.

4.7.3 Benchmark Setting Results

Recall that the benchmark setting represents the DRG with known KFK relationships without spurious connections. No results are shown for the baselines *JoinAll* and *JoinAll+F* in Figure 4.5 on the *school* dataset, due to the increased computation time. For *school* dataset, which follows a star schema, and the join cardinality is not 1:1, the number of possible join paths for *JoinAll* and *JoinAll+F* is $P = 15!$ (Equation (4.3)). As a result, those baselines did not finish within the given time constraint.

Effectiveness. AutoFeat shows an average accuracy increase of 16% across all datasets and algorithms. The AutoFeat's effectiveness aligns with our expectations as it navigates through longer join paths, leveraging transitive joins to identify more relevant features. This contrasts ARDA, which is limited to star schemata. Given its capability to handle

Figure 4.5: Benchmark Setting - **Top**: Depicts the average runtime using pastel shades, where contrasting colours illustrate the fraction of time dedicated to feature selection within the overall runtime. **Bottom**: Displays accuracy per dataset, averaged across all evaluated tree-based ML algorithms. Numerical values on the bars indicate the total number of joined tables. Horizontal lines signify the highest achieved accuracy, providing a clear visual differentiation of performance metrics.

transitive joins, we expected that MAB would be more effective than the results show. Compared to the *JoinAll* and *JoinAll+F* baselines, `AutoFeat` results in similar accuracy. This shows that `AutoFeat` finds the right features to augment a given base table.

Efficiency. The efficiency of `AutoFeat` is noteworthy. On average, `AutoFeat` operates 4.5x faster than ARDA and 12x faster than MAB. Referencing the *credit* dataset from Figure 4.5, `AutoFeat` is 4x faster than ARDA and 7x faster than MAB, even when the increase in accuracy is marginal. Using heuristics to identify relevant features contributes to the superior efficiency of `AutoFeat`, a stark contrast to ARDA and MAB, which incorporate the ML model into their feature discovery process. Given that the *JoinAll* baseline does not perform feature selection, we can only compare the total runtime, where `AutoFeat` is on average 3x faster than *JoinAll*. In contrast, the feature selection time of the *JoinAll+F* baseline costs less than one second since it performs feature selection once for a single wide table. However, on average, its total runtime is 3x slower than `AutoFeat`.

Path Analysis. Figure 4.5 illustrates the number of datasets each approach uses to find the most relevant features. ARDA is limited to star schemata and joins all the datasets directly connected to the base table. Thus, the number of datasets used to find the most related features is, at most, the maximum number of directly connected neighbours. Although MAB can handle transitive joins, it does not explore the schemata in depth. This limitation results from the fact that MAB requires the same join column name (i.e., PK-FK with the same names). `AutoFeat` requires fewer datasets while achieving higher accuracy, except for *covertype*, *eyemove*, and *steel*. Here, `AutoFeat` joins more tables than ARDA and MAB because `AutoFeat` explores the schemata in depth. We observed that the most relevant

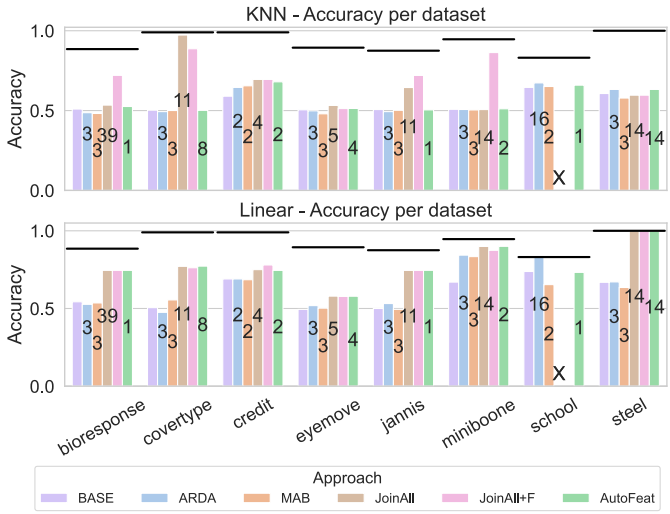


Figure 4.6: Benchmark Setting: The figure depicts the accuracy metrics for each dataset when applied to KNN and Linear Regression models. Numerical values on the bars indicate the total number of joined tables. Horizontal lines signify the highest achieved accuracy, providing a clear visual differentiation of performance metrics.

features reside via transitive joins; thus, most tables joined by AutoFeat are only required to navigate the join path search space.

Evaluation on Non-Tree-Based Models. Figure 4.6 shows the effectiveness of AutoFeat on non-tree based ML models: K-Nearest Neighbours (KNN) and Linear Regression with L1 regularisation (LR). For LR, AutoFeat achieves top performance together with *JoinAll* and *JoinAll+F* baselines across most datasets, with ARDA surpassing the *school* dataset. Dataset characteristics influence the effectiveness of KNN: it underperforms on smaller datasets due to insufficient neighbours for reliable decision-making (e.g., *credit*, *eyemove*, *steel*, *school*). Conversely, on larger datasets with fewer features (e.g., *covertype*, *jannis*, *miniboone*), or on high-dimensional datasets (e.g., *bioresponse*), KNN converges effectively with the *JoinAll* and *JoinAll+F* baselines.

4.7.4 Data Lake Setting Results

Recall that we apply an automated dataset discovery method to find the relationships between tables in a data lake setting. The once snowflake-like schemata transform into a densely connected graph in this context. Figure 4.7 and Figure 4.8 do not show results for the baselines *JoinAll* and *JoinAll+F* as the number of possible *JoinAll* paths in this scenario is extremely large (Equation (4.3)).

Effectiveness. AutoFeat consistently shows equal or superior effectiveness to other approaches (Figure 4.7). On average, AutoFeat outperforms ARDA by 12% and MAB by 2%. In a data lake setting with numerous spurious connections, AutoFeat adeptly prunes tables that introduce noise, indicating the critical role of join ordering for data augmentation.

Efficiency. The AutoFeat approach maintains impressive efficiency, even within a data

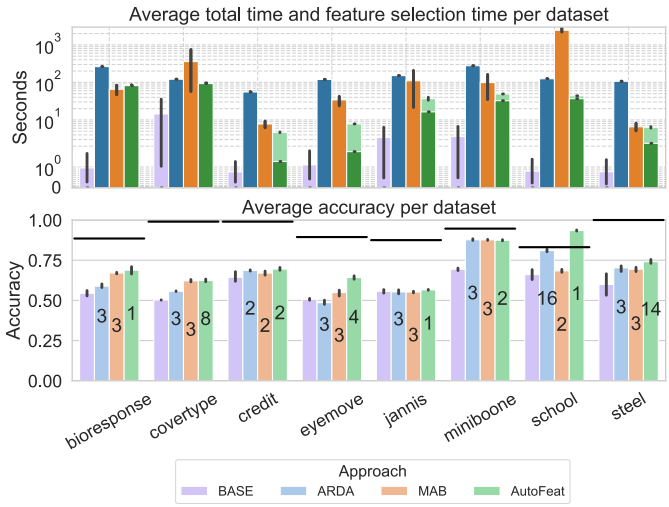


Figure 4.7: Data Lake Setting - **Top**: Depicts the average runtime using pastel shades, where contrasting colours illustrate the fraction of time dedicated to feature selection within the overall runtime. **Bottom**: Displays accuracy per dataset, averaged across all evaluated tree-based ML algorithms. Numerical values on the bars indicate the total number of joined tables. Horizontal markers signify the highest achieved accuracy, providing a clear visual differentiation of performance metrics.

lake setting. On average, `AutoFeat` operates 10x faster than MAB and 3x faster than ARDA while delivering comparable or superior accuracy (Figure 4.7). In this context, we explore various join column pairs to identify the path enriched with the most relevant features, subsequently increasing the time invested in feature discovery. However, using heuristics, instead of relying on machine learning models to predict feature importance, effectively counters the overall time spent, sustaining our approach's efficiency.

Path Analysis. Figure 4.7 illustrates expected behaviours. With the snowflake schemata disrupted, where dataset discovery algorithms yield many spurious connections, ARDA indiscriminately joins all the datasets neighbouring the base table. MAB suffers from the same limitation as in the *benchmark setting*, where it restricts its joins to tables sharing the same join column name, thereby inhibiting the exploration of transitive connections. `AutoFeat` can prune out irrelevant tables and explore the join path search space in depth (e.g., *jannis* and *steel* datasets). Once again, we noticed that the high number of joined datasets results from the fact that the most relevant features are in a multi-hop join path.

Evaluation on Non-Tree-Based Models. Figure 4.8 shows the performance of `AutoFeat` on non tree based ML models. In the data lake scenario, the baselines may introduce irrelevant features; thus, KNN does not achieve optimal results. KNN suffers challenges due to possible noise when irrelevant tables are joined, leading to worse distance measurements. Conversely, LR shows `AutoFeat` outperforming baselines in the majority of cases (i.e., five out of eight datasets), with ARDA maintaining a consistent lead on the *school* dataset.

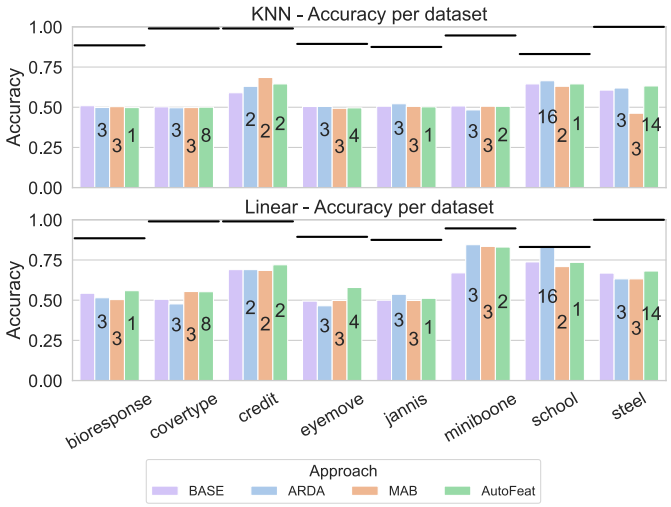


Figure 4.8: Data Lake Setting: The figure illustrates the accuracy metrics for each dataset when applied to KNN and Linear Regression models. Numerical values on the bars indicate the total number of joined tables. Horizontal lines signify the highest achieved accuracy, providing a clear visual differentiation of performance metrics.

4.7.5 Summary of Results

On a snowflake schema, AutoFeat can explore the join path space in depth, searching for relevant features. On a densely connected multi-graph, AutoFeat can explore different join column possibilities and follow the path with the highest data quality in search of the most relevant features for augmentation. On both configurations, AutoFeat is more efficient for feature discovery than the baselines.

AutoFeat is an approach best suited to tree-based ML models that can mitigate the consequences of high dimensionality. Kernel-based ML models and linear models suffer more from the "curse of dimensionality", where data points become sparse, and the distance between points loses meaning, which increases the difficulty of finding meaningful patterns in the data, resulting in degraded performance.

The results show that AutoFeat is a more efficient and effective method for feature discovery over long join paths on tree-based ML models.

4.7.6 Alternative Dataset Division Strategies

We designed multiple dataset division strategies to create the snowflake schemata, and consequently longer join paths. The default division strategy, also used for the *benchmark* setting, is called the "short reverse correlation". With this strategy, the most correlated feature is pushed towards the leaf nodes of the join tree, thus forcing the algorithms to search in depth for the most predictive feature. We acknowledge that this strategy creates an advantage for our approach, AutoFeat, thus, we created additional experiments with alternative dataset division strategies. Table 4.4 provides a summary of the total number of joinable tables and the maximum tree depth for each dataset for each table division strategy.

Table 4.4: Overview of the number of joinable tables and the maximum join tree depth for each dataset divided using one of the alternative table division strategies: correlation-based, random overlap, random tree.

Dataset	Correlation-Based		Random Tree		Random Overlap	
	# Joinable Tables	Max Join Tree Depth	# Joinable Tables	Max Join Tree Depth	# Joinable Tables	Max Join Tree Depth
bioresponse	420	6	308	10	848	11
covertype	50	4	40	5	87	6
credit	8	2	11	3	35	6
eyemove	24	3	15	3	40	6
jannis	55	4	39	7	97	7
miniboone	51	4	35	8	80	7
steel	34	4	20	4	50	4

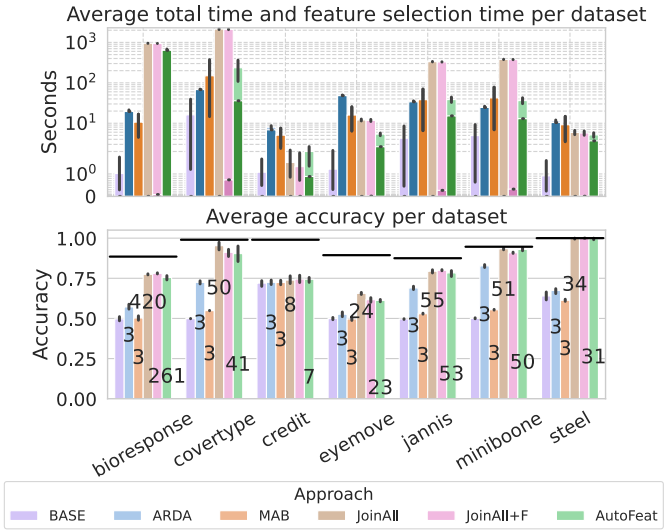


Figure 4.9: Correlation-Based - **Top**: Depicts the average runtime using pastel shades, where contrasting colours illustrate the fraction of time dedicated to feature selection within the overall runtime. **Bottom**: Displays accuracy per dataset, averaged across all evaluated tree-based ML algorithms. Numerical values on the bars indicate the total number of joined tables. Horizontal markers signify the highest achieved accuracy, providing a clear visual differentiation of performance metrics.

Correlation-Based. Contrary to the default strategy, the correlation-based table division places the most correlated feature in a table that is directly joinable with the base table. With this strategy, we test the accuracy of finding the table with the most predictive features regardless of the length of the join path.

The results of the experiments are summarised in Figure 4.9. In terms of accuracy, `AutoFeat` performs similarly to *JoinAll* and *JoinAll+F* approaches, which means that `AutoFeat` finds the most important features for augmentation, outperforming *ARDA* and *MAB*. However, in this setting, for *bioresponse* dataset which contains 420 joinable tables, `AutoFeat` is exponentially slower than *ARDA* and *MAB*. This decrease in efficiency stems from the fact that, during the search for the best features, `AutoFeat` performs joins on 261 out of the

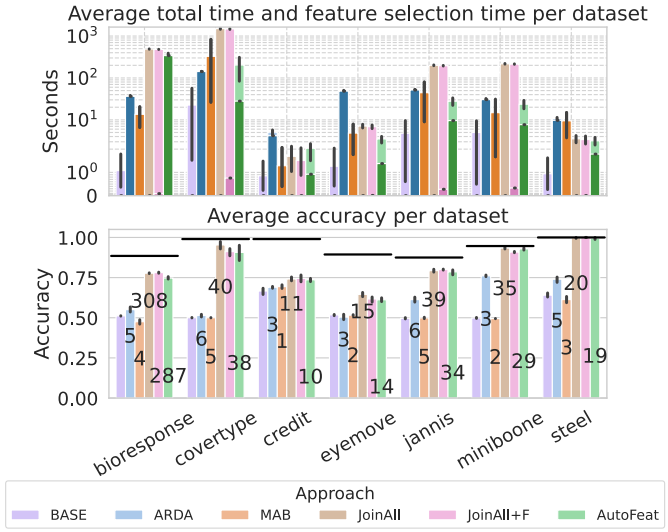


Figure 4.10: Random Tree - **Top**: Depicts the average runtime using pastel shades, where contrasting colours illustrate the fraction of time dedicated to feature selection within the overall runtime. **Bottom**: Displays accuracy per dataset, averaged across all evaluated tree-based ML algorithms. Numerical values on the bars indicate the total number of joined tables. Horizontal markers signify the highest achieved accuracy, providing a clear visual differentiation of performance metrics.

total 420 tables. In contrast, *ARDA* and *MAB* only join the three tables directly connected to the base table. We also observe that AutoFeat exhibits a greedy behaviour and performs many more joins than *ARDA* and *MAB*, which are limited to star schemata.

Random Tree. The random tree table division strategy employs vertical splits performed randomly, resulting in new tables with no overlap between them. Subsequently, each of these smaller tables is recursively subdivided until only tables containing a single PK column and one additional column from the original input table remain. These final tables are referred to as leaf tables.

Figure 4.10 shows the results of the experiments using the datasets divided with the random tree strategy. The effectiveness of AutoFeat is barely changed in comparison with the correlation-based strategy and even the *benchmark setting*. In terms of efficiency, we notice the same behaviour on the *bioresponse* dataset, where AutoFeat joins 287 out of 308 tables, which leads to a high runtime. AutoFeat exhibits the same greedy behaviour of joining more tables searching for the most important features for augmentation as seen for the correlation-based strategy.

Random Overlap. The random overlap table division strategy is similar to the random tree strategy, with the addition of allowing tables to have overlapping columns. The results of the experiments are presented in Figure 4.11, showing no deviation in behavior compared to the previous two table division strategies.

These results indicate that AutoFeat exhibits greedy behavior, prioritizing effectiveness over efficiency.

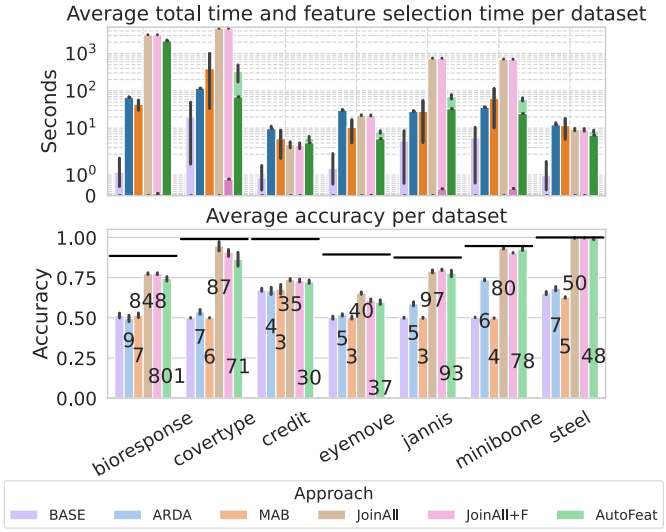


Figure 4.11: Random Overlap - **Top**: Depicts the average runtime using pastel shades, where contrasting colours illustrate the fraction of time dedicated to feature selection within the overall runtime. **Bottom**: Displays accuracy per dataset, averaged across all evaluated tree-based ML algorithms. Numerical values on the bars indicate the total number of joined tables. Horizontal markers signify the highest achieved accuracy, providing a clear visual differentiation of performance metrics.

4.7.7 Parameter Sensitivity Analysis

`AutoFeat` relies on two hyper-parameters: the null value ratio (τ) and the maximum number of features to select (κ). The next experiments aim to demonstrate `AutoFeat`'s sensitivity to variations in τ and κ and their influence on effectiveness and efficiency.

Sensitivity to κ. Figure 4.12 illustrates the change in accuracy and feature selection time with different values of κ, denoting the maximum number of features to be selected. As κ varies from 2 to 6, there is a notable 4% increase in accuracy at the cost of an additional 2.5 seconds in feature selection time. However, when κ ranges from 6 to 10, the increase in accuracy is less than 1%, despite the same increment of 2.5 seconds in time. A similar trend is observed when κ varies within the interval [15, 20]. We recommend selecting a value for κ within the range [10, 15], as this provides a high accuracy level with a tolerable trade-off of a few seconds. Any $\kappa > 15$ does not significantly improve accuracy or efficiency but could potentially risk overfitting the ML model.

Sensitivity to τ. Our pruning strategy is contingent on data quality, such that a table is pruned if the ratio of null values to the total number of values exceeds a threshold τ. Figure 4.13 displays the average accuracy and feature selection time across datasets for different values of the hyper-parameter τ within the range of [0.05, 1.0], incremented by steps of 0.05. Most datasets appear to be relatively insensitive to the threshold, barring two exceptions: *covertype* (shown in orange) and *school* (illustrated in pink). In Figure 4.14 and Figure 4.15, we provide a closer look at these two datasets.

When $\tau = 1$, it leaves no room for null values, implying that the tables perfectly match on the join keys, thus leading to peak accuracy. However, the *school* dataset depicted in

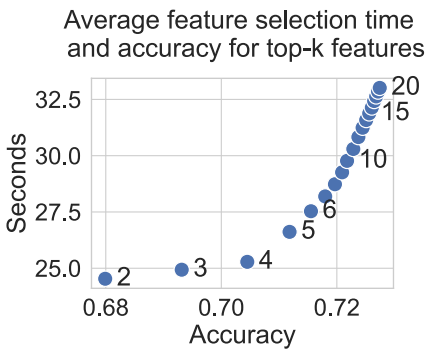


Figure 4.12: The figure shows the feature selection time and accuracy for top-κ features, averaged over datasets and ML models.

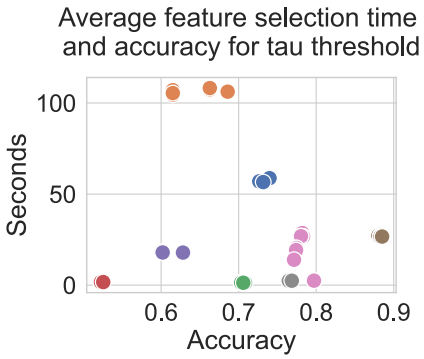


Figure 4.13: The figure shows the feature selection time and accuracy for every τ threshold in the $[0, 1]$ interval, averaged over datasets and ML models.

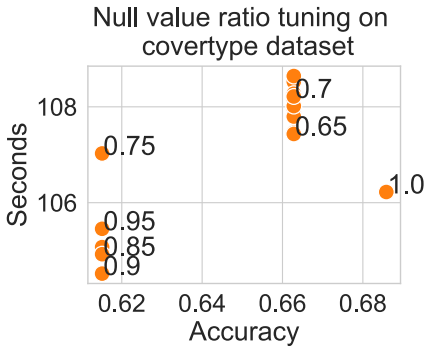


Figure 4.14: The figure illustrates the results of null value ratio tuning for *covertype* dataset in terms of accuracy and feature selection time averaged over ML models.

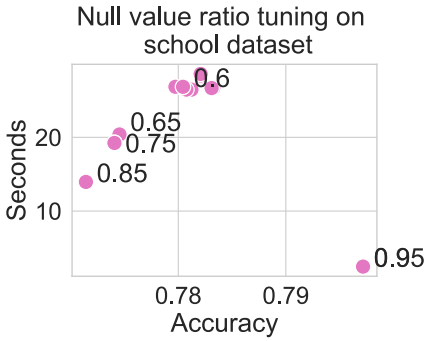


Figure 4.15: The figure illustrates the results of null value ratio tuning for *school* dataset in terms of accuracy and feature selection time averaged over ML models.

Figure 4.15 yields no output when $\tau = 1$, indicating no tables with perfect matches on join keys. Hence, although $\tau = 1$ can result in optimal accuracy in certain instances, it is overly restrictive and fails to generalise across other tables.

The data indicates that for τ within the range $[0.05, 0.6]$, the results tend to cluster around similar accuracy values or feature selection times. This suggests that not many tables are pruned during this interval, and imputation methods have little effect on accuracy. However, for $\tau > 0.6$, we observe a decrease in accuracy (up to 4% less) and time. Reducing feature selection time implies that more tables are pruned at this stage. The decrease in accuracy signifies potential bias introduced by the imputation strategies. We recommend $\tau = 0.65$ as a balance between accuracy and feature selection time.

The hyper-parameter κ directly influences our model's effectiveness, with higher κ leading to improved results. Meanwhile, the hyper-parameter τ greatly impacts the model's efficiency, highlighting the role of our pruning strategy.

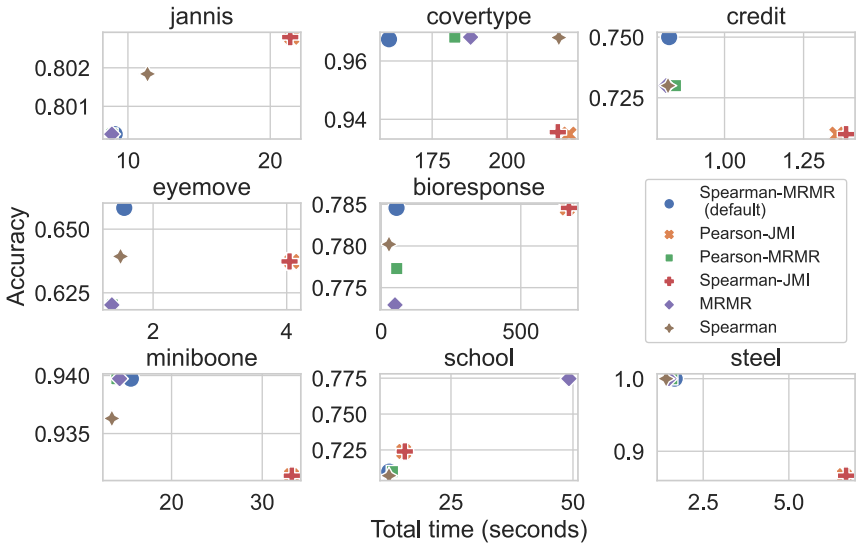


Figure 4.16: The figure shows the accuracy and total time on the ablation study with different configurations of AutoFeat for every dataset.

4.7.8 Ablation Study

AutoFeat uses two core methods: Spearman to measure relevance and MRMR redundancy. To show that AutoFeat is indeed more efficient and more effective thanks to these methods, we perform an ablation study, where we use different configurations of AutoFeat. We replace MRMR with JMI, Speaman with Pearson, or both one by one. Finally, we turn off the computation of relevance or redundancy.

The results of the ablation study are presented in Figure 4.16. The variants of AutoFeat which use JMI are at least two times slower than AutoFeat, consistent with Section 4.5.4. However, both Pearson-JMI and Spearman-JMI are less accurate than AutoFeat on five out of eight datasets and marginally more accurate on *school* and *jannis*. Compared with AutoFeat, the Spearman-only version results in either degradation in time or accuracy and achieves similar results only on *school* and *steel* datasets.

Pearson-MRMR and MRMR versions exhibit similar behaviour. They are either less accurate (i.e., they fail to find all the relevant features) or slower (i.e., they retain too many features). However, on three of the datasets, they result in similar performance with AutoFeat. One reason for this behaviour is the fairly similar schemata of these datasets, which contain the same amount of joinable tables. However, on *school* dataset, MRMR is the slowest but also the most accurate. Here, the schemata are star-shaped, which results in fewer opportunities for deep pruning. MRMR fails to discard features, exhibiting a behaviour similar to *JoinAll* baseline.

The proposed version of AutoFeat (i.e., Spearman-MRMR) is the most efficient version with minimal accuracy loss.

4.8 Conclusion

We have introduced AutoFeat, a novel approach for automatic transitive feature discovery. AutoFeat addresses a critical gap in current data science practices by navigating the complex join space, effectively augmenting a base table with relevant features across various datasets. It exploits a relevance-aware algorithm for scoring features, not relying on specific ML models, making the approach broadly applicable and versatile. The outcomes of our research allow us to respond affirmatively to the second research question:

***RQ2**: How can we enhance the automation of the feature discovery process to create high-quality datasets for machine learning applications?*

By implementing an automated feature discovery method, we have successfully created high-quality datasets for training machine learning models. AutoFeat distinguishes itself by its in-depth search for relevant features by exploring transitive joins. Furthermore, our findings reveal that AutoFeat achieves a level of effectiveness comparable to the exhaustive *JoinAll* strategy when applied to tree-based ML models. Yet, it does so in a significantly reduced timeframe. This confirms that automating the feature discovery process, which consists of identifying related tables and joinable columns, executing table joins, and selecting features, can create datasets that meet high-quality standards. Furthermore, the feature discovery process and the resulting dataset significantly improve the performance of machine learning models.

In the next part, we explore the user's role as a domain expert in the feature discovery process. Therefore, in the next chapter, we describe our user study with 19 data practitioners and report our findings in contrast with the theoretical feature discovery pipeline.

4.8.1 Future Work

Several directions remain open for future exploration. We aim to integrate more complex feature selection strategies that could improve the feature's relevance and overall performance. We plan to explore dynamic hyper-parameter tuning, allowing the algorithm to adapt to different data landscapes and tasks.

Moreover, as datasets grow in complexity and volume, more efficient methods of navigating join paths and reducing the complexity of the problem space should be investigated. We are particularly interested in how graph-based models and representations could be used to achieve this. Other improvement possibilities might arise after evaluating our approach to real data lake scenarios within organisations, where we envision that more aggressive pruning strategies might be required.

III

Human-in-the-Loop Feature Discovery

5

Feature Discovery: a User Study

Multiple works in data management research focus on automating the processes of data augmentation and feature discovery to save users from having to perform these tasks manually. Yet, this automation often leads to a disconnect with the users, as it fails to consider the specific needs and preferences of the actual end-users of data management systems for machine learning. To explore this issue further, we conducted 19 semi-structured, think-aloud use-case studies based on a scenario in which data specialists were tasked with augmenting a base table with additional features to train an ML model. In this chapter, we share key insights into the practices of feature discovery on tabular data performed by real-world data specialists derived from our user study. Our research uncovered differences between the user assumptions reported in the literature and the actual practices, as well as some areas where literature and real-world practices align.

This chapter is based on the following workshop paper and research paper currently under review:

- *Andra Ionescu, Zeger Mouw, Efthimia Aivaloglou, and Asterios Katsifodimos. "Key Insights from a Feature Discovery User Study". HILDA@SIGMOD 2024* [90]
- *Andra Ionescu, Zeger Mouw, Efthimia Aivaloglou, and Asterios Katsifodimos. "Feature Discovery for Machine Learning: a User Study". under review*

5.1 Introduction

The assumption that the input data of a Machine Learning (ML) model is typically given as a single table has been refuted for a long time. In reality, valuable predictive features might be scattered across various database tables or files within an open data repository or data lake [144]. In order to discover and use related tables scattered in a data lake, the database community has proposed multiple schema matching and dataset discovery techniques [17, 58, 106]. Given a large set of tabular datasets, the natural next step towards training an ML model is to discover predictive features. A significant and ongoing research effort is dedicated to developing automated methods to create training datasets for machine learning (ML) applications. This process, named feature discovery, builds upon the exploration and integration steps from dataset discovery and relies on feature selection approaches to select only the most relevant features for an ML task [36, 53, 129, 205]. This process typically starts with a query table that contains a target variable. Then, through an exploratory process, relevant candidates from a data repository are augmented to improve ML model effectiveness.

User-centric research in data management has progressively expanded its focus, delving into the complexity of interactive user interfaces [140] and the usage of data collected through crowdsourcing methodologies [116, 175]. These efforts underscore the critical role of the user as an integral component within the data management process. Historically, the focus on technological advancements has often led to user engagement playing a more supporting role, facilitating the evolution and refinement of these advancements [130]. However, contemporary studies within this community increasingly acknowledge the centrality of the users as a fundamental part of driving the research.

Existing research continues to operate under various assumptions regarding user workflows within the feature discovery pipeline. Some state-of-the-art works [58, 112] have incorporated user studies in their evaluation scenarios. Yet, the methods and approaches they offer are not always grounded in empirical evidence from actual user workflows. Despite this progress, a noticeable gap remains: the user perspective is lost as more automated feature discovery and augmentation approaches are developed. By understanding how users interact with and perceive the feature discovery process, we can develop more intuitive and effective methods for identifying and integrating relevant features. With this chapter, we build the foundational knowledge to answer our third research question.

***RQ3**: Can human expertise and domain knowledge enhance the automatic feature discovery process?*

Therefore, we aim to answer the following sub-questions:

How do data scientists and engineers in the real world perform feature discovery when asked to train an ML model from tabular data residing in a data lake?

Does the real-life process align with the theoretical one reported in the literature?

To answer these questions, we study, understand and integrate real-world user experiences and requirements, aiming to raise awareness of real users' needs through data management research. To this end, we contribute with the first qualitative, think-aloud

user study on how data scientists and engineers with hands-on experience perform feature discovery for training an ML model based on tabular data. We engaged a diverse set of 19 participants from various organizations, presenting them with a small-scale feature discovery task. The interviews allowed us to capture a nuanced understanding of how these professionals interact with feature discovery challenges in real-world scenarios.

5.2 Related work

The literature review reveals various workflows and pipelines that have been meticulously examined in various studies.

The Data Science Pipeline. The concept of a data science pipeline, encompassing steps from data acquisition through to cleaning, curation, and modelling, is well-documented [15]. This pipeline has been explored from multiple perspectives. Theoretical studies resulted in extensive literature reviews [15, 41]. On the empirical side, existing pipelines from platforms such as Kaggle and GitHub have been analyzed for their real-world applicability [15, 167]. Furthermore, a qualitative perspective has been adopted, where insights were gathered from interviews with industry practitioners, enriching the understanding of these pipelines in practical settings [97, 104, 143, 167, 168, 180, 183, 200].

The Data Pipeline. The data pipeline concept, characterised as a sequence beginning from a data source and ending at a point where the processed data is delivered [162], has been extensively studied from various viewpoints, similar to the data science pipeline. In-depth qualitative interviews have been instrumental in conceptualising the data processing and preparation activities [100, 158]. The insights gained from these studies highlight the complexities in data pipeline management, emphasising the need for continuous evaluation and adaptation in response to evolving challenges and technological advancements.

The Machine Learning Pipeline. Machine learning serves as another example of a domain where the pipeline concept has been explored in depth through qualitative research. Many studies have focused on conceptualising the machine learning operations pipeline [109, 172, 173], and creating an automated machine learning workflow [40].

User Studies in Data Management. In data management, qualitative research methods have been extensively employed to assess the effectiveness and efficiency of different data management systems [163, 165], or evaluate emerging frameworks [204]. This highlights the pivotal role of qualitative research in understanding and refining data management tools and practices. Beyond qualitative research, the use of crowd workers for data labelling and application testing has become increasingly prevalent [116, 175]. This approach leverages the collective efforts of a diverse group of individuals to enhance the quality and accuracy of data and provide practical insights into application usability and functionality. Furthermore, the expertise of human professionals is often called upon for the critical evaluation of frameworks [73, 140].

Together, these diverse methodologies, ranging from qualitative research to crowdsourced testing and expert evaluation, underscore the multi-faceted approach necessary for effectively developing and assessing data management systems and frameworks. While these interview studies are comprehensive and provide valuable insights, they typically employ open-ended questions or retrospectively analyse existing projects within an organization. Our study, however, adopts a hands-on, practical use-case scenario approach,

placing participants directly in front of a real task. This design aims to immerse the participants in their typical workflows, allowing us to capture data from their actual, hands-on work processes. To the best of our knowledge, this is the first research paper to study the feature discovery pipeline and perform user studies using a use-case scenario.

5.3 Preliminaries

This section describes the preliminary steps of the feature discovery pipeline. Recall that feature discovery is the process which discovers and augments relevant features for improving ML models [144]. Given a query table with a label column, feature discovery retrieves candidates from a data repository focusing on the features that can improve an ML model's performance [144]. Therefore, feature discovery emerges as a process that intersects various steps of the data management for machine learning pipeline and extends across the broader data science workflow.

According to the literature, feature discovery contains several key steps, such as data exploration, data integration, feature selection and ML modelling, as illustrated in Figure 5.1. Moreover, data processing steps, such as aggregation, sampling, and cleaning, can be found in between the data preparation steps – data exploration and data integration – as named in the data science and machine learning pipelines [41, 173, 183, 207].

Data Exploration. Data exploration is the process of navigating a collection of datasets with the aim of extracting knowledge [81, 151]. The process makes use of user interactions, such as exploring visualisation techniques and interfaces. The process usually leverages databases, where the storage format and indexes can make the exploration more efficient.

Data Integration. Data integration is the problem of combining various datasets from diverse sources and providing a unified view of the data [115]. Integrating tables is possible through join or union operations or related table search [103]. In feature discovery and data augmentation, the data integration step uses joins to extend a table with more features [36, 129].

Data Processing. The data processing step concerning aggregation and sampling can be found right before the data integration step, as a preparatory process before the integration [36]. Another data processing step can be found right after the data integration step, as a cleaning process before feature selection, to ensure the data quality [93].

Feature selection. Feature selection methods are classified into three categories: wrapper, embedded, and filter [139]. The *filter feature selection* methods are independent of the learning algorithm and rely on the general characteristics of data to compute statistics and use them to rank the features. The *wrapper feature selection* methods function in a manner similar to cross-validation, where the employed model is used to test the accuracy of the feature subset. The wrapper methods are known to be useful but slow, as the model has to be constantly trained and evaluated [119]. The *embedded feature selection* is one of the optimal approaches to feature selection, which uses the advantages of both filter and wrapper methods [114]. In feature discovery, the wrapper feature selection or a hybrid approach are the most employed methods [36, 53, 129].

Machine Learning Modeling. The routine tasks of a machine learning pipeline include model planning, selection and mining. Next, the model is trained with labelled data points

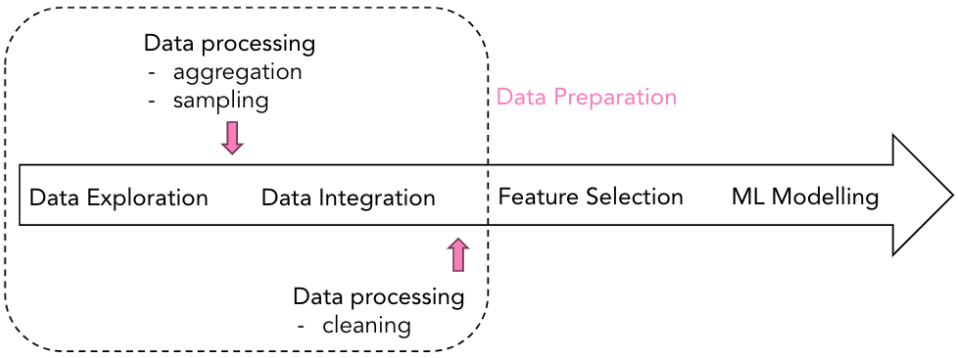


Figure 5.1: The feature discovery pipeline according to the literature.

and tested with new data (i.e., not used for training). Finally, after optimisations and tuning, the model is used to solve problems in an unknown setup [15]. Feature discovery uses the ML modelling step in various stages of the augmentation: (*i*) during the workflow as a main component of the augmentation process, or (*ii*) as an evaluation step at the end of the process.

5.4 User-Study Design

The goal of this study is to explore how data practitioners perform feature discovery in a real-life scenario. Our focus is on observing the steps of the feature discovery workflow and comparing them with the theoretical workflow and the assumptions from the literature. To this end, we conducted 19 semi-structured think-aloud interview sessions with different data practitioners from diverse organisations. We describe the participants, the use case scenario, the interview process, and the data processing pipeline.

5.4.1 Participants

We recruited individuals who work with data and perform data integration tasks, such as joining tables, creating augmented tables and using them for analysis. Participants were invited via an open call for interviews, which was disseminated through communication channels such as Twitter and LinkedIn, as well as via the authors' professional network. An anonymised description of our participants is shown in Table 5.1.

We recruited participants with diverse roles in the organisation (Figure 5.2a), such as data engineers, data analysts, data scientists, machine learning engineers and others with different titles at the moment of the interview but with experience in data engineering or analysis. We aimed to interview a balanced number of data experts per role for potential role-specific workflow differences to be represented in the results.

The participants varied in their years of experience working with data, as illustrated in Figure 5.2b, industry section and company size, and education. We used the years of experience as guidelines to further organise the participants into juniors (i.e., less than three years of experience), mediors (i.e., between three and five years of experience), and

Table 5.1: The table provides an anonymised description of the interviewed participants. We present their current role and their latest degree, and we categorise their years of experience with data into *Junior, Medior, Senior*. The participants work in organisations of various sizes, and diverse industry sectors.

#	Role	Education	Data XP	Org. Size	Industry Sector
P1	ML Platform Eng.	Masters	Senior	Medium	Software Dev.
P2	Bio Info. Eng.	Bachelors	Junior	Small	Software Dev.
P3	Data Engineer	Bachelors	Senior	Medium	Software Dev.
P4	Data Scientist	Masters	Junior	Large	Banking
P5	Data Scientist	Masters	Senior	Medium	Media Broadcast
P6	ML Platform Eng.	PhD	Senior	Medium	Software Dev.
P7	Data Analyst	Masters	Medior	Large	IT Serv.
P8	Data Engineer	Bachelors	Senior	Large	Consumer Serv.
P9	Data Scientist	Masters	Senior	Large	E-learning
P10	ML Engineer	Masters	Junior	Medium	Financial Serv.
P11	ML Platform Eng.	Masters	Senior	Medium	Software Dev.
P12	Data Scientist	Masters	Medior	Large	Software Dev.
P13	Data Engineer	Masters	Senior	Large	Retail
P14	Lead Engineer	Masters	Medior	Small	Software Dev.
P15	Data Engineer	Masters	Senior	Large	Financial Serv.
P16	CEO	MBA	Senior	Small	Supply Chain
P17	Data Engineer	PhD	Senior	Large	Software Dev.
P18	Data Analyst	Masters	Senior	Large	Retail
P19	ML Engineer	Masters	Junior	Small	IT Serv.

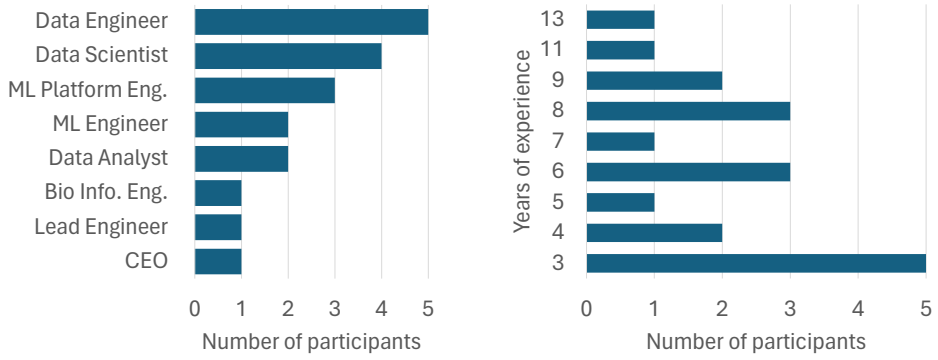
seniors (i.e., more than five years of experience). As such, four participants identified themselves as juniors, three as medior, and the rest as seniors (Table 5.1).

When asked about their latest degree, the majority of our participants have a master's degree in diverse areas of computer science; three have a bachelor's degree in business and technology, computer science or bioinformatics, and two of the participants have a doctorate degree in computer science. These statistics are illustrated in Figure 5.3a.

We also used the number of employees mentioned by the participants to organise the companies into small businesses (i.e., less than 100 employees), medium businesses (i.e., between 100 and 1500 employees), and large businesses (i.e., more than 1500 employees). According to Figure 5.3b, we have four participants from small organisations, six from medium organisations and nine from large organisations.

5.4.2 The Use Case Scenario

We offer a concise overview of the use case scenario, as illustrated in Figure 5.4. The use case scenario contains 17 tables representing a small dataset about schools used for state-of-the-art data augmentation evaluation [36, 93, 129]. Central to this scenario is the base table (i.e., named *base* in Figure 5.4), which is the primary focus for feature discovery. This table is used for binary classification tasks, where the target prediction — specifically the *class* feature depicted in Figure 5.4 — represents the performance of each school on a



(a) Distribution of the roles in the company.

(b) Distribution of the years of experience with data.

Figure 5.2: Statistics about a) the roles of participants in the moment of the study, and b) the years of experience as reported by participants.

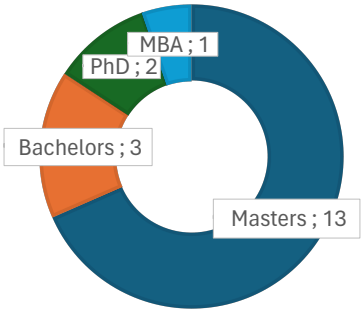
standardised test based on student attributes. When applied to this base table, a decision tree ML model yields a baseline accuracy of 0.69.

The objective is to enrich the base table with additional relevant features extracted from the other 16 tables. The augmented features aim to improve the accuracy of a tree-based ML model. For efficiency and to streamline the process, we facilitated the participants by providing a curated subset of seven tables following their preliminary investigation. These selected tables are highlighted in Figure 5.4. The optimal outcome of the feature discovery task is to augment the base table with features from two specific tables (e.g. *2010_Gen_Ed_Survey_Data, Schools_Progress_Report_2012-2013*), which should increase the accuracy of the initial ML model to 0.83.

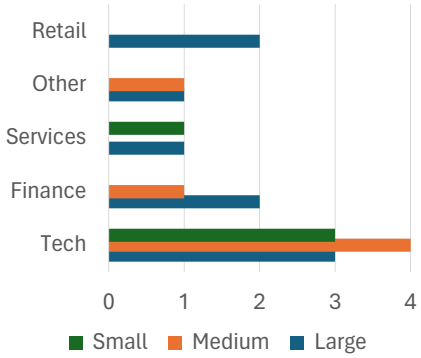
5.4.3 Interview Process

The interview was structured as a think-aloud use-case scenario, where the participants were presented with a dataset in CSV format and were asked to augment a base table with more features from the other remaining tables (Section 5.4.2). The participants worked on their own machines, using their own sets of tools, and were asked to share their screens so that we could capture information on their workflow. Participants were encouraged to think aloud at the beginning and during the interview, explaining the motivations and expectations related to their actions.

The interview was semi-structured; thus, we did not strictly follow a formalised list of questions. Instead, during the use-case scenario, we asked open-ended questions, allowing for a discussion with the participants about the task that they were observed to be performing. Our list of open-ended questions spanned the four steps of the feature discovery pipeline: (*i*) exploration, with interest in the dataset characteristics they usually work with; (*ii*) integration, looking for the key steps of the integration such as the join columns, type and problems, and also their own experience with joining tables, (*iii*) feature selection, observing how the feature selection process is performed and what tools are used,



(a) Distribution of the last completed degree.



(b) Distribution of the industry sector and company size.

Figure 5.3: Statistics about a) the industry sector where participants work and the corresponding company size, and b) education as reported by the participants.

and (*iv*) evaluation, observing if and how the augmented dataset is evaluated. Besides these steps, we asked semi-structured follow-up questions and structured questions about their education and experience.

Before the interview, the participants were asked to provide their consent to participate in the study, to share their screen and record their screen and voice. We conducted all interviews via video conferences. The interviews lasted between 45 and 60 minutes. The first two interviews were used to inform the study scope and refine the protocol questions, which remained unchanged for the other interviews. After 16 interviews, the information started converging, and no new insights emerged, reaching saturation [66]. The interviews were transcribed using automatic transcription software and were afterwards manually corrected. The screen recordings were used to annotate the transcripts with notes on the participants' actions. The annotated transcripts were anonymised and used for subsequent processing.

5.4.4 Data Processing

We analysed the data resulting from the interviews using the thematic analysis methodological framework. In thematic analysis, the determination of themes (i.e., patterns) can be both theory-driven and data-driven [16]. We used the theoretical feature discovery workflow, summarised in Section 5.3, to derive the main themes, and we used the interview data to generate the sub-themes, also known as codes.

We derived four *a priori* themes from the feature discovery pipeline: exploration, integration, feature selection, and evaluation. We finalised our set of codes *a posteriori*, extracted inductively from the interview data using the qualitative analysis tool ATLAS.ti[1]. In total, we used 60 codes to label different aspects of each step in the pipeline. We extracted a total of 1088 quotes, each labelled with one or multiple codes.

[1] https://atlasti.com/

base
DBN
School Name
School Type
Total Parent Response Rate (%)
Total Teacher Response Rate (%)
Total Student Response Rate (%)
class

Base table

- # rows: **1775**
- Target prediction: ***class***
- Accuracy: **0.69**

Augmented base table

- Accuracy: **0.83**
- Best features:
  - 2010_Gen_Ed_Survey_Data
  - Schools_Progress_Report_2012-2013

Table Name	# Features	# Rows
2010_Gen_Ed_Survey_Data	300	1598
2013_NYC_School_Survey	10	1775
ap	5	259
crime	30	1920
disc	230	1666
esl	16	28466
gender	17	47736
math	16	28479
oss	12	1744
pe	14	1600
qr	3	4795
s2tr	3	1572
sat	6	461
Schools_Progress_Report_2012-2013	21	1818
transfer	20	56
yabc	20	23

Figure 5.4: The figure illustrates statistics about the use case scenario. On the left side, we present the base table and its columns, while on the right side, we present an overview of the entire dataset: each table name, the corresponding number of features and rows. The yellow-highlighted rows represent the filtered list of candidate tables for augmentation.

Our themes and codes describe the following steps and activities: (*i*) goal setting – problem definition, understanding the target prediction, creating a baseline ML model, (*ii*) exploration – understanding the query table, understanding the entire dataset, (*iii*) integration – data integration pipeline, primary-key foreign-key joins, identifying the join column, the join type and the problems arising from using the specific join type, tools, (*iv*) feature selection – feature engineering steps and tools, manual or automated selection, (*v*) evaluation – model building and training, (*vi*) data preparation – any operation executed with the aim to process the data, (*vii*) documentation – the process of documenting the datasets, data catalogues, recommendations, and tools. Besides these themes, we also have a special follow-up theme, defined *a priori*. With the follow-up, we gathered data about education, experience, tools, and the problems they encounter on a daily basis.

In the description of the findings, the participants have been assigned numbers and are referred to as P1 to P19. The quotes that are included in the description are verbatim.

5.5 Feature Discovery Pipeline: Findings

This section introduces the empirical feature discovery pipeline derived from our interview data. The pipeline encompasses five key phases: Goal Setting (including hypothesis formulation and understanding the target variable), Data Exploration (focusing on base table analysis and dataset overview), Data Integration (covering join operations and

pipelines), Feature Selection (involving feature engineering and selection processes), and Dataset Evaluation. A process often regarded as a step in the pipeline, Data Preparation is a key process through which the users ensure that each of the five steps in the pipeline uses high-quality data. Each phase is critical for constructing a comprehensive understanding of the datasets and guiding the augmentation process for the development of effective ML models.

5.5.1 Goal Setting

Our findings reveal that, contrary to the literature, the first step of the feature discovery process is not data exploration but rather (*i*) defining a goal, (*ii*) clarifying the target prediction, or (*iii*) consulting with clients to understand the scope of the problem. This initial goal-setting phase is commonly seen in pipelines focused on dataset search, as seen in Aurum [58], or in typical data science workflows [41]. However, beyond this context, goal-setting as an initial step of the pipeline is not typically observed in feature discovery or data augmentation pipelines. The reason for this might be unintentional oversight or implicitly assuming that goal-setting is an inherent part of the process [41].

On the other hand, our participants indicated that having the goal already defined affects the subsequent steps in the process, such as data preparation. We observed that just describing the task was insufficient for the participants, who required much more information, such as information about the column names (i.e., what each column represents), in-depth details about the target variable and its meaning, and baseline accuracy. We provide a detailed description of these findings in the subsequent points.

– *Setting a Goal or Hypothesis.* Four participants begin the process with a clear hypothesis or goal, often in the form of a business KPI (P8), a hypothesis (P7), or already specified in a design document (P17). Having the goal already defined affects the subsequent steps in the process, such as data preparation (P2).

> *"We take the data to check our findings and not the other way around. We don't go through all the data to check correlations and then act from there."* (P7)

– *Understanding the Target Variable.* While the participants familiarised themselves with the dataset, they specifically focused on gaining a deeper understanding of what the *true* and *false* values signify within the context of the target feature. Nine out of 19 participants (9/19) are interested in knowing how the target feature relates to the overall goal of the analysis and emphasise the importance of understanding the target variable to select relevant features and build an effective machine learning model.

> *"So you can, of course, do arbitrary augmentations and integrations, but at the end of the day, the real thing that matters is if the features that you have are in some way related to the thing you're trying to predict."* (P2)

– *Discussing with the Client.* Another example of goal setting is via discussing with the client. When receiving data from clients, the first step is to understand the meaning of each column to mitigate the risks associated with using potentially misunderstood data. This understanding is crucial to avoid building models that may not be reproducible or relevant in the future (7/19).

> *"First thing you always do is talk to the client because it's very risky to use columns that might be very discriminating. But if you don't know what they mean, then you don't know what your model is working on, and then you might be building something that's not reproducible in the future."* (P14)

The participants also describe scenarios where errors in client-provided data are identified and addressed. These errors are typically inspected manually, and findings are communicated to the client, highlighting discrepancies with the expected schema.

– *Creating a Baseline ML model.* Almost half of the participants emphasised the importance of establishing a baseline ML model before inspecting and exploring other datasets. The baseline serves as a reference point for understanding the effectiveness of the model with minimal features and without data augmentation (8/19).

The feature discovery process starts by formulating a clear hypothesis or goal – A step often excluded from data and ML pipelines, the users spend time formulating their goal and hypothesis before any other subsequent steps.

5.5.2 Data Exploration

The literature highlights that finding relevant tables within a vast collection of datasets is an arduous and time-intensive task, as users are unaware of the relationships between the tables [36, 38, 58, 112].

In reality, users typically have a clear understanding of the location of the data and of the overall problem they need to solve with data. This is frequently achieved by examining available documentation, which acts as a road map through the data. Moreover, users actively seek the expertise of colleagues with business insight or firsthand experience with data collection, providing invaluable context. When the data originates from clients, the business problems tend to be well-articulated, further guiding the users in their quest to find meaningful information. These resources empower the users and enable them to navigate the potential data overwhelm.

Once the goal is set, the next step in the workflow of data practitioners is to explore and understand the datasets. The exploration phase consists of two steps: (*i*) exploring and understanding the base table (i.e., the table that will be augmented with more features), and (*ii*) exploring, understanding and relating the rest of the datasets with the base table. Moreover, we present findings regarding the total time spent on the exploration and the data processing techniques used in the exploration step.

– *Exploring and Understanding the Base Table.* In our observations, we noted that the participants aim to acquire an in-depth understanding of the base table, its columns, and corresponding values. To accomplish this, they employ various methods. These methods include seeking insights from individuals with domain knowledge, relying on their own domain expertise, or applying their intuitive understanding.

> **Collaboration** – The collaboration between data workers and other members of organizations has been a significant focus of research. For instance, studies have shown that the scientific collaboration between biomedical scientists and data scientists can be successful when efforts are made to establish common ground and

shared processing methodologies [131]. In large organizations, where data is spread across various sources, strong collaboration is often required between data workers and other organisational members, such as IT staff who assist in locating and delivering datasets [97], and business personnel who help define goals and requirements [98]. Although our study did not primarily focus on the collaboration between organizational members, we observed that data exploration is inherently a collaborative process. Our findings indicate that when faced with unclear data, data workers commonly ask knowledgeable colleagues and consult with other team members or even different departments for clarification, as noted in 6 out of 19 cases.

> *"For example, I would go back to wherever I got this data set and say, can we spend 20 minutes running through what each of these columns mean?"*(P16)

In our setting, and based on the literature, we did not offer any documentation or information about the datasets. Therefore, the participants had to rely on their intuition and knowledge to try to understand the base table.

Knowledge – Domain knowledge is key in determining which aspects of a problem are relevant and which types of data can predict certain behaviours. A business context often deepens the understanding of what needs to be predicted and identifies the factors that might influence it. The team size also influences the expected knowledge someone has about their datasets (4/19).

> *" I currently work in a relatively small team. We all know the data we're working with, so there's also some domain knowledge just assumed when you're working with these tables."* (P10)

Intuition – The participants discuss the approach of exploring a dataset they are not familiar with, emphasising the use of common sense to navigate unknown data. They mention the importance of intuitively reasoning about the problem to identify correlations between tables or data, especially when aiming to predict specific outcomes. They resort to experimentation and intuition to form hypotheses about the significance of data (4/19).

> *"Kind of use my common sense since I don't have any business logic or business knowledge about this specific dataset to figure out what other features in the other tables might be interesting to try out."* (P1)

One participant would even consider it irresponsible to rely on their own intuition to solve the task without any other context.

> *"It feels a little irresponsible even to push forward without having a lot more context."* (P16)

– *Exploring and Understanding the Collection of Datasets.* Once the participants create a general overview of the base table, they start exploring the rest of the datasets. We noticed a few patterns of exploration.

Browsing – Most of the participants (17/19) start the exploration of the remaining tables by just browsing the data with the aim of understanding the context. The activities related to browsing and understanding the data include: (*i*) exploring all the files to get a general sense of the data, sometimes starting with the first file from the folder (i.e., the top file), regardless of the sorting criteria, and then systematically exploring the rest of the data, (*ii*) loading all data using diverse tools, preferably a database, and in the absence of it, working with pandas[2], (*iii*) examining the file name and using the intuition to understand its meaning, (*iv*) understanding the content of the data, skimming through the tables.

> *"I have no clue what I'm doing because I've never looked at these data, so I'm just opening some tables to check what is inside because otherwise, I have no clue about the data I'm looking at."* (P7)

Assessing the Relevance with the Base Table – During their exploration, the participants tried to relate any other table with the base table. Similarly to the base table exploration step, the participants would first inspect the documentation. The documentation is used as a crucial tool for understanding the contents and relevance of different tables, aiding in the efficient and effective exploration of data. It provides context and clarity, helping them make informed decisions about which data to include in their analysis (4/19).

> *" Usually, we're using documentation on what the table is about. So it's easy to not go to data and see what's in the column, but read the description."* (P13)

In the absence of documentation, the participants would try to seek help from someone with domain or business knowledge (P18).

> *"From an analytics perspective, my approach here would be to try and find someone who can explain to me what these columns mean."* (P18)

The participants rely on a combination of intuition, existing knowledge, and exploratory data analysis to identify potential relationships between datasets, focusing on factors they deem relevant or potentially influential for their analysis. Their activities include identifying potential correlations, assessing the column names, focusing on specific data points, judging data usefulness based on the contents, cross-referencing tables for common values and prioritising data based on assumed importance (14/19).

> *"So what could be interesting is some information on schools and maybe some information on teachers and parents. So maybe* `school progress report` *is interesting."* (P8)

[2] `https://pandas.pydata.org/`

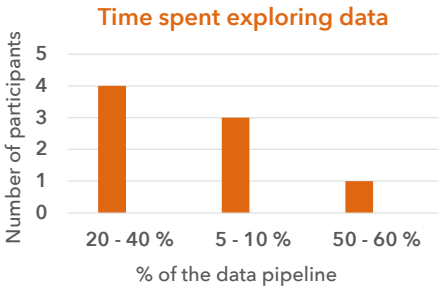


Figure 5.5: The figure illustrates statistics about the time allocation for the data exploration step reported by participants.

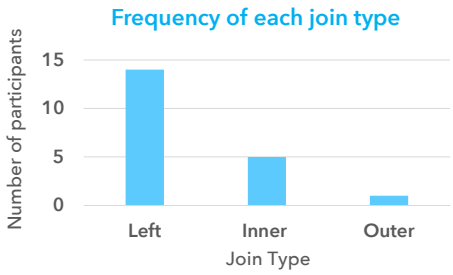


Figure 5.6: The figure illustrates statistics about the frequency of each type of join reported by participants in the data integration step.

During the dataset exploration, automatic processes to compute the relatedness of tables were mentioned a few times (3/19). More specifically, one approach was to find schema matches by comparing the column names in the CSV files (P18). Additionally, examining the correlation between the features could help determine the relationship between the features (P2, P7).

– *Exploration Time.* When questioned about the amount of time typically dedicated to data exploration *"How much time do you usually spend on the exploration?"*, 9 out of 19 participants indicated that data exploration often occupies a substantial part of the timeline of a project. These results are summarised in Figure 5.5. A significant number of responses fell within the 20-40% time range for exploration, followed by 5-10% for datasets that are familiar, and 50-60% time allocation for exploration when working with new datasets or domains.

Data exploration is a collaborative and intuitive process – users will always rely on the domain and business knowledge of data owners or use their own knowledge and intuition.

5.5.3 Data Integration

We observed a blended transition between the exploration and integration steps. While exploring the data, the participants instinctively searched for columns to relate to and joined the tables.

The insights from our study, particularly regarding the data integration step, resonate with the established findings in the field. Mirroring the methods outlined in COCOA [53], a few participants used automatic techniques such as computing feature correlations to determine the relevance of different tables. The participants' approach to joining all available tables or just the *"most helpful"* ones echoes the findings of [103, 112]. Furthermore, consistent with the practices documented in the literature, our participants typically work with primary key-foreign key relationships [58, 112]. Before joining tables, our participants also used data aggregation, a step that aligns with the workflow proposed in ARDA [36].

– *Joining Datasets.* Lacking descriptive documentation for dataset columns, the participants rely on their intuition and experience to identify candidate columns for join keys (16/19). Their approach involves:

- conducting statistical analysis to verify the uniqueness of the columns (9/16);

  > *"I'm looking to make sure that this DBN code is actually the unique identifier."* (P16)

- assessing the reliability and relevance for joining (6/11);

  > *"I need a way to connect these tables together, and I also need to make sure that I don't join things that are not joinable."* (P4)

- identifying specific column names which indicate the presence of a relationship (6/16);

  > *"If the column name is the same and the format in this case seems like a very specific format, so there's a very high likelihood that this is actually the join key. I might be wrong, but at first glance, that seems to be pretty straightforward."* (P1)

Our datasets did not contain explicit PK-FK relationships but a set of disconnected datasets – we left PK-FK relationships to be discovered by the participants. However, we asked our participants about their day-to-day analysis, such as joining PK-FK relations. We found that primary key and foreign key connections are the main ways for almost half of the participants to join datasets.

> *"I'm just saying that the primary key and the foreign key if you have it, it's good because it's kind of an easier way to explore and try to add the datasets together. If you don't have it, then a whole conversation starts about how you create a custom key for your datasets that can be used in both tables."* (P17)

Participants reported that joining on primary keys is common, but in some cases, fuzzy matching or other methods may be used. Most data from relational databases have primary keys, but there are exceptions, such as stand-alone tables in big data scenarios. In data lakes, primary keys are not explicitly stated or enforced, so validation is important.

> *"If I'm assuming that a primary key or a key is unique, I would make sure to validate that because otherwise, you can generate duplicates."* (P18)

– *Join Type.* The process of integrating the tables involves joining the tables. According to the literature, the feature discovery and data augmentation processes involve performing a *left* join to assure that the number of rows is preserved [36, 53, 129]. Figure 5.6 contrasts the theoretical approach described in the literature with the actual workflow of participants while integrating tables.

Left Join – The most popular join type is *the left join* (14/19) to ensure the completeness of the base dataset, allowing for the addition of data (e.g. features) from other tables without losing any rows from the base table.

> *"Because I'm assuming we only wanna keep the base table. So, if there are extra rows in the feature tables or related tables, I don't want those rows because they don't have a class value, right? I can't use them to train the model."* (P14)

> *"Left join ensures that we keep all of the rows from the base table, which is our base data set. So, because we want to enhance the data there. We don't want to eliminate any schools from the data set."* (P10)

Inner Join – The second most recorded answer is *the inner join* (4/19) because they want to *"make sure my data has as few new values as possible"* (P1), or *"to just keep the common data"* (P11) and even *"to see if there is any data left"* (P19).

Outer Join – The least recorded answer is *the outer join* (P4). The specific participant reported that they wanted to retrieve all the features.

– *Data Integration Pipeline.* The literature on feature discovery and data augmentation techniques takes different approaches to joining plans or pipelines. Some works choose to join all tables up to a budget, then apply feature selection [36], while others have an iterative process of joining one table at a time and testing its usefulness [129]. We have observed a similar pattern among our participants.

Join All Tables – The strategy of joining all tables at once is driven by the desire for comprehensive data analysis (4/19), simplifying the initial data processing stages (6/19), and leveraging machine learning algorithms' feature selection capabilities (3/19).

> *"I would say quick and dirty is literally add everything. And since we're using decision trees, decision trees have this inherent property of doing feature selection. So we could say, we trust that the decision tree picks the right features and just go with that, but in most cases, that doesn't really work."* (P1)

The methodology involves using a common key to merge tables, followed by data cleaning, aggregation, and iterative model training and enhancement (4/19). This approach allows for a thorough exploration of the data, uncovering potential insights that might not be apparent when analyzing tables in isolation (3/19) - *"If you join everything together first, then you always have the freedom to look at all of the columns in the context of each other"* (P10).

Join Tables One by One – This step-by-step process allows for careful examination of the impact of each table on the overall dataset (5/19). Most participants (4 out of the 5 that discussed this process) indicated that it is particularly beneficial in

complex or large datasets where an incremental approach can provide more clear insights than a bulk join. By joining tables sequentially, they can pinpoint which datasets enhance the accuracy and predictive power of the ML model (3 out of 5 participants), and they can also observe how each new data source contributes positively to the analysis (3 out of 5 participants).

> *"So I assume that I already have a baseline. I add the new feature, I train the model and I see if it improves the baseline, right? And then it improves. But how much? And then I ask myself like is it worth the effort?"* (P9)

Data integration process aligns with literature – Our findings regarding the integration step are perfectly aligned with state-of-the-art literature on data integration, proving that this step in the data pipeline captures the practices and habits of users in real life.

5.5.4 Feature Selection

Subsequent to the integration step is feature selection. Initially, the augmented dataset undergoes processing, coupled with the application of feature engineering techniques. Following this, diverse techniques of feature selection are applied.

The literature suggests that data professionals struggle with the feature selection process, as the data volume is too high [112]. In our study, we observed that while exploring the dataset, or after the feature engineering step, a few participants (6/19) discussed manually selecting specific columns relevant for the base table, and thus creating the augmented table for evaluation - *"So for this table, I'd only keep* `borough` *and* `enrollment` *[columns]."* (P18). Moreover, data professionals often have an iterative process, ensuring that each feature included contributes positively to the model's performance and overall accuracy - *"For each table I do that [i.e., cost-benefit analysis] and seeing how much it improves."* (P9).

The approach to automatic feature selection and model training is iterative and data-driven. Some participants rely on using the ML models to assess feature importance - *"Initially I would just throw a random forest or a boosted tree at it because it comes for free with feature selection."* (P1), followed by careful analysis and pruning of features. Overall, the process involves balancing automated feature selection methods, manual analysis, and continuous testing to achieve an optimal set of features for the ML model.

Feature selection is rarely decoupled from the ML model – Feature selection and ML modelling are rarely treated separately. After thoroughly manipulating and engineering the dataset, users often rely on the ML model to select the best features.

5.5.5 Dataset Evaluation

The dataset evaluation and model training step are reported a few times alongside processes such as hyper-parameter tuning. The participants report an iterative process, starting from a simple model. When the join pipeline implies joining the table one by one, the participants mention that *"If I [i.e., the model] start degrading, I would stop pretty much."* (P15).

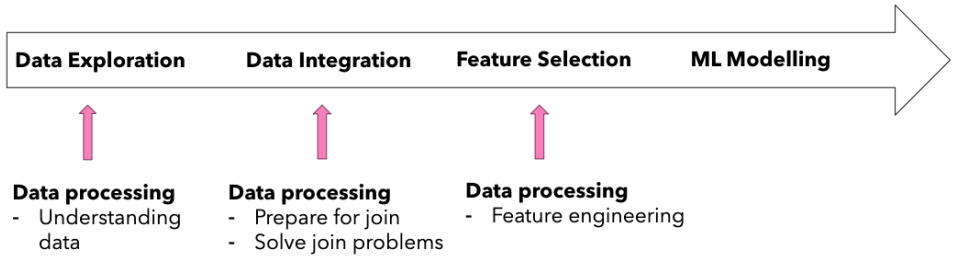


Figure 5.7: Our study reveals that data processing is an integral part of the workflow instead of a single step in the pipeline.

The dataset evaluation is an implicit process that happens at the end of the feature discovery pipeline. The goal of the use case scenario, presented to every participant, was to find relevant features for augmentation such that the accuracy of a tree-based ML model increases. It is worth mentioning that some participants do not engage in the machine learning process in their daily workflow, thus this step has been omitted during the interview.

5.5.6 Data Processing

Our observations reveal that data processing is an iterative and integral part of the entire workflow rather than a one-time step as presented in literature [15, 27]. Data processing is consistently revisited and initiated at any point in the pipeline throughout the various stages of the participants' workflow. Figure 5.7 summarises our findings.

We observed participants engaging in data processing early during the exploration phase, which aids in a deeper understanding of the datasets. They use data quality metrics such as the number of unique or missing values. They also look into the data distribution and apply appropriate data balancing techniques based on the distribution. They use data cleaning and standardisation approaches, sampling, and pay close attention to the data types. These techniques and approaches reflect a comprehensive methodology for processing and analyzing data, emphasising the importance of cleanliness, clarity, and understanding of the dataset for effective analysis.

Data processing is also present during the integration phase, where it serves the dual purpose of preparing the data for effective joining (e.g., data cleaning and aggregating) and tackling any issues that emerge as a result of the integration. Any type of join can create diverse problems in the data, especially when the joins are not one-to-one. We categorised the issues encountered by our participants (13/19), which can arise during or after performing joins in data analysis, into a few key areas.

> **Data Integrity and Quality Concerns** – The participants report that careful consideration is required on how to aggregate or group data to maintain meaningful relationships, as a left join can introduce *null* values and even duplicate data when the relation is one-to-many causing data redundancy and potential skewing of analysis (9/19). This has also been reported in data augmentation research papers [36, 53, 129]. Moreover, multiple (5/19) participants emphasised the importance of

always checking the join result, as joining on columns that are not consistent (e.g., case sensitivity issues) or not unique identifiers will lead to incorrect joins and missing values.

> *"The more features you want to add to your table, the more rows you need to have. If it's a very simple [ML] problem with just a few features, I would be comfortable having fewer rows, but the wider the dataset becomes, I would argue the more examples [rows] you need."* (P1)

Computational Limitations and Scalability Issues – One participant reports that in situations where large datasets are processed, using database systems or chunking methods is a more efficient data processing approach (P6). Join operations, especially in big data contexts, can be computationally expensive and time-consuming, impacting resources and potentially incurring costs (3/19).

> *"I would look through the tables and join each one on the base table. Yes, I think this is what I would do at first, assuming this data is small enough, and my computer memory would allow me to do this very data-sciency thing."* (P6)

Moreover, data processing is linked with the feature selection process. Here, it is part of feature engineering, where the quality and relevance of features are enhanced and tailored to meet the specific requirements of the analysis. In what follows, we describe the techniques mentioned by 14 out of 19 participants, ordered by their frequency of occurrence during the interviews.

Duplicate Data: Drop vs Aggregate – The most occurring techniques involve either removing duplicated columns or data or aggregating them meaningfully, depending on their impact on the analysis.

> *"Definitely getting rid of of duplicates would be either grouping and aggregating somehow, or just dropping the ones that are so often. Also, if I see the duplicates are like 1% of the data, I'll just drop them. I don't go through the hassle of understanding why there are duplicates."* (P18)

Null Values: Drop vs Impute – The presence of null values requires assessing their significance in each feature. Next, one can decide to retain or drop features depending on the percentage of nulls and their relevance to the use case. Three out of four participants consider the removal of features with excessively high null values (e.g., over 60-70%) unless the absence of data itself carries meaningful information, while another participant considers imputation methods such as using the median or mean.

> *"It's also not the end of the world to remove a couple of rows in order to get your data clean."* (P14)

Drop the ID column Identifying and removing identifier columns (e.g., IDs) before the final training step of the model is crucial, as the IDs are typically not informative for the ML model and can lead to issues such as overfitting.

Normalisation and Encoding The normalization and encoding techniques should be tailored to the characteristics of the data and the requirements of the specific machine learning models being used - *"Ideally I also want to know what values we're dealing with and what's the appropriate way to encode the columns."* (P2). Text columns receive special consideration, as some decide to encode them, while others decide to remove them - *"So most of the string data types [features] I would not consider"* (P19).

The process extends beyond feature selection to ensure that the final dataset has a high quality for the machine learning algorithms. This multifaceted approach to data processing highlights its significance as a dynamic, adaptive process in a data pipeline.

Data processing is an iterative process and not a single step in the pipeline – Our participants used data processing (e.g., data manipulation, data engineering) at every step of their pipeline.

5.6 Inside the Utility Drawers

During the interview, we collected data regarding the characteristics of datasets known to the participants, as referenced in Section 5.6.1, through a series of structured questions and think-aloud sessions. Additionally, information was gathered about the tools employed in their workflow, as detailed in Section 5.6.2.

5.6.1 Dataset Characteristics

Prominent dataset characteristics observed include the type of input commonly managed by the participants, the dimensions of their datasets, and details regarding the naming conventions they employ.

– *Input Type.* Our use case scenario contained 17 CSV files. When asked *"Do you usually work with CSV files?"*, 16 participants responded to the question. We ordered the answers by their frequency (i.e., the number of participants who mentioned the specific input type).

Databases – The participants usually work with data from databases and export it using SQL into a CSV format - *"Often data is in databases like data lake, data warehouse."* (P9).

Parquet – Four participants mentioned *Parquet* files - *"It's quicker to load, but it also keeps the dtypes, the correct types that you've defined before saving it as a parquet file. So that's why I really like it and especially when you're dealing with big files."* (P4).

CSV – The CSV file type was mentioned mostly as a common type to export data from the databases - *"I would have to work with CSV exports from other people's databases that I would then load in pandas or something like this, but a lot of the times we worked directly with databases."* (P1)

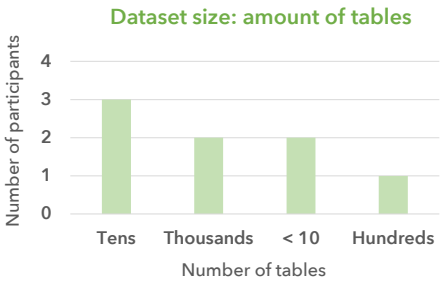


Figure 5.8: The figure illustrates statistics about the dimensions of the datasets: range and frequency of the number of tables reported by participants.

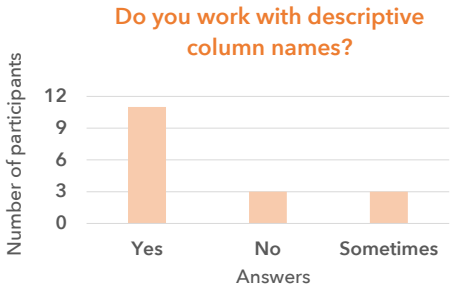


Figure 5.9: The figure illustrates the distribution of answers about column names in datasets reported by participants.

Others – Besides the three input types described, participants also mention working with JSON files - *"we've been working a lot with JSON on the current role."* (P6), or *"sometimes we use a Google Sheets."* (P7), and tabular data *"in a data lake in Amazon S3, in AWS World or tables and views and workload in Snowflake."* (P8).

5

– *Dataset Size.* The majority of the participants provided information about the size of the data they typically handle, specifically the number of tables involved. These details are summarised in Figure 5.8. Briefly, it was found that the most common range for the number of tables was in the tens, with the highest reported number being 40 tables.

– *Naming Conventions.* The participants wished to have more context about the data they were exploring. We noticed that in the absence of dataset context, they attempted to deduce it based on these premises: file names and column names.

File Names. During the exploration step, we observed that the participants used the file names to create a context for the table and used their intuition and knowledge to understand the content of the table. One of the approaches was *"going for the more meaningful named files."* (P18). However, in general, the reaction we observed resonated with P10, who said *"the naming of the tables is also not 100% clear, at least not for someone who's not very familiar with the context."*.

Column Names. When asked about the column names they usually encounter, the participants answered with *Yes, No, Sometimes*, as illustrated in Figure 5.9.

> *"There is a review process because if you want the data to be used by others, you try to put yourself in their shoes."* (P9)

A common practice is to clean the data and use appropriate naming conventions.

> *"You would immediately clean up the column name so you would have no spaces in column names; everything would be lowercase, just to make it easy for anyone else to work with. In a work setting, we're not thinking about answering one question, but knowing that we're gonna have to answer all of these questions again and again."* (P16)

Table 5.2: Overview of the tools **used** by the participants in their workflow during the use case. *Category* represents our own organisation of the tools based on their similarities, and *Count* represents the number of participants using each tool.

Category		Name	Count
Programming Language		Python	9
		SQL	5
Environment	**Environments**	Python Env	5
		Conda	2
	Package Management	Requirements.txt	1
		Poetry	1
	Python Package	Pandas	11
		Dusk	1
		Numpy	2
		Great Expectations	1
Analysis Frameworks	**IDE**	VS Code	10
		PyCharm	2
	Notebooks	Jupyter	10
		Google Colab	1
		Hex	1
File Exploration		File System (Linux, MacOS)	2
		Sheets (Excel, Google)	2
		Terminal	2
Other Tools		GitHub Copilot	2
		ChatGPT	1
		Spark	4
		Databricks	2
		DuckDB	2
		BigQuery	1
		DBT	1

5.6.2 Tools to Support the Pipeline

During the execution of the entire use case scenario, the participants employed a diverse array of tools. These tools frequently occur in conversation as participants engage in a think-aloud process, providing insights into their tool preferences and usage. We documented these tools and have compiled a comprehensive overview, which can be found in Table 5.2. This table is a culmination of our observations coupled with discussions held with 18 of the 19 participants, offering a detailed perspective on the tools used throughout the interview.

Programming Language – The most popular programming language in our interview was `Python` (9 participants). Although many participants mentioned that *"I'm most familiar with Python"* (P12), some of them (5 participants) *"[I] prefer to interact with data in SQL"* (P16). Moreover, the data shows that even though some participants prefer `Python`, they also know `SQL` and use both of them interchangeably depending on the situation - *"I learnt SQL before I learnt pandas"* (P3).

Environment – Most of the participants who mentioned `Python` as their preferred programming language also reported that *"For the ease of analysis, I would start by defining an environment"* (P1). The reported environment management systems are `conda` and the default `python venv`. Some of the `Python` packages installed and used were `pandas, dusk, numpy` and `Great Expectations` Python package for testing.

Analysis Frameworks – Once the environments are defined, the participants analyse the data in an Integrated Development Environment (IDE) such as `Visual Studio Code` or `PyCharm` - *"I pretty much never work with notebooks."* (P19). However, 12 out of 19 participants used notebooks *" [...] to do some quick exploration of some data"* (P3) or *"[...] proof of concepts"* (P5).

> *"My workflow is first up with Jupyter lab, and then I would switch over to Visual Studio Code and put a lot of these Jupyter Notebook codes into Python files."* (P4)

File Exploration – Some participants used the IDEs even to browse the CSV files and installed specific IDE extensions to help them visualise the CSV files. Besides the IDEs, the participants used different tools to browse through the files, such as the file system, sheets (Excel, Google) and even the terminal - *"For me, it's easier [to use the terminal], but you can also open it with Excel."* (P9).

Other Tools – The participants reported the usage of multiple tools during their day-to-day workflow such as AI tools (e.g., ChatGPT, GitHub Copilot) - *"I use copilot every day"* (P19), databases (e.g. DuckDB), and frameworks to help them analyse the data (e.g. Spark, BigQuery, DBT, Databricks) - *"Normally I will do this [data processing] into BigQuery."* (P7).

5.7 Daily Challenges & Wishes

In this section, we describe the unique aspects of the data-centric challenges faced by the participants. We highlight the challenges that arise from handling vast quantities of data (Section 5.7.1), present the more mainstream problems encountered by data professionals (Section 5.7.2), and report the optimal set of tools and methodologies that, if implemented, could streamline the feature discovery process, making it more efficient and effective (Section 5.7.3).

5.7.1 Big Data Scenario

> *"That's going to be a very long and arduous task, and I might be hating myself for choosing to do so."* (P3)

We challenged the participants and asked *"Would you follow the same workflow if the dataset had one thousand tables?"* with the aim of understanding what the process would be in a big data scenario. The answers were very diverse.

Exploring Data Manually – Although the task would take significantly more time, six participants would still rely on a manual exploration of the tables, browsing

through the tables *"I would still browse through a few tables and try to figure out if there's a common structure to the tables."* (P10), and filtering the tables - *"I would hope that the names of the tables made sense, and if they made sense, I would select the ones that made the most sense first and try to go with that."* (P4)

Automating the Exploration Process – *"1000 tables. I haven't been in such an environment before where I had to do that, so I assume doing that by hand wouldn't be feasible."* (P12). Therefore, the participants discussed methods of automating the process, which involved extracting the column names and computing their overlap and frequency of occurrence.

Other Strategies – Among other strategies for feature discovery in a big data scenario are: relying on the documentation, asking for help from domain experts, relying on their intuition or simply not solving the problem - *"I'm even struggling with this amount of tables to be honest, because again I don't know the domain."* (P8).

5.7.2 Daily Data Problems

We aimed to uncover the specific issues that participants regularly face in their daily data pipeline management. Through our interview, they disclosed a range of problems from several key areas: data quality, the sheer volume of data, consistency in data formats and structures, and the availability of data for timely access and analysis. Furthermore, they highlighted issues stemming from the systems they use, which often involve complex processes that add layers of difficulty to their daily data-handling tasks.

Data Quality – Six participants reported data quality as being one of their biggest issues. Data quality issues can be an inherent issue or generated by the source of the data - *"It's based on real life, so it has inherent messiness"* (P4). Other quality problems involve data imbalance and even the cleaning process by itself.

Data Volume – Four participants report issues associated with a large volume of data such as synchronisation issues - *"Millions and millions of rows that are being logged."* (P5), and scalability issues - *"volume is a big consideration that comes to mind whenever you're trying to do any kind of intelligent processing. "* (P17).

Data Consistency and Availability – Three participants report issues with data consistency, which originate from using different notations or formats (i.e., specifically timestamps) - *"Especially as organizations grow, it's hard to get everybody to kind of follow the same [formats]."* (P18), and with the data availability, an issue prominent in the data integration step - *"you might end up with too few data points in the end."* (P10).

Systems Issues – Four participants describe the issues produced by systems and complex processes, such as misconfigurations in deployment and improper testing leading to a chain of errors - *"Everything is breaking in the domino effect."* (P7), and lack of optimisations, especially at the query level.

Documentation – The lack of documentation and data catalogues which describe the data types, formats, and relationships among the tables is another issue encountered in the daily workflow - *" In my experience, very few companies maintain solid*

data catalogue that gives you some sort of description for columns" (P1), *"It requires sort of a culture shift within companies."* (P11).

5.7.3 Ideal Tools & Workflow

"I wish people would put more effort into documenting things." (P9)

During our follow-up questions, the participants were asked to describe the tools they envisaged as being most effective in lightening their workload within the context of our specific use case scenario. Following, we present the tools reported by 17 participants.

Tools for Analysis and Visualisations The participants wish they had a tool to automate the analysis of the tables such as `Spark`, and tools which can perform *"some basic aggregations on the data that tells maybe a more complete story than just looking at the columns"* (P3), *"make an overview of what kind of columns or features you're dealing with, so you get basic statistics like min, max, median, mean, sort of distribution"* (P2), or *"something that runs in the background to pre-compute some of these values, and then would automatically suggest that these [features] can be removed"* (P19). Visualisation tools are also among the tools which help the participants to better understand the data - *"if these [relationships between tables] were somehow visualised the way that an entity relationship diagram is visualised"* (P18).

Documentation A reoccurring topic during the interviews has been the documentation of datasets and columns (10/15). The participants acknowledge the benefit of having documentation or data catalogues with descriptions about each dataset and columns - *"I think one of the most powerful things you can do is kind of keep data in context together as much as you can."* (P16). They mention tools such as `Amundsen`, `Collibra` as being good candidates for documenting the datasets, or even simpler tools such as `Google Docs, Wikipedia` - *"Like [a] Wikipedia [page]: if you have that for a specific set of datasets, and then you can also explore them fast within a web UI, that will be great."* (P17).

Database Tools The participants report that working with databases instead of CSV files would have been easier. They also mention existing tools and applications that would have been useful in our use case scenario, such as `DBeaver`, which is *"[...] a nice application where I can connect to the database and run SQL queries."* (P5), `DBT` and `SQLAlchemy`.

Regarding the workflow, participants expressed that a better understanding of the context and domain knowledge could significantly help in task resolution. Additionally, they emphasised the value of improved collaboration between teams responsible for processing and distributing data, as this would greatly simplify their daily workflow. Lastly, the introduction of automated tools for data processing was highlighted as a key factor that could considerably ease their workload, with one participant notably stating *"[this] would make my life a lot easier."* (P4).

When asked if they would rely on an automated tool for feature discovery and data augmentation, 15 out of 19 participants answered this question, and the answers were

quite diverse. We had three categorical negative answers and two straightforward positive answers. The rest of the participants would use and trust an automatic tool if and only if they would be able to: (*i*) interfere with the process to verify the results and control different steps by adding their input, and (*ii*) have access to insights into the *modus operandi* – the tool should generate easy to understand and to verify explanations.

5.8 Discussion

In this section, we discuss our study findings.

5.8.1 Documentation is the Source of Truth

The theme of documentation, or the lack of documentation, has been a recurring and significant topic throughout our interviews. Participants frequently highlighted their reliance on documentation for various tasks, such as exploration and understanding. They viewed documentation as a foundational source of truth, essential for grasping notations, definitions, data formats, and relationships between tables.

> *"Usually, we're using the documentation on what the table is about. So it's easy not to go to data and see what's in the column but to read the description."* (P13)

The literature, however, focuses primarily on creating automated frameworks for documenting the code and computational data science notebooks [181, 182], on automating documentation to improve the reproducibility of experiments [164] without considering the importance of documenting datasets.

Our study suggests that documentation and data catalogues are crucial in understanding and effectively working with datasets. A data catalogue with detailed descriptions of tables and a dictionary or index for clarification is highly beneficial, aiding in the interpretation and utilization of data. The value of having clear, detailed documentation is underscored for both understanding the context of the data and meeting specific requirements in data processing and presentation to the final customer (9/19).

> *"Unless I'm coming into a totally green field data team, there's always some existing data sets and some level of documentation around the data."* (P16)

However, our participants acknowledged that lacking comprehensive and up-to-date documentation of datasets is a common issue in many organizations. This gap often hinders efficient data management and understanding, leading to data usage and interpretation challenges.

This reliance on documentation stresses its vital role in any data pipeline. Well-maintained documentation can enhance the efficiency and accuracy of data-related tasks, serving as a guide and reference for the entire data management lifecycle. Steps towards making documentation more accessible have emerged, such as automatic approaches for data versioning with explanations [176]. The discussion leads to a broader need for more robust, accessible, and regularly updated documentation practices, suggesting that enhancing documentation quality and availability could significantly improve data management processes across various organizations.

5.8.2 Differences in Workflow Based on Role

During our interviews, we observed subtle variances in the participants' workflows. We are aware that this might be influenced by our biases regarding the structure of the pipeline, which we defined prior to the interview (i.e., data exploration, integration, feature selection, and machine learning modelling). We noticed that those with roles such as data engineers and analysts allocate a greater portion of their time to achieve a high degree of understanding of the datasets compared to data science and machine learning engineering participants. This group (i.e., data engineers and analysts) also exhibited a stronger proficiency in SQL over Python. On the other hand, data scientists and machine learning engineers demonstrated a more direct approach to feature discovery. They prioritise establishing the working environment and setting a benchmark for machine learning model accuracy early in the process.

5.9 Conclusion

Our qualitative study with 19 data practitioners provides invaluable insights into the real-world challenges and workflows of feature discovery. With this chapter, we answer our two research sub-questions, which support our **RQ3**:

How do data scientists and engineers in the real world perform feature discovery when asked to train an ML model from tabular data residing in a data lake?

Does the real-life process align with the theoretical one reported in the literature?

Our research indicates that data practitioners (e.g., data scientists, data engineers, data analysts, and ML engineers) depend on their specialised business and domain expertise, alongside their intuitive understanding, to tackle the challenges of feature discovery. Additionally, while some recourse to automated methods and tools to facilitate this process, a significant amount of work is still conducted manually. This finding underscores the importance of embedding user experiences and specific needs into developing and enhancing feature discovery methodologies. Doing so will ensure that these techniques are grounded in the practical realities faced by users and are tailor-made to enhance their efficacy and efficiency in navigating the complex landscape of feature discovery.

Furthermore, we found that the pipeline followed by our participants diverges partially from the theoretical pipeline presented in the literature. While the data integration step aligns closely with existing literature, the other steps are misrepresented or not fully captured. This discrepancy highlights the need for researchers to develop more intuitive and effective data management strategies that better address real-world challenges and workflows faced by data professionals.

In the next chapter, we integrate users directly into the feature discovery pipeline by developing a human-in-the-loop feature discovery approach. This approach will not only involve users in the iterative cycles of feature discovery but also seek to leverage their unique insights and expertise to refine and enhance the process. Following the design and implementation of this strategy, we will conduct an evaluation to assess its effectiveness and impact. This evaluation will serve as the basis for addressing our third research question, ultimately contributing to a deeper understanding of how human involvement can optimise feature discovery methodologies.

6

Feature Discovery: a Human-in-the-Loop Approach

Data scientists and engineers use automated feature discovery over tabular datasets to add new features from different datasets and databases and enrich training data. By surveying data practitioners, we have observed that automated feature discovery approaches do not allow data scientists to use their domain knowledge during the feature discovery process. In this chapter, we introduce the first user-driven human-in-the-loop feature discovery method called `HILAutoFeat`*, which effectively combines automated feature discovery with user-driven insights. The code of our tool is open source at* https://github.com/delftdata/hci-auto-feat

This chapter is based on the following demonstration paper and open-source resources:

- *Andra Ionescu, Zeger Mouw, Efthimia Aivaloglou, Rihan Hai, and Asterios Katsifodimos. "Human-in-the-Loop Feature Discovery for Tabular Data". CIKM 2024 [91]*
- *Source-code* [89] *and data* [84]

6.1 Introduction

The long-standing presumption that the training data for an ML model is a single table does not hold true. In practice, essential predictive features often reside across multiple database tables or files, which could be part of an extensive open data repository or a data lake [70, 144, 146]. Currently, there is significant ongoing research dedicated to developing methods that automate the discovery and augmentation of tabular features for ML model training [36, 53, 129, 205]. This process, named *feature discovery*, builds upon the exploration and integration steps from dataset discovery [17, 53, 58, 112] and relies on feature selection strategies to select only the most relevant features for a given ML task.

In data lakes lacking primary-key/foreign-key constraints, employing dataset discovery [56, 106] is an essential first step for feature discovery, which reveals table relationships [36, 93] such as joinability [17, 112] or unionability [56, 146]. However, dataset discovery approaches often produce false positives [106], leading to joins that yield irrelevant tables filled with unrelated data. The issue is exacerbated when two datasets are joinable through multiple columns, a scenario where state-of-the-art feature discovery approaches typically fall short [36, 129].

The automated systems for feature discovery for ML either focus on strategies for improving the correlation metrics [53], maximising relevance while minimising redundancy [93], or integrating the ML model directly into the augmentation process to ensure feature compatibility [36, 129]. However, our recent user study [90] (Chapter 5) has revealed two critical issues with fully automated approaches. While data scientists find the automated feature discovery methods to be very useful for finding relevant features to train their ML models, they also lose control over which features are included in the training data for a given model. This issue is two-fold. First, fully automated feature discovery methods do not leverage the user's domain expertise, which can be pivotal in discovering important predictive features. Second, the fully automated methods can incorporate features that should not be part of the training data due to government regulations and company policies (e.g., privacy, bias).

Example: *Take as an example a dataset for predicting if an employee is suitable for promotion. In this dataset, the ML model has access to features such as education, years of experience, technical skills, and soft skills. Augmenting this dataset with personal information about the employees, such as age, gender, and nationality, and given the high amount of male employees the company already has, an algorithm trained with the gender feature can be biased to generate a favourable decision for male employees. Here, human input is crucial in determining the correct set of relevant and related features for augmentation.*

Therefore, we ask our third research question:

> ***RQ3****: Can human expertise and domain knowledge enhance the automatic feature discovery process?*

Departing from the black-box automated approaches, to answer **RQ3**, we have extended our automated tool `AutoFeat` [93] (Chapter 4) to incorporate user feedback and involvement during the feature discovery process. With this human-in-the-loop approach, named `HILAutoFeat`, we address the reported issues from our user study (Chapter 5). `HILAutoFeat` leverages the strengths of automated feature discovery methods while providing a platform for data scientists to use their domain expertise and business knowledge.

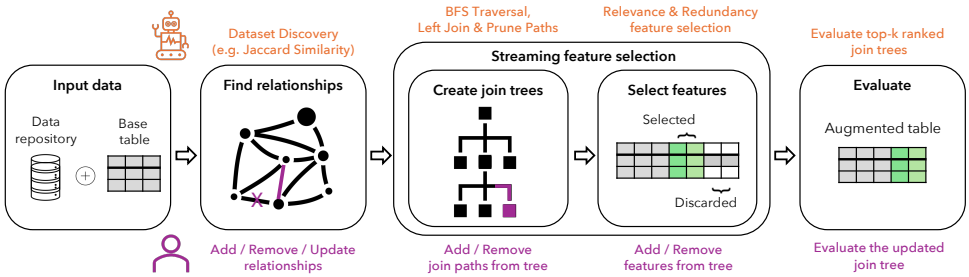


Figure 6.1: HILAutoFeat pipeline: automatic workflow and user-driven workflow.

HILAutoFeat allows users to control essential steps: users can filter the discovered relationships and the join paths and adjust the selected features while observing the effects of these updates over the augmented dataset in real time. To the best of our knowledge, HILAutoFeat is the first user-driven, semi-automated feature discovery tool that dynamically adjusts to user feedback. Specifically designed for data scientists and analysts across various fields, users reported that this tool provides a more efficient augmentation process and yields effective results through the added benefit of modifying the data (i.e., relationships, join trees, features) at any given point.

6.2 System Overview

We have developed a user-driven human-in-the-loop feature discovery approach, HILAutoFeat, which extends our automated feature discovery system, AutoFeat [93]. The primary objective of feature discovery is to enhance a base table by adding new features that significantly increase the predictive accuracy of a target ML model. HILAutoFeat streamlines the process of selecting and integrating relevant tables from a dataset collection into the base table, based on the user's input, whose domain expertise can potentially change the outcome of the augmentation process. Additionally, HILAutoFeat employs heuristic-based feature selection strategies to eliminate redundant or irrelevant features from this augmented table. By doing so, HILAutoFeat notably enhances the efficiency and accuracy of subsequent ML operations.

Figure 6.1 illustrates the automated feature discovery process and the user interactions which are available at every stage in the pipeline. Our human-in-the-loop feature discovery approach provides the following functionalities to the user: (1) refining dataset relationship (Section 6.2.1), (2) manipulating join trees (Section 6.2.2), and (3) refining feature sets (Section 6.2.3). We also discuss how HILAutoFeat maintains the high efficiency of AutoFeat through various scalability enhancements in Section 6.2.4.

6.2.1 Refine Dataset Relationships

In a data lake with hundreds or thousands of tables, the number of relationships between these tables for a fully connected graph is $n*(n-1)/2$, where n is the number of vertices. For a multi-graph, the number can be much higher. HILAutoFeat maps the relationships between tables using similarity scores generated by a dataset discovery algorithm. Cur-

rently, we support our schema matching tool suite Valentine [106]. With the automated feature discovery process, spurious relations are eliminated. However, the remaining relationships are not guaranteed to be relevant to augment the base table. `HILAutoFeat` enables the users to adjust the relationships discovered by the automatic process. The users possess domain knowledge [90] and can immediately recognise which tables are beneficial for the augmentation.

By default, `HILAutoFeat` displays a graph with the strongest relationships between tables. Then, it enables the user to adjust the similarity threshold, visualise all relationships, and refine them (i.e., delete, update, or add an additional one). These functionalities enhance user control over the process and allow for a more tailored and precise dataset construction, accommodating specific analytical needs and objectives.

6.2.2 Manipulate Join Trees

After establishing the relationships, the next step in the pipeline is streaming feature selection. In this feature selection approach, the features follow a streaming process: a new batch of features arrives with every new join. The automatic feature discovery approach computes two steps in the same streaming feature selection iteration: creates join trees and selects the features. For `HILAutoFeat`, we deconstruct this process such that the user can update the join trees and the subsequent feature sets and actively observe the impact of each join and feature on the performance of the ML model.

The join trees are an early representation of the augmented table, as each node in the tree represents a table with the associated join column. We use Breadth First Search (BFS) traversal to navigate through the graph of table relationships. With BFS, we first join directly connected datasets and then proceed to join datasets that are farther away. This order of joining is crucial because it allows us to prioritise the most relevant datasets in the early stages of the traversal. Through the BFS traversal, we form join paths of varying lengths by sequentially *left*-joining the tables. The choice of a *left* join is strategic, aiming to preserve the original number of tuples and, more importantly, to maintain the number and distribution of classes in the target variable. We discuss other traversal and join strategy options in Section 4.4.

We refine the join trees by pruning any spurious paths. We employ similarity-based pruning – where the join column with the highest similarity score is selected, and data quality-based pruning – which involves discarding join paths that surpass a pre-defined threshold for a non-null value ratio. Each join tree is then ranked by a linear function derived from two distinct feature selection methods measuring relevance and redundancy. In Section 4.5, we provide an in-depth analysis of our feature selection methods.

With `HILAutoFeat`, we open the black box automatic approach and empower users by giving them control over and insights into the process. Users can actively influence the augmentation process by adding or removing paths from a join tree, acting as an external knowledge source for the algorithm. For example, the user applies his domain knowledge and removes a path (i.e., table) from the join tree, which had the potential to bias the algorithm. Additionally, `HILAutoFeat` is enhanced with an explainability function. At each algorithm step, `HILAutoFeat` provides users with a comprehensive understanding of the process. This transparent approach fosters a deeper understanding and trust, empowering users to make informed decisions while fine-tuning their data.

6.2.3 Refining Feature Sets

At the heart of HILAutoFeat lies the balance between relevance and redundancy, which is crucial to its effectiveness. In the context of ML, relevance is divided into two categories: strong relevance, where removing a key feature negatively affects the optimal set of features, and weak relevance, where less important features impact the output upon removal. Redundant features, on the other hand, are those that offer no new information and can be interpreted as a duplication of relevant features. For HILAutoFeat, we apply the Spearman correlation to assess feature importance, while the MRMR metric is used to identify and manage redundancy, ensuring the model operates efficiently and effectively [93].

The most granular operations the user can make are at the feature level. They can view the collection of selected and discarded features associated with a join tree and make updates by either adding or removing features. At this stage, users can leverage their domain expertise or business knowledge to prioritise more critical features. By modifying the feature set, the users not only alter the augmented dataset but also significantly impact the accuracy of the ML model. Refining the feature set introduces a higher degree of customization and precision to the feature discovery process. It allows for a dynamic interplay between automated feature selection and human judgement, ensuring the final dataset is rich in relevant features and aligned with specific analytical goals and business objectives.

6.2.4 Scalability

In our approach, we rely on external dataset discovery techniques for the initial computation of table relationships, a phase that constitutes the most extensive duration within the process. Nevertheless, it is noteworthy that dataset discovery methods can be efficiently scaled to accommodate thousands of tables, enhancing both accuracy and computational speed, as shown in JOSIE [206].

When running HILAutoFeat with hundreds or thousands of tables, our methodology incorporates pruning strategies to eliminate irrelevant tables, such as similarity-based pruning and data quality-based pruning. Consequently, the number of tables in a join tree will not approach the thousand mark, given that most tables will be irrelevant to the base table targeted for augmentation. Furthermore, we have implemented a relationship-caching method, eliminating the need to recompute these connections for future usage.

In addition, our experimental evaluations with the automated approach reveal that our strategy offers a superior performance speed compared to existing state-of-the-art automated dataset augmentation and feature discovery techniques (Section 4.7). This efficiency is maintained despite the integration of human interaction within the process, as user involvement does not affect the computation time for constructing the join trees.

HILAutoFeat uses hyper-parameters to ensure that the curated set of features remains manageable for the user. Accordingly, a maximum of κ features is chosen from each table. In scenarios with large data repositories, HILAutoFeat relies on evaluating the relevance and redundancy metrics for features, thus ensuring the construction of an optimal feature set.

6.3 User-Study Design

The objective of this study is to evaluate whether our human-in-the-loop approach for feature discovery effectively addresses the concerns raised by participants in previous interviews (Chapter 5), specifically regarding their reliance on and trust in an automated approach. We aim to gather information on several aspects: the functionality of the tool – the ease of controlling the steps within the human-in-the-loop pipeline and the usefulness of the explanations provided, the overall effectiveness of the tool, and the ease of using domain/business knowledge. To achieve this, we conducted five follow-up semi-structured interviews with participants who were involved in our earlier study. In the following sections, we will describe the participants, the survey structure, the interview process, and the data processing pipeline.

6.3.1 Participants

For this interview, we invited the participants who had previously indicated that they would not trust or use an automatic tool for feature discovery and augmentation, as well as those who stated they would rely on such a tool only if they could maintain control over the process and have insights into it. We sent a total of seven invitations, and five participants agreed to attend this second round of interviews. To ensure consistency, we retained the same participant identifiers as noted in the previous study (Section 5.4). Therefore, the participants in this study are P3, P4, P11, P14, and P17.

6.3.2 Setup & Scenario

Our aim for the interviews was to gather data on the informed perceptions and experiences of the users rather than their first impressions of the tool. For this reason, before the interviews, we invited the users to explore the tool functionality. This was done via a Kaggle Notebook[1], which describes in detail the steps a user would make in using `HILAutoFeat`. In order to minimise the required time investment for the users and to ensure the comprehensiveness of our human-in-the-loop approach, we provided this notebook to our participants as reading material accompanying the invitation to this interview. We divided the notebook into four sections, according to the pipeline presented in Figure 6.1: input data, finding relationships – to test our design decisions presented in Section 6.2.1, computing join trees – a more extensive section which comprises the designs explained in both Section 6.2.2 and Section 6.2.3, and the evaluation. The Notebook demonstrates the following scenario.

Scenario: *A user aims to augment a dataset by adding features to enhance the accuracy of a tree-based ML model. The default ML model is LightGBM, which is a part of the AutoGluon AutoML framework [52]. Users, however, have the flexibility to select their desired model from the range of models supported by AutoGluon. The user starts with a base table that includes a target variable for binary classification and promising features for the ML model. Additionally, the user has access to a data repository containing multiple tables, either relevant or irrelevant, for augmentation.*

[1] `https://www.kaggle.com/code/zegermouw2/human-in-the-loop-tabular-data-augmentation`

Table 6.1: Overview of the questions asked during the interview and the corresponding **goals**. We also present the *follow-up questions* regarding suggestions for improvement, the working interface and the trust in the human-in-the-loop approach.

Goal		Question
Functionality	Control	What step in this stage are you missing? What other functionality?
	Insights	Is this information helpful? What else would you like to see here?
Knowledge		Would you use your domain/business knowledge in this step?
Effectiveness		Is this approach easier than manual processing?
Follow-up	Suggestions	What other functionalities would you like to see that we haven't discussed yet?
	Interface	Would you prefer a GUI or a notebook for this tool?
	Trust	Would you use and trust this tool to do your data augmentation and feature discovery task?

6.3.3 Interview Process

To ensure that all participants were familiar with the tool such that they express their informed opinion, we preceded each interview with a short presentation of the the steps, where our participants had the opportunity to ask questions. This was followed by the actual semi-structured interview. We asked questions about (*i*) **the functionality**, such as the quality of insights and the degree of control, (*ii*) **the effectiveness** of the tool, and (*iii*) gathered feedback about the ease of adding their **knowledge** into the pipeline. Although we had prepared a set of questions, summarised in Table 6.1, we could deviate from our questions, allowing for a discussion with the participants about the task we were presenting. Thus, the participants were encouraged to motivate and explain their answers so that we could better understand their opinions. We ended the interview with a series of **follow-up** questions. We asked the participants to suggest features and functionalities that can improve the tool and workflow. Moreover, we gathered feedback about the preferred working interface (i.e., python library or user interface), and we inquired about the trust and willingness to use this tool instead of a fully automated tool or manually processing the data.

Before the interview, the participants were asked to provide their consent to participate in the study and record their voice. We conducted all interviews via video conference, which lasted between 30 and 40 minutes. The first interview was used to inform the study scope and refine the protocol questions, which remained unchanged for the other interviews. The interviews were transcribed using automatic transcription software and then manually corrected. Finally, the transcripts were anonymised and used for subsequent processing.

6.3.4 Data Processing

We analysed the data resulting from the interviews using the thematic analysis methodological framework. In thematic analysis, determining themes (i.e., patterns) can be both

theory-driven and data-driven [16]. For the current research, the main themes were theory-driven, while sub-themes (i.e., referred to below as codes) within these main themes were data-driven, generated from the interview data. This application of a thematic analysis approach is fitting since we build upon existing knowledge on the user preferences (Chapter 5), yet openly examine how this takes form with a specific tool to analyse.

First, seven themes were derived. We derived three themes based on the goal of the study: (1) the enhanced functionality by adding insights and offering control, (2) the effectiveness of the tool, and (3) using the domain and business knowledge throughout the pipeline. Next, we derived an additional four themes from each step in the pipeline (Figure 6.1): (4) input data, (5) find relationships, (6) compute join trees and (7) evaluation.

Secondly, the open coding phase was initially conducted on two interviews to define several labels within these seven themes. Two researchers labelled together two interviews, developing labels based on the data within the determined themes. During this phase, the interview transcripts were fully read, and the labels were extracted from the transcripts and linked to themes. Due to the inductive approach, open coding per each theme resulted in a broad identification of ideas and views. We finalised our set of codes *a posteriori*, extracted inductively from the interview data using the qualitative analysis tool ATLAS.ti[2]. We created codes for *computational concerns* and *suggestions* mentioned during each step in the pipeline and for the follow-up questions regarding *trust* and *working interface*. In total, we used 22 codes to label different aspects of each theme. We extracted a total of 137 quotes, each labelled with one or multiple codes. The quotes that are included in the description are verbatim.

6.4 Findings

In this section, we present our findings grouped by each step in the pipeline illustrated in Figure 6.1. We will report participants' feedback on various aspects of functionality, including their perceived level of control over the process and the adequacy of insights provided. Additionally, we will discuss their perspectives on the tool's effectiveness compared to manual or fully automated processes, as well as the extent to which their domain and business knowledge contributes to the process.

6.4.1 Input Data

The input data step is relatively straightforward. Once the user specifies the base table, target column and dataset repository, they can proceed to remove additional tables from the workflow and inspect the collection of tables that `HILAutoFeat` will process. Next, we present our findings.

Functionality & Knowledge. Concerning the depth of the insights, we received numerous suggestions on how participants expected to visualise the tables and the type of information missing from our current insights. On the other hand, the positive feedback centred on the ability to remove tables based on their domain knowledge and inspect how this operation affects the final augmented table.

"If you've got a repository of a large number of files and you know specifically

[2] `https://atlasti.com/`

there are one or two that are not good for your use case, then being able to remove these ones and look at the repository is useful." (P3)

Although none of the participants identified any deficiencies in the current method of controlling the data input step, we received a suggestion for enhancing functionality by enabling users to work with a view of the table instead.

Effectiveness. Our participants reported that it is easy to load a dataset repository, remove tables, and visualise a quick overview, especially compared with the manual process and especially to the automatic process, which is usually a black box. By having control over the tables in the dataset repository, rather than working blindly with the entire dataset, the participants began to trust the process.

6.4.2 Find Relationships

Recall that finding the relationships between tables is computed automatically upon the specification of the necessary hyper-parameters. Once this process is completed, the user can control the subsequent processes, such as adding, removing, or updating any relationship. Moreover, the user can visualise these relationships and receive an explanation of how they were computed (Section 6.2.1).

Functionality & Knowledge. Participants noted that the functionality we introduced, particularly the ability to update relationships, is very helpful. They appreciated being able to use their knowledge to remove relationships, finding this feature especially beneficial.

"For some of these tables, assuming that you have some context, try to adjust some of the relationships." (P17)

We were praised for limiting the operations to only the three essentials (i.e., add, remove, update), thus ensuring that the user is not overwhelmed. The participants did not identify any deficiencies in how they could control this step. However, we received several recommendations for enhancing the functionality related to insights into the process. These suggestions included improving the visualization of relationships, adding the ability to edit table names, and enabling the visualization of non-matching tables.

Effectiveness. The human-in-the-loop approach for identifying relationships has been reported to be much easier than the manual process, as it accelerates the workflow. Additionally, the interface for visualising relationships is considered an added benefit.

6.4.3 Compute Join Trees

Recall that computing the join trees is entirely automatic. However, the users are presented with a visual representation of these join trees accompanied by an explanation of how they have been generated. The users can modify a selected join tree by adding or removing a node from the tree as explained in Section 6.2.2. Moreover, they can also visualise the set of selected and discarded features, which they can manipulate by adding or removing any feature (Section 6.2.3).

Functionality. Our participants appreciated the functionality for controlling the process, such as adding and removing tables (i.e., nodes in the tree) or features, and expressed overall satisfaction with the workflow. Our participants deemed the insights provided at this stage of the pipeline the most useful.

> *"I think the analysis that you show is rather complete and provides really useful information."* (P3)

They reported that seeing the scores for each metric associated with each feature was particularly helpful, and these scores enhanced their trust in the tool. The explanations accompanying the visual representations were found to be easy to follow, and the visualization of the entire workflow helped them gain a comprehensive understanding of the data. Suggestions for improving the insights focused on visualising join trees. Participants recommended incorporating dynamic and interactive views, sorting features based on different criteria, and adding options for data preparation methods.

Effectiveness. Participants reported that the entire pipeline is significantly faster than the manual process, resulting in substantial time savings. They highlighted that the automated aspects of the pipeline reduce the need for repetitive and labour-intensive tasks, allowing them to focus on more critical and strategic aspects of their work. Additionally, the participants appreciated the reduced cognitive load that typically accompanies manual processes. By automating routine tasks, they could allocate their cognitive resources more effectively towards analysing results and deriving insights.

6.4.4 Evaluation

The human-in-the-loop evaluation of the join trees includes several options: evaluating the updated tree, the best-performing tree according to the automatic approach, or all top-k generated trees. Users have the flexibility to choose between these evaluation possibilities at any stage in the workflow. This flexibility allows users to tailor the evaluation process to their specific needs and preferences, ensuring that they can focus on the most relevant aspects of the data. Furthermore, we have enabled the automated materialization of the join tree, providing users with the capability to download the tree and perform additional processes independently. Next, we report our findings.

Functionality & Knowledge. Participants reported that evaluating the tree after each update is very helpful. The ability to materialise the join tree and visualise it was also beneficial, as it allowed them to use their domain knowledge to verify whether all the desired features were taken into account. Participants suggested several enhancements to further improve control over the process. These suggestions included the ability to customise the machine learning model and support for asynchronous runs. Customising the ML model would allow users to tailor the analysis to better fit specific requirements or preferences, while asynchronous runs would enable users to continue other tasks without waiting for the process to complete. While participants found the insights provided by the tool to be helpful, they identified a need for better visualization formats for the results and feature importance.

Effectiveness. All the participants reported that this entire workflow is significantly easier and more convenient compared to the manual process.

> *"I cannot imagine you can get something better by manually doing this. Even if you spend time, but, again, I would probably try the first time to do it both ways, see how it goes and then just use the tool from there."* (P17)

6.4.5 Follow-up & Remarks

Next, we report the results from our follow-up questions, including participants' suggestions, preferred working interface, and trust in the tool. Additionally, we present insights derived from discussions with participants during the pipeline demonstration, such as computational concerns.

Suggestions. As part of the follow-up questions regarding additional remarks and suggestions, all participants emphasised the importance of integrating the tool with databases and providing support for file types beyond those demonstrated. Enabling direct connections to databases could facilitate smoother data import, thus reducing the overhead of working with dataframes. Additionally, supporting a wider range of file types would make the tool more accessible and valuable for users working with diverse datasets.

Working Interface. We asked the participants what their ideal prototype for `HILAutoFeat` would be: a Python library in Jupyter Notebook, as demonstrated, or a graphical user interface. All participants reported that a graphical user interface is not necessary. They found the tool in its current format to be much easier to understand, provided that some of the visual insights are improved. This feedback suggests that enhancing the existing visualizations within the Jupyter Notebook environment would be more beneficial than developing a separate interface. Participants appreciated the direct integration with their existing workflows and the flexibility offered by the Python library format.

Trusting the Tool. Participants reported that breaking down the process into distinct steps helps them use the tool more effectively. Additionally, providing clear explanations of the process and scoring enhances their trust in the tool. Overall, they indicated that they would use the tool to verify assumptions and distil information.

Computational Concerns. A few computational concerns were reported, such as the potential for long training times during the evaluation, which can result in failures and difficulties in handling large datasets with the chosen dataframe tool. Participants noted that prolonged training times can hinder productivity, as they may need to wait for extended periods or experience interruptions due to system failures. Additionally, the difficulty in managing large volumes of data can lead to performance bottlenecks and limit the tool's scalability.

6.5 Conclusion

Our findings suggest that `HILAutoFeat` offers the users the possibility to distil information. Given a large collection of tables, manually performing feature discovery implies inspecting the tables and selecting the most relevant features for the augmentation. This manual process takes a tremendous amount of effort and time to complete. Assuming that the user is unaware of the quality and relevance of the information in the data repository, `HILAutoFeat` helps users by automatically filtering out irrelevant tables. Now, the users have an overview of the relevant tables and features and can choose the most suitable join tree for their subsequent processes.

Moreover, assuming that the dataset repository is smaller (i.e., as presented to our participants) and that the user knows the datasets, our findings suggest that `HILAutoFeat` and the users can work together. While the users rely on `HILAutoFeat` to perform automatic time-consuming computations, they can apply domain and business knowledge to

enhance the feature discovery process and create the best-performing augmented dataset. These findings support and answer our last research question:

***RQ3**: Can human expertise and domain knowledge enhance the automatic feature discovery process?*

We conclude that incorporating the user's domain knowledge is instrumental in shaping a significantly more robust and tailored training dataset, thereby enhancing the overall effectiveness of our feature discovery approach.

7

Conclusion

This thesis researched methods and approaches for designing and developing high-quality training datasets for ML models. Distancing from the traditional model-centric paradigm that prioritises model development, we adopted a data-centric paradigm focused on engineering high-quality datasets. Throughout this thesis, we have researched the effect of integrating dataset discovery approaches in data marketplace platforms to facilitate high-quality data acquisition and designed and developed two feature discovery approaches: automated and human-in-the-loop.

This concluding chapter summarises our main findings. We discuss the limitations encountered during our research, critically reflecting on the areas where our work could be improved or expanded, present potential future work directions, and express our final remarks.

7.1 Summary

In this section, we present a summary for each part of the thesis, highlighting the results and contributions of our work and revisiting the research questions that guided our study.

7.1.1 Tabular Data Acquisition with Data Marketplaces

In *Part I* of this thesis, we explored the evolving landscape of data marketplace platforms in data-centric AI, emphasising their role in facilitating the acquisition of high-quality datasets for ML applications. We followed the examples from data lakes, where finding related datasets is facilitated through metadata management and dataset discovery approaches [70, 146]. As such, in the first part of the thesis, we asked:

> ***RQ1**: How can dataset discovery approaches enable and facilitate data acquisition in data marketplace platforms?*

To address **RQ1**, in Chapter 2, we explored data marketplace platforms and the data acquisition process within these platforms. Our research revealed that while considerable focus is on developing pricing algorithms for data assets, less attention is given to discovering relevant and related assets. This gap highlighted the opportunity for further research into dataset discovery approaches within data marketplace platforms.

To better understand the needs and challenges faced by users of data marketplace platforms, we conducted a survey involving 122 participants representing both data providers and consumers. The survey aimed to uncover the specific requirements and difficulties encountered by these users. Our findings indicated significant market interest and demand for a comprehensive portfolio of services supporting users in publishing and purchasing assets. For data providers, the main concerns involved asset pricing and the development of contracts, which represented significant barriers to entering the marketplace. On the other hand, data consumers were primarily concerned with the ease of discovering and acquiring assets and the quality of the data they obtained.

Guided by the concerns and needs identified through our survey, we developed an open-source data market platform. As such, in Chapter 3, we introduced the `Topio` platform, designed to enhance the exploration and discovery of data assets for both providers and consumers. This platform incorporated and enhanced existing methodologies for data profiling, dataset search and discovery, and data recommendations, making these capabilities available through open-source libraries.

Our preliminary usability evaluation uncovered challenges, particularly regarding the pricing of assets. We opted not to provide a pricing solution, leaving this decision to the providers, which was a struggle for them. However, the evaluation also highlighted the ease with which data consumers could purchase assets. Furthermore, our demonstrations to a diverse audience showcased the capability of the platform to allow users to discover relevant datasets effortlessly. Users could engage with the platform in various ways, from casual browsing to deep analysis within a notebook environment, thus facilitating effective data acquisition. By integrating dataset discovery approaches, the `Topio` platform not only aided users in finding the data they needed more efficiently but also encouraged the exploration of new datasets that might not have been initially considered.

7.1.2 Automated Feature Discovery

In *Part II* of this thesis, we addressed the challenge of automatically discovering relevant features for augmentation within a vast data repository. The process of feature discovery has traditionally been addressed through distinct components: identifying related datasets using dataset discovery approaches, using data integration methods to join or union these datasets, and subsequently applying feature selection techniques to choose the most relevant features (e.g., columns) for an ML model [36, 93]. Consequently, these crucial data preparation steps have been excluded from the AutoML pipeline, which aims to automate ML applications for real-world problems. Existing approaches to automate dataset augmentation or feature discovery have offered limited support for users and have predominantly relied on the capabilities of the ML model alone [36, 129]. Given these considerations, the research question we asked in the second part of this thesis is:

> ***RQ2**: How can we enhance the automation of the feature discovery process to create high-quality datasets for machine learning applications?*

To this end, in Chapter 4, we introduced a novel approach to feature discovery, named `AutoFeat`. This approach distinguished itself by exploring transitive join paths, going beyond the single-hop paths (e.g., star schemata) typically supported by ARDA [36]. Furthermore, `AutoFeat` employed heuristics to assess feature relevance and generated a ranked

list of features, which was a more efficient method compared to the approaches that executed the ML model after every feature augmentation [36, 129]. Moreover, `AutoFeat` enhanced flexibility in joining datasets by supporting multiple join columns for each pair of tables, transforming the joinability into a *multigraph* structure, in contrast to the single join possibility supported by existing methods [36, 129].

Our evaluation of `AutoFeat` against the existing works [36, 119] and the simple, yet exhaustive, approaches involving joining all tables and then applying feature selection algorithms demonstrated that `AutoFeat` achieves similar effectiveness in identifying relevant features for ML models. These results were accomplished in a fraction of the time compared to our baselines, highlighting the efficacy of our pruning strategies and heuristic-based ranking function. Finally, the datasets augmented by `AutoFeat` showed enhanced quality for ML applications, proving the substantial benefits of our innovative approach to automated feature discovery.

7.1.3 Human-in-the-Loop Feature Discovery

In *Part III* of this thesis, we shifted our focus towards the user experience in automated feature discovery systems. While automation alleviated the burden of manual tasks, it often created a gap between the functionality of the system and the actual needs and preferences of end-users. This disconnect could be attributed to the inability to consider the specific expertise of users, which could otherwise make the process more tailored and effective. The challenge, therefore, lay in integrating user-centric design principles into the automated feature discovery process to ensure its efficiency while aligning closely with user requirements and enhancing user engagement. Hence, we asked:

> ***RQ3**: Can human expertise and domain knowledge enhance the automatic feature discovery process?*

To answer our final research question, **RQ3**, in Chapter 5, we conducted 19 semi-structured, think-aloud use-case studies involving data specialists (e.g., data scientists, data engineers and ML engineers). They were tasked with augmenting a base table with additional features to train an ML model. Participants worked on their machines using their preferred set of tools, allowing us to observe their real-life methods for data augmentation and feature discovery without introducing bias by suggesting tools or techniques.

The findings revealed that participants predominantly employed a hybrid approach to feature discovery. During the initial data exploration phase, they manually browsed through tables, constructing mind-maps to note important features, connections between tables, and potential join keys. The data exploration step seamlessly transitioned into the integration step, where participants began to employ automated methods to determine join keys and assess correlations between columns. Further automation was observed in the data preparation processes, which they revisited throughout the entire pipeline. However, the feature selection phase remained hybrid; participants manually reviewed the integrated tables to verify the join results while mapping out the most relevant features. This manual verification was complemented by training the ML model to compare the feature importance scores.

A significant takeaway from the study was the participants' preference for maintaining control over the process. Despite recognising the benefits of automation, they expressed

a reluctance to rely entirely on fully automated tools. They valued the ability to directly interact with the data, which allowed them to gain insights and ensure the integrity of the process. This preference highlighted the need for developing advanced tools that balance automation with user control, ensuring that data specialists could leverage the strengths of both approaches to optimise their workflows.

In response to the insights gained from our user study, in Chapter 6, we developed `HILAutoFeat`, a human-in-the-loop approach for feature discovery. This system enhanced the automated methods previously discussed in *Part II* by incorporating user control at every step of the process. This allowed users to add their expertise and knowledge to the workflow. We integrated features such as explanations and visualisations, giving users deeper insights into the automated processes. To evaluate `HILAutoFeat`, we surveyed a sub-group of participants from the previous study. Although we had prepared a set of questions, we could deviate from our questions, allowing for a discussion with the participants about the task we were presenting. Thus, the participants were encouraged to motivate and explain their answers so that we could better understand their opinions.

The findings from this survey indicated that the human-in-the-loop feature discovery approach not only enhanced the users' confidence in their results through increased control but also alleviated the burden of manual exploration through insightful automation. Participants appreciated being able to "see inside" the black-box automated processes, which helped them trust the results. In conclusion, the human-in-the-loop methodology proved essential in developing high-quality training datasets, as it balanced the efficiency of automation with the effectiveness of human expertise.

7.2 Limitations

We acknowledge multiple limitations that affected this thesis. First, our research was based on interviews conducted with a relatively small number of professionals. Even though we had included professionals with varying levels of experience and occupying different roles, their workflow might not have been representative of that of the general population.

Second, regarding the internal validity of our studies, a threat is that the answers participants gave to the open-ended questions might not have represented their actual opinions and workflows due to of biases and memory limitations. Maybe socially desirable answers were given since the researchers involved in developing the tool interviewed the participants, thus making the participants reluctant to express negative opinions. Even though we offered training and explanations before the interviews, the expressed opinions might have differed if the participants were more familiar with the tool or had experienced using it in real-life scenarios. We partially circumvented this bias by including in the study design the use case scenario and direct observation of participant actions on their regular machines. Still, participant actions may have been affected by observer expectancy biases. Moreover, the use case setup may have affected the observed workflows. For this reason, the use case was selected to represent a real-world scenario, and follow-up questions were included in the interview protocol to capture participant reflections on more complex scenarios.

Third, concerning data processing, in the case of a thematic analysis, decisions on the approach are guided and limited by the underlying aim of the study [19]. The dif-

ferent workflows and opinions of the participants were integrated yet described extensively, giving context and providing quotes to illustrate and substantiate our interpretation. Nonetheless, this study has yielded multiple undocumented insights into how data practitioners operate in the real world. Those insights can impact how database research is shaped in the future.

Furthermore, the datasets used for our evaluations, though sourced from previous studies or well-known and widely used open-source platforms (e.g., OpenML), may not have mirrored the complexities and nuances of real-world data. This aspect of our research design could impact the applicability of our findings to a broader context. Moreover, our experimental approach involved using modified datasets specifically to highlight the limitations of other methods, which might not provide a complete picture of how these approaches would perform under typical real-world conditions.

7.3 Future Directions

Following, we present directions for future work based on our research findings.

7.3.1 Data Marketplace Platforms

Fair Markets. Future research and development in data marketplace platforms should prioritise addressing the concerns of data providers, particularly regarding the pricing of data assets. Existing research offers various pricing mechanisms designed to support data providers and consumers [57, 133, 156], which could be further refined or adapted to the evolving market dynamics. Moreover, methods developed to prevent arbitrage should continue to be enhanced to maintain fair trading conditions [25]. More research in this direction is needed to create a fair market for data.

Data Protection. The rise of generative AI poses new challenges, mainly the ease of generating datasets that might compromise the quality of data available in the marketplaces. These artificially generated datasets could mislead data consumers into purchasing inaccurate and non-representative data. Therefore, extensive research is required to develop robust mechanisms verifying data quality and authenticity in marketplaces. Protecting the integrity of data assets is another crucial concern, and techniques such as watermarking offer potential solutions [3].

Transparent Data Sources. Finally, ethical considerations and potential biases in data selling must be carefully managed. It is essential to ensure that the origins of data assets are transparent and ethically sound. Issues such as data being sourced without the owners' consent, as seen with data generated by users of web browsers often gathered without their explicit knowledge, must be addressed. Implementing stricter regulatory compliance and ethical standards for data acquisition and sales could help safeguard the interests of all stakeholders involved in the data marketplace. The General Data Protection Regulation (GDPR) is an example of regulating and protecting the data. However, its implementation is localised [61].

Privacy-Preserving Data Acquisition. Data acquisition continues to be a complex area of research, mainly due to privacy concerns, which rank among the primary worries for users of data market platforms [108, 192]. Data providers often hesitate to share detailed

information about their assets beyond basic metadata, including data samples. This reluctance stems from a desire to protect proprietary information and maintain privacy. However, without the ability to review the data, data consumers may be reluctant to make purchases. The risk perceived in buying unseen data is significant, as it could lead to expenses that do not meet their specific data needs or expectations. Privacy-preserving methods have already been proposed in the literature [80, 108], and future work should focus on integrating these methods into discovery and acquisition approaches.

Enhanced User Experience. From our demonstrations, we could derive that future data market platforms should incorporate more flexible query capabilities. Supporting multiple query types, such as allowing users to upload a table and then automatically suggesting other related and relevant tables for acquisition and augmentation, could significantly enhance user experience and platform utility.

7.3.2 Feature Discovery

Deployment in Marketplaces. Building on our previous research on data marketplace platforms, the automatic feature discovery approach presents a promising path for enhancing data acquisition in these marketplaces. By combining this approach with an improved query mechanism, data consumers could benefit significantly, as it allows them to purchase specific features from a dataset rather than acquiring the entire dataset.

Deployment in Production. Deploying this method in an industry setting would also provide substantial benefits for future research. Such real-world applications would allow us to critically evaluate our design decisions within a practical context and observe the scalability challenges firsthand. Future work on automated feature discovery methods should consider the vast volume of evolving heterogeneous data, ensuring that the solutions developed are robust enough to handle real-world complexities.

Integration with AutoML. Integrating feature discovery into the AutoML pipeline represents a natural progression for this automatic approach, offering users a comprehensive end-to-end solution for ML tasks within data repositories. While specific steps from the data preparation pipeline have already been automated with the intention of integration into the AutoML pipeline, further research is needed to fully unify all steps into a cohesive automated system [63, 99, 171]. Data quality is a critical factor in the successful adoption of automated feature discovery in AutoML pipelines. Future initiatives should focus on understanding the impact of data quality on data augmentation processes and developing methods to automatically enhance this quality.

Finally, future work could leverage automatic feature discovery to mitigate bias within datasets actively. This could be further enhanced by incorporating user input into the process, allowing for a more tailored approach. We explored a human-in-the-loop approach in Part III of this thesis, aiming to refine the process of automatic feature discovery.

7.3.3 User-Centric AI

Large Scale User Studies. Future work should include an in-depth analysis of data specialists' workflows. To achieve this, conducting large-scale user studies is essential. These studies will provide valuable insights into the processes and tools used by data specialists, enabling us to better understand their needs and the challenges they face. Such research

will also help identify areas where additional support and research could streamline their workflows and improve overall efficiency.

Industry Collaboration. Further analysis and deployment within an industry setting would greatly benefit the human-in-the-loop feature discovery approach. By observing how users interact with the tool in real-world scenarios, we can better understand its limitations and the enhancements required to facilitate broader adoption. Exploring these areas will enhance the utility of the tool we designed and developed, ensure its robustness and user-friendliness, and encourage wider acceptance and use within the professional community.

7.4 Final Remarks

This thesis has advanced the research and development within the data management community guided by the principles and objectives of data-centric AI. However, our findings and contributions extend beyond data-centric approaches and pave the way for creating a synergy between data-centric, model-centric, and user-centric AI. By exploring the intersections of these paradigms, future research can balance the strengths of these areas, producing more robust, effective and intuitive AI solutions.

Bibliography

References

[1] Antragama Ewa Abbas, Wirawan Agahari, Montijn Van de Ven, Anneke Zuiderwijk, and Mark De Reuver. 2021. Business data sharing through data marketplaces: A systematic literature review. *Journal of Theoretical and Applied Electronic Commerce Research* 16, 7 (2021), 3321–3339.

[2] Anish Agarwal, Munther Dahleh, and Tuhin Sarkar. 2019. A marketplace for data: An algorithmic solution. In *EC'19*. 701–726.

[3] Rakesh Agrawal and Jerry Kiernan. 2002. Watermarking relational databases. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 155–166.

[4] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2021. PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *J. Mach. Learn. Res.* 22 (2021), 82:1–82:6.

[5] Noura AlNuaimi, Mohammad Mehedy Masud, Mohamed Adel Serhani, and Nazar Zaki. 2020. Streaming feature selection algorithms for big data: A survey. *Applied Computing and Informatics* 18, 1/2 (2020), 113–135.

[6] Abolfazl Asudeh and HV Jagadish. 2020. Fairly evaluating and scoring items in a data set. *Proceedings of the VLDB Endowment* 13, 12 (2020), 3445–3448.

[7] Abolfazl Asudeh and Fatemeh Nargesian. 2022. Towards distribution-aware query answering in data markets. *Proceedings of the VLDB Endowment* 15, 11 (2022), 3137–3144.

[8] Santiago Andrés Azcoitia and Nikolaos Laoutaris. 2022. A Survey of Data Marketplaces and Their Business Models. *SIGMOD Rec.* 51, 3 (2022), 18–29.

[9] Wolfgang Badewitz, Christoph Hengesbach, and Christof Weinhardt. 2022. Challenges of pricing data assets: a literature review. In *2022 IEEE 24th Conference on Business Informatics (CBI)*, Vol. 1. IEEE, 80–89.

[10] Magdalena Balazinska, Bill Howe, and Dan Suciu. 2011. Data markets in the cloud: An opportunity for the database community. *Proceedings of the VLDB Endowment* 4, 12 (2011), 1482–1485.

[11] James K Batcheller and Femke Reitsma. 2010. Implementing feature level semantics for spatial data discovery: Supporting the reuse of legacy data using open source components. *Computers, Environment and Urban Systems* 34, 4 (2010), 333–344.

[12] Roberto Battiti. 1994. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks* 5, 4 (1994), 537–550.

[13] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. 2023. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf* 2, 3 (2023), 8.

[14] Sagar Bharadwaj, Praveen Gupta, Ranjita Bhagwan, and Saikat Guha. 2021. Discovering Related Data at Scale. *PVLDB* 14, 8 (2021), 1392–1400.

[15] Sumon Biswas, Mohammad Wardat, and Hridesh Rajan. 2022. The art and practice of data science pipelines: A comprehensive study of data science pipelines in theory, in-the-small, and in-the-large. In *Proceedings of the 44th International Conference on Software Engineering*. 2091–2103.

[16] Erik Blair. 2015. A reflexive exploration of two qualitative data coding techniques. *Journal of Methods and Measurement in the Social Sciences* 6, 1 (2015), 14–29.

[17] Alex Bogatu, Alvaro AA Fernandes, Norman W Paton, and Nikolaos Konstantinou. 2020. Dataset discovery in data lakes. In *ICDE*. IEEE, 709–720.

[18] Alex Bogatu, Norman W Paton, Mark Douthwaite, and Andre Freitas. 2022. Voyager: Data discovery and integration for data science. *Journal of Data and Information Quality (jul 2022). https://doi. org/10.48786/edbt* (2022).

[19] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[20] Maarten A Breddels and Jovan Veljanoski. 2018. Vaex: big data exploration in the era of gaia. *Astronomy & Astrophysics* 618 (2018), A13.

[21] Dan Brickley, Matthew Burgess, and Natasha Noy. 2019. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In *The World Wide Web Conference*. 1365–1375.

[22] Alan Bundy and Lincoln Wallen. 1984. Breadth-first search. *Catalogue of artificial intelligence tools* (1984), 13–13.

[23] Michael J. Cafarella, Alon Halevy, and Nodira Khoussainova. 2009. Data integration for the relational web. *PVLDB* (2009), 1090–1101.

[24] Sonia Castelo, Rémi Rampin, Aécio SR Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. 2021. Auctus: a dataset search engine for data discovery and augmentation. (2021).

[25] Raul Castro Fernandez. 2022. Protecting data markets from strategic buyers. In *Proceedings of the 2022 International Conference on Management of Data*. 1755–1769.

[26] Raul Castro Fernandez, Jisoo Min, Demitri Nava, and Samuel Madden. 2019. Lazo: A cardinality-based method for coupled estimation of jaccard similarity and containment. *ICDE* 2019-April (2019), 1190–1201.

[27] Chengliang Chai, Jiayi Wang, Yuyu Luo, Zeping Niu, and Guoliang Li. 2022. Data management for machine learning: A survey. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2022), 4646–4667.

[28] Gromit Yeuk Yin Chan, Tung Mai, Anup B Rao, Ryan A Rossi, Fan Du, Cláudio T Silva, and Juliana Freire. 2021. Interactive Audience Expansion on Large Scale Online Visitor Data. In *27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2021*. Association for Computing Machinery, 2621–2631.

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

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

[31] Shuchi Chawla, Shaleen Deep, Paraschos Koutrisw, and Yifeng Teng. 2019. Revenue maximization for query pricing. *Proceedings of the VLDB Endowment* 13, 1 (2019), 1–14.

[32] Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2147–2157.

[33] Lingjiao Chen, Bilge Acun, Newsha Ardalani, Yifan Sun, Feiyang Kang, Hanrui Lyu, Yongchan Kwon, Ruoxi Jia, Carole-Jean Wu, Matei Zaharia, et al. 2023. Data acquisition: A new frontier in data-centric AI. *arXiv preprint arXiv:2311.13712* (2023).

[34] Lingjiao Chen, Paraschos Koutris, and Arun Kumar. 2019. Towards Model-based Pricing for Machine Learning in a Data Marketplace. In *SIGMOD*. 1535–1552.

[35] Yiling Chen, Nicole Immorlica, Brendan Lucier, Vasilis Syrgkanis, and Juba Ziani. 2018. Optimal data acquisition for statistical estimation. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. 27–44.

[36] Nadiia Chepurko, Ryan Marcus, Emanuel Zgraggen, Raul Castro Fernandez, Tim Kraska, and David Karger. 2020. ARDA: automatic relational data augmentation for machine learning. *PVLDB* (2020), 1373–1387.

[37] Yunfei Chu, Jiangchao Yao, Chang Zhou, and Hongxia Yang. 2022. *Graph Neural Networks in Modern Recommender Systems*. Springer Nature Singapore.

[38] Tianji Cong, James Gale, Jason Frantz, HV Jagadish, and Çağatay Demiralp. 2022. WarpGate: A Semantic Join Discovery System for Cloud Data Warehouse. *arXiv preprint arXiv:2212.14155* (2022).

[39] Corinna Cortes, Lawrence D Jackel, and Wan-Ping Chiang. 1994. Limits on learning machine accuracy imposed by data quality. *Advances in Neural Information Processing Systems* 7 (1994).

[40] Anamaria Crisan and Brittany Fiore-Gartland. 2021. Fits and starts: Enterprise use of automl and the role of humans in the loop. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.

[41] Anamaria Crisan, Brittany Fiore-Gartland, and Melanie Tory. 2020. Passing the data baton: A retrospective analysis on data science work and workers. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1860–1870.

[42] Bing Tian Dai, Nick Koudas, Beng Chin Ooi, Divesh Srivastava, and Suresh Venkatasubramanian. 2006. Rapid identification of column heterogeneity. In *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 159–170.

[43] Mark de Reuver, Hosea Ofe, Wirawan Agahari, Antragama Ewa Abbas, and Anneke Zuiderwijk. 2022. The openness of data platforms: a research agenda. In *Proceedings of the 1st International Workshop on Data Economy*. 34–41.

[44] Joost de Winter, Samuel Gosling, and J. Potter. 2016. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological Methods* 21 (2016), 273–290. `https://doi.org/10.1037/met0000079.supp`

[45] Yuri Demchenko, Wouter Los, Cees de Laat, et al. 2018. Data as economic goods: Definitions, properties, challenges, enabling technologies for future data markets. *ITU Journal: ICT Discoveries* 2, 23 (2018).

[46] Dong Deng, Raul Castro Fernandez, Ziawasch Abedjan, Sibo Wang, Michael Stonebraker, Ahmed K Elmagarmid, Ihab F Ilyas, Samuel Madden, Mourad Ouzzani, and Nan Tang. 2017. The Data Civilizer System.. In *Cidr*.

[47] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[48] Hong-Hai Do and Erhard Rahm. 2002. COMA—a system for flexible combination of schema matching approaches. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 610–621.

[49] Xin Luna Dong, Barna Saha, and Divesh Srivastava. 2012. Less is more: Selecting sources wisely for integration. *Proceedings of the VLDB Endowment* 6, 2 (2012), 37–48.

[50] Yuyang Dong, Kunihiro Takeoka, Chuan Xiao, and Masafumi Oyamada. 2021. Efficient joinable table discovery in data lakes: A high-dimensional similarity-based approach. In *ICDE*. IEEE, 456–467.

[51] Rakkrit Duangsoithong and Terry Windeatt. 2009. Relevance and redundancy analysis for ensemble classifiers. In *Machine Learning and Data Mining in Pattern Recognition: 6th International Conference, MLDM 2009, Leipzig, Germany, July 23-25, 2009. Proceedings 6*. Springer, 206–220.

[52] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. 2020. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505* (2020).

[53] Mahdi Esmailoghli, Jorge-Arnulfo Quiané-Ruiz, and Ziawasch Abedjan. 2021. COCOA: COrrelation COefficient-Aware Data Augmentation.. In *EDBT*. 331–336.

[54] Sabri Eyuboglu, Bojan Karlaš, Christopher Ré, Ce Zhang, and James Zou. 2022. dcbench: a benchmark for data-centric AI systems. In *Proceedings of the Sixth Workshop on Data Management for End-To-End Machine Learning*. 1–4.

[55] Ronald Fagin, Phokion G Kolaitis, and Lucian Popa. 2005. Data exchange: getting to the core. *ACM Transactions on Database Systems (TODS)* 30, 1 (2005), 174–210.

[56] Grace Fan, Jin Wang, Yuliang Li, and Renée J Miller. 2023. Table Discovery in Data Lakes: State-of-the-art and Future Directions. In *Companion of the 2023 International Conference on Management of Data*. 69–75.

[57] Raul Castro Fernandez. 2022. Protecting Data Markets from Strategic Buyers. (2022).

[58] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *ICDE*. 1001–1012.

[59] Raul Castro Fernandez, Pranav Subramaniam, and Michael J Franklin. 2020. Data market platforms: trading data assets to solve data problems. *Proceedings of the VLDB Endowment* 13, 12 (2020), 1933–1947.

[60] Tim Furche, Georg Gottlob, Leonid Libkin, Giorgio Orsi, and Norman W Paton. 2016. Data Wrangling for Big Data: Challenges and Opportunities.. In *EDBT*, Vol. 16. 473–478.

[61] Michal S Gal and Oshrit Aviv. 2020. The competitive effects of the GDPR. *Journal of Competition Law & Economics* 16, 3 (2020), 349–391.

[62] Sainyam Galhotra, Yue Gong, and Raul Castro Fernandez. 2023. METAM: Goal-Oriented Data Discovery. *arXiv preprint arXiv:2304.09068* (2023).

[63] Joseph Giovanelli, Besim Bilalli, and Alberto Abelló Gamazo. 2021. Effective data pre-processing for AutoML. In *Proceedings of the 23rd International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP): co-located with the 24th International Conference on Extending Database Technology and the 24th International Conference on Database Theory (EDBT/ICDT 2021): Nicosia, Cyprus, March 23, 2021*. CEUR-WS. org, 1–10.

[64] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data?. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[65] Venkat Gudivada, Amy Apon, and Junhua Ding. 2017. Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software* 10, 1 (2017), 1–20.

[66] Greg Guest, Arwen Bunce, and Laura Johnson. 2006. How many interviews are enough? An experiment with data saturation and variability. *Field methods* 18, 1 (2006), 59–82.

[67] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of machine learning research* 3, Mar (2003), 1157–1182.

[68] Rihan Hai, Sandra Geisler, and Christoph Quix. 2016. Constance: An intelligent data lake system. In *Proceedings of the 2016 international conference on management of data*. 2097–2100.

[69] Rihan Hai, Christos Koutras, Andra Ionescu, Ziyu Li, Wenbo Sun, Jessie van Schijndel, Yan Kang, and Asterios Katsifodimos. 2023. Amalur: Data Integration Meets Machine Learning. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 3729–3739.

[70] Rihan Hai, Christos Koutras, Christoph Quix, and Matthias Jarke. 2023. Data Lakes: A Survey of Functions and Systems. *IEEE TKDE* (2023).

[71] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. 2017. Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications* 73 (2017), 220–239.

[72] Minbiao Han, Jonathan Light, Steven Xia, Sainyam Galhotra, Raul Castro Fernandez, and Haifeng Xu. 2023. A Data-Centric Online Market for Machine Learning: From Discovery to Pricing. *arXiv preprint arXiv:2310.17843* (2023).

[73] Maeda F Hanafi, Miro Mannino, and Azza Abouzied. 2019. A collaborative framework for structure identification over print documents. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–8.

[74] Mark Harrower and Matt Bloch. 2006. MapShaper. org: A map generalization web service. *IEEE Computer Graphics and Applications* 26, 4 (2006), 22–27.

[75] Teruaki Hayashi and Yukio Ohsawa. 2020. TEEDA: an interactive platform for matching data providers and users in the data marketplace. *Information* 11, 4 (2020), 218.

[76] James Hendler, Jeanne Holm, Chris Musialek, and George Thomas. 2012. US government linked open data: semantic. data. gov. *IEEE Intelligent Systems* 27, 03 (2012), 25–31.

[77] Xuegang Hu, Peng Zhou, Peipei Li, Jing Wang, and Xindong Wu. 2018. A survey on online feature selection with streaming features. *Frontiers of Computer Science* 12 (2018), 479–493.

[78] Samuel H Huang. 2015. Supervised feature selection: A tutorial. *Artif. Intell. Res.* 4, 2 (2015), 22–37.

[79] Yan Huang, Shashi Shekhar, and Hui Xiong. 2004. Discovering colocation patterns from spatial data sets: a general approach. *IEEE TKDE* 16, 12 (2004), 1472–1485.

[80] Nick Hynes, David Dao, David Yan, Raymond Cheng, and Dawn Song. 2018. A demonstration of sterling: a privacy-preserving data marketplace. *Proceedings of the VLDB Endowment* 11, 12 (2018), 2086–2089.

[81] Stratos Idreos, Olga Papaemmanouil, and Surajit Chaudhuri. 2015. Overview of data exploration techniques. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 277–281.

[82] Andra Ionescu. 2023. OpertusMundi Discovery Service. `https://doi.org/10.5281/ZENODO.12755203`

[83] Andra Ionescu. 2024. Automated Feature Discovery. `https://doi.org/10.5281/ZENODO.12755373`

[84] Andra Ionescu. 2024. Daatset for testing Feature Discovery. `https://doi.org/10.5281/ZENODO.12755408`

[85] Andra Ionescu. 2024. Dataset for testing the OpertusMundi Discovery Service. `https://doi.org/10.5281/ZENODO.12755251`

[86] Andra Ionescu, Alexandra Alexandridou, Leonidas Ikonomou, Kyriakos Psarakis, Kostas Patroumpas, Georgios Chatzigeorgakidis, Dimitrios Skoutas, Spiros Athanasiou, Rihan Hai, and Asterios Katsifodimos. 2023. Topio Marketplace: Search and Discovery of Geospatial Data. In *EDBT*.

[87] Andra Ionescu, Rihan Hai, Marios Fragkoulis, and Asterios Katsifodimos. 2022. Join Path-Based Data Augmentation for Decision Trees. In *IEEE ICDEW*. IEEE, 84–88.

[88] Andra Ionescu, A Katsifodimos, and GJPM Houben. 2021. Interactive Data Discovery in Data Lakes. In *VLDB PhD Workshop*, Vol. 2971. CEUR-WS.

[89] Andra Ionescu and Zeger Mouw. 2024. Human-in-the-Loop Feature Discovery. `https://doi.org/10.5281/ZENODO.12755486`

[90] Andra Ionescu, Zeger Mouw, Efthimia Aivaloglou, and Asterios Katsifodimos. 2024. Key Insights from a Feature Discovery User Study. In *Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics*. 1–5.

[91] Andra Ionescu, Zeger Mouw, Fenia Aivaloglou, Rihan Hai, and Asterios Katsifodimos. 2024. Human-in-the-Loop Feature Discovery for Tabular Data. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*.

[92] Andra Ionescu, Kostas Patroumpas, Kyriakos Psarakis, Georgios Chatzigeorgakidis, Diego Collarana, Kai Barenscher, Dimitrios Skoutas, Asterios Katsifodimos, and Spiros Athanasiou. 2023. Topio: An Open-Source Web Platform for Trading Geospatial Data. In *International Conference on Web Engineering*. Springer, 336–351.

[93] Andra Ionescu, Kiril Vailev, Florena Buse, Rihan Hai, and Asterios Katsifodimos. 2024. AutoFeat: Transitive Feature Discovery over Join Paths. In *ICDE*. IEEE, 1861–1873.

[94] Abhinav Jain, Hima Patel, Lokesh Nagalapatti, Nitin Gupta, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Ruhi Sharma Mittal, and Vitobha Munigala. 2020. Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 3561–3562.

[95] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. 2017. Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology* 2, 4 (2017).

[96] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the sigchi conference on human factors in computing systems*. 3363–3372.

[97] Sean Kandel, Andreas Paepcke, Joseph M Hellerstein, and Jeffrey Heer. 2012. Enterprise data analysis and visualization: An interview study. *IEEE transactions on visualization and computer graphics* 18, 12 (2012), 2917–2926.

[98] Eser Kandogan, Aruna Balakrishnan, Eben M Haber, and Jeffrey S Pierce. 2014. From data to insight: work practices of analysts in the enterprise. *IEEE computer graphics and applications* 34, 5 (2014), 42–50.

[99] Shubhra Kanti Karmaker, Md Mahadi Hassan, Micah J Smith, Lei Xu, Chengxiang Zhai, and Kalyan Veeramachaneni. 2021. Automl to date and beyond: Challenges and opportunities. *ACM Computing Surveys (CSUR)* 54, 8 (2021), 1–36.

[100] Stephen Kasica, Charles Berret, and Tamara Munzner. 2023. Dirty Data in the Newsroom: Comparing Data Preparation in Journalism and Data Science. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.

[101] Maxat Kassen. 2013. A promising phenomenon of open data: A case study of the Chicago open data project. *Government information quarterly* 30, 4 (2013), 508–513.

[102] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems* 30 (2017).

[103] Aamod Khatiwada, Roee Shraga, Wolfgang Gatterbauer, and Renée J Miller. 2022. Integrating Data Lake Tables. *Proceedings of the VLDB Endowment* 16, 4 (2022), 932–945.

[104] Miryung Kim, Thomas Zimmermann, Robert DeLine, and Andrew Begel. 2016. The emerging role of data scientists on software development teams. In *Proceedings of the 38th International Conference on Software Engineering*. 96–107.

[105] Ron Kohavi and George H John. 1997. Wrappers for feature subset selection. *Artificial intelligence* 97, 1-2 (1997), 273–324.

[106] Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, and Asterios Katsifodimos. 2021. Valentine: Evaluating Matching Techniques for Dataset Discovery. In *ICDE*. IEEE, 468–479.

[107] Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, and Dan Suciu. 2015. Query-based data pricing. *Journal of the ACM (JACM)* 62, 5 (2015), 1–44.

[108] Vlasis Koutsos, Dimitrios Papadopoulos, Dimitris Chatzopoulos, Sasu Tarkoma, and Pan Hui. 2021. Agora: A privacy-aware data marketplace. *IEEE Transactions on Dependable and Secure Computing* 19, 6 (2021), 3728–3740.

[109] Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. 2023. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access* (2023).

[110] Agustinus Kristiadi, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann, and Asja Fischer. 2019. Incorporating Literals into Knowledge Graph Embeddings. In *The Semantic Web - ISWC*. Springer, 347–363.

[111] Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 3609–3619.

[112] Arun Kumar, Jeffrey Naughton, Jignesh M Patel, and Xiaojin Zhu. 2016. To join or not to join? Thinking twice about joins before feature selection. In *SIGMOD*. 19–34.

[113] Javier Lacasta, Javier Nogueras-Iso, Rubén Béjar, Pedro R Muro-Medrano, and F Javier Zarazaga-Soria. 2007. A web ontology service to facilitate interoperability within a spatial data infrastructure: Applicability to discovery. *Data & Knowledge Engineering* 63, 3 (2007), 947–971.

[114] Thomas Navin Lal, Olivier Chapelle, Jason Weston, and André Elisseeff. 2006. Embedded methods. In *Feature extraction*. Springer, 137–165.

[115] Maurizio Lenzerini. 2002. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 233–246.

[116] Aristotelis Leventidis, Jiahui Zhang, Cody Dunne, Wolfgang Gatterbauer, HV Jagadish, and Mirek Riedewald. 2020. QueryVis: Logic-based diagrams help users understand complicated SQL queries faster. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2303–2318.

[117] Guoliang Li. 2017. Human-in-the-loop data integration. *Proceedings of the VLDB Endowment* 10, 12 (2017), 2006–2017.

[118] Haiguang Li, Xindong Wu, Zhao Li, and Wei Ding. 2013. Group feature selection with streaming features. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 1109–1114.

[119] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. 2017. Feature selection: A data perspective. *ACM computing surveys (CSUR)* 50, 6 (2017), 1–45.

[120] Peng Li, Xi Rao, Jennifer Blase, Yue Zhang, Xu Chu, and Ce Zhang. 2021. CleanML: A study for evaluating the impact of data cleaning on ml classification tasks. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 13–24.

[121] Xijun Li, Jianguo Yao, Xue Liu, and Haibing Guan. 2017. A first look at information entropy-based data pricing. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2053–2060.

[122] Yanying Li, Haipei Sun, Boxiang Dong, and Hui Wendy Wang. [n. d.]. Cost-efficient Data Acquisition on Online Data Marketplaces for Correlation Analysis. *Proceedings of the VLDB Endowment* 12, 4 ([n. d.]).

[123] Yifan Li, Xiaohui Yu, and Nick Koudas. 2021. Data acquisition for improving machine learning models. *Proceedings of the VLDB Endowment* 14, 10 (2021), 1832–1844.

[124] Yifan Li, Xiaohui Yu, and Nick Koudas. 2021. Data Acquisition for Improving Machine Learning Models. *Proc. VLDB Endow.* 14, 10 (2021), 1832–1844.

[125] Yifan Li, Xiaohui Yu, and Nick Koudas. 2024. Data Acquisition for Improving Model Confidence. *Proceedings of the ACM on Management of Data* 2, 3 (2024), 1–25.

[126] Fan Liang, Wei Yu, Dou An, Qingyu Yang, Xinwen Fu, and Wei Zhao. 2018. A survey on big data market: Pricing, trading and protection. *Ieee Access* 6 (2018), 15132–15154.

[127] Fan Liang, Wei Yu, Dou An, Qingyu Yang, Xinwen Fu, and Wei Zhao. 2018. A Survey on Big Data Market: Pricing, Trading and Protection. *IEEE Access* 6 (2018), 15132–15154.

[128] Dongxiao Liu, Cheng Huang, Jianbing Ni, Xiaodong Lin, and Xuemin Sherman Shen. 2022. Blockchain-cloud transparent data marketing: Consortium management and fairness. *IEEE Trans. Comput.* 71, 12 (2022), 3322–3335.

[129] Jiabin Liu, Chengliang Chai, Yuyu Luo, Yin Lou, Jianhua Feng, and Nan Tang. 2022. Feature augmentation with reinforcement learning. In *ICDE*. IEEE, 3360–3372.

[130] Yuze Lou and Michael Cafarella. 2022. Enabling useful provenance in scripting languages with a human-in-the-loop. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–7.

[131] Yaoli Mao, Dakuo Wang, Michael Muller, Kush R Varshney, Ioana Baldini, Casey Dugan, and Aleksandra Mojsilović. 2019. How data scientists work together with domain experts in scientific collaborations: To find the right answer or to ask the right question? *Proceedings of the ACM on Human-Computer Interaction* 3, GROUP (2019), 1–23.

[132] Mark Mazumder, Colby Banbury, Xiaozhe Yao, Bojan Karlaš, William Gaviria Rojas, Sudnya Diamos, Greg Diamos, Lynn He, Alicia Parrish, Hannah Rose Kirk, et al. 2024. Dataperf: Benchmarks for data-centric ai development. *Advances in Neural Information Processing Systems* 36 (2024).

[133] Sameer Mehta, Milind Dawande, Ganesh Janakiraman, and Vijay Mookerjee. 2021. How to sell a data set? Pricing policies for data monetization. *Information Systems Research* 32, 4 (2021), 1281–1297.

[134] Zhengjie Miao, Yuliang Li, and Xiaolan Wang. 2021. Rotom: A meta-learned data augmentation framework for entity matching, data cleaning, text classification, and beyond. In *Proceedings of the 2021 International Conference on Management of Data*. 1303–1316.

[135] Renée J. Miller, Fatemeh Nargesian, Erkang Zhu, Christina Christodoulakis, Ken Q. Pu, and Periklis Andritsos. 2018. Making Open Data Transparent: Data Discovery on Open Data. *IEEE Data Eng. Bull.* 41, 2 (2018), 59–70.

[136] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. 2018. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics* 19, 6 (2018), 1236–1246.

[137] Krešimir Mišura and Mario Žagar. 2016. Data marketplace for Internet of Things. In *2016 International Conference on Smart Systems and Technologies (SST)*. IEEE, 255–260.

[138] Pantelis Mitropoulos, Kostas Patroumpas, Dimitrios Skoutas, Thodoris Vakkas, and Spiros Athanasiou. 2021. BigDataVoyant: Automated Profiling of Large Geospatial Data.. In *EDBT/ICDT Workshops*.

[139] Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. 2002. Feature selection algorithms: A survey and experimental evaluation. In *ICDM*. IEEE, 306–313.

[140] Oscar Moll, Manuel Favela, Samuel Madden, Vijay Gadepally, and Michael Cafarella. 2023. SeeSaw: interactive ad-hoc search over image databases. *Proceedings of the ACM on Management of Data* 1, 4 (2023), 1–26.

[141] Tomasz Mucha and Timo Seppala. 2020. Artificial intelligence platforms–a new research agenda for digital platform economy. (2020).

[142] Heiko Müller, Sonia Castelo, Munaf Qazi, and Juliana Freire. 2021. From papers to practice: the openclean open-source data cleaning library. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2763–2766.

[143] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How data science workers work with data: Discovery, capture, curation, design, creation. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–15.

[144] Fatemeh Nargesian, Abolfazl Asudeh, and HV Jagadish. 2022. Responsible Data Integration: Next-generation Challenges. In *SIGMOD*. 2458–2464.

[145] Fatemeh Nargesian, Erkang Zhu, Renée J. Miller, Ken Q. Pu, and Patricia C. Arocena. 2019. Data Lake Management: Challenges and Opportunities. *Proc. VLDB Endow.* 12, 12 (2019), 1986–1989.

[146] Fatemeh Nargesian, Erkang Zhu, Ken Q. Pu, and Renee J. Miller. 2018. Table union search on open data. *PVLDB*, 813–825.

[147] Felix Naumann. 2014. Data profiling revisited. *ACM SIGMOD Record* 42, 4 (2014), 40–49.

[148] Marius F Niculescu, DJ Wu, and Lizhen Xu. 2018. Strategic intellectual property sharing: Competition on an open technology platform under network effects. *Information Systems Research* 29, 2 (2018), 498–519.

[149] Edobor Osagie, Mohammad Waqar, Samuel Adebayo, Arkadiusz Stasiewicz, Lukasz Porwol, and Adegboyega Ojo. 2017. Usability evaluation of an open data platform. In *Proceedings of the 18th annual international conference on digital government research*. 495–504.

[150] Paul Ouellette, Aidan Sciortino, Fatemeh Nargesian, Bahar Ghadiri Bashardoost, Erkang Zhu, Ken Pu, and Renée J. Miller. 2021. RONIN: Data Lake Exploration. *Proc. VLDB Endow.* 14, 12 (2021), 2863–2866.

[151] Paul Ouellette, Aidan Sciortino, Fatemeh Nargesian, Bahar Ghadiri Bashardoost, Erkang Zhu, Ken Q Pu, and Renée J Miller. 2021. RONIN: data lake exploration. *Proceedings of the VLDB Endowment* 14, 12 (2021).

[152] Kazim Rifat Özyilmaz, Mehmet Doğan, and Arda Yurdakul. 2018. IDMoB: IoT data marketplace on blockchain. In *2018 crypto valley conference on blockchain technology (CVCBT)*. IEEE, 11–19.

[153] Marko Palviainen and Jutta Suksi. 2023. Data marketplace research: A review of the state-of-the-art with a focus on smart cities and on edge data exchange and trade. In *2023 Smart City Symposium Prague (SCSP)*. IEEE, 1–7.

[154] Thorsten Papenbrock, Tanja Bergmann, Moritz Finke, Jakob Zwiener, and Felix Naumann. 2015. Data Profiling with Metanome. *Proc. VLDB Endow.* 8, 12 (Aug 2015), 1860–1863. https://doi.org/10.14778/2824032.2824086

[155] Hima Patel, Nitin Gupta, Naveen Panwar, Ruhi Sharma Mittal, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Srikanta Bedathur, and Vitobha Munigala. 2022. Automatic Assessment of Quality of your Data for AI. In *Proceedings of the 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*. 354–357.

[156] Jian Pei. 2020. Data Pricing–From Economics to Data Science. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3553–3554.

[157] Jian Pei. 2022. A Survey on Data Pricing: From Economics to Data Science. *IEEE Trans. Knowl. Data Eng.* 34, 10 (2022), 4586–4608.

[158] Fahad Pervaiz, Aditya Vashistha, and Richard Anderson. 2019. Examining the challenges in development data pipeline. In *Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies*. 13–21.

[159] Neoklis Polyzotis and Matei Zaharia. 2021. What can data-centric ai learn from data and ml engineering? *arXiv preprint arXiv:2112.06439* (2021).

[160] André Pomp, Alexander Paulus, Andreas Burgdorf, and Tobias Meisen. 2021. A semantic data marketplace for easy data sharing within a smart city. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4774–4778.

[161] Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2021. Are neural rankers still outperformed by gradient boosted decision trees? (2021).

[162] Aiswarya Raj, Jan Bosch, Helena Holmström Olsson, and Tian J Wang. 2020. Modelling data pipelines. In *2020 46th Euromicro conference on software engineering and advanced applications (SEAA)*. IEEE, 13–20.

[163] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, Vol. 11. NIH Public Access, 269.

[164] Sergey Redyuk. 2019. Automated documentation of end-to-end experiments in data science. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2076–2080.

[165] El Kindi Rezig, Lei Cao, Giovanni Simonini, Maxime Schoemans, Samuel Madden, Nan Tang, Mourad Ouzzani, and Michael Stonebraker. 2020. Dagger: a data (not code) debugger. In *CIDR 2020, 10th Conference on Innovative Data Systems Research, Amsterdam, The Netherlands, January 12-15, 2020, Online Proceedings*.

[166] Yuji Roh, Geon Heo, and Steven Euijong Whang. 2019. A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (2019), 1328–1347.

[167] Adam Rule, Aurélien Tabard, and James D Hollan. 2018. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[168] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. 2021. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.

[169] Sebastian Schelter, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Biessmann, and Andreas Grafberger. 2018. Automating large-scale data quality verification. *Proceedings of the VLDB Endowment* 11, 12 (2018), 1781–1794.

[170] Fabian Schomm, Florian Stahl, and Gottfried Vossen. 2013. Marketplaces for Data: An Initial Survey. *SIGMOD Record* 42, 1 (2013), 15.

[171] Vraj Shah and Arun Kumar. 2019. The ML data prep zoo: Towards semi-automatic data preparation for ML. In *Proceedings of the 3rd International Workshop on Data Management for End-to-End Machine Learning*. 1–4.

[172] Shreya Shankar, Rolando Garcia, Joseph M Hellerstein, and Aditya G Parameswaran. 2022. Operationalizing machine learning: An interview study. *arXiv preprint arXiv:2209.09125* (2022).

[173] SHREYA SHANKAR, ROLANDO GARCIA, JOSEPH M HELLERSTEIN, and ADITYA G PARAMESWARAN. 2023. "We have no idea how models will behave in production until production": How engineers operationalize machine learning. (2023).

[174] Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal* 27, 3 (1948), 379–423.

[175] Roee Shraga. 2022. HumanAL: Calibrating human matching beyond a single task. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–8.

[176] Roee Shraga and Renée J Miller. 2023. Explaining Dataset Changes for Semantic Data Versioning with Explain-Da-V. *Proceedings of the VLDB Endowment* 16, 6 (2023), 1587–1600.

[177] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

[178] Robert Tarjan. 1972. Depth-first search and linear graph algorithms. *SIAM journal on computing* 1, 2 (1972), 146–160.

[179] Ryan J Urbanowicz, Melissa Meeker, William La Cava, Randal S Olson, and Jason H Moore. 2018. Relief-based feature selection: Introduction and review. *Journal of biomedical informatics* 85 (2018), 189–203.

[180] April Yi Wang, Anant Mittal, Christopher Brooks, and Steve Oney. 2019. How data scientists use computational notebooks for real-time collaboration. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–30.

[181] April Yi Wang, Dakuo Wang, Jaimie Drozdal, Xuye Liu, Soya Park, Steve Oney, and Christopher Brooks. 2021. What makes a well-documented notebook? a case study of data scientists' documentation practices in kaggle. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–7.

[182] April Yi Wang, Dakuo Wang, Jaimie Drozdal, Michael Muller, Soya Park, Justin D Weisz, Xuye Liu, Lingfei Wu, and Casey Dugan. 2022. Documentation matters: Human-centered ai system to assist data science code documentation in computational notebooks. *ACM Transactions on Computer-Human Interaction* 29, 2 (2022), 1–33.

[183] Dakuo Wang, Justin D Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-ai collaboration in data science: Exploring data scientists' perceptions of automated ai. *Proceedings of the ACM on human-computer interaction* 3, CSCW (2019), 1–24.

[184] Tingting Wang, Shixun Huang, Zhifeng Bao, J Shane Culpepper, Volkan Dedeoglu, and Reza Arablouei. 2024. Optimizing Data Acquisition to Enhance Machine Learning Performance. *Proceedings of the VLDB Endowment* 17, 6 (2024), 1310–1323.

[185] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*. AAAI Press, 1112–1119.

[186] Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 6382–6388.

[187] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. 2023. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal* 32, 4 (2023), 791–813.

[188] Ian H Witten, Eibe Frank, Mark A Hall, Christopher J Pal, and MINING DATA. 2005. Practical machine learning tools and techniques. In *Data Mining*, Vol. 2.

[189] Wen Xia, Hong Jiang, Dan Feng, Fred Douglis, Philip Shilane, Yu Hua, Min Fu, Yucheng Zhang, and Yukun Zhou. 2016. A comprehensive study of the past, present, and future of data deduplication. *Proc. IEEE* 104, 9 (2016), 1681–1710.

[190] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems* 33 (2020), 6256–6268.

[191] Mohamed Yakout, Kris Ganjam, Kaushik Chakrabarti, and Surajit Chaudhuri. 2012. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD*. 97–108.

[192] Jian Yang and Chunxiao Xing. 2019. Personal data market optimization pricing model based on privacy level. *Information* 10, 4 (2019), 123.

[193] Arda Yenipazarli. 2021. The marketplace dilemma: Selling to the marketplace vs. selling on the marketplace. *Naval Research Logistics (NRL)* 68, 6 (2021), 761–778.

[194] Kui Yu, Xindong Wu, Wei Ding, and Jian Pei. 2016. Scalable and accurate online feature selection for big data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11, 2 (2016), 1–39.

[195] Lei Yu and Huan Liu. 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*. 856–863.

[196] Lei Yu and Huan Liu. 2004. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research* 5 (2004), 1205–1224.

[197] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, et al. 2018. Accelerating the machine learning lifecycle with MLflow. *IEEE Data Eng. Bull.* 41, 4 (2018), 39–45.

[198] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, and Xia Hu. 2023. Data-centric ai: Perspectives and challenges. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 945–948.

[199] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. 2023. Data-centric artificial intelligence: A survey. *arXiv preprint arXiv:2303.10158* (2023).

[200] Amy X Zhang, Michael Muller, and Dakuo Wang. 2020. How do data science workers collaborate? roles, workflows, and tools. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–23.

[201] Meihui Zhang and Kaushik Chakrabarti. 2013. Infogather+ semantic matching and annotation of numeric and time-varying attributes in web tables. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 145–156.

[202] Yi Zhang and Zachary Ives. 2020. Finding Related Tables in Data Lakes for Interactive Data Science. In *SIGMOD*. 1951–1966.

[203] Yi Zhang and Zachary G Ives. 2019. Juneau: data lake management for Jupyter. *Proc. VLDB Endow.* 12, 12 (2019).

[204] Jinjin Zhao, Avigdor Gal, and Sanjay Krishnan. 2023. Data Makes Better Data Scientists. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–3.

[205] Zixuan Zhao and Raul Castro Fernandez. 2022. Leva: Boosting machine learning performance with relational embedding data augmentation. In *Proceedings of the 2022 International Conference on Management of Data*. 1504–1517.

[206] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J Miller. 2019. Josie: Overlap set similarity search for finding joinable tables in data lakes. In *SIGMOD*. 847–864.

[207] Marc-André Zöller and Marco F Huber. 2021. Benchmark and survey of automated machine learning frameworks. *Journal of artificial intelligence research* 70 (2021), 409–472.

List of Figures

List of Tables

Curriculum Vitæ

Andra Ionescu

01-06-1994 Born in Caracal, Romania

Professional Experience

10/2024 – 11/2024 **Developer Relations Advocate**
DuckDB Labs Amsterdam – Amsterdam, The Netherlands

10/2019 – 01/2020 **Research Intern**
ING Nederland – Amsterdam, The Netherlands

10/2018 – 09/2019 **Software Engineer**
Maistering – Rotterdam, The Netherlands

09/2017 – 09/2018 **Java Developer**
Onior Group – Den Haag, The Netherlands

01/2017 – 08/2017 **Junior Software Engineer**
Computaris – Bucharest, Romania

07/2015 – 12/2016 **Junior Developer**
Rinf Tech – Bucharest, Romania

Education

2020 – 2024 **Doctor of Philosophy (PhD)**, Computer Science
Delft University of Technology, The Netherlands

2017 – 2020 **Master of Science (MSc)**, Computer Science
Delft University of Technology, The Netherlands

2014 – 2017 **Bachelor of Science (BSc)**, Computer Science and Engineering
Politehnica University of Bucharest, Romania

Awards

2023 **Best Demonstration Award**
The 26th International Conference on Extending Database Technology (EDBT)

Academic Service

2025	PC Member	DOLAP
2024	PC Member	TaDa
2024	Workshop and tutorial chair	DEBS
2023	Co-organiser and workshop chair	DBML@ICDE
2022	Co-organiser	XAISS
2022	Co-organiser	Alice&Eve
2022	Publicity Chair	DBML@ICDE
2021	Publicity Chair	DBDBD

List of Publications

1. **Andra Ionescu**, Zeger Mouw, Efthimia Aivaloglou, Rihan Hai, Asterios Katsifodimos, *Human-in-the-Loop Feature Discovery for Tabular Data*, in The Conference on Information and Knowledge Management (CIKM), 2024.

2. **Andra Ionescu**, Zeger Mouw, Efthimia Aivaloglou, Asterios Katsifodimos, *Key Insights from a Feature Discovery User Study*, in Workshop on Human-In-the-Loop Data Analytics Co-located with Special Interest Group on Management of Data (HILDA@SIGMOD), 2024.

3. **Andra Ionescu**, Kiril Vasilev, Florena Buse, Rihan Hai, Asterios Katsifodimos, *AutoFeat: Transitive Feature Discovery over Join Paths*, in International Conference on Data Engineering (ICDE), 2024.

4. **Andra Ionescu**, Alexandra Alexandridou, Leonidas Ikonomou, Kyriakos Psarakis, Kostas Patroumpas, Georgios Chatzigeorgakidis, Dimitrios Skoutas, Spiros Athanasiou, Rihan Hai, Asterios Katsifodimos, *Topio Marketplace: Search and Discovery of Geospatial Data*, in International Conference on Extending Database Technology (EDBT), 2023.

5. Rihan Hai, Christos Koutras, **Andra Ionescu**, Ziyu Li, Wenbo Sun, Jessie van Schijndel, Yan Kang, Asterios Katsifodimos, *Amalur: Data Integration Meets Machine Learning*, in International Conference on Data Engineering (ICDE), 2023.

6. **Andra Ionescu**, Kostas Patroumpas, Kyriakos Psarakis, Georgios Chatzigeorgakidis, Diego Collarana, Kai Barenscher, Dimitrios Skoutas, Asterios Katsifodimos, Spiros Athanasiou, *Topio: An Open-Source Web Platform for Trading Geospatial Data*, in International Conference on Web Engineering (ICWE), 2023.

7. Rihan Hai, Christos Koutras, **Andra Ionescu**, Asterios Katsifodimos, *Amalur: Next-generation Data Integration in Data Lakes*, in Conference on Innovative Data Systems Research (CIDR), 2022.

8. **Andra Ionescu,** Rihan Hai, Marios Fragkoulis, Asterios Katsifodimos, *Join Path-Based Data Augmentation for Decision Trees*, in International Conference on Data Engineering Workshops (ICDEW), 2022.

9. Christos Koutras, Kyriakos Psarakis, George Siachamis, **Andra Ionescu**, Marios Fragkoulis, Angela Bonifati, Asterios Katsifodimos, *Valentine in Action: Matching Tabular Data at Scale*, in Very Large Data Bases (VLDB), 2021.

10. Christos Koutras, George Siachamis, **Andra Ionescu**, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, Asterios Katsifodimos, *Valentine: Evaluating Matching Techniques for Dataset Discovery*, in International Conference on Data Engineering (ICDE), 2021.

11. **Andra Ionescu**, *Interactive Data Discovery in Data Lakes*, in Very Large Data Bases (VLDB) PhD Workshop, 2021.

Included in this dissertation.

SIKS Dissertation Series

Since 1998, all dissertations written by PhD. students who have conducted their research under the auspices of a senior research fellow of the SIKS research school are published in the SIKS Dissertation Series.

2016
01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines
02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow
03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support
04 Laurens Rietveld (VUA), Publishing and Consuming Linked Data
05 Evgeny Sherkhonov (UvA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers
06 Michel Wilson (TUD), Robust scheduling in an uncertain environment
07 Jeroen de Man (VUA), Measuring and modeling negative emotions for virtual training
08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data
09 Archana Nottamkandath (VUA), Trusting Crowdsourced Information on Cultural Artefacts
10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms
11 Anne Schuth (UvA), Search Engines that Learn from Their Users
12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems
13 Nana Baah Gyan (VUA), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach
14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization
15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments
16 Guangliang Li (UvA), Socially Intelligent Autonomous Agents that Learn from Human Reward
17 Berend Weel (VUA), Towards Embodied Evolution of Robot Organisms
18 Albert Meroño Peñuela (VUA), Refining Statistical Data on the Web
19 Julia Efremova (TU/e), Mining Social Structures from Genealogical Data
20 Daan Odijk (UvA), Context & Semantics in News & Web Search
21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22 Grace Lewis (VUA), Software Architecture Strategies for Cyber-Foraging Systems
23 Fei Cai (UvA), Query Auto Completion in Information Retrieval

24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach
25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VUA), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media
28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control
29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30 Ruud Mattheij (TiU), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web content monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33 Peter Bloem (UvA), Single Sample Statistics, exercises in learning from just one example
34 Dennis Schunselaar (TU/e), Configurable Process Trees: Elicitation, Analysis, and Enactment
35 Zhaochun Ren (UvA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry
38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40 Christian Detweiler (TUD), Accounting for Values in Design
41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance
42 Spyros Martzoukos (UvA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44 Thibault Sellam (UvA), Automatic Assistants for Database Exploration
45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46 Jorge Gallego Perez (UT), Robots to Make you Happy
47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
48 Tanja Buttler (TUD), Collecting Lessons Learned
49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis

	50	Yan Wang (TiU), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains
2017	01	Jan-Jaap Oerlemans (UL), Investigating Cybercrime
	02	Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
	03	Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
	04	Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
	05	Mahdieh Shadi (UvA), Collaboration Behavior
	06	Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
	07	Roel Bertens (UU), Insight in Information: from Abstract to Anomaly
	08	Rob Konijn (VUA), Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery
	09	Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text
	10	Robby van Delden (UT), (Steering) Interactive Play Behavior
	11	Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment
	12	Sander Leemans (TU/e), Robust Process Mining with Guarantees
	13	Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology
	14	Shoshannah Tekofsky (TiU), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
	15	Peter Berck (RUN), Memory-Based Text Correction
	16	Aleksandr Chuklin (UvA), Understanding and Modeling Users of Modern Search Engines
	17	Daniel Dimov (UL), Crowdsourced Online Dispute Resolution
	18	Ridho Reinanda (UvA), Entity Associations for Search
	19	Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval
	20	Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility
	21	Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
	22	Sara Magliacane (VUA), Logics for causal inference under uncertainty
	23	David Graus (UvA), Entities of Interest — Discovery in Digital Traces
	24	Chang Wang (TUD), Use of Affordances for Efficient Robot Learning
	25	Veruska Zamborlini (VUA), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search
	26	Merel Jung (UT), Socially intelligent robots that understand and respond to human touch
	27	Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
	28	John Klein (VUA), Architecture Practices for Complex Contexts

29 Adel Alhuraibi (TiU), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"
30 Wilma Latuny (TiU), The Power of Facial Expressions
31 Ben Ruijl (UL), Advances in computational methods for QFT calculations
32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives
33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics
35 Martine de Vos (VUA), Interpreting natural science spreadsheets
36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging
37 Alejandro Montes Garcia (TU/e), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
38 Alex Kayal (TUD), Normative Social Applications
39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR
40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems
41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle
42 Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets
43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval
44 Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering
45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement
46 Jan Schneider (OU), Sensor-based Learning Support
47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration
48 Angel Suarez (OU), Collaborative inquiry-based learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations
02 Felix Mannhardt (TU/e), Multi-perspective Process Mining
03 Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction
04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks
05 Hugo Huurdeman (UvA), Supporting the Complex Dynamics of the Information Seeking Process
06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems
07 Jieting Luo (UU), A formal account of opportunism in multi-agent systems
08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems
09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations

10 Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology
11 Mahdi Sargolzaei (UvA), Enabling Framework for Service-oriented Collaborative Networks
12 Xixi Lu (TU/e), Using behavioral context in process mining
13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future
14 Bart Joosten (TiU), Detecting Social Signals with Spatiotemporal Gabor Filters
15 Naser Davarzani (UM), Biomarker discovery in heart failure
16 Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children
17 Jianpeng Zhang (TU/e), On Graph Sample Clustering
18 Henriette Nakad (UL), De Notaris en Private Rechtspraak
19 Minh Duc Pham (VUA), Emergent relational schemas for RDF
20 Manxia Liu (RUN), Time and Bayesian Networks
21 Aad Slootmaker (OU), EMERGO: a generic platform for authoring and playing scenario-based serious games
22 Eric Fernandes de Mello Araújo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
23 Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis
24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections
26 Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology
27 Maikel Leemans (TU/e), Hierarchical Process Mining for Scalable Software Analysis
28 Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel
29 Yu Gu (TiU), Emotion Recognition from Mandarin Speech
30 Wouter Beek (VUA), The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

---

2019 01 Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification
02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty
03 Eduardo Gonzalez Lopez de Murillas (TU/e), Process Mining on Databases: Extracting Event Data from Real Life Data Sources
04 Ridho Rahmadi (RUN), Finding stable causal structures from clinical data
05 Sebastiaan van Zelst (TU/e), Process Mining with Streaming Data
06 Chris Dijkshoorn (VUA), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets
07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms
08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes

09 Fahimeh Alizadeh Moghaddam (UvA), Self-adaptation for energy efficiency in software systems
10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
12 Jacqueline Heinerman (VUA), Better Together
13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses
15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16 Guangming Li (TU/e), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts
18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence
21 Cong Liu (TU/e), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection
22 Martin van den Berg (VUA),Improving IT Decisions with Enterprise Architecture
23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24 Anca Dumitrache (VUA), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
25 Emiel van Miltenburg (VUA), Pragmatic factors in (automatic) image description
26 Prince Singh (UT), An Integration Platform for Synchromodal Transport
27 Alessandra Antonaci (OU), The Gamification Design Process applied to (Massive) Open Online Courses
28 Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
29 Daniel Formolo (VUA), Using virtual agents for simulation and training of social skills in safety-critical circumstances
30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31 Milan Jelisavcic (VUA), Alive and Kicking: Baby Steps in Robotics
32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
33 Anil Yaman (TU/e), Evolution of Biologically Inspired Learning in Artificial Neural Networks
34 Negar Ahmadi (TU/e), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES

	35	Lisa Facey-Shaw (OU), Gamification with digital badges in learning programming
	36	Kevin Ackermans (OU), Designing Video-Enhanced Rubrics to Master Complex Skills
	37	Jian Fang (TUD), Database Acceleration on FPGAs
	38	Akos Kadar (OU), Learning visually grounded and multilingual representations
2020	01	Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour
	02	Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models
	03	Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding
	04	Maarten van Gompel (RUN), Context as Linguistic Bridges
	05	Yulong Pei (TU/e), On local and global structure mining
	06	Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support
	07	Wim van der Vegt (OU), Towards a software architecture for reusable game components
	08	Ali Mirsoleimani (UL),Structured Parallel Programming for Monte Carlo Tree Search
	09	Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research
	10	Alifah Syamsiyah (TU/e), In-database Preprocessing for Process Mining
	11	Sepideh Mesbah (TUD), Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models
	12	Ward van Breda (VUA), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment
	13	Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming
	14	Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases
	15	Konstantinos Georgiadis (OU), Smart CAT: Machine Learning for Configurable Assessments in Serious Games
	16	Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling
	17	Daniele Di Mitri (OU), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences
	18	Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems
	19	Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems
	20	Albert Hankel (VUA), Embedding Green ICT Maturity in Organisations
	21	Karine da Silva Miras de Araujo (VUA), Where is the robot?: Life as it could be
	22	Maryam Masoud Khamis (RUN), Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar
	23	Rianne Conijn (UT), The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging
	24	Lenin da Nóbrega Medeiros (VUA/RUN), How are you feeling, human? Towards emotionally supportive chatbots

	25	Xin Du (TU/e), The Uncertainty in Exceptional Model Mining
	26	Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization
	27	Ekaterina Muravyeva (TUD), Personal data and informed consent in an educational context
	28	Bibeg Limbu (TUD), Multimodal interaction for deliberate practice: Training complex skills with augmented reality
	29	Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference
	30	Bob Zadok Blok (UL), Creatief, Creatiever, Creatiefst
	31	Gongjin Lan (VUA), Learning better – From Baby to Better
	32	Jason Rhuggenaath (TU/e), Revenue management in online markets: pricing and online advertising
	33	Rick Gilsing (TU/e), Supporting service-dominant business model evaluation in the context of business model innovation
	34	Anna Bon (UM), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development
	35	Siamak Farshidi (UU), Multi-Criteria Decision-Making in Software Production
2021	01	Francisco Xavier Dos Santos Fonseca (TUD),Location-based Games for Social Interaction in Public Space
	02	Rijk Mercuur (TUD), Simulating Human Routines: Integrating Social Practice Theory in Agent-Based Models
	03	Seyyed Hadi Hashemi (UvA), Modeling Users Interacting with Smart Devices
	04	Ioana Jivet (OU), The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning
	05	Davide Dell'Anna (UU), Data-Driven Supervision of Autonomous Systems
	06	Daniel Davison (UT), "Hey robot, what do you think?" How children learn with a social robot
	07	Armel Lefebvre (UU), Research data management for open science
	08	Nardie Fanchamps (OU), The Influence of Sense-Reason-Act Programming on Computational Thinking
	09	Cristina Zaga (UT), The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children's Collaboration Through Play
	10	Quinten Meertens (UvA), Misclassification Bias in Statistical Learning
	11	Anne van Rossum (UL), Nonparametric Bayesian Methods in Robotic Vision
	12	Lei Pi (UL), External Knowledge Absorption in Chinese SMEs
	13	Bob R. Schadenberg (UT), Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning
	14	Negin Samaeemofrad (UL), Business Incubators: The Impact of Their Support
	15	Onat Ege Adali (TU/e), Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm
	16	Esam A. H. Ghaleb (UM), Bimodal emotion recognition from audio-visual cues
	17	Dario Dotti (UM), Human Behavior Understanding from motion and bodily cues using deep neural networks

	18	Remi Wieten (UU), Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks
	19	Roberto Verdecchia (VUA), Architectural Technical Debt: Identification and Management
	20	Masoud Mansoury (TU/e), Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems
	21	Pedro Thiago Timbó Holanda (CWI), Progressive Indexes
	22	Sihang Qiu (TUD), Conversational Crowdsourcing
	23	Hugo Manuel Proença (UL), Robust rules for prediction and description
	24	Kaijie Zhu (TU/e), On Efficient Temporal Subgraph Query Processing
	25	Eoin Martino Grua (VUA), The Future of E-Health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications
	26	Benno Kruit (CWI/VUA), Reading the Grid: Extending Knowledge Bases from Human-readable Tables
	27	Jelte van Waterschoot (UT), Personalized and Personal Conversations: Designing Agents Who Want to Connect With You
	28	Christoph Selig (UL), Understanding the Heterogeneity of Corporate Entrepreneurship Programs
2022	01	Judith van Stegeren (UT), Flavor text generation for role-playing video games
	02	Paulo da Costa (TU/e), Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey
	03	Ali el Hassouni (VUA), A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare
	04	Ünal Aksu (UU), A Cross-Organizational Process Mining Framework
	05	Shiwei Liu (TU/e), Sparse Neural Network Training with In-Time Over-Parameterization
	06	Reza Refaei Afshar (TU/e), Machine Learning for Ad Publishers in Real Time Bidding
	07	Sambit Praharaj (OU), Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics
	08	Maikel L. van Eck (TU/e), Process Mining for Smart Product Design
	09	Oana Andreea Inel (VUA), Understanding Events: A Diversity-driven Human-Machine Approach
	10	Felipe Moraes Gomes (TUD), Examining the Effectiveness of Collaborative Search Engines
	11	Mirjam de Haas (UT), Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers' engagement with robots and tasks during second-language tutoring
	12	Guanyi Chen (UU), Computational Generation of Chinese Noun Phrases
	13	Xander Wilcke (VUA), Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented Knowledge
	14	Michiel Overeem (UU), Evolution of Low-Code Platforms

15 Jelmer Jan Koorn (UU), Work in Process: Unearthing Meaning using Process Mining
16 Pieter Gijsbers (TU/e), Systems for AutoML Research
17 Laura van der Lubbe (VUA), Empowering vulnerable people with serious games and gamification
18 Paris Mavromoustakos Blom (TiU), Player Affect Modelling and Video Game Personalisation
19 Bilge Yigit Ozkan (UU), Cybersecurity Maturity Assessment and Standardisation
20 Fakhra Jabeen (VUA), Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media
21 Seethu Mariyam Christopher (UM), Intelligent Toys for Physical and Cognitive Assessments
22 Alexandra Sierra Rativa (TiU), Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations
23 Ilir Kola (TUD), Enabling Social Situation Awareness in Support Agents
24 Samaneh Heidari (UU), Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values
25 Anna L.D. Latour (UL), Optimal decision-making under constraints and uncertainty
26 Anne Dirkson (UL), Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences
27 Christos Athanasiadis (UM), Emotion-aware cross-modal domain adaptation in video sequences
28 Onuralp Ulusoy (UU), Privacy in Collaborative Systems
29 Jan Kolkmeier (UT), From Head Transform to Mind Transplant: Social Interactions in Mixed Reality
30 Dean De Leo (CWI), Analysis of Dynamic Graphs on Sparse Arrays
31 Konstantinos Traganos (TU/e), Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management
32 Cezara Pastrav (UU), Social simulation for socio-ecological systems
33 Brinn Hekkelman (CWI/TUD), Fair Mechanisms for Smart Grid Congestion Management
34 Nimat Ullah (VUA), Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change
35 Mike E.U. Ligthart (VUA), Shaping the Child-Robot Relationship: Interaction Design Patterns for a Sustainable Interaction

---

2023 01 Bojan Simoski (VUA), Untangling the Puzzle of Digital Health Interventions
02 Mariana Rachel Dias da Silva (TiU), Grounded or in flight? What our bodies can tell us about the whereabouts of our thoughts
03 Shabnam Najafian (TUD), User Modeling for Privacy-preserving Explanations in Group Recommendations
04 Gineke Wiggers (UL), The Relevance of Impact: bibliometric-enhanced legal information retrieval

05 Anton Bouter (CWI), Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization, Including Real-World Medical Applications
06 António Pereira Barata (UL), Reliable and Fair Machine Learning for Risk Assessment
07 Tianjin Huang (TU/e), The Roles of Adversarial Examples on Trustworthiness of Deep Learning
08 Lu Yin (TU/e), Knowledge Elicitation using Psychometric Learning
09 Xu Wang (VUA), Scientific Dataset Recommendation with Semantic Techniques
10 Dennis J.N.J. Soemers (UM), Learning State-Action Features for General Game Playing
11 Fawad Taj (VUA), Towards Motivating Machines: Computational Modeling of the Mechanism of Actions for Effective Digital Health Behavior Change Applications
12 Tessel Bogaard (VUA), Using Metadata to Understand Search Behavior in Digital Libraries
13 Injy Sarhan (UU), Open Information Extraction for Knowledge Representation
14 Selma Čaušević (TUD), Energy resilience through self-organization
15 Alvaro Henrique Chaim Correia (TU/e), Insights on Learning Tractable Probabilistic Graphical Models
16 Peter Blomsma (TiU), Building Embodied Conversational Agents: Observations on human nonverbal behaviour as a resource for the development of artificial characters
17 Meike Nauta (UT), Explainable AI and Interpretable Computer Vision – From Oversight to Insight
18 Gustavo Penha (TUD), Designing and Diagnosing Models for Conversational Search and Recommendation
19 George Aalbers (TiU), Digital Traces of the Mind: Using Smartphones to Capture Signals of Well-Being in Individuals
20 Arkadiy Dushatskiy (TUD), Expensive Optimization with Model-Based Evolutionary Algorithms applied to Medical Image Segmentation using Deep Learning
21 Gerrit Jan de Bruin (UL), Network Analysis Methods for Smart Inspection in the Transport Domain
22 Alireza Shojaifar (UU), Volitional Cybersecurity
23 Theo Theunissen (UU), Documentation in Continuous Software Development
24 Agathe Balayn (TUD), Practices Towards Hazardous Failure Diagnosis in Machine Learning
25 Jurian Baas (UU), Entity Resolution on Historical Knowledge Graphs
26 Loek Tonnaer (TU/e), Linearly Symmetry-Based Disentangled Representations and their Out-of-Distribution Behaviour
27 Ghada Sokar (TU/e), Learning Continually Under Changing Data Distributions
28 Floris den Hengst (VUA), Learning to Behave: Reinforcement Learning in Human Contexts
29 Tim Draws (TUD), Understanding Viewpoint Biases in Web Search Results

2024 01 Daphne Miedema (TU/e), On Learning SQL: Disentangling concepts in data systems education
02 Emile van Krieken (VUA), Optimisation in Neurosymbolic Learning Systems
03 Feri Wijayanto (RUN), Automated Model Selection for Rasch and Mediation Analysis
04 Mike Huisman (UL), Understanding Deep Meta-Learning
05 Yiyong Gou (UM), Aerial Robotic Operations: Multi-environment Cooperative Inspection & Construction Crack Autonomous Repair
06 Azqa Nadeem (TUD), Understanding Adversary Behavior via XAI: Leveraging Sequence Clustering to Extract Threat Intelligence
07 Parisa Shayan (TiU), Modeling User Behavior in Learning Management Systems
08 Xin Zhou (UvA), From Empowering to Motivating: Enhancing Policy Enforcement through Process Design and Incentive Implementation
09 Giso Dal (UT), Probabilistic Inference Using Partitioned Bayesian Networks
10 Cristina-Iulia Bucur (VUA), Linkflows: Towards Genuine Semantic Publishing in Science
11 withdrawn
12 Peide Zhu (TUD), Towards Robust Automatic Question Generation For Learning
13 Enrico Liscio (TUD), Context-Specific Value Inference via Hybrid Intelligence
14 Larissa Capobianco Shimomura (TU/e), On Graph Generating Dependencies and their Applications in Data Profiling
15 Ting Liu (VUA), A Gut Feeling: Biomedical Knowledge Graphs for Interrelating the Gut Microbiome and Mental Health
16 Arthur Barbosa Câmara (TUD), Designing Search-as-Learning Systems
17 Razieh Alidoosti (VUA), Ethics-aware Software Architecture Design
18 Laurens Stoop (UU), Data Driven Understanding of Energy-Meteorological Variability and its Impact on Energy System Operations
19 Azadeh Mozafari Mehr (TU/e), Multi-perspective Conformance Checking: Identifying and Understanding Patterns of Anomalous Behavior
20 Ritsart Anne Plantenga (UL), Omgang met Regels
21 Federica Vinella (UU), Crowdsourcing User-Centered Teams
22 Zeynep Ozturk Yurt (TU/e), Beyond Routine: Extending BPM for Knowledge-Intensive Processes with Controllable Dynamic Contexts
23 Jie Luo (VUA), Lamarck's Revenge: Inheritance of Learned Traits Improves Robot Evolution
24 Nirmal Roy (TUD), Exploring the effects of interactive interfaces on user search behaviour
25 Alisa Rieger (TUD), Striving for Responsible Opinion Formation in Web Search on Debated Topics
26 Tim Gubner (CWI), Adaptively Generating Heterogeneous Execution Strategies using the VOILA Framework
27 Lincen Yang (UL), Information-theoretic Partition-based Models for Interpretable Machine Learning

28 Leon Helwerda (UL), Grip on Software: Understanding development progress of Scrum sprints and backlogs
29 David Wilson Romero Guzman (VUA), The Good, the Efficient and the Inductive Biases: Exploring Efficiency in Deep Learning Through the Use of Inductive Biases
30 Vijanti Ramautar (UU), Model-Driven Sustainability Accounting
31 Ziyu Li (TUD), On the Utility of Metadata to Optimize Machine Learning Workflows
32 Vinicius Stein Dani (UU), The Alpha and Omega of Process Mining
33 Siddharth Mehrotra (TUD), Designing for Appropriate Trust in Human-AI interaction
34 Robert Deckers (VUA), From Smallest Software Particle to System Specification - MuDForM: Multi-Domain Formalization Method
35 Sicui Zhang (TU/e), Methods of Detecting Clinical Deviations with Process Mining: a fuzzy set approach
36 Thomas Mulder (TU/e), Optimization of Recursive Queries on Graphs
37 James Graham Nevin (UvA), The Ramifications of Data Handling for Computational Models
38 Christos Koutras (TUD), Tabular Schema Matching for Modern Settings
39 Paola Lara Machado (TU/e), The Nexus between Business Models and Operating Models: From Conceptual Understanding to Actionable Guidance
40 Montijn van de Ven (TU/e), Guiding the Definition of Key Performance Indicators for Business Models
41 Georgios Siachamis (TUD), Adaptivity for Streaming Dataflow Engines
42 Emmeke Veltmeijer (VUA), Small Groups, Big Insights: Understanding the Crowd through Expressive Subgroup Analysis
43 Cedric Waterschoot (KNAW Meertens Instituut), The Constructive Conundrum: Computational Approaches to Facilitate Constructive Commenting on Online News Platforms
44 Marcel Schmitz (OU), Towards learning analytics-supported learning design
45 Sara Salimzadeh (TUD), Living in the Age of AI: Understanding Contextual Factors that Shape Human-AI Decision-Making
46 Georgios Stathis (Leiden University), Preventing Disputes: Preventive Logic, Law & Technology
47 Daniel Daza (VUA), Exploiting Subgraphs and Attributes for Representation Learning on Knowledge Graphs
48 Ioannis Petros Samiotis (TUD), Crowd-Assisted Annotation of Classical Music Compositions

---

2025 01 Max van Haastrecht (UL), Transdisciplinary Perspectives on Validity: Bridging the Gap Between Design and Implementation for Technology-Enhanced Learning Systems
02 Jurgen van den Hoogen (JADS), Time Series Analysis Using Convolutional Neural Networks
03 Andra-Denis Ionescu (TUD), Feature Discovery for Data-Centric AI