Human Interaction in Tabular Data Augmentation in Data Science Workflows

by

Zeger Mouw

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday April 26, 2024 at 10:00 AM.

Student number:	4595270	
Project duration:	September 18, 2023 – April 26, 2024	
Thesis committee:	Assistant Prof. dr. ir. Asterios Katsifodimos,	TU Delft, Thesis Advisor
	Assistant Prof. dr. ir. Efthimia Aivaloglou	TU Delft, Daily Supervisor
	PhD Cand. ir. Andra Ionescu ,	TU Delft, Daily Co-Supervisor
	Assistant Prof. dr. ir. Merve Gürel	External Committee Member

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

TU Delft

Abstract

The advancement of artificial intelligence (AI) has led to an increased demand for both a greater volume and quality of data. In many companies, data is dispersed across multiple tables, yet AI models typically require data in a single table format. This necessitates the merging of these tables and the selection of optimal features for the model, a process known as Tabular Data Augmentation (TDA). With the rapid growth of TDA, automated tools have been developed to streamline this process. However, these state-of-the-art tools often make assumptions about user workflows that may not align with the actual needs of data specialists, potentially making them efficient yet not fully user-friendly. Additionally, without thorough evaluation through user studies, these tools may overlook critical steps in the TDA process.

This thesis is divided into two main parts. The first part is dedicated to uncovering the assumptions and oversights within current TDA research through an exhaustive review of recent literature. This is followed by conducting interviews with 19 data specialists. These discussions aim to verify the identified assumptions and reveal any missing elements in state-of-the-art research. The second part focuses on creating a new tool to meet the requirements identified from validated assumptions and the gaps discovered. This tool is then subjected to evaluation interviews to assess its effectiveness.

The findings indicate that data specialists prefer a TDA tool that offers enhanced control and deeper insights into the data augmentation process. To meet these preferences, Human in the Loop AutoTDA was developed, embodying the desired functionalities. Feedback from the evaluation phase confirmed that data specialists find Human in the Loop AutoTDA suitable for their TDA workflows, marking a significant advancement in the field.

Acknowledgement

This thesis concludes my Master of Computer Science at the TU Delft. During my studies, I worked on many projects and documents. However, I have never worked on a single document alone. This thesis was the most challenging project of all. During this project, I learned many things, from extensively reading documents and fetching the correct information to conducting interviews and developing software to meet the user's needs. This is exactly what I want to do later: listen to the users and the business and develop software to meet their needs.

The first to acknowledge is Andra. She helped me every day with every problem I encountered and guided me through a project from which I had no clue what to do in the beginning. Andra, I am very glad that you were my Daily Co-Supervisor and that you worked day and night on our projects. Furthermore, I would like to thank Asterios and Fenia. Even though Asterios' calendar was always fully booked, he searched for some free time to help me with my software problems. Fenia helped me with every aspect of the empirical part of my thesis. Also, Fenia, thanks for the great snacks you always had in your office. I also thank Merve for the extra time to evaluate my thesis and defence.

I would like to thank my family for always supporting me through this project and trying to help me with parts that didn't concern any software-related topics. Also, I would like to thank my girlfriend Luna for supporting me during the hard times and making studying more fun. I love you all.

Contents

List of Figures

List of Tables

1

Introduction

AI's increasing influence in our daily lives often goes unnoticed until errors occur, highlighting the critical importance of accuracy in AI-driven decisions. A notable example of such an error occurred in 2021 when the Dutch government resigned following the "De toeslagenaffaire" scandal [20]. In this incident, an AI system erroneously labelled tax recipients as fraudsters by using the income among others as input [39], leading to significant personal and political fallout. While the ethical dimensions of AI use merit separate consideration, this event underscores the essential need for accurate AI predictions, which depend on high-quality data that accurately reflects real-world scenarios.

Organizations frequently possess extensive datasets stored across tables containing valuable information that could enhance AI decision-making. The challenge lies in efficiently consolidating this data into a unified table suitable for AI analysis. Data specialists traditionally undertake this labour-intensive process, manually sifting through tables to identify relevant data, determining the most effective ways to combine them, and assessing their performance within AI models. This task is not only time-consuming but also prone to human error.

Imagine you're setting up an essential Machine Learning (ML) model to forecast customer orders in a coffee shop. Refer to 2.2: Introduction to Machine Learning for an introduction to how ML operates. Your primary dataset for this model is indicated in 1.1. Your objective is to predict the orders for customers 5 and 6. Analyzing historical data, the model identifies that the day of the week and the moods of customers 1 and 3 closely match those of customers 5 and 6, suggesting they might choose similar items. However, the actual orders differ, with Cold Brew Coffee for one and Hot Latte for the other, hinting that the initial data might be insufficient.

Searching for a more comprehensive dataset, you review 100 tables and select two that appear beneficial: table 1.2 and table 1.3. Table 1.2 enriches your base dataset with the order dates and customers' genders linked via customer IDs. This addition allows you to understand when each order was placed and the gender of each customer.

Moreover, table 1.3 provides insights into the weather conditions on specific dates. By integrating this table with your base dataset through the order dates, you gain valuable context about the weather during each order. This new layer of data reveals that weather conditions significantly influence coffee preferences, while gender seems unrelated to order patterns and is thus excluded from refining the prediction accuracy. This nuanced approach to data integration offers a more informed basis for your ML model to predict customer orders more accurately.

During this exploration, we sought new attributes to reflect better real-world complexities—something that is probably not done correctly in" De toeslagenaffaire." This process is called Tabular Data Augmentation (TDA).

Table 1.1: Example table of coffee orders in a coffee shop

Customer ID	Day of Week	Mood	Order
1	Weekday	Happy	Hot Latte
2	Weekday	Stressed	Cold Brew Coffee
3	Weekend	Happy	Smoothie
4	Weekend	Stressed	Spiced Chai Latte
5	Weekday	Happy	?
6	Weekend	Happy	?

Table 1.2: Example table of customer information of the coffee shop

Customer ID	Date of order	Gender
1	19-02-2024	M
2	20-02-2024	W
3	24-02-2024	W
4	25-02-2024	M
5	27-02-2024	M
6	02-03-2024	W

Table 1.3: Example table of weather forecasts on specific dates

Date	Weather Conditions
19-02-2024	Cold
20-02-2024	Hot
24-02-2024	Normal
25-02-2024	Warm
27-02-2024	Hot
02-03-2024	Cold

1.1. Assumptions and Gaps in SOTA TDA

TDA represents an emerging field that has quickly gained traction, sparking widespread discussion and leading to the development of numerous tools and methodologies designed to streamline the work of data scientists. However, user assumptions regarding the TDA process have accompanied this rapid innovation. These user assumptions are critical in shaping these tools' development and intended functionality, but they are not backed by user studies.

Assumptions in the TDA domain vary widely, encompassing every aspect of the data management process—from data collection and cleaning to data integration. They are fundamental, as they directly influence the capabilities and limitations of the tools developed for TDA. When these assumptions do not align with the practical realities faced by data specialists, there's a risk of creating tools that, while theoretically impressive, fall short of expectations in real-world applications. Moreover, the absence of user studies may lead to unnoticed gaps in state-of-the-art applications. This disconnect emphasizes the importance of tool development in actual experiences and ensuring their effectiveness and utility in live projects.

1.2. Conducting Interviews and a implementing Novel Tool

In response to the identified disconnect between the theoretical assumptions underpinning TDA tools and their practical utility in the hands of data specialists, this thesis presents a novel solution to bridge this gap. Through extensive interviews with data professionals, this research has culled valuable insights into users' real-world challenges and expectations when employing TDA methodologies. Leveraging these insights, a new TDA tool has been developed, specifically designed to align with data specialists' actual workflows and correct the prevalent misconceptions about the TDA process. This tool not only embodies the correct assumptions identified through our research but also integrates advanced features and functionalities directly informed by the needs and feedback of the data science community. By focusing on the practical realities of data management and analysis, the tool offers a more effective and user-centric approach to data augmentation, enhancing the efficiency and accuracy of data-driven decision-making. The development and validation of this tool represent a significant step towards closing the gap between theoretical expectations and practical experiences in TDA, potentially setting a new standard for how data augmentation tools are designed and implemented.

1.3. Research Questions

The study is structured in two main phases. Initially, it's essential to identify which user assumptions associated with contemporary techniques do not hold. Understanding the user assumptions documented in the existing literature is a prerequisite. The subsequent goal is to verify the accuracy of these assumptions. Achieving this sets the groundwork for effectively addressing the first research question.

The following research phase concentrates on creating a new tool to bridge the identified gaps and align with the verified workflow in TDA. Utilizing the insights obtained from Research Question 1 (RQ1), we aim to define precisely what data specialists are looking for in the application, including the specific functionalities that should be incorporated.

The research questions are as follows:

Part 1 What user assumptions are invalid in the current state-of-the-art TDA tools?

- a What user assumptions are made during the research of state-of-the-art TDA?
- b How do data specialists execute their workflows within TDA, and what challenges do they encounter?

Part 2 What new tool can fit the gaps in TDA and is usable by data specialists?

- a What features and functionalities should a new approach incorporate to support data specialists' workflows in TDA?

1.4. Contributions

The contributions of this thesis are as follows:

C1: Comprehensive Overview of Assumptions in State-of-the-Art TDA Research
This thesis provides an in-depth overview of the assumptions made regarding the workflows and methodologies within the research of state-of-the-art Tabular Data Augmentation (TDA) applications, elucidating the gap between theoretical expectations and practical realities.

C2: Empirical Analysis of TDA Workflows
Through interviews with data specialists, this study presents an empirical analysis of the workflows used in TDA, identifying key practices, challenges, and discrepancies between existing assumptions and the real-world execution of TDA processes.

C3: Development of a Novel TDA Tool
A pivotal contribution of this research is developing a new tool designed to align closely with the validated workflows and needs of data specialists in TDA, crafted based on insights gained from direct engagement with professionals in the field.

C4: Evaluation and Future Directions
The thesis concludes by thoroughly evaluating the newly developed TDA tool, assessing its efficacy in streamlining data specialists' workflows and enhancing the TDA process. It also proposes future directions for further refinement and research, setting the stage for ongoing advancements in the field.

1.5. Thesis Structure

This thesis starts with the Preliminaries chapter, where essential concepts, definitions, and methodologies relevant to TDA are introduced, establishing a solid foundation for the subsequent analysis. The Background & Related Work chapter delves into the existing body of TDA research, identifying prevalent assumptions that inform the basis of our inquiry. This research is divided into two parts. The first part concerns validating user assumptions and identifying gaps in state-of-the-art applications. In this part, the Methods chapter shows the methods used for collecting the user assumptions and conducting a user study to verify these assumptions and find gaps in the state-of-the-art literature. The Results shows the results found by those methods. The first part is concluded in Part 1 Conclusions. The second part is focused on finding

a solution that fits the proper requirements and workflow of the data specialists. The chapter: Tool Support for Human-in-the-loop Data Augmentation describes how the novel tool is developed. It shows the design and explains the implementation. To verify this tool, the chapter Tool Evaluation describes how this tool is evaluated. The second part is concluded in Part 2 Conclusions. The thesis concludes with the Discussion, Limitations and Future Work chapter, where the study's findings are critically examined, discussing their implications while acknowledging the research's constraints and suggesting avenues for future research.

2

Preliminaries

In this chapter, we delve into the foundational concepts that underpin our research. Section 2.1 provides a comprehensive exploration of Relational Data, laying the groundwork for understanding the structured datasets central to our study. Following this, Section 2.2 introduces the principles of Machine Learning, offering insights into the algorithms and methodologies that drive predictive analytics and data modelling.

2.1. Introduction to Relational Data

In Chapter 1, we discussed an example involving consolidating data from multiple sources. This section will delve into the concept of relational data, providing a detailed explanation of its principles and applications.

2.1.1. Explanation of Tabular Data

Every ML model needs tabular data as input. For ML models that work with images. The data is converted to tabular data. Tabular data is a structured data format that organizes information into rows and columns, resembling a table. An example of university courses is shown in table 2.1. In tabular data:

- Every row represents a record, and each column has a specific feature of the record. Just like in this table, every row represents a university course. The column represents different attributes of that course.
- All data within a column share the same type. Column #students cannot contain a textual value or a boolean.

2.1.2. Integrating Tables

Integrating tables refers to combining data from two or more separate tables into a single, unified table. There are multiple ways to combine the data. In the scope of this project, we will only describe the most used operations: Join and Union

Table 2.1: Example table of University Courses

Course ID	Course Name	Professor	#students	Examination
MA1340	Probability and Statistics	Prof. Keegan	25	True
PH3210	Liquid Transportations	Prof. Basely	10	True
LP2222	Law and Policies	Prof. Smith	30	False
HW1234	Defense against the Dark Arts	Prof. Snape	50	False

Table 2.2: Example table of other University Courses

Course ID	Course Name	University.Professor	#Students	Examination
CS1004	Introduction to Cryptography	Prof. Turing	200	True
AR3005	Archaeology	Prof. Jones	10	False

Table 2.3: Result of a unioned table

Course ID	Course Name	University.Professor	#Students	Examination
MA1340	Probability and Statistics	Prof. Keegan	25	True
PH3210	Liquid Transportations	Prof. Basely	10	True
LP2222	Law and Policies	Prof. Smith	30	False
HW1234	Defense against the Dark Arts	Prof. Snape	50	False
CS1004	Introduction to Cryptography	Prof. Turing	200	True
AR3005	Archaeology	Prof. Jones	10	False

UNION

The UNION operation stands out for its simplicity and efficacy in aggregating data from tables with the same structure. For a successful UNION, it is imperative that each table involved has an identical number of columns, with corresponding columns across the tables holding compatible data types. This ensures that when the data is combined, each column aligns precisely, allowing for a seamless aggregation of the tables under one another. If you union table 2.1 with table 2.2, the result is 2.3.

JOIN

This operation combines rows from tables based on similar values in a related column between them. The resulting table has columns from all the tables joined in the process. The join type determines how rows from each table are matched and which are included in the result set. 4 types of joins are mainly used. Each type of join is described by an example of the join with (the left) table 2.1 and (the right) table 2.4 on the columns *Professor*.

- **Inner Join**
  An inner join selectively combines rows from two tables so that only those rows with matching values in both tables are included in the result set. Rows without corresponding matches in either table are excluded. As demonstrated in Table 2.5, executing an inner join on Table 2.1 and Table 2.4 yields a result of two rows. This outcome arises because only the *Professor* column entries for *Prof. Keegan* and *Prof. Smith* match across both tables.
- **Left Join**
  A left join combines rows from two tables, ensuring that all rows from the left table are included in the result set, regardless of whether there is a matching value in the right table. For rows in the left table with no corresponding match in the right table, the result set includes null (empty) values for the columns from the right table. Demonstrated in Table 2.6, when performing a left join between Table 2.1 and Table 2.4, all entries from the left table (*University Courses*) are retained. This includes rows for which there is no matching *Professor* in the right table (*Professors*), thereby ensuring that the dataset's integrity from the perspective of the left table is maintained, with missing matches filled in as nulls.

Table 2.4: Example table of Professor details

Professor	Age	#Classes
Prof. Keegan	56	14
Prof. Johnson	43	2
Prof. Smith	31	10
Prof. Oak	58	1

Table 2.5: Resulting table after an Inner Join

Course ID	Course Name	Professor	#Students	Examination	Professor	Age	#Classes
MA1340	Probability and Statistics	Prof. Keegan	25	True	Prof. Keegan	56	14
LP2222	Law and Policies	Prof. Smith	30	False	Prof. Smith	31	10

Table 2.6: Resulting table after a Left Join

Course ID	Course Name	Professor	#Students	Examination	Professor	Age	#Classes
MA1340	Probability and Statistics	Prof. Keegan	25	True	Prof. Keegan	56	14
PH3210	Liquid Transportations	Prof. Basely	10	True			
LP2222	Law and Policies	Prof. Smith	30	False	Prof. Smith	31	10
HW1234	Defense Against the Dark Arts	Prof. Snape	50	False			

- **Right Join**
  A right join combines rows from two tables so that all rows from the right table are included in the result set, even if there are no matching values in the left table. Where a row from the right table does not have a matching counterpart in the left table, the result set will include null values for the columns from the left table. As illustrated in Table 2.7, executing a right join between Table 2.1 and Table 2.4 ensures that every entry from the right table (*Professors*) is preserved. This process results in a comprehensive dataset that includes all the details from professors, filling in any gaps from the left table (*University Courses*) with null values for unmatched entries, thereby providing a full representation from the perspective of the right table.
- **Outer Join**
  A full outer join merges rows from two tables, ensuring that rows from both tables are included in the result set, regardless of whether there are matching values in the other table. Rows that do not have a corresponding match in either table are also included, with null values filling in for the missing data from the non-matching side. Demonstrated in Table 2.8, conducting a full outer join on Table 2.1 and Table 2.4 yields a comprehensive dataset that retains all entries from both tables. This approach ensures no information is lost; entries from *University Courses* without a matching *Professor* in *Professors*, and vice versa, are all preserved, with nulls indicating the absence of a match. This method provides a complete view by combining the inclusivity of both left and right joins, showcasing the full spectrum of data across the joined tables.

The context and objectives of the data query determine the selection of a specific join type. For scenarios requiring the inclusion of all records from a particular table, a left or right join would be appropriate, depending on which table's records need to be fully represented. Conversely, an inner join is preferable when the priority is to ensure no null values in the result set, even at the expense of excluding some rows. Alternatively, if the goal is to capture all data from both tables, accepting null values for unmatched records, then an outer join is the most suitable choice.

2.1.3. Relationships

In the examples previously discussed, each item was unique, appearing only once across both databases. However, real-world data often includes instances where items recur multiple times, leading to various inter-table relationships. Understanding these relationships is crucial for effectively managing and querying relational databases. Below, we outline the three primary relationships that can exist between tables:

- **One-to-One Relationship**

Table 2.7: Resulting table after a Right Join

Course ID	Course Name	University.Professor	#Students	Examination	Professor.Professor	Age	#Classes
MA1340	Probability and Statistics	Prof. Keegan	25	True	Prof. Keegan	56	14
LP2222	Law and Policies	Prof. Smith	30	False	Prof. Smith	31	10
					Prof. Johnson	43	2
					Prof. Oak	58	1

Table 2.8: Resulting table after an Outer Join

Course ID	Course Name	Professor	#Students	Examination	Professor	Age	#Classes
MA1340	Probability and Statistics	Prof. Keegan	25	True	Prof. Keegan	56	14
LP2222	Law and Policies	Prof. Smith	30	False	Prof. Smith	31	10
PH3210	Liquid Transportations	Prof. Basely	10	True			
HW1234	Defense Against the Dark Arts	Prof. Snape	50	False			
					Prof. Johnson	43	2
					Prof. Oak	58	1

  - **Description:** This scenario mirrors what was illustrated in the earlier examples. In a one-to-one relationship, each row in one table corresponds to exactly one row in another, ensuring a one-to-one mapping without duplicating records.
  - **Example:** Every Professor teaches only one course. There are no professors with the same name.

- **One-to-Many or Many-to-One Relationship**
  - **Description:** These relationships occur when a single row from one table associates with multiple rows in another. In a one-to-many relationship, a single row in the first table can correspond to several rows in the second table. Conversely, a many-to-one relationship might involve multiple rows in the first table matching a single row in the second. This results in the duplication of data of one of the tables in the resulting table.
  - **Example:** A professor teaches multiple courses, and no professors have the same name; Or every Professor teaches one course, but multiple professors have the same name.

- **Many-to-Many Relationship**
  - **Description:** When there is both a one-to-many and a many-to-one relationship. Multiple rows in one table could be matched to multiple rows in the other.
  - **Example:** A professor can teach multiple courses, and there are multiple professors with the same name.

2.1.4. Matching Techniques

In the sections above, we discussed values in rows that ”match”. With matching, we meant that two values are the same. However, in some cases, rows can be matched even if the values are similar but not the same.

Hard Join

The joins utilized in the preceding sections, characterized by their strict matching criteria, are called ”hard joins.” In this context, rows are only paired if their values are precisely identical—for example, ”Prof. Keegan” would only match with ”Prof. Keegan” and not with ”Professor Keegan”. This exact matching method is stringent, ensuring that only rows with perfectly matching values are joined together.

Fuzzy Join

Fuzzy joins are employed to link rows based on nearly identical values, accommodating slight variations in representation. An illustrative example is matching ”Professor Keegan” with ”Prof Keegan,” where the values closely resemble each other but are not exact matches. Various algorithms, such as the Jaccard distance [16], N-Gram similarity [16], and Hamming Distance [21], offer methods to quantify the degree of similarity between such strings, facilitating the fuzzy joining process. However, a notable challenge with fuzzy joins is the potential for erroneously linking rows that are not genuinely related due to overestimating the similarity between distinct values.

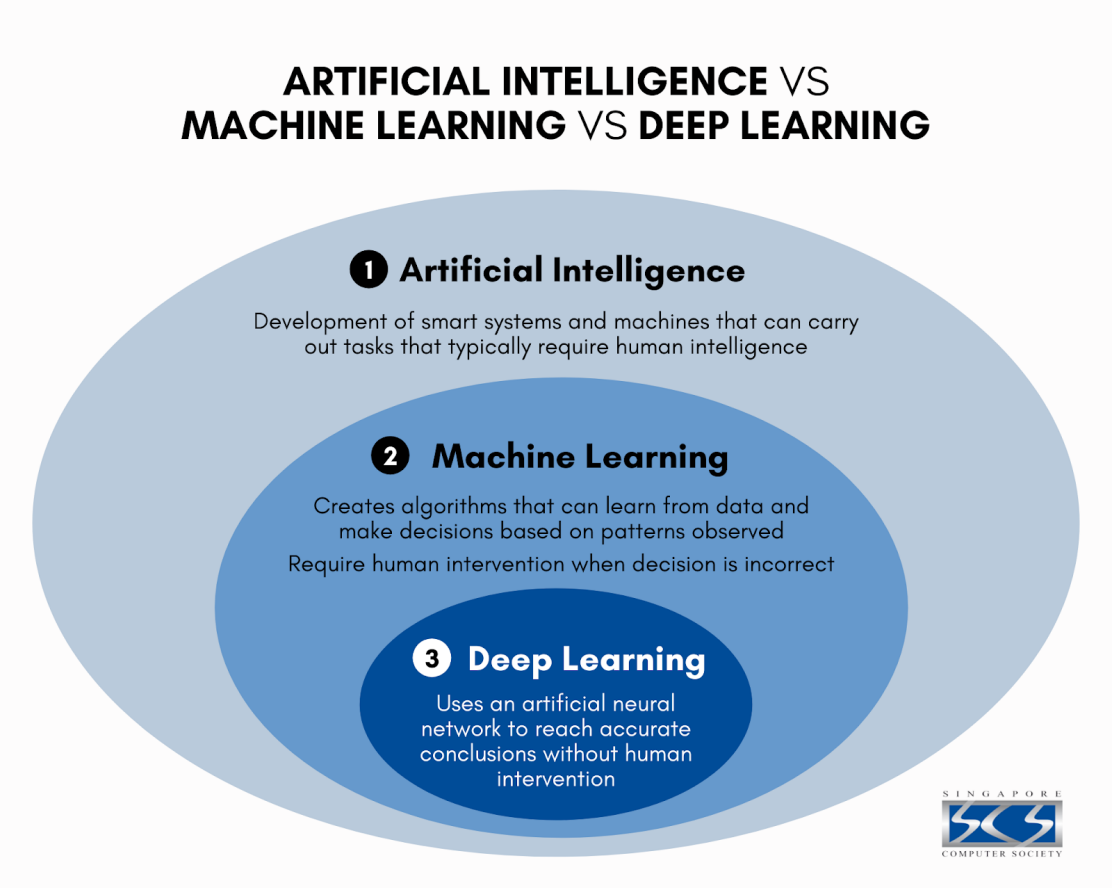


Figure 2.1: An overview of AI and ML from [43]

Semantic Join

Semantic joins enable the association of rows based on the similarity in meaning of their values, even when they appear quite distinct. For instance, ”Holland” and ”The Netherlands” represent markedly different terms in text but often carry the same meaning in various contexts. Data entries with nominally divergent values can be intelligently linked through semantic joins, acknowledging and leveraging their underlying semantic equivalence [24] [19].

2.2. Introduction to Machine Learning

As shown in figure 2.1, Machine Learning (ML) is a subset of artificial intelligence (AI) focused on building systems that learn from data. Instead of being explicitly programmed to perform a task, these systems learn patterns and insights directly from data, enabling them to make decisions or predictions based on the data they receive.

2.2.1. Types of Machine Learning

This section delves into the primary types of Machine Learning: Supervised, Unsupervised, and Reinforcement Learning. Each type employs distinct mechanisms for learning and making predictions or decisions tailored to specific tasks. By exploring these foundational categories, we can better understand how ML algorithms interpret data, adapt to new information, and solve complex problems. A visual overview of the ML techniques is shown in figure 2.2

Supervised Learning

Supervised learning is a type of Machine Learning where the algorithm is trained on a labelled dataset. This means that for each piece of data in the training set, the desired output (label) is known. The goal of supervised learning is to learn a mapping from inputs to outputs, enabling

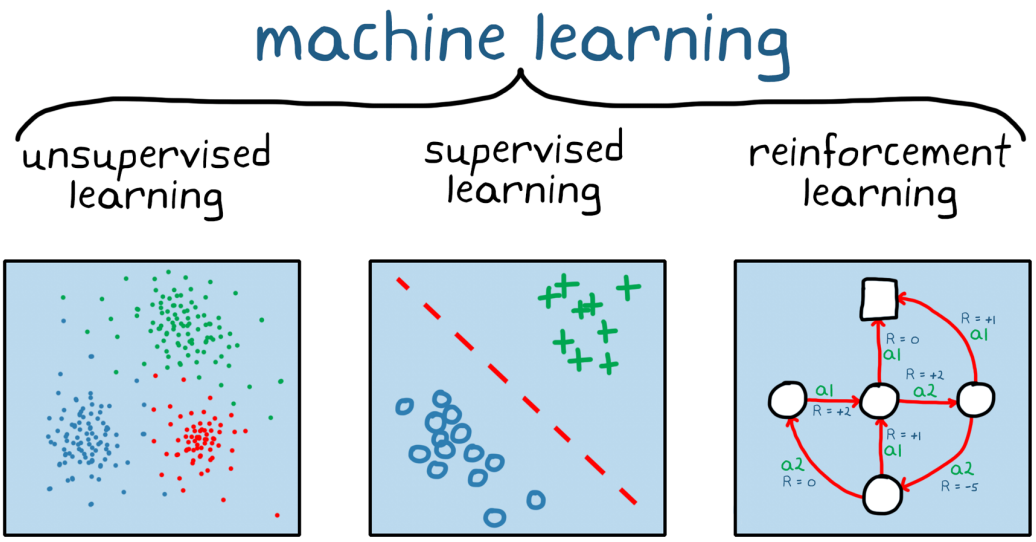


Figure 2.2: Different types of ML techniques from [45]

the model to make accurate predictions or decisions for unseen data based on this learned mapping. Supervised learning can solve a classification problem where the output variable is a category, e.g., predict a coffee order, or a regression problem where the output is a real value, e.g., a coffee's price.

Unsupervised Learning

Unsupervised learning is a type of machine learning where the algorithm is trained on data without predefined labels. In contrast to supervised learning, where each training example is associated with a label or output, unsupervised learning algorithms must find structure in the input data on their own. The primary goal is to uncover hidden patterns, correlations, or clusters within the dataset without guidance on what to look for.

Reinforcement Learning

Reinforcement Learning (RL) is a paradigm within machine learning where an agent learns to make decisions by interacting with an environment. Unlike supervised learning, which relies on a dataset with correct answers (labels), or unsupervised learning, which seeks patterns in data without labels, reinforcement learning is based on the concept of agents learning from the consequences of their actions through trial and error, aiming to maximize some notion of cumulative reward. A robot who solves a maze is an example of Reinforcement Learning.

2.2.2. Bias in ML

When engaging in supervised machine learning, the process fundamentally involves the model attempting to predict outcomes based on input data, subsequently comparing its predictions against actual outcomes to refine its learning algorithms. When datasets are merged using left join operations across tables, an unintended consequence can be the generation of duplicate rows within the resulting dataset. These duplicates can disproportionately amplify the presence of certain pieces of information, misleading the model to overvalue these repeated data points during its training phase. This misrepresentation can significantly bias the model's understanding, as it perceives these over-represented data attributes as more prevalent or influential than in the broader dataset [27]. The model's predictive accuracy may suffer, especially when faced with new, unseen data, because it has been tuned to an imbalanced dataset that doesn't accurately mirror the real-world distribution of data it's meant to interpret. This skewed training can lead to overfitting on the duplicated data points, reducing the model's effectiveness in generalizing its learned insights to make accurate predictions in varied contexts beyond its training environment. Aggregation and sampling are possible solutions to overcome this problem.

2.2.3. Null Values

When performing joins in data preprocessing for machine learning, it's common to merge datasets from different sources or tables based on a key column. While inner joins exclude rows with unmatched keys in either table, left joins and right joins include all rows from the left or right table, respectively. These joins potentially introduce empty (null) values in the dataset for unmatched keys. These null values result from the absence of corresponding data in the joined table and must be handled appropriately before training a machine learning model. The presence of null values introduced by left or right joins can significantly impact the quality of the machine learning model. If not adequately addressed, these nulls can lead to biased models, reduced accuracy, and unexpected behaviour during prediction. Therefore, it's essential to adopt strategies to manage these null values effectively. Imputation and exclusion are examples of solutions for this problem.

3

Background & Related Work

This chapter comprehensively explores the scholarly works and research findings pertinent to Tabular Data Augmentation (TDA) and its stages. It shows the academic and practical contributions that have shaped the understanding and methodologies within TDA.

3.1. End-to-End TDA applications

First, we delve into applications that offer comprehensive TDA solutions. We gain insights into the operational mechanics and methodologies underpinning TDA through an in-depth analysis of these applications.

Arda [7] presents itself as a comprehensive end-to-end system designed for feature discovery, initiating its process with a base table as input. An external data discovery system identifies and delegates potential joins, enabling Arda to manage large-scale and significantly noisy datasets adeptly. The system distinguishes between soft joins (such as a nearest neighbour, linear interpolation, and resampling) and hard joins for data integration. Arda initially generates a coreset by selectively sampling rows from the base table to enhance efficiency across all phases, significantly reducing processing time. Further optimisation is achieved through sketching techniques that streamline the dataset before formulating a join plan. Arda's execution phase skillfully addresses challenges like table grouping, handling of soft keys, managing missing values, and processing one-to-many joins. The subsequent feature selection phase employs Arda's proprietary Random Injection Feature Selection method, underscoring the system's capability to minimise user intervention and automate the feature discovery process effectively.

Like Arda, Cocoa [18] leverages an external system for identifying relevant data sources. However, while Arda requires the identification of potential joins, Cocoa streamlines the process by requiring users to provide only the base table, a specific query, and a target column. Utilising DataXformer [1], Cocoa efficiently identifies the top-k tables that are most joinable with the base table. Cocoa constructs a lightweight JoinMap for each candidate table, detailing the connections between rows across tables. This preparatory step allows the subsequent Data Augmenter component to utilise the JoinMap for assessing correlations between external and target columns. Crucially, this approach means actual data joining is reserved for only those columns demonstrating significant correlation, ensuring efficiency. The Data Augmenter then enriches the base table with these highly correlating columns from the candidate tables, optimising the augmentation process.

In their work, Ionescu et al. introduce Autofeat [22], an Automated Feature Discovery system designed to enhance the feature engineering process. The system initiates with inputs, including a base table, a target column, and the dataset's name. Crucially, relationships among data are determined in an automated fashion and are subsequently stored within a database to streamline the process. Following this preparatory step, tables are methodically integrated according to predefined join trees. A significant aspect of this integration involves assessing the increase

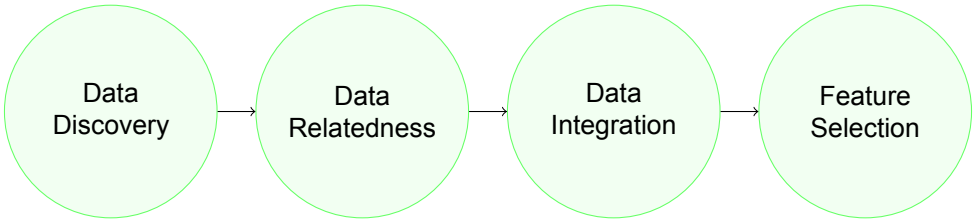


Figure 3.1: Visual Representation of the TDA stages

in null values resulting from each join. The join is discontinued if the growth is excessive to preserve data quality. Moreover, the system evaluates the added features for their relevance and non-redundancy; features failing to meet these criteria are excluded from the final dataset to avoid information overload. Ultimately, the system conducts a comprehensive evaluation of all formulated join trees, ensuring the optimisation of the dataset for further machine learning tasks.

The applications discussed in the previous section adhere to a consistent workflow [7] [18] [22], which can be systematically divided into four distinct stages as seen in figure 3.1

3.2. Research focused on Data Discovery

In today's data-driven business landscape, most companies find themselves sitting atop large data repositories, much of it siloed and disconnected, stored across numerous folders and drives. Within these sprawling data reserves lies untapped potential, as a significant portion of this data exists in tabular formats—rows and columns of valuable information that remain isolated. Dealing with sizable datasets often makes manual searching for relevant data impractical [5]. The first step in the TFD workflow, Data Discovery, addresses this challenge head-on. It involves systematically exploring these data repositories to identify and catalogue all instances of tabular data. By sifting through the digital folders, Data Discovery aims to uncover these hidden data assets, setting the stage for their transformation into actionable insights. This phase ensures that no valuable data is left unconsidered, beginning a journey from disparate data points to cohesive, integrated datasets.

In Dataset search: a survey [6], Chapman et al. show the relevant types of data and methods used. For each discipline, the search methodologies are described.

1. **Databases:** In the context of databases, the search process is highly structured, revolving around querying, planning, optimisation, and execution stages. This methodology is fundamental for structured data storage and retrieval systems, enabling efficient and precise data queries within relational databases.
2. **Information Retrieval** : Information Retrieval encompasses the search and retrieval of information from a corpus of documents or other types of objects. Using statistical and semantic techniques, IR systems can understand and match the relevance of documents to user queries.
3. **Entity-centric Search**: Entity-centric search organises information around specific entities and their relationships, enhancing search precision by leveraging structured data. It is primarily utilised within the semantic web and knowledge discovery domains, facilitating complex queries over interconnected data. Resource Description Format (RDF) is the standard model for representing and exchanging entity data.
4. **Table Search**: According to Chapman et al., table search is identified as a specific component of the broader tabular search process. While tabular search primarily focuses on the augmentation of tables, table search aims to identify tables relevant to the user's needs. This objective can be accomplished using keyword queries to find potential relevant tables.

3.3. Research focused on Data Relatedness

In the Data Relatedness phase, the focus shifts to unravelling the intricate connections between various tables identified during the Data Discovery phase. To accurately calculate and understand the relationships between tables, sophisticated methods such as COMA[11] or the Jaccard similarity are employed. These techniques are instrumental in identifying potential linkages by analysing patterns, structures, and content similarities within and across datasets. Notably, these complex calculations of data relatedness can be efficiently performed in the background without impeding the workflow of data scientists. This approach allows for the continuous analysis and updating of relationships as new data becomes available, ensuring that the model's understanding of the data landscape remains current. The results of these calculations, capturing the nuanced relationships between tables, can then be stored in a dedicated storage system.

In Valentine [28], Koutras et al. focus on the critical evaluation of schema-matching methods, aiming to assess their effectiveness and efficiency within the domain of dataset discovery. By developing Valentine, the authors facilitate large-scale, automated experiments tailored specifically to tabular data, addressing a gap in the data integration process. They examine how datasets could be relatable by unionable, view-unionable, joinable and semantically joinable relations. Through evaluating six state-of-the-art matching algorithms, such as COMA [12] and Jaccard, and a baseline method, Koutras et al. unveil the varying degrees of effectiveness these techniques exhibit across different scenarios, marking a significant step towards systematising the assessment of dataset discovery methods.

In their work on Aurum [5], Fernandez et al. constructed an Enterprise Knowledge Graph (EKG) designed to manage data relations efficiently. Their approach achieves O(n) efficiency through a two-step process, where n represents the number of columns. Initially, Aurum comprehensively reads and summarises the data into a profile, leveraging data parallelism to enhance algorithm speed. In the second step, relationships between these profiles are established, utilising LSH to attain O(n) efficiency. PK/FK candidates are also found. The system incorporates Resource Efficient Signature Sampling (RESS) to track changes and maintain the EKG. Users can employ the Source Retrieval Query Language, enabling keyword-based querying within columns or the dataset and facilitating searches for similar content.

Warpgate [8] is built with the assumption that data scientists who work with a Cloud Data Warehouse (CDW) do not have good knowledge of their data. Like Aurum, Warpgate uses LSH indexing to find similar data. On the other hand, Aurum uses relationships like similarity to find related tables, whereas Warpgate uses a semantic understanding to find related columns using column embeddings. Warpgate encodes the columns into a high-dimensional vector to find potential joins that are not immediately visible or outside the column's database. Warpgate is implemented in Sigma Workbooks to give the user a GUI.

Aurum and Warpgate are focused on joining tables together, but sometimes, you want other relationships between tables. In Finding Related Tables[10], Das Sarma et al. use relatedness to find candidate tables. They compare the tables by Entity Complement (union) and Schema Complement (join) metrics. Like Aurum and Warpgate, Finding Related Tables uses hash functions with buckets (LSH). Within these buckets, each table pair must satisfy a series of filtering conditions selected by the method of relatedness. Furthermore, the authors show that tables that don't rank high with the query table but do with the highly ranked table are often still related.

Like in Finding Related Tables, D3L [3] also researches joinable and unionable relatedness. D3L uses column names, values, formats, embeddings and domains to calculate the relatedness between tables. D3L computes a minhash for all attributes except domains, employing LSH indexes for each attribute to group columns into corresponding buckets. These grouped columns, sharing common features within a bucket, are considered candidate pairs. The distances between the five features are placed between candidate pairs in a 5-dimensional Euclidean space. With a linear regression model, the weights of the features are calculated to find the k-most related tables.

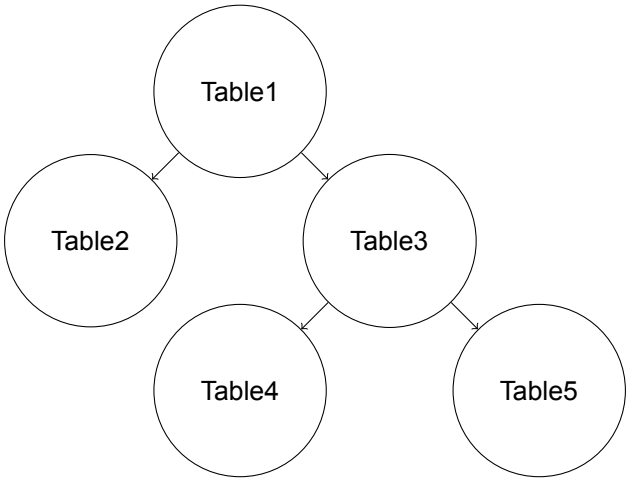


Figure 3.2: An example of an elementary Join Tree

3.4. Research focused on Data Integration

In the Data Integration phase of Tabular Feature Discovery (TFD), the groundwork laid in the Data Relatedness stage comes to fruition. Since machine learning models typically operate on a singular, consolidated table, synthesising the disparate tables into a unified dataset is necessary. Leveraging the relationships identified between tables, relationships calculated, and possibly stored during the Data Relatedness phase, this stage embarks on the strategic process of table combination. Various combinations of related tables are explored to construct potentially augmented tables, starting with a base table. When tables are merged in the process of Data Integration within Tabular Feature Discovery (TFD), this structured amalgamation is referred to as a ”Join Tree.” This terminology draws from the concept of a data structure where nodes represent the individual tables and edges denote the relationships that connect these tables. An example of a Join Tree is illustrated in 3.2

Every machine learning model necessitates tabular data for its input. After identifying potential tables for merging, the subsequent phase is to integrate these tables effectively. This section describes different methodologies for accurately amalgamating the chosen candidate tables, ensuring a cohesive dataset is prepared for modelling.

In Integrating Data Lake Tables [23], the authors created a framework called ALITE. This framework uses Full Disclosure (FD) to integrate tables. In Full Disclosure, the tables are first outer unioned. This could result in null values and duplicated values. With subsumption and complementation, respectively, these problems are reduced. Outer union is only possible when you know how to union the columns. Therefore, the method creates integration IDs using column embeddings and clustering.

Octopus [4] is a framework focused on integrating tables found on the web. It transforms the data from the tables into the correct format to ensure an equi-join is available. First, it tries to find candidate joinable row pairs. The algorithm tries to learn a transformation for each candidate joinable row. The transformation can consist of the following operators: Split, SelectK, Concat, Substring and Constant. It tries to generalise the best across all the row pairs. This results in a join that covers the most extensive set of rows.

3.5. Research focused on Feature Selection

Incorporating multiple tables into the final dataset can significantly expand its size, potentially including features already known to be extraneous for the model’s needs. Proactively eliminating these unnecessary features can lead to reductions in computational costs and enhance the model’s efficiency [33] [31] [29]. Therefore, selecting features is essential when training an ML model. There are three methods for selecting features. The filter method [32] selects features

based on statistics, the wrapper method uses learning models to choose its features [34], and the embedded method uses both.

In Efficient Feature Selection via Analysis of Relevance and Redundancy [47], a separation is made between feature relevance and feature redundancy. Because of this separation, an efficient way of creating a subset First, the framework checks for strongly relevant, weakly relevant and irrelevant features. It makes a subset of the strongly and weakly relevant features. Afterwards, the framework tries to eliminate redundant features. It removes features that are weakly relevant and have a strong correlation with other features.

AutoFeature [34] is a system which uses Reinforcement Learning to select features. It uses an exploration-exploitation strategy with trial and error to choose features that have performed well and have not yet been established. The system uses two algorithms to execute this strategy. The Multi-armed Bandit (MAB) pulls tables and calculates the score. Each iteration rewards or penalises a table. The algorithm uses an Upper Confidence Bound to prevent a local optimum. The Deep Q Network (DQN) uses a neural network to select the tables and features. Whereas the MAB is static and uses the same selection strategy, the DQN is more flexible in changing its approach.

Part I

Part 1: Validating User Assumptions

4

Methods

Tabular Data Augmentation represents a rising area of research that, despite its novelty, has developed sophisticated tools informed by assumed workflows, actions, and preferences of specialists engaged in TDA tasks. While foundational to current state-of-the-art applications, these assumptions have not been empirically validated through user studies. To ensure these tools accurately reflect the needs and practices of professionals in the field, it is imperative first to identify, categorise, and critically analyse these underlying assumptions. This chapter is dedicated to undertaking this essential process, aiming to bridge the gap between theoretical assumptions and practical utility in TDA.

4.1. Strategy for Collecting and Analysing Assumptions in state of the art Literature

To adequately explore the Tabular Data Augmentation (TDA) landscape and its applications, we identify assumptions highlighted in contemporary literature, explicitly targeting works published from 2015 onwards. This cutoff is strategically chosen to balance the inclusion of current insights while acknowledging the rapid pace of technological advancement in software and data systems. By imposing this boundary, we aim to filter out outdated assumptions that may no longer hold relevance due to advancements in data processing technologies and methodologies. On the other hand, setting the boundary too recently could omit valuable insights that remain applicable despite not being the latest developments. 2015 serves as a pragmatic threshold, recognising that while the past nine years have witnessed a significant evolution in data systems, they have not necessarily ushered in revolutionary changes that completely invalidate prior research. Our investigation begins with a detailed examination of the literature outlined in 3: Background & Related Work, focusing exclusively on studies published in or after 2015.

For instance, our analysis begins with the study ”ARDA: Automatic Relational Data Augmentation for Machine Learning” by Chepurko et al. [7]. Our examination documents every user-related assertion that is not supported by empirical research. Our methodology for navigating the literature includes a strategic approach to paper review: we invariably start with the introduction, which sometimes contains assumptions that motivate the study. Following the introduction, we survey each chapter’s title and introductory sections, vigilant for indications that the content pertains to the TDA workflow. Should we identify hints that a chapter discusses aspects of the TDA process, we deduce that it may also harbour user assumptions regarding this workflow, prompting a thorough review of said chapter.

We systematically apply this methodology to all the studies in the Background & Related Work. Upon reviewing these ten papers, we have identified 16 distinct assumptions. Given the relevance of the literature in the Background & Related Work to our research topic, it stands to reason that works cited in these papers could also hold valuable insights and potentially harbour assumptions about users within the TDA workflow. Consequently, we examine the Related

Table 4.1: Number of assumptions found per literature paper

Authors, Title	**Number of Assumptions found**
Kumar et al., To Join or Not to Join?: Thinking Twice about Joins before Feature Selection	9
Fernandez et al., Aurum: A Data Discovery System	6
Chepurko et al., ARDA: automatic relational data augmentation for machine learning.	3
Zhang & Ives, Finding Related Tables in Data Lakes for Interactive Data Science	3
Khatiwada et al., SANTOS: Relationship-based Semantic Table Union Search	3
Cong et al., Warpgate: A Semantic Join Discovery System for Cloud Data Warehouses	3
Eberius et al., Top-k Entity Augmentation using Consistent Set Covering	3
Esmailoghli et al., COCOA: Correlation Coefficient-Aware Data Augmentation	2
Zhu et al., JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes	2
Abedjan et al., DataXFormer: A robust transformation discovery system	2
Liu et al. Feature Augmentation with Reinforcement Learning	1
Khatiwada et al., Integrating Data Lake Tables	1
Sha et al., Are Key-Foreign Key Joins Safe to Avoid when Learning High-Capacity Classifiers	1
Das Sarma et al., Finding related tables	1
Nargesian et al., Table union search on open data	1
Yu et al., Feature Selection for high-dimensional data: A fast correlation-based filter solution	1

Work sections for each piece of literature we've initially reviewed, extending our search to include the references cited therein. This expansion of our literature base is undertaken using a snowballing technique, which, though potentially endless, necessitates a predetermined point of convergence to halt the expansion. We establish this stopping criterion as the moment no new assumptions—or, more precisely, no assumptions with novel implications—are unearthed. This ensures a thorough yet focused review, allowing us to capture a comprehensive spectrum of user assumptions without the process becoming unbounded.

Upon reaching the convergence point in our research, we identified 43 assumptions regarding user interactions within the TDA workflow. These assumptions were distributed across 16 distinct papers. The detailed compilation of these assumptions is presented in table 4.1, offering a consolidated view of the prevalent notions and insights derived from the current academic discourse on this subject.

4.2. Goals of the User Study

The previous chapter has unveiled that the Tabular Data Augmentation (TDA) workflow has numerous undiscovered aspects. Initially, it is essential to validate the assumptions made therein. Moreover, existing tools for TDA presuppose a standardised workflow among data specialists, a premise yet to be substantiated through research. Consequently, there is a keen interest in elucidating the actual workflow adopted by these professionals. It is conceivable that significant insights were overlooked in the existing literature on TDA, prompting a need to identify any prevailing gaps. Most crucially, a question of practical utility arises: would data specialists prefer to leverage a tool for conducting TDA? Answering this question will enable us to determine the improvements necessary for making state-of-the-art applications more accessible and valuable for data specialists. Thus, our research objectives are to:

1. Determine which user assumptions about TDA, as documented in the literature, do not hold.
2. Identify the authentic workflow followed by data specialists in TDA.
3. Uncover any omissions or gaps present in the TDA literature.
4. Assess the willingness of data specialists to adopt a tool for TDA and understand the improvements required to meet their needs.

4.3. Empirical Method Selection

To gather the necessary data and achieve our research objectives, it is imperative to develop a study design tailored to draw out detailed responses to our inquiries. This section describes the methodology employed in the study's design, ensuring a structured approach to collecting insights to address our goals.

4.3.1. Quantitative or Qualitative Methods

Given the nature of our research objectives, which seek to delve into the details of data specialists' workflows and preferences, a qualitative approach is more suitable than a strictly quantitative one. Quantitative data, characterised by numerical and measurable variables, might not fully capture the depth and nuance required to understand the practices and attitudes towards TDA tools. For instance, a simple yes/no response to whether a data specialist would rely on a TDA tool lacks the richness of context and reasoning that qualitative data can provide. Similarly, exploring the workflow of a data specialist during TDA cannot be effectively encapsulated through quantitative means alone.

Therefore, our study leans towards a qualitative methodology to gather comprehensive, text-based data that offers insights into data specialists' thought processes, experiences, and specific needs. However, this emphasis on qualitative research does not preclude including quantitative elements. Questions like "With how many tables do you normally work?" are quantitative inquiries within a primarily qualitative framework. Such questions add valuable context and a measurable dimension to the otherwise qualitative exploration, enriching our understanding of data specialists' TDA practices.

4.3.2. Candidate Empirical Methods

It is essential to analyse various empirical methods suited for qualitative data collection. This section provides an overview of potential qualitative methodologies and elaborates on the rationale behind selecting a specific approach.

- **Survey** [9] [14]: Surveys, commonly implemented through questionnaires, stand out as a prevalent method for data collection. They can also be conducted as structured interviews, wherein a large group of participants is presented with identical questions. This approach allows for the aggregation of responses, creating conclusions based on the frequency and patterns of answers relative to the total population size. However, a notable limitation of surveys, particularly in their questionnaire format, is their restrictiveness. Respondents are confined to the options provided, and the opportunity for spontaneous, in-depth exploration through follow-up questions is absent. This can constrain the depth of understanding, especially in qualitative research, where the nuances of participant experiences and perspectives are invaluable.
- **Interviews** [41] [14]: Interviews are a versatile method for qualitative data collection, offering various formats that cater to different research needs and objectives. These formats include structured, semi-structured, and unstructured interviews, each with its unique approach and level of control over the conversation.
  - **Structured Interviews**: In this format, the interviewer adheres closely to a predetermined script, asking each participant the same set of questions in the same order. The structured interview is highly controlled, with little to no deviation from the script. This method ensures consistency across interviews, making it easier to compare re-

sponses [13]. However, its rigidity may limit the depth of insight into complex or nuanced topics since it doesn't allow follow-up questions that could explore unexpected answers in greater detail.

  - **Semi-Structured Interviews**: Semi-structured interviews balance structured and unstructured formats. While the interviewer has a set of prepared questions, there is flexibility to diverge from the script based on the interviewee's responses.[2] This approach allows for exploring topics that arise naturally during the conversation, facilitating a deeper understanding of the participant's perspectives, experiences, and insights. [13]
  - **Unstructured Interviews**: Unstructured interviews are the least controlled, characterised by open-ended questions that serve as conversation starters rather than a predetermined list of questions to be followed rigidly. The interviewer follows up with questions based on the interviewee's answers, allowing the conversation to flow more naturally and unpredictably. This format is particularly useful for exploring personal opinions, memories, or impressions, offering rich, detailed data that might not emerge in more structured settings. [13]

- **Case study/Observation** [2] [9] [14] [41]: During a case study, also known as Participant Observation, participants engage in activities that simulate real-world scenarios they are meant to execute. This method collects data on the participant's behaviour for in-depth analysis. It is essential for participants to verbalise their thoughts and reasoning processes throughout the activity—known as "thinking aloud"—to capture their cognitive experiences, challenges, and problem-solving strategies. One of the primary advantages of this approach is its ability to provide rich, contextual insights into the practical application of skills, decision-making processes, and workflows. The disadvantage is that data analysis gathered from participant observation is labour-intensive and time-consuming. It requires a detailed examination of the qualitative data collected, often leading to prolonged data processing and interpretation periods. Furthermore, case studies are open to researcher bias. The researcher's perspectives, expectations, and preconceptions can influence the analysis and conclusions.
- **Experiment** [9] [14]: Experiments are conducted in a thoroughly controlled environment, allowing researchers to manipulate one or more variables to explore their effects and test hypotheses concerning these variables. This method divides a sample from the research population into groups, where each group is exposed to different conditions or variable outcomes. By maintaining control over all other variables, researchers aim to isolate the effects of the manipulated variables, thereby enabling a detailed analysis of cause-and-effect relationships. One significant challenge is the potential for results to occur by chance, leading to incorrect assumptions about causality.

4.3.3. A Mix of Empirical Methods

When undertaking empirical research, employing a singular method is not a requisite [14] [9]. Integrating multiple methodologies can enrich the depth and breadth of findings. For instance, interviews can be incorporated into experiments or case studies, providing a diverse approach to data collection. This combination allows researchers to observe behaviours and outcomes in a controlled or real-world setting and delve into the participants' perceptions, experiences, and rationales through direct questioning.

4.3.4. Choice for an Empirical Method

Given our study's objectives, combining a case study and interviews is the most suitable approach. This hybrid methodology allows us to capture the workflow of data specialists in a dynamic, real-world context. Through the case study, we can observe participants navigating tasks that simulate actual TDA processes, enabling us to gather comprehensive insights into their routines, strategies, and challenges.

Incorporating interviews into the case study enhances the depth of our data collection. By engaging participants in dialogue during the case study, we can delve deeper into their thought

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 4.1: A visual description of the statistics about the use case scenario.

processes, asking targeted questions that explain their actions and decisions. This interactive element allows us to explore aspects of their workflow that might not be observable through the use case alone.

4.4. Design of the User Study

Engaging them with a practical TDA challenge is essential to ensure a thorough analysis of data specialists' workflows in Tabular Data Augmentation (TDA). A frequently referenced TDA problem within academic literature involves augmenting a table from the "school" dataset, as noted in works by Chepurko et al. [7] and Liu et al. [34]. This dataset, accessible through a specified online repository [1], comprises 17 tables, with the "base" table serving as the primary focus for augmentation. Figure 4.1 visually represents the use-case scenario.

The task for participants involves enhancing the "base" table to optimise it for a binary classification task. This problem is presented with minimal guidance to encourage participants to apply their knowledge, strategies, and creative thinking to the augmentation process. The instructions provided are designed to be open-ended, allowing participants to approach the problem in a manner that reflects their natural workflow and decision-making processes. This setup aims to simulate a real-world scenario as closely as possible, thereby facilitating an authentic examina-

[1] https://surfdrive.surf.nl/files/index.php/s/OnjF1It4JWh5Qy8

tion of their TDA methodologies and techniques. The given instructions to the participants are as follows:

”You are given a dataset with 17 tables. One of the tables is our target table, which we want you to augment with more features. We call this table ”the base table”, which contains a target feature with classes for binary classification. Your task is to add features from the remaining 16 tables, increasing the accuracy of a tree-based ML model. The base table is called ”base.csv”. You can consider all the features in the base table as good features to keep with the target feature ”class”. It is important that you talk during your process and describe every step in detail.”

Participants can ask questions about the query table, crucial for understanding the machine learning (ML) model prediction objective. This allowance ensures participants have a clear grasp of the task's goal, aiding in the design of their augmentation strategy for the ”base” table. However, to maintain the study's integrity and closely mimic real-world scenarios where data specialists may have limited information, participants are restricted from asking questions about the other data within the dataset. This approach helps highlight the creativity, resourcefulness, and strategic thinking employed by data specialists when augmenting data under realistic constraints.

To maximise the information gathered from participants while respecting their time constraints, we decided to limit the duration of each meeting to an hour. This approach balances the need for detailed insights with the practicality of participant availability and willingness to contribute to our study.

4.4.1. Hypothetical Workflow and Questions

To gain a comprehensive understanding of the participants' thought processes and the challenges they encounter during the case study, it's crucial to engage them in a dialogue that encourages them to articulate their reasoning and decision-making. This approach is essential for capturing the cognitive aspects of their workflow that are not immediately observable through actions alone. Therefore, we plan to ask targeted questions throughout the use case to stimulate ”thinking aloud” from the participants. If a participant becomes silent or overly focused, prompting questions such as ”What are you doing now?” or ”Why are you doing this?” will be employed to maintain a continuous stream of verbal feedback.

Given the time-sensitive nature of the case study and the impracticality of formulating questions on the fly, we will pre-arrange our questions based on the stages of a hypothetical in-depth workflow tailored to the TDA problem. These stages will be almost similar to the categories of assumptions previously collected. This alignment allows for a more efficient and structured analysis of the assumptions post-case study by correlating the insights gathered during both phases.

The outlined hypothetical workflow, depicted in 4.2, serves as a guide for structuring our questions. For each step of the workflow, we will develop specific questions designed to gather detailed responses that reveal the participants' approach, thought processes, and any difficulties they encounter.

- **Query Data Understanding**: In this first stage, the assumption is that participants will initiate their task by examining the base table closely, focusing on understanding the query table designated for this specific TDA challenge. Participants are expected to examine the columns of the base table to understand its structure, content, and underlying semantics.
  - **Do you normally have column names that make sense or have names like C1, C2 or some other jibberish naming? And what do you do in this case?**
  - **Do you work with CSV files or something else?**
  - **(Assuming they found the ID DBN: How do you know DBN is the ID?**
  - **How is the ID named usually in your data? How do you know it is the ID?**
- **Input Size**: As participants begin their examination of the dataset, we anticipate a systematic approach where they assess the tables individually, a process that transitions into the

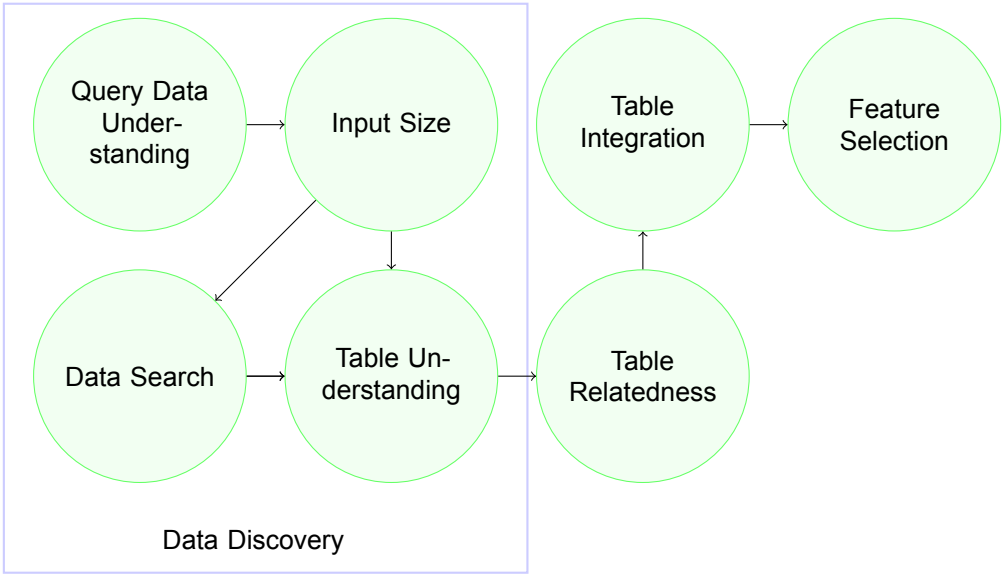


Figure 4.2: In-Depth Hypothetical Workflow of TDA

Table Understanding phase. The feasibility of such a detailed review is primarily attributed to the dataset's small size, containing 17 tables, which allows for an in-depth exploration without overwhelming the participants. However, we introduce a hypothetical scenario by asking, "What would you do if you had a thousand tables?" This question considers strategies for managing a significantly more extensive and more complex dataset. This will shift their focus towards the Data Search phase.

  - **If you were given a thousand tables, would you do the same? What would you do differently?**

- **Data Search**: In this phase, we anticipate that participants will describe a specific strategy for managing a dataset containing many tables. Their response to the hypothetical scenario will offer valuable insights into their capacity for scalability and efficiency in data management. After describing their approach, the participants will receive a selection of candidate tables. These tables, identified during the Data Search phase as potentially containing relevant features for the task, are highlighted in yellow in figure 4.1. This will streamline the process and conserve time.

  - **If you were given a thousand tables, how would you search?**
  - **Do you have a catalogue or something with more information about the tables?**
  - **What input would you use to search?**
  - **Do you ever search for keywords in the data, not the columns?**
  - **What metrics of relevance do you use?**

- **Table Understanding**: At this stage, with the selected set of candidate tables highlighted, participants are expected to examine these tables to determine the type and nature of the information each contains. This critical analysis concerns the data and the insights and implications that the data may hold for the TDA task.

  - **What is your approach? Now that you have a small set of candidate tables?**
  - **What are you looking for?**
  - **How much time are you spending on exploration? (% of data pipeline)**

- **Table Relatedness**: At this step, participants are expected to shift their focus towards determining the relationships between the candidate tables and the base table. This step is

crucial for identifying how these tables can be integrated to augment the base table effectively. Participants might employ various methods to discern these relationships, including using similarity metrics that quantitatively assess the degree of correspondence between datasets or leveraging domain knowledge to infer connections based on the content and context of the tables.

  - **How did you compute your similarity score? What does it mean?**
  - **Why do you think the tables are related? Would you do the same if the table had one million rows?**

- **Table Integration** At this step, participants are poised to actualise the integration of selected tables with the base table, a crucial step in the data augmentation process. Having identified relevant tables in the Table Relatedness phase, the focus now shifts to the technical aspect of merging these tables into a cohesive dataset; identifying the correct join keys results in combining data from different sources to maintain data integrity and relevance. The integration task is straightforward in our scenario, with a shared primary key, ”DBN,” across all tables.

  - **What tool or method do you use to join the tables?**
  - **What type of join do you use? (Inner, left, equi, soft)**
  - **How do you know the join result is correct?**
  - **If they do left joins. What do you do with duplicates?**
  - **Do you normally work with PK-FK connections?**
  - **After the first join, what is the next step?**
  - **What do you do with the rest of the tables?**
  - **What happens if you have too many tables to join?**

- **Feature Selection**: Following the successful integration of tables into a comprehensive dataset, participants enter the phase of Feature Selection. This stage involves examining the expanded dataset, which is now rich with many features resulting from the previous integration. The task is to discern which of these features is most likely to enhance the predictive capability of the machine learning model. They can achieve this by using quantitative analysis, which involves calculating scores, or by using domain expertise, which requires knowledge about the dataset.

  - **Why do you think these features are useful for the training model?**
  - **Is there something more that you would do before training?**

It is important to note that the outlined workflow is hypothetical and guides anticipated participant actions. We recognise that participants may adopt different approaches based on their expertise and preferences. Such deviations are valuable, offering unique insights into alternative data augmentation strategies and workflows.

However, to ensure a comprehensive exploration of the TDA process within the constraints of the study’s timeframe, participants who spend excessive time on a particular step may be gently prompted to proceed to subsequent stages. This approach balances the need for detailed observation with the broader goal of understanding the overall TDA workflow, ensuring that the study yields insights across all critical phases of the data augmentation process.

4.4.2. Follow-up Questions

In addition to the structured exploration of the Tabular Data Augmentation (TDA) workflow, our study incorporates a set of follow-up questions designed to enrich our understanding of hypothetical ideal scenarios and the participants’ education and experiences.

Ideal Scenario

These questions probe the participants' conceptualisations of an optimal environment for Tabular Data Augmentation (TDA) and explore their desires for tools, resources, or conditions to significantly enhance their efficiency, accuracy, or satisfaction in performing TDA tasks.

- **In an ideal world, what would make your work on this task easier?**
- **What would be an ideal tool to have? What features should the tool have?**
- **Would you rely on a tool which does the joins and selects the features for you automatically?**
- **What other problems do you encounter in your data pipeline?**

Education and Experience

Understanding our participants' educational background and professional experience is vital for accurately assessing the validity and applicability of our study's results. By gathering detailed information on the participants' levels of education, areas of study, and the extent of their practical experience with data analysis, machine learning, and, specifically, Tabular Data Augmentation (TDA), we can better contextualise their performance and the insights derived from the study.

Education:

- **What is your most recent degree?**
- **What is the major (or minor, if applicable)?**

Experience:

- **What is the industry sector of the company where you currently work for?**
- **What is the company size? (Roughly)**
- **What is your current role in the company?**
- **What is your official title in the company? And is this what you are doing?**
- **For how long have you been working with data? (all professional experience)**
- **When was the last time you had such a data augmentation/integration task?**
- **Who gives you the tasks?**
- **Who do you communicate with? At what phase and why?**
- **Do you usually understand all the data that you work with? (Quantify this in a percentage)**

4.5. Setting up the Environment for the Study Execution

To ensure the integrity and validity of our research, we have to imitate the real workflows of the data specialists as closely as possible. This means that the data specialists have to use their preferred tool and their preferred environment and that they can execute their workflow normally. This section outlines the strategic approach and considerations to achieve an optimal research environment, aiming to facilitate a focused data collection process.

The case study and interviews are facilitated through video conferencing on Microsoft Teams [2], leveraging its screen-sharing feature to gain a direct view of participants' approach in addressing the use case. Additionally, Microsoft Teams enables video and audio recording during these sessions. This capability is invaluable as it allows us to propose questions without the immediate need for note-taking. Consequently, these recordings can be reviewed and analysed post-meeting to extract and assess the insights and thought processes shared by the participants. After transcriptions were created, the participants' recordings were deleted to ensure their privacy.

[2] https://www.microsoft.com/en-us/microsoft-teams

4.5.1. Strategies for Conducting the User Study

Several strategies will be implemented to ensure the study is conducted unbiasedly and encourage participants to share their thoughts and workflows freely. These approaches create an environment where participants feel comfortable and uninfluenced in their responses.

- It is important to clarify that there are no right answers. Software developers sometimes mistakenly believe that anyone coming to interview them (or observe them) is there to evaluate them. [41] [13]
- Capture the interest of the potential interviewee with a brief explanation of your research, and if appropriate, send them the interview schedule. [40]

4.6. Participant Collection Approach

For our study, we aimed to recruit participants who were ideally engaged with tabular data augmentation (TDA) in their daily tasks. Given TDA's specificity and novelty, finding such participants proved challenging. Therefore, we broadened our criteria to include individuals with expertise in any aspect of data workflow.

We sourced participants through our professional networks and social media channels. Additionally, we employed the snowball sampling method, as outlined by Warren et al.[44] and Rowley et al. [40], where existing participants recommended other potential contributors. Our social media outreach involved posts calling for data experts to participate in our study, highlighting the voluntary nature of their involvement and its significance to academic research. We encouraged sharing these posts to reach a wider audience.

Identifying the ideal number of study participants is a complex task. Rowley et al. [40] suggest that around 12 participants typically constitute a perfect sample size. Nevertheless, we opted to define our criterion for adequate data collection based on data saturation. This approach means we continue recruiting and interviewing until no new information emerges from additional conversations or until analysing further interviews becomes impractical within our time constraints.

4.7. Processing the Interviews

After transcribing the interviews, we analysed them, employing a methodology to achieve consensus on thematic labels. Initially, we identified key labels corresponding to distinct steps in the workflow, as outlined in Table 4.2. This process began with two researchers analyzing the first interview separately, utilizing the qualitative analysis tool ATLAS.ti[3]. This step involved assigning labels to specific quotes and then jointly reviewing these labels to resolve any discrepancies.

Once a consensus on the labels was reached, we applied them to the subsequent interviews. During this phase, we also encountered statements that did not neatly fit into predefined workflow steps. These instances were classified according to their inherent significance, thus allowing for a more nuanced understanding of the data.

This approach of coding and thematic convergence proved instrumental in synthesizing the viewpoints of our participants, thereby enabling a unified examination of shared experiences and perceptions within our dataset.

[3] https://atlasti.com

5

Results

5.1. Cataloging Assumptions: Insights and Illustrations

This section will provide some examples of assumptions and highlight why they might be relevant to validate. All the assumptions are displayed in the appendix: A

In ARDA [7], for instance, we discovered the following assumptions:

> ”Moreover, there might be hundreds of related tables, creating much work for the user.”

This assumption suggests that users manage extensive datasets comprising numerous tables, underlining the utility of an automated TDA process that streamlines the workflow. Such automation would alleviate the need for manual review of each table, saving significant time and effort. If this assumption does not hold, it implies that users might favour a hands-on approach to the TDA workflow, particularly if the dataset is manageable in size.

> ”Traditionally, a user has had to invest significant effort in deciding what data can usefully augment a classification problem.”

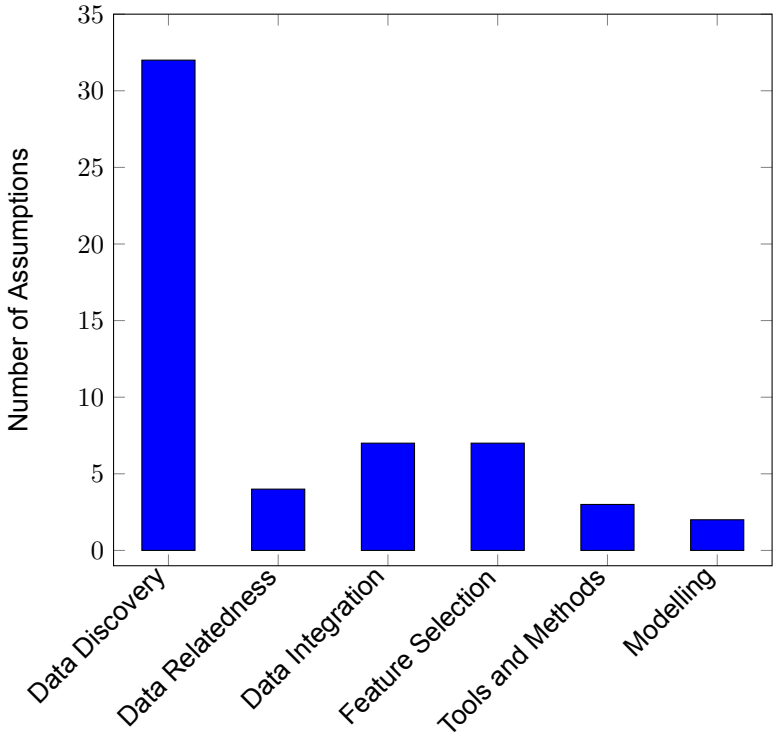
This assumption argues that users face challenges in determining the usability of their data, which could be due to a lack of understanding of the data or the overwhelming volume of it. If accurate, this scenario underscores the value of an automated TDA system that can make integration decisions, potentially revealing new insights by efficiently combining data sources. Conversely, suppose data specialists can quickly and confidently decide on the utility of their data. In that case, the data-relatedness and integration process may be quick, diminishing the necessity for an automated TDA tool.

> ”Users often have a hard time distinguishing a semantically meaningful join from a meaningless join if they don’t know details about the join itself;”

This assumption suggests that users struggle to link tables without detailed knowledge of the data relationships. If accurate, users must understand specific details about the data beyond mere column names to execute confidently joins that enhance their dataset. This requirement would encompass a deeper insight into values, data types, distribution patterns, and uniqueness within the data. On the contrary, if this assumption does not hold, it indicates that users are adept at performing joins effortlessly without requiring extensive background information on the datasets in question.

Categorising these assumptions proves invaluable during analysis, facilitating the identification

Figure 5.1: This figure shows the number of assumptions found and categorised by the TDA workflow stage and other aspects



of gaps, patterns, differences, and similarities within the TDA workflow. This systematic approach allows us to pinpoint which workflow segments contain the most assumptions, offering insights into potential areas for significant improvement. Moreover, this categorisation simplifies correlating assumptions with specific statements from data specialists during interviews. By tackling the analysis category by category, we can streamline our investigation, making it more manageable and focused and ultimately enhancing our understanding of where the TDA process can be optimised to meet the needs of its users better.

To determine these categories, we used the stages explained in 3: Background & Related Work:

1. Data Discovery
2. Data Relatedness
3. Data Integration
4. Feature Selection

While collecting the assumptions, we identified three assumptions related to tools and methods and two concerning modelling. The distribution of assumptions across these four stages is depicted in 5.1. Notably, the Data Discovery stage encompasses many assumptions, indicating a need for more categorisation within this area. To address this issue and enhance the clarity and utility of our analysis, it becomes imperative to develop subcategories for Data Discovery. By breaking down this stage into more specific domains, we can better organise the assumptions, facilitating a deeper understanding of the challenges and needs associated with each aspect of Data Discovery.

Upon further examination and sub-categorisation of the Data Discovery phase, we identified five distinct subcategories within Data Discovery.

- **Data Aquisition/Collection:** These assumptions focus on searching for data, utilising a data catalogue, and acquiring external data.

> ”We hypothesise that during interactive data science tasks, users often want to search the data lake, not only by keyword, but using a table, to find other related tables.” [49]

- **Data Needs:** These assumptions relate to what users aim to find during their data search, including complete tables, specific column names, or certain types of files.

> ”An analyst who is building the stock change prediction model may start with a search for tables that include metrics of relevance, such as stock prices and mentions in social media (schema similarity).” [5]

- **Input Size:** These assumptions relate to the volume of data users need to handle, noting that their approach may vary significantly between managing ten tables versus a thousand.

> ”Moreover, there might be hundreds of related tables, creating a large amount of work for the user.”[7]

- **Data Preparation:** These assumptions involve preparing data for subsequent calculations in the pipeline, including tasks such as wrangling, cleaning, and generating new data.

> ”When integrating data from multiple sources, there is often a need to perform transformations.”[1]

- **Data Exploration:** These assumptions concern comprehending newly discovered data and the time required for users to accomplish this understanding.

> ”We thus assume (query column) q is selected explicitly by the user.” [18]

With the introduction of these new categories, we now have a total of ten categories. The distribution of assumptions across these categories is detailed in 5.2.

With the assumptions now more evenly distributed across the categories, we can conduct a more thorough analysis after completing the interviews.

5.2. Overview of the Participants

We successfully conducted 19 interviews. Through follow-up questions regarding their educational background and professional experiences, we gathered some data concerning the education and experience of each participant. This information has been compiled into an overview presented in table 5.1. For the purpose of anonymity and ease of reference throughout the study, each participant has been assigned a unique identifier, ranging from P1 to P19.

5.3. Key Findings of the Interviews

After applying our convergence strategy and analysing the interviews, we ended up with 66 labels, as explained in 4.7. With these labels, we could label 1088 quotes from the participants. These labels are in Appendix B. Following the data processing phase, applying numerous labels provided significant insights. In this section, we will focus on presenting the findings that contribute to answering our research questions.

5.3.1. Query Data Understanding

As we assumed, all (19/19) participants started by looking at the base table. The participants tried to acquire an in-depth understanding of the base table, its columns and its values.

Figure 5.2: This figure shows the number of assumptions found and categorised by in-depth functionalities in the TDA workflow stage

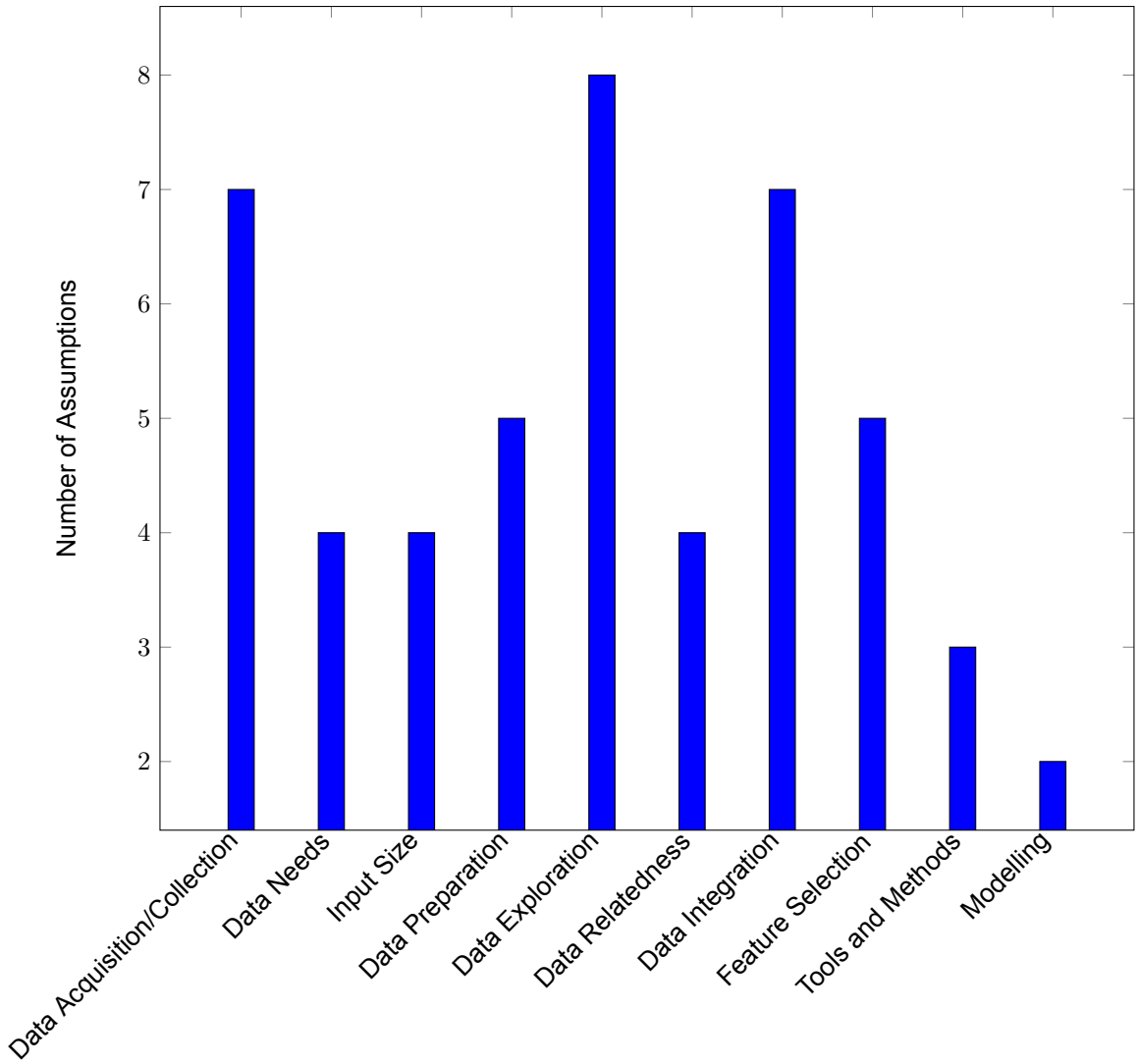


Table 5.1: The table provides an anonymised description of the interview participants.

#	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.

Input Type

We asked the participants *"Do you usually work with CSV files?"*. From these questions, we could get their data types.

- **Database**: nine participants mentioned they work with Databases.

  "Cause you normally use databases and SQL"

- **CSV**: There were no participants who normally work with CSV files. However, it was mentioned as an export product of databases for ML training.

  "Often it's data in databases[...], but then for training, usually you prepare a SQL query and then you get the CSV file".

- **Parquet**: Four participants mentioned Parquet.

  "With Parquet Files, it is quicker to load and it keeps the dtypes before saving. It is especially nice when you're dealing with big files."

- **Other**: There were other input types mentioned like JSON, Google Sheets or a Data Lake.

  "So I work mostly with tabular data, yes. Not specifically CSV files"

Understanding the Target Variable

When the base table was given, nine participants started looking at the target feature. They tried to get an understanding of what the true and false values meant in the context of the table.

"What does class indicate? What is the target? What does it mean"

Baseline ML Model

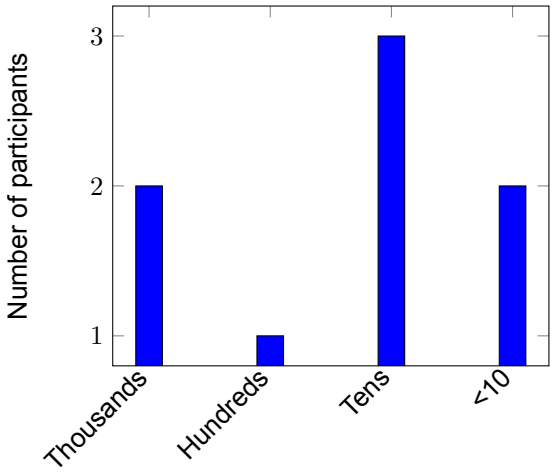
7 participants started by creating a baseline for the model. So they could see if adding or removing features would increase or decrease the model's performance.

"Maybe it will be good to have some sort of baseline, just to check how well we're doing without anything else" (P20)

5.3.2. Input Size

This stage, while somewhat prompted by our direct question, "If you were given a thousand tables, would you do the same? What would you do differently?" nonetheless offered valuable insights into participants' data handling capabilities and preferences.

Figure 5.3: Average amount of tables in the datasets of participants



One thousand tables

There were four approaches for the one thousand tables problem that occurred the most in the responses.

- **Asking Experts**: Five participants explained that they would contact someone in the company or someone who has knowledge over this large dataset.

  > ”There was usually somebody else who would tell me ”hey, actually you don’t need to look at a thousand sables, there’s actually like 10-15 of them that could potentially be more relevant” (P1)

- **Table names**: Six participants would approach this problem manually. They would review the tables and see if they have a similar table name to the base table.

  > ”If I had a thousand tables, I would kind of hope that the names of the tables made sense. I would pick the ones that mad the most sense first and try to go with that.” (P4)

- **Column names**: Two participants said that they would look manually for column names in all tables.

  > ”Try to start with the column name and to find if the column names are in the other tables as well.” (P2)

- **Catalogue**: Three participants explained that they would utilise a catalogue where the context about the tables are stored.

  > ”I would hope that this data would come of some sort of documentation or index. Just to get an understanding for the data” (P11).

- **Automatic** Two participants describe how they would write a script that does this automatically.

  > ”I would start by writing a script. I would look at the first five or so, just look at the structure. Do they all have the same unique identifying column? If so, I would run a script to verify that right?” (P14)

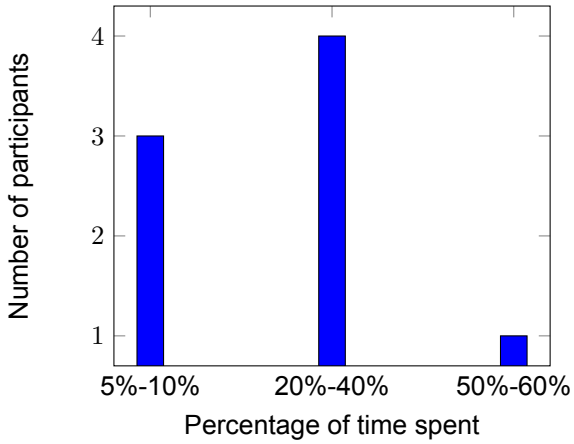
Dataset Size

During the interviews, several participants mentioned the number of tables they regularly work with. We describe this as the Dataset Size. An overview of these answers is shown in figure 5.3

5.3.3. Data Search

When given the candidate tables, most of the participants (17/19) started exploring them by searching the dataset.

Figure 5.4: Average time spent on Data Exploration by participants



Browsing

The process of browsing through the candidate takes, encompasses several key activities:

- Conducting an initial review of all files to gain an overview of the dataset, often beginning with the first file in the directory, regardless of how the files are organised, followed by an examination of the remaining data.

  > ”At this point, I’m just trying to get a view, maybe on what data we have available to us” (P3)

- Employing various tools to load the data, with a preference for databases, or utilising Pandas in their absence

  > ”The first thing I want to do is load in all the data” (P14)

- Analysing the file names to infer their significance based on intuition

  > ”Yeah, like the crime is super important. So definitely I will choose crime” (P13)

- Familiarise themselves with the data’s content by briefly reviewing the tables.

  > ”Yeah, so I’m just going through the files to get a sense of what’s in there” (P4)

Exploration Time

When we asked the question: *”How much time do you usually spend on the data exploration?”*, we noticed that 9/19 participants told us that the data exploration takes a lot of their time during a project. The results for these answers are shown in:

5.3.4. Table Understanding

Upon receiving the candidate tables, all participants (19/19) examined the contents and implications of the data presented. Afterwards, they tried to understand that data. Their strategies for this process varied, encompassing several methods:

- **Dependence on Documentation**: Access to detailed table descriptions and a catalogue greatly facilitates data interpretation and application. Clear, documentation is invaluable for fetching data (9/19).

  > ”I think most of the time when presented with a dataset, you’re given some sort of context on where that dataset came from” (P16)

- **Consultation with Domain Experts**: When faced with ambiguous data, participants often sought insight from more knowledgeable colleagues or reached out across teams or departments for clarification. This approach underscores the value of collaborative knowledge-sharing in demystifying data complexities (6/19).

  > ”Normally, I would ask someone in the company if the know more”(P5)

- **Utilising Own Domain Knowledge**: The application of personal expertise and understanding of the relevant domain plays a critical role in understanding data. Knowledge of the business context enriches comprehension of the predictive targets and the variables likely to impact them. (4/19).

  ”Here is where the domain knowledge plays a role” (P18)

- **Reliance on Intuition**: Several participants noted the importance of intuitive judgment to traverse and make sense of new information. This approach involves leveraging common sense and intuitive analysis to hypothesise relationships within the data. (4/19).

  ”I’m assuming this is some district-related data that they named disc” (P8)

5.3.5. Table Relatedness

Once the participants had developed a comprehensive understanding of the tables, their next step involved determining how these tables related to the base table.

Finding Relatedness

First, it is necessary to find the possible relationships between the tables. Several strategies were employed for this purpose:

- Documentation: As used in the Data Search, documentation plays a vital role in clarifying the contents and significance of various tables. It also helps by serving as a guide offering the necessary context to make choices regarding data relatedness (4/19).

  ”You would need some sort of business knowledge to figure out if the feature names match with whatever is in your base table” (P1)

- Common knowledge: Participants leveraged their common knowledge and insights across datasets, focusing on the analysis of column names to identify similar columns. They also looked for shared values within these columns as a method to uncover potential correlations. (14/19)

  ”You should reason about which tables are relevant in the first place. So I would assume that maybe for example gender” (P12)

- Automatic matching: In their data workflow some participants highlighted the use of automated techniques to assess how tables are related (3/19). One specific strategy involved identifying relatedness through a comparison of column names across CSV files (P18). Furthermore, calculating the correlation among features was mentioned as a useful method for understanding the relatedness between different data elements (P2, P7).

  ”I could also kind of look at what the column names are in the CSV files and see whether there’s a match between them. This could be automated in some way” (P18)

Relatedness Between Columns

If a possible relationship between tables was found, there were multiple strategies to see how the columns were related (16/19):

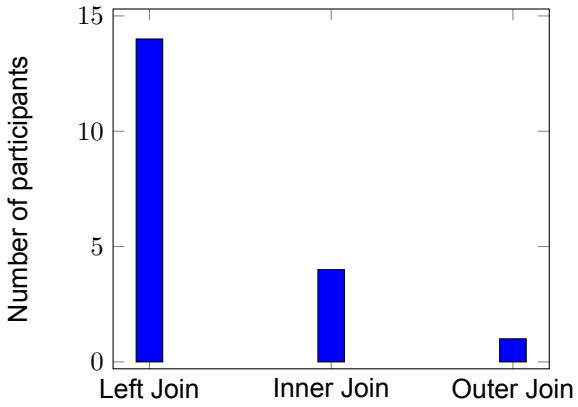
- **Uniqueness**: Performing statistical analyses to confirm the uniqueness of the columns or IDs of columns. (9/16).

  ”I’m looking to see if there is like a unique identifier, for example, if I can join” (P11)

- **Column similarity**: Observing recurring column names across tables suggests the presence of similar columns (6/16).

  ”See the DBN is pretty much present in most of them, so I will use this information as a join key” (P15)

Figure 5.5: Types of joins being used by participants during table integration



5.3.6. Data Integration

After the participants found some relatedness between tables, they wanted to integrate these tables together.

Primary and Foreign Keys

In our study, participants encountered PK-FK (Primary Key-Foreign Key) hard joins, which were not previously introduced to them. Upon inquiring about their familiarity with PK-FK joins, the majority (7/10) confirmed they regularly utilise such joins, while the remaining participants (3/10) indicated they employ fuzzy joins in their work.

> ”In cases where I’m trying to do that text matching, I would like to use a fuzzy match” (P12)

Type of Join

When the participants executed the joins, we noticed that there were different join types being used. An overview of the amount of join types is shown in

- **Left Join**:

  > ”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**:

  > ”I would do an inner join to just keep the common data” (P11)

- **Outer Join**:

  > ”It should be an outer join, right? Caus we wanna have all the features.” (P4)

Data Quality

When implementing a Left Join, there is a possibility of null values and duplication in the base table. These occurrences can introduce bias in the ML model. Several participants (9/19) enlightened this during the workflow.

> ”I just check if the count of the outcome is the same as the count of the left table. And if it’s like increased, then I’ll realise there was some duplicates here” (P18)

Some participants (5/19) were also concerned with the result of the table after the join.

> ”And then I would inspect if this went correctly. And it seems to be the case” (P12)

Join Pipeline

In the literature, there are two approaches to the join pipeline. One is where all the tables are joined and afterwards, the features are selected.[7] The other approach is joining the table one by one while selecting the features and evaluating the model between each join. Our participants followed the same approaches. Most of them (13/19) tried to join all the tables together.

> "So my pipeline in this case would be to just join all and then see what you get out of it" (P19)

Other participants (5/19) joined the tables one by one.

> "So this is the survey data and we join it with our existing table and you could train a classifier again ... to see if your performance goes up." (P12)

5.3.7. Feature Selection

Following the data integration phase, participants focused on ensuring that the final table contained the appropriate features. To achieve this, they engaged in feature engineering and feature selection processes.

Feature Engineering

Most of the participants (14/19) applied or mentioned Feature Engineering methods to improve the features in the table.

- **Remove Duplicates** : Duplicate columns are redundant, therefore it is useful to remove them. (3/14)

  > "If you need to get rid of the duplicates somehow, you either drop them base on some criteria or you group by the school name and then kind of figure out a way to aggregate the metrics." (P18)

- **Remove Null Values**: If features have too many null values. They do not contain useful information. Therefore, it is useful to drop the column or impute the null values in the column. (5/14)

  > "I would probably run some summary statistics just like there is a column full of NANS or anything that I wanna drop." (P16)

- **Remove ID's**: ID's are always unique. Therefore, they do not bring any useful information to an ML model and can even result in overfitting. (4/14)

  > "And in the last step to just throw away the ID's, because that is usually never informative for a model and it's very like that the model might actually overfit on those specific IDs." (P1)

- **Encoding**: Encoding techniques are useful when 3dealing with textual values since ML models do not understand text. Every categorical value is mapped to a vector. (4/14)

  > "I'm thinking of a very simple machine learning approach where you have a feature vector for each school and then train them a little bit based on that." (P6)

- **Normalisation**: Normalisation reshapes a numerical column to a standard scale. This results in a more accurate prediction. (3/14)

  > "So you should normalise this probably between zero and one or maybe standard deviation" (P12)

Feature Selection

When ending up with a large table, it is necessary to select features to maintain the performance of the ML model. Some participants (6/19) discussed that manually selecting columns.

> "We could drop certain columns in order to not confuse ourselves with too many columns that are not relevant in this case." (P10)

Where more participants (8/19) discussed the possibility of automatic feature selection.

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

> ”Initially, I would just throw a random forest or boosted tree, because it comes for free with feature selection.” (P1)

5.3.8. Tools

During the interviews, the participants described a lot of tools that we use in their daily work. An overview of their tools is shown in table 5.2

5.3.9. Ideal World

During the follow-up questions, we wanted to investigate what problems the data specialists have in their daily jobs and what solutions could overcome these problems.

Daily Problems

When we proposed the question: ”What other problems do you encounter in your data pipeline?”, we got answers that fall under 3 categories.

- **Documentation**: Several participants (5/19) encounter problems with a lack of documentation.

  > ”A common across data engineering teams is the lack of documentation.” (P1)

- **Quailty of Data**: Most of the participants (8/19) have problems with the quality of the data.

  > ”The main problems have to do with dat quality and the data coming in from the data source. And of course, it’s rubbish in and rubbish out for ML models. So it’s very important to have good quality of data.” (P11)

- **Size and synchronisation of data**: The amount of data can be troubling for data specialists. It can slow down the pipeline or create synchronisation issues. (6/19)

  > ”Millions and millions of rows that are being logged daily. There can be synchronisation issues.”(P5)

Tool for TDA

With these problems, we would like to know what kind of tool would help them with the use case that we gave them.

- **Databases**: A few (3/19) participants explained that databases would help them with this process.

  > ”Some capability of maybe running ad hox SQL so that I can already start writing some SQL code just to also explore the data.” (P8)

  ”

- **Documentation**: Again, a lot of the participants (5/19) wanted extra documentation, so they could base their decisions on this documentation.

  > A wiki or even just a Google doc with all that information would be great. I think one of the most powerful thinks you can do is kind of keep data in context together as much as you can.” (P16)

- **Automatic processes**: Some participants (3/19) explained that they would like some automatic processes that could speed this process up.

  > ”I think something that runs on the background and pre-compute some of these values and then would automatically suggest.” (P19)

Rely on Automatic TDA

Now after the entire manual TDA was done. We asked 16 of the 19 participants the question: ”Would you rely on a tool that does this TDA process automatically for you?” Most of them (12/16) said they would use this. The rest of the participants said that they do not trust such a tool to do this process.

Out of all the participants, several (8/16) gave the criteria that the tool has to explain why it made certain decisions. The participants want full insights in the tool.

> ”So I would if and that’s a big if it’s explained all the way through, right?” (P14)

> ”If I can have insights and it’s explainable and Understand why did what it did and I can see if there is a floor or not.” (P5)

5.4. Differences found with the Interviews and Assumptions

Now that we have both the assumptions and the interview data collected. We can analyse them and see the differences between the theoretical background and the real-life situation. This section shows the interesting valid and invalid assumptions.

5.4.1. Data Discovery

Within the Data Discovery assumptions, we observed that assumptions highlighted in Arda[7] and Aurum[5] revolved around the significant amount of time users spend locating the necessary data for their projects. Consistent with these assumptions, our interviews revealed that users typically allocate about 30% of their time in the data pipeline to Data Discovery.

The assumptions include statements regarding the data quality of the datasets users handle. Our interviews showed that 8 out of 19 participants encountered issues with data quality in their daily tasks. Thus, it can be concluded that data quality indeed poses a significant challenge within a TDA workflow.

In Warpgate [8], an assumption was made that only a few individuals have a thorough understanding of their data and its interrelationships. Additionally, Aurum [5] suggested that users often search for datasets based on specific schemas or key attributes. However, our interviews indicated that many participants rely on various resources to comprehend their data: documentation (9/19), consultations with domain experts (6/19), knowledge (4/19), and intuition (4/19). This demonstrates that data specialists prioritise understanding the significance of tables over merely searching for specific attributes within them.

5.4.2. Data Relatedness

In COCOA [18], it was mentioned that the correlation coefficient is widely used in data science. However, our interviews revealed that only 2 out of 19 participants employed correlation to calculate relationships between tables. Instead, most participants (14/19) relied on common knowledge to determine these relationships.

5.4.3. Data Integration

In Alite [23] and "To Join or Not to Join" [31], it was stated that data specialists often join all tables in one operation. Our interviews confirmed this finding, as 13 out of 19 participants indicated that they typically join all tables and proceed with the feature selection process.

Moreover, our interviews revealed that 7 out of 10 participants use primary key-foreign key (PK-FK) joins, while 3 out of 10 utilise fuzzy joins. The assumptions in the literature primarily focused on PK-FK joins. This suggests that the applications mentioned in [5] and [31] apply to approximately 70% of data specialists.

5.4.4. Feature Selection

In the feature selection phase, there were no assumptions that we could validate.

5.4.5. Overal

Overall, the literature predominantly focuses on automatic Tabular Data Augmentation (TDA) applications, with assumptions built upon the use of large datasets. However, as illustrated in Figure 5.3, it's evident that the majority of participants do not work with such extensive datasets. More importantly, participants commonly utilise a catalogue to understand the data, indicating that they do not heavily rely on automated processes like correlation and column name comparison.

When posed with the question of whether they would rely on a tool that automates tasks like those described in [7], [18], and [22], the majority of participants expressed reluctance. They cited concerns over these tools operating as black boxes, indicating a preference for more transparent and understandable processes in their workflow.

6

Part 1 Conclusions

In the first part of this research, we embarked on an examination of the existing assumptions around state-of-the-art TDA research. Our inquiry was guided by two pivotal questions: firstly, we identified the assumptions made by users during their interaction with TDA technologies, aiming to discern whether these presumptions were justified or misplaced. Secondly, we delved into the workflows of data specialists utilizing TDA, paying particular attention to the methodologies they employ and the challenges they face.

To address these research questions, we adopted a methodological approach known as snowballing to gather assumptions from existing literature. Following this, interviews with data specialists were conducted to both validate these assumptions and uncover gaps within the current state-of-the-art literature.

Our findings show that most participants work with small datasets and currently undertake each step of the Tabular Data Augmentation (TDA) workflow manually. While they utilize catalogues, domain experts, or common knowledge to guide their process, a pattern consistent with the assumptions, we also recognize that this manual approach is time-consuming for them.

Given these challenges, an automatic tool could significantly aid in streamlining their workflow. However, such a tool mustn't be a black box. Participants expressed a strong desire for transparency and insights into the tool's decision-making process. They value their understanding of the data and wish to comprehend why the tool makes certain decisions. Therefore, any automatic tool must provide users with the necessary insights and explanations to align with their expertise and preferences.

Part II

Part 2: Tool Development for Human In The Loop TDA

7

Tool Support for Human-in-the-loop Data Augmentation

While existing state-of-the-art Tabular Data Augmentation (TDA) applications perform commendably, executing the TDA workflow efficiently and producing well-augmented tables, a gap remains in addressing user needs. Predominantly, these applications operate in a 'black box' manner, lacking transparency and user engagement in the process. Our research, enriched by insights from interviews, highlights a clear user preference for an algorithm that delivers results and elucidates the steps to achieve those results. Users desired to observe intermediate values and tailor the process to their specific requirements. This chapter introduces a novel tool designed to fulfil these needs, offering a more interactive and user-centric approach to the TDA algorithm, ensuring transparency and adaptability at every stage of the data augmentation process.

7.1. Requirements for the tool

Building on the findings from Chapter 6, it's evident that Automatic TDA tool users seek transparency rather than a black-box algorithm. It's crucial to delineate a detailed set of requirements to tailor this tool to user preferences. These guidelines will steer the development journey, guaranteeing that the tool fulfils user expectations and delivers on its promised capabilities. Central to crafting a white-box algorithm are features that offer users control and insights. Moreover, the efficacy of the tool is crucial. Below are the principal requirements for crafting the design of the tool:

- **Effective TDA Process**: The tool should effectively address the users' needs in a TDA workflow.
- **Insights**: Users should have access to intermediate results within the TDA process, allowing for a transparent view of how data is being transformed and augmented.
- **Control**: The tool must allow users to adjust and customise intermediate values according to their specific needs or preferences, enhancing their adaptability to various use cases.

7.2. Design of the tool

To meet these specified requirements, a detailed and well-thought-out design is essential. While we have the option to develop an entirely new solution from the ground up, it would necessitate the creation of bespoke TDA techniques, essentially duplicating efforts already accomplished in existing state-of-the-art methodologies [7] [18] [22]. Given that these established methods have demonstrated promising results, it makes practical sense to leverage these available open-source tools, thereby avoiding the unnecessary replication of work and capitalising on proven strategies.

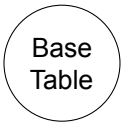


Figure 7.1: Example of a Join Tree with the paths: [Base Table]

We have chosen AutoFeat [22] to implement our Control and Insight features, guided by several compelling reasons:

1. Open Source Availability: AutoFeat's open-source nature allows us to directly incorporate its established methods into our work, enhancing our efficiency by utilising pre-existing solutions.
2. Access to Expertise: The AutoFeat development team is part of our research group, providing invaluable insights into the tool's architecture and functionalities and facilitating a deeper understanding of its operational mechanics.
3. Performance Efficiency: When the relationships are calculated, AutoFeat demonstrates remarkable speed in the augmentation, ensuring a swift analysis process and a more user-friendly experience due to reduced waiting times for outcomes.

7.2.1. AutoFeat's workflow

AutoFeat is a method designed for feature discovery. It employs multi-hop, transitive join paths to identify pertinent features that can enhance a base table. It achieves this by selectively adding highly relevant and non-redundant features and optimising the information value and uniqueness within the augmented dataset.

Relationships Calculations

AutoFeat initiates its process by identifying connections within the dataset, utilising the Valentine [28] package. Valentine offers a variety of methods to assess similarities between columns across different tables, including Coma[12], Cupid [35], Similarity Flooding [37], Distribution-based methods [48], and Jaccard-Levenshtein. AutoFeat constructs a Dataset Relation Graph (DRG) based on the chosen method. When the similarity score between two columns surpasses a predefined threshold, AutoFeat draws an edge between the two corresponding nodes in the DRG, assigning the edge a weight equal to the similarity score. These inter-column relationships are catalogued within a Neo4J[1] Graph Database. This setup ensures efficient query processing, such as retrieving all neighbouring nodes of a given node (i.e., nodes connected by an edge).

Graph Traversal

Beginning with the base table, AutoFeat employs a Breadth-First Search (BFS) strategy to navigate through the graph, prioritising the exploration of all available paths simultaneously rather than delving deep into a singular path. This exploration starts with stratified sampling of the base table, enhancing the algorithm's speed and overall efficiency. For each neighbouring node identified in the graph, AutoFeat attempts to add that node to all discovered combinations of nodes that have been calculated already. The resultant paths from these operations are referred to as join paths. This terminology is used because documenting the sequence of joins provides a clear map of the journey through the graph. Illustratively, the process of integrating two tables into the base table can be visualised in 7.1, 7.2, and 7.3, with the specific join paths detailed in the figures' captions. Crucially, every join path incorporates the base table, as it contains the essential labels for the machine learning model.

AutoFeat employs Left Joins to preserve the original number of tuples and distribute the label within the base table. This approach maintains the integrity of the dataset by keeping the initial structure and information unchanged. To address the potential issue of duplicates that might arise from one-to-many and many-to-many relationships, duplicates that could bias the results, AutoFeat converts these relationships into one-to-one joins. It accomplishes this by randomly

[1] https://neo4j.com/

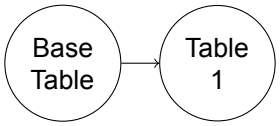


Figure 7.2: Example of a Join Tree with the paths: [Base Table], [Base Table-Table 1]

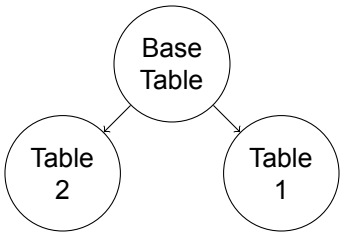


Figure 7.3: Example of a Join Tree with the paths: [Base Table], [Base Table-Table 1], [Base Table-Table 2], [Base Table-Table 1, Base Table-Table 2]

selecting a single row from the duplicated rows. This careful handling of join operations is crucial for maintaining the quality and reliability of the augmented data.

For each join operation, AutoFeat performs a data quality assessment. The operation is cancelled if the resulting dataset from any of the join paths exhibits excessive null values post-join. The threshold for ”too many” null values is determined at the onset of the algorithm’s initialisation, ensuring a predefined standard of data integrity is maintained throughout.

Additionally, every feature within the newly added table is evaluated for relevance and redundancy. This dual-check mechanism ensures that only features which pass these criteria are retained in the join. Features failing to demonstrate relevance or exhibiting redundancy are excluded, safeguarding the quality and effectiveness of the augmented dataset.

- **Relevance:** A feature’s relevance assessment is quantified through either Spearman or Pearson correlation coefficients [46]. A correlation value of 0 indicates complete irrelevance, suggesting that the feature has no impact and, thus, does not contribute to the model’s predictive capability. Conversely, a correlation value of 1 signifies perfect correlation, which, paradoxically, is also undesirable as it may indicate redundancy. Features that exhibit correlation values within the range of 0 and 1 are deemed to have passed the relevance test, suggesting they hold potential value for enhancing the model’s performance without introducing unnecessary noise or redundancy.
- **Redundancy**: Following the relevance evaluation, each feature that passes this initial test undergoes further analysis to determine its uniqueness. This step involves checking whether the feature duplicates another feature already deemed relevant. The purpose is to ensure the augmented dataset does not contain redundant information, which could dilute the model’s efficiency. A feature is excluded from the join if it is identified as redundant. This approach helps maintain a streamlined, high-quality, informative dataset without unnecessary replication.

Path Evaluation

After potential join paths are collected, each path undergoes an evaluation phase using the machine learning (ML) model. For each path, the joins are materialised to create consolidated datasets. However, tables are sampled before executing the joins to maintain high efficiency and effectively manage computational resources. This strategic sampling ensures that the size of the data remains manageable without compromising the integrity of the evaluation.

Utilising AutoGluon [17], a powerful automated machine learning tool, the accuracy of each materialised join is assessed. This step is crucial for determining each join path’s effectiveness in enhancing the ML model’s predictive performance. Based on the accuracy measurements obtained from AutoGluon, the algorithm then ranks the join paths, providing a list of the top_k

paths. These paths are ranked according to their contribution to model accuracy, enabling the selection of the most impactful datasets for model training and evaluation.

7.2.2. Autofeat's TDA Workflow

Based on the insights gathered from our interviews, it became evident that users tend to follow a workflow consistent with what is documented in the literature, as depicted in figure 3.1. This observation suggests that the pipeline employed by AutoFeat could be effectively adopted for our purposes. AutoFeat is conceptualised around three primary components: Relationship Calculations, Graph Traversal, and Path Evaluation, each corresponding to distinct phases of the TDA workflow.

- In the Relationship Calculations phase, the algorithm identifies and assesses the connections between all columns across the dataset. This step corresponds to the initial stages of the TDA workflow, namely Data Discovery and Data Relatedness.
- In the Graph Traversal phase, the algorithm navigates through all potential join paths, systematically merging the tables identified in each path. This process encapsulates the Data Integration stage of the TDA workflow. Concurrently, as the joins are executed, each feature within the newly integrated table undergoes evaluation for its relevance and redundancy. Only the features that successfully meet the established criteria are retained. This process effectively integrates the Feature Selection stage into the pipeline.
- The Path Evaluation component, though not traditionally a part of the standard TDA workflow, plays a crucial role in our approach by enabling an empirical assessment of how each join path impacts the performance of the machine learning (ML) model. This stage is instrumental in quantifying the effectiveness of the data integration and feature selection processes, providing a direct measure of the augmented dataset's predictive capability. By including Path Evaluation in our pipeline, we ensure a comprehensive validation mechanism that tests the practical utility of each join path within the context of ML modelling.

For each stage of our pipeline—Relationship Calculations, Graph Traversal, and Path Evaluation—we aim to integrate control and insights functionalities. This enhancement is designed to give users visibility into the intermediate steps of each stage, thereby demystifying the process and empowering users to make informed adjustments as needed.

7.2.3. Notebook

To effectively interact with and manipulate intermediate values between components, users must have the capability to store these values in memory and access the interim results. Implementing a solution compatible with notebook environments would be particularly advantageous. This setup allows users the flexibility to execute components on an individual basis and to rerun these components with modified parameters as needed. Insights from our interviews indicate that 12 participants already incorporate notebooks into their data workflows, suggesting that a notebook-compatible approach would seamlessly integrate into and enhance their current practices.

7.2.4. Human in the Loop AutoTDA

Our innovative approach is called Human in the Loop AutoTDA. This tool is outlined in the figure: 7.4, presenting a design that integrates user interaction within the AutoFeat algorithm. This application acts as an API for AutoFeat, facilitating a dynamic interface where users can execute specific functions within a notebook environment. This capability allows for real-time adjustments and insights, allowing users to fine-tune parameters and closely observe the algorithm's processing stages.

7.3. Implementation of the Tool

This section delves into the five key components of our solution alongside an illustration of a fully automated approach. We will detail how users can apply control and access intermediate values for deeper insights into each component. To facilitate ease of use and accessibility, our solution is encapsulated within the autotda package, available on the Python Package Index

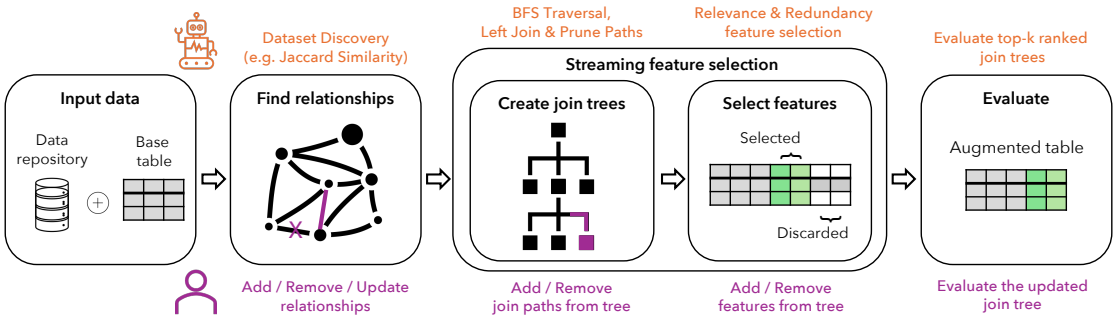


Figure 7.4: Human in the Loop AutoTDA workflow

(PyPi) at https://pypi.org/project/autotda/. This distribution ensures users can readily implement our solution within their data workflows. In our implementation, we changed the name join paths, used in AutoFeat, to **join trees**.

7.3.1. Object Oriented Approach

We adopted an Object-Oriented approach for this design, ensuring that all data is encapsulated within objects. Given that the AutoFeat workflow unfolds across three distinct stages, we aimed for these objects to represent each phase comprehensively. Consequently, in our proposed solution, we crafted four specific objects:

- **Weight:** The objects are designed to encapsulate table relationships. Each object specifies a link from a column in one table to a column in another, accompanied by a weight attribute. This weight quantifies the strength of the relationship, providing a metric for assessing the significance of the connection between the specified columns across the tables.

```
        class Weight():
            from_table: str
            to_table: str
            from_col: str
            to_col: str
            weight: float
```

- **Join:** A Join object represents a connection between two tables, established when the algorithm identifies a pathway linking them. Each join is defined by a linkage from a column in one table to a column in another. Additionally, every Join object encapsulates the null ratio of the resulting table after performing the join. It also includes scores for relevance and redundancy of the features in the ”to table,” providing a quantitative assessment of the join’s impact on data quality and information value.

```
        class Join():
            from_table: str
            to_table: str
            from_col: str
            to_col: str
            non_null_ratio: str
            rel_red: dict
            rel_red_discarded: dict
```

- **Tree:** The Tree object symbolises a join tree within the workflow, embodying a structured amalgamation of joins designed to merge multiple tables. Each Tree is characterised by its selected features and the designated join keys that define how tables are interconnected. It also contains a list of Join objects that delineate the specific pathway through which the tables are joined. Additionally, a rank is attributed to each Tree, derived from evaluating the relevance and redundancy scores of the features within the joined tables. This ranking quantifies the join tree’s overall efficacy and information integrity in the workflow context.

```
        class Tree:
```

```
            features: list = []
            join_keys: list = []
            joins: list = []
            rank: float = 0
```

- **Result:** A Result object is produced after evaluating a join tree within the workflow. Each Result is intrinsically linked to a single join tree, embodying the outcomes of that specific configuration of joins. Moreover, the Result encapsulates key performance metrics of the evaluated model, including the accuracy of predictions made by the model. It also holds important values regarding the features within the model, providing insights into which features contribute most significantly to the model's predictive capabilities. This structure allows for a comprehensive assessment of the effectiveness of each join tree configuration in enhancing the model's performance.

```
        class Result():
            tree: Tree
            accuracy: float
            feature_importance: dict
            model: str
            model_full_name: str
```

7.3.2. Initialisation

First, the user needs to initialise the pinitialise

```
from autotda.autofeat_class import TDA
autoTDA = TDA(repository_location="../data/benchmark")
autoTDA.set_base_table(base_table="school/base.csv", target_column="class")
```

Within the initialisation of the TDA() constructor, users must define the path to their data repository. This specification informs the system of the precise location to retrieve the necessary data. Additionally, users must provide the name of the dataset and the base table they wish to augment; for example, in our context, these would be specified as ”school” for the dataset and ”base.csv” for the base table. Finally, users are tasked with identifying the target column within the base table.

7.3.3. Automatic TDA

Users who prefer to execute the algorithm seamlessly without engaging in intermediate steps can utilise the *augment_dataset()* function. This function streamlines the process, enabling dataset augmentation in a single operation.

```
autoTDA.augment_dataset()
def augment_dataset(self, algorithm="GBM", relationship_threshold: float = 0.5,
    non_null_threshold=0.5, matcher="coma",
                        top_k_features: int = 10,
                        top_k_trees: int = 3, explain=True, verbose=True, use_cache=
                            True):
        self.relationship_threshold = relationship_threshold
        self.matcher = matcher
        if use_cache:
            if os.path.isfile(f"{self.weights_location}/{self.base_table}_{
                relationship_threshold}_{matcher}_weights.txt"):
                if verbose:
                    print("Reading␣from␣cache␣file:␣" + f"{self.weights_location}/{self
                        .base_table}_{relationship_threshold}_{matcher}_weights.txt")
                    self.read_relationships(f"{self.weights_location}/{self.base_table}
                        _{relationship_threshold}_{matcher}_weights.txt")
            else:
                self.find_relationships(relationship_threshold=relationship_threshold,
                    matcher=matcher,
                                        explain=explain, verbose=verbose)
        else:
            self.find_relationships(relationship_threshold=relationship_threshold,
                matcher=matcher,
                                    explain=explain, verbose=verbose)
```

```
        self.compute_join_trees(top_k_features=top_k_features, explain=explain,
            non_null_threshold=non_null_threshold,
                                verbose=verbose)
        self.evaluate_trees(algorithm=algorithm, top_k_trees=top_k_trees, explain=
            explain)
        return self.materialise_join_tree(self.get_best_result().tree.id)
```

This function orchestrates the complete TDA workflow, initiating a verification for any pre-existing cache of previously calculated relationships. If such a cache exists, it leverages the *read_relationships* function to load these relationships efficiently, thus bypassing the need for recalculating them. This feature significantly accelerates the process, allowing users to make adjustments without the redundancy of re-executing those calculations. Without a cached relationships file, the function identifies relationships anew by invoking the *find_relationships* function. Following this, it advances to construct join trees and select features through the *compute_join_trees* function. The workflow ends with the *evaluate_trees* function, which assesses all the candidate trees generated in the preceding step, determining their efficacy and relevance to the TDA objectives.

The *augment_dataset()* function produces an augmented table that achieves optimal accuracy as assessed by the *evaluate_trees* function. This output details the features' relevance and redundancy scores, outlines the journey through the tables, and highlights the value of the chosen features. Users who prefer not to utilise the automated algorithm and wish to engage with the process incrementally have the option to invoke the *find_relationships* individually, *compute_join_trees*, and *evaluate_trees* functions for a step-by-step execution.

7.3.4. Input Data

A notable limitation identified in AutoFeat is its lack of flexibility regarding dataset input. Users are confined to selecting a single dataset without the option for modification or customisation. Recognising the importance of adaptability in data analysis, we propose to expand the input capabilities in our solution, enabling users to have greater control over the selection and configuration of dataset inputs.

The *set_dataset_repository* function configures the source datasets from which the tool retrieves data, accepting an array to allow users to specify multiple directories. In our case, we have chosen only the School dataset. Users looking to include tables from outside standard repositories can use the *add_table* function to designate the specific table and its corresponding dataset. Similarly, to exclude a table from the repository, users can employ the *remove_table* function.

```
autoTDA.set_dataset_repository(dataset_repository=["school"])
autofeat.add_table("weather/rain.csv")
autofeat.remove_table("school/qr.csv")
```

7.3.5. Find Relationships

Determining the strength of relationships between tables can be utilised by the *find_relationships* function. This function offers flexibility in setting up the relationship discovery process, including selecting a matcher method (initially, only Coma and Jaccard methods are available). Users can also define the threshold level for establishing connections between tables, activate an explanation feature to gain insights into the tool's decision-making and results and save the calculated weights in a storage file for future reference or analysis.

```
autoTDA.find_relationships(matcher="coma", relationship_threshold=0.5, explain=True,
    use_cache=True)
```

Control

When you're confident about the relationships between tables, whether connected or not, you can refine these relationships through specific functions. Utilising *remove_relationships()* allows you to eliminate existing connections you deem inaccurate. With *update_relationships()*, you can better modify the details of current relationships to reflect their true nature. Meanwhile, *add_relationships()* enables you to introduce new relationships between previously unlinked tables, ensuring a more accurate representation of table connections.

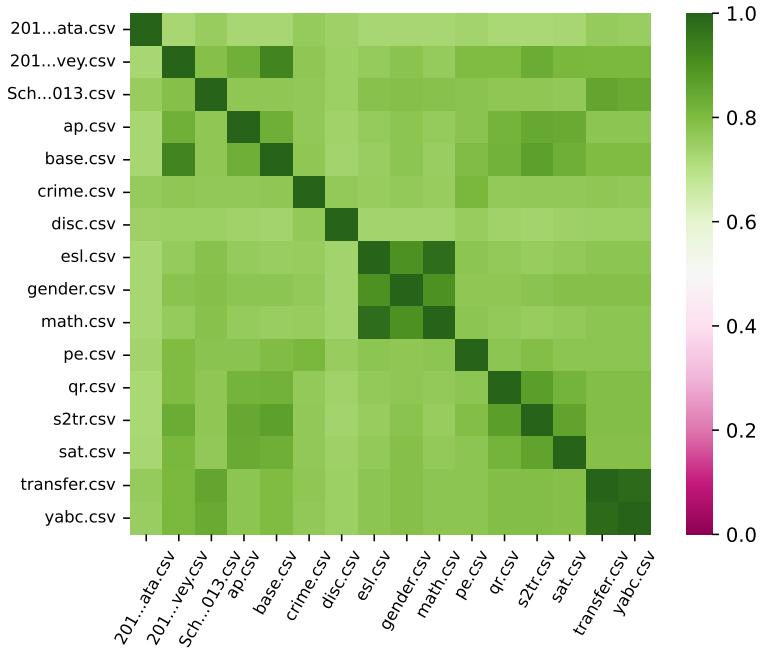


Figure 7.5: Result of showing the best relationships between all tables

```
autoTDA.remove_relationship(table1="school/ap.csv", col1="DBN", table2="credit/
    table_0_0.csv", col2="DBN")
autoTDA.update_relationship(table1="school/ap.csv", col1="DBN", table2="school/qr.csv",
     col2="DBN Name", weight=1)
autoTDA.add_relationship(table1="school/ap.csv", col1="SchoolName", table2="school/qr.
    csv", col2="School Name", weight=0.2)
```

Insights

To explore the optimal relationships among all tables, you can invoke the *display_best_relationships()* function. This function showcases the highest relationship score across all tables, with each score representing the most significant connection found between any columns of those tables. The outcomes of this function are illustrated in detail, as depicted in figure 7.5, providing a clear visualisation of the strongest table-to-table relationships within your dataset.

To examine the relationship scores between the columns of two specific tables, the *display_table_relationship* function is available. Executing this function generates a visualisation similar to the one mentioned earlier, but this time with the columns of the two tables plotted along the axes. This detailed representation facilitates a granular view of how each column from one table relates to the columns of another, providing insights into the inter-columnar relationships. The results from this operation are captured in a figure, as illustrated in figure 7.6, offering a clear and concise depiction of these connections.

For those interested in delving deeper into the specifics of the relationship between two tables, the *explain_relationship* function serves this purpose. By calling this function, you receive a detailed table listing all the relationships between the selected tables. The output of this function is systematically organised and is shown in table 7.1, offering an explicit explanation and visualisation of the relationships in question.

7.3.6. Compute Join Trees

Users can invoke the *compute_join_trees* function to initiate the computation of join trees. This function accepts parameters to customise the process, including the *top_k_features* value to specify the number of features to be included in the trees, the *non_null_ratio_threshold* to set a threshold for the maximum allowable nulls in a join to be included, and the *explain* flag to indicate

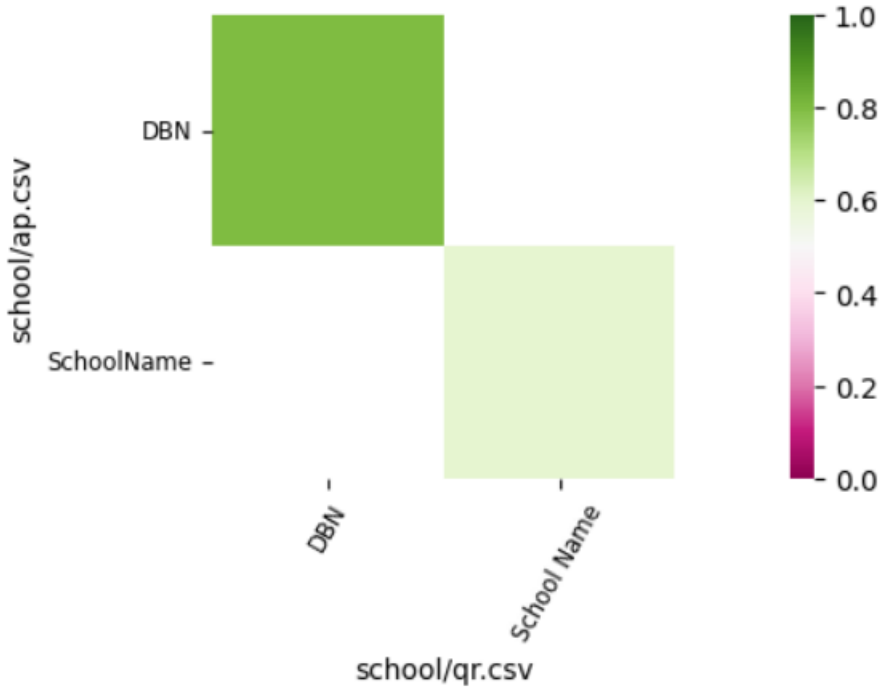
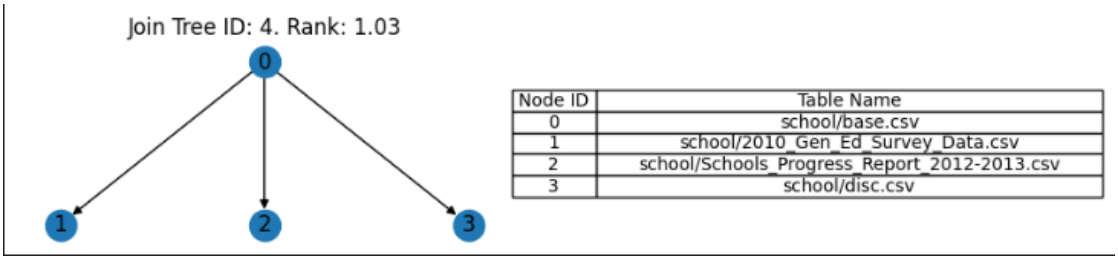


Figure 7.6: Result of showing the relationships between table ap.csv and qr.csv

Table 7.1: Result of running *explain_relationship* with tables ap.csv and qr.csv

From Column	To Column	Weight
ap.csv.DBN	qr.csv.DBN	0.8
qr.csv.DBN	ap.csv.DBN	0.8
ap.csv.SchoolName	qr.csv.School Name	0.6
qr.csv.School Name	ap.csv.SchoolName	0.6



Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/Schools_Progress_Report_2012-2013.csv
3	school/disc.csv

Figure 7.7: Example of join tree 4 with three integrated joins and a rank of 1.03

whether the function should provide explanations for its decisions.

```
autoTDA.compute_join_trees(self, top_k_features = 10, non_null_ratio_threshold=0.8,
    explain=True)
```

Insights

By utilising the *display_join_trees()* function, you can view all the generated join trees, where each node represents a table, and every edge signifies a join connecting two tables. This visual representation helps us understand the complex relationships and pathways through which data from different tables can be integrated. An instance of such a displayed join tree is provided as an example, which can be seen in figure 7.7, offering a clear and structured visualisation of how the tables are interconnected within the generated join trees.

To delve into the specifics of a join tree, including its constituent joins and their respective non-null ratios, you can utilise the *explain_tree* function. This function provides a textual output detailing each join within the tree, emphasising the non-null ratio to offer insights into the data completeness across the joins.

```
autoTDA.explain_tree(tree_id=5)

The join tree starts at the table school/base.csv.
The join tree has the following joins:
         from (table.column) school/base.csv.DBN to (table.column) school/2010
             _Gen_Ed_Survey_Data.csv.DBN with Non-Null ratio 0.87.
         from (table.column) school/base.csv.DBN to (table.column) school/
             Schools_Progress_Report_2012-2013.csv.DBN with Non-Null ratio 0.96.
         from (table.column) school/base.csv.DBN to (table.column) school/disc.csv.DBN
             with Non-Null ratio 0.86.
         from (table.column) school/base.csv.DBN to (table.column) school/esl.csv.DBN
             with Non-Null ratio 0.63.
The rank of the tree is 1.03.
```

To explore the features included in a join tree, the *show_features* function is available for use. When the *show_discarded_features* parameter is set to True, this function not only displays the features that have been retained but also those that were discarded during the process. This comprehensive view allows a deeper understanding of the feature selection mechanism and the rationale behind excluding certain features. A concise illustration of this functionality, including both retained and discarded features, is provided as an example in figure: 7.8, offering insight into the feature composition of the join tree.

```
autoTDA.show_features(tree_id=5, show_discarded_features=False)
```

Control

Users are provided with three specific functions to manage the configuration of join trees effectively. Initially, users can eliminate a join from a tree. As an illustrative case, one might remove the table *school/disc.csv* from the join tree, represented in figure 7.7. This function allows for the refinement and optimisation of the join tree structure by excluding tables that may not contribute to the desired analysis or model performance.

```
autoTDA.remove_join_path_from_tree(tree_id=5, table="school/disc.csv")
```

Selected features:

Key	Non-Null Ratio	Relevance	Redundancy
school/2010_Gen_Ed_Survey_Data.csv.studentsurvey	0.873168	0.270114	1
school/Schools_Progress_Report_2012-2013.csv.2012-2013 ENVIRONMENT GRADE	0.959977	0.446768	1
school/Schools_Progress_Report_2012-2013.csv.2011-12 PROGRESS REPORT GRADE	0.959977	0.221042	0.614142
school/Schools_Progress_Report_2012-2013.csv.2012-2013 OVERALL GRADE	0.959977	0.206767	0.592583
school/Schools_Progress_Report_2012-2013.csv.2010-11 PROGRESS REPORT GRADE	0.959977	0.202085	0.517268

Figure 7.8: Example of features selected by tree 5 and their non-null ratio, relevance and redundancy scores

Additionally, users are empowered to override algorithmic decisions regarding feature selection selectively. They can choose to retain features that the algorithm may have discarded and, conversely, opt to exclude initially selected features. This level of customisation ensures that users maintain control over the composition of features, allowing them to tailor the feature set according to their specific needs.

```
autoTDA.move_features_to_selected(tree_id=5, features=["school/esl.csv.Grade"])
autoTDA.move_features_to_discarded(tree_id=5, features=["school/esl.csv.Grade"])
```

7.3.7. Evaluate Trees

After the join trees have been computed, users can gauge their effectiveness within the context of a machine learning (ML) model by using the *evaluate_trees* function. This functionality allows users to choose the specific ML model they prefer for evaluation. Additionally, it enables them to decide on the quantity of join trees they aim to evaluate, thereby offering a structured approach to assess the impact of various join tree configurations on the model's performance. This step is crucial for identifying the most effective data integration strategies to enhance predictive accuracy and model efficiency.

```
autoTDA.evaluate_trees(algorithm='GBM', top_k_trees= 3)
```

Insights

To review the performance outcomes of the join trees, the *get_all_results* function can be utilised. This function fetches and returns a collection of tuples, where each tuple comprises the ID of a join tree alongside its evaluated accuracy. This comprehensive overview allows users to quickly compare the effectiveness of different join tree configurations, facilitating informed decisions on which tree structures yield the best results in terms of accuracy for their specific machine learning models.

```
results = autoTDA.get_all_results()
print(results)

prints: [(1, 0.6845070422535211), (9, 0.8056338028169014), (10, 0.8140845070422535)]
```

For instances where only the top-performing result is needed, the *get_best_result* function can be invoked. This command fetches the most optimal outcome from the evaluated join trees. Users can print these particulars to delve into more specific details, such as the joins that constitute this best result, thereby gaining insights into the structure and characteristics of the join tree that achieved the highest accuracy in the evaluation.

```
    Result of tree: 10 with model LightGBM

         rank:1.09

         with tree:

Rank: 1.09

From base table: school/base.csv

Join Trees:
```

```
12
13 +---------------------+-----------------------------------------------------+------------------+
14
15 | From Table.Column   | To Table.Column                                     |   Non-Null
       Ratio |
16
17 +=====================+=====================================================+==================+
18
19 | school/base.csv.DBN | school/2010_Gen_Ed_Survey_Data.csv.DBN              |
       0.873168 |
20 ....
21 (more joins included here)
22 ....
23 Accuracy:0.81
24
25 Feature importance:
26
27 school/Schools_Progress_Report_2012-2013.csv.2012-2013 ENVIRONMENT GRADE :
       0.20563380281690144
28 ....
29 (more features included here)
30 ....
```

Once satisfied with the outcome, the final result can be transformed into a practical format by executing the *materialise_join_tree* function. This action converts the selected join tree into a pandas DataFrame, facilitating further analysis or integration with other data processing workflows.

```
1 dataframe = autoTDA.materialise_join_tree(1)
```

8

Tool Evaluation

Having implemented a tool, the next critical step is to verify its practicality and alignment with the specific needs of data professionals, especially in the context of Tabular Data Augmentation (TDA). This chapter will detail the strategy and process for conducting an evaluative study on the Human in the Loop AutoTDA tool. The outcomes of this evaluation will enable us to identify which predefined criteria have been fulfilled and which have not been met.

8.1. Software Evaluation Techniques in the Literature

This section contains a brief overview of the possible software evaluation techniques.

8.1.1. Quantitative vs Qualitative Evaluation

Initially, our evaluation strategy necessitates a decision between quantitative and qualitative approaches. Quantitative evaluation hinges on objective metrics and measurable attributes, offering numerical data that reflect the tool's performance. In contrast, qualitative evaluation relies on subjective analysis, assessing the tool's effectiveness through observations, interviews, and user experiences. This choice is critical, as it shapes our evaluation framework and determines whether our focus will be on statistically significant outcomes or in-depth, narrative insights into user interactions with the tool.

We have to use a qualitative approach to fulfil the other evaluation objectives since those objectives are subjective.

8.1.2. Evaluation Techniques

There are three main techniques for conducting a qualitative software evaluation. [25] [26].

- **Experiment**
  A qualitative experiment is structured to test hypotheses or explore the effects of specific variables (in this context, tool features) on outcomes in a controlled setting. The goal is to understand the impact of these variables on user experience, usability, and satisfaction. In our situation, this would mean that the user does what we tell them to do.
- **Case study**
  A case study is an in-depth investigation of a particular instance or scenario involving the tool in a real-world context. The goal is to comprehensively understand the tool's use, effectiveness, and integration into existing workflows within a specific setting.
- **Survey**
  A qualitative survey is an assessment technique by individuals who have utilized or examined the methods/tools in focus. In this approach, the necessary features are translated into distinct questions within the survey.

When deciding among experiments, case studies, and surveys for evaluating tools or methods,

understanding the benefits and limitations of each approach is crucial. Here's a refined comparison:

- **Experiment**
  - Advantages:
    - **Consistency in Execution:** Participants engage in identical workflows, simplifying the analysis process.
    - **Controlled Environment:** Conducted in settings where variables are controlled, ensuring focused feedback on evaluation questions.
  - Disadvantages:
    - **Potential Overdirection:** There's a risk of overly guiding participants, which may influence their natural responses.
    - **Limited Realism:** May not accurately reflect everyday usage scenarios, affecting the applicability of results to real-world conditions.
- **Case Study**
  - Advantages:
    - **Exploratory Flexibility:** Allows for a deeper, more exploratory examination of user interactions with the tool.
    - **Contextual Depth:** Provides a comprehensive view of the tool's usage in real settings, capturing nuanced user experiences.
  - Disadvantages:
    - **Generalization Challenges:** Findings are often case-specific, complicating the extrapolation of results to a wider audience.
    - **Data Collection Complexity:** The uniqueness of each case study can make gathering and analyzing data more challenging.
- **Survey**
  - Advantages:
    - **Broad Reach:** Can engage a diverse user base without direct interaction.
    - **Encourages Candidness:** Anonymity promotes honest feedback, potentially unearthing genuine user sentiments.
  - Disadvantages:
    - **Insight Depth:** May not achieve the depth of insight afforded by case studies or the detailed observation of experiments.
    - **Feedback Limitations:** Responses might lack clarity or detail, with limited opportunities for follow-up or clarification.

8.2. Tool Evaluation Objectives

As discussed in 6: Part 1 Conclusions, participants expressed a preference against using a tool that functions as a black box, indicating a desire for greater control and transparency. Consequently, during our evaluation, we aim to determine if Human in the Loop AutoTDA meets participants' expectations for control and insight. Additionally, we seek to identify any new challenges associated with this tool to inform future improvements. Ultimately, we wish to assess whether participants now consider Human in the Loop AutoTDA a viable addition to their TDA workflow. This results in the following objectives:

- Assessing whether the tool provides adequate control.
- Examining the depth of insights offered by the tool.
- Identifying any additional issues that emerge.

Question	Interest of question
Is this approach easier than manual processing? Why?	Effectiveness
Is this information helpful? What else would you like to see here?	Insights
What step in this stage are you missing? What other functionality?	Control

Table 8.1: Questions asked during the stages in the evaluation

Question	Interest of question
What other functionalities would you like to see that we haven't discussed yet?	Control
Would you prefer a GUI or a notebook for this tool?	Control
Would you now use/trust this tool to do your TDA? (With and without your additions) Can you motivate/explain your answer?	Effectiveness

Table 8.2: Follow-Up questions at the end of the evaluation

- Evaluating the tool's overall effectiveness.

8.3. Tool Evaluation Protocol

To assess our tool effectively and meet our evaluation objectives, we initially considered three methodologies outlined in chapter 8.1.2. Our first attempt involved a case study that lasted about an hour, during which we observed users interacting with the tool. This approach provided insights into user behaviours and potential issues. However, a trial run revealed that one hour was insufficient for users to gain the necessary experience for a meaningful case study.

As a result, we shifted our evaluation strategy to a survey method. Before conducting interviews, we will prepare a notebook containing pre-executed code and its outputs. This preparation will streamline the interview process, avoiding delays and ensuring a smooth execution. The notebook showcases key functionalities pertinent to assessing the tool's control, insights, and effectiveness, as detailed in figure 7.4. These functionalities are critical for understanding how the tool operates at each stage, documented in Appendix C. During the survey; we will pose questions designed to evaluate the tool's control, insights, and effectiveness, as listed in table 8.1. Additionally, we have prepared a set of follow-up questions for the end of the interview, outlined in table 8.2. We assume that additional issues arise when the participant has questions about a topic or disagrees with a functionality.

This revised approach is estimated to take around 40 minutes, making it more feasible for participants to help. While this method may not provide the depth of evaluation that a case study could offer, it enables a practical and achievable assessment of the tool.

8.4. Participants for the Evaluation

For this case study, selecting the right participants is crucial to mirror real-world scenarios closely. We aim to recruit individuals with a Data Science and Machine Learning background to ensure they have the foundational knowledge required for this evaluation. However, it's crucial that they have not previously used Human in the Loop AutoTDA, as we are interested in observing their initial interactions with the tool. In previous interviews, seven respondents indicated their preference for a tool capable of automatic Feature Discovery, which allowed for greater control and transparency. We found five willing participants from this group of seven respondents willing to help. Although this group size is relatively small, their perspectives may echo the sentiments of a more significant segment of data specialists. By involving participants who share this viewpoint, we aim to assess whether the API's design, offering enhanced oversight and manipulation capabilities, aligns with their expectations and if it influences their willingness to adopt the tool. To compare the participants' answers with their answers in the interviews of chapter 4, we use

the same participant ID. In the invitation for the evaluation, we asked the participants to read already the workbook and see a five-minute video demonstration of the tool.

8.5. Key findings of the Evaluation

After conducting the evaluation interviews, we had to analyse these interviews. The approach for this process is the same as for the interviews in the first part, as described in section 4.7. Two persons label the first interview independently, then consent to a standardised labelling approach. With this agreement, the rest of the interviews can be labelled. After the labelling, the findings were summarised by the corresponding objectives. The findings of these objectives are described in this chapter, followed by corresponding quotes from the evaluation interviews.

8.5.1. Evaluation of User Control Adequacy

During the evaluation, participants navigated through various stages, and at each juncture, they were inquired about any functionalities or steps they found lacking. Here's a synthesis of their feedback, organized by the stage in the tool's workflow:

Input Data Stage

Nearly all participants (4 out of 5) were satisfied with the functionality offered during this stage. Moreover, two participants wanted the tool to support file types beyond CSV (P11, P17).

Finding Relationships Stage

Similarly, the majority (4 out of 5) felt that the control provided in this stage met their needs. Yet, one participant wished for the capability to assign aliases to tables for easier reference (P4).

Computing Join Trees Stage

In this phase, a majority (3 out of 5) reported no missing control functionalities. Nonetheless, one participant suggested the addition of functionality to modify join trees by incorporating joins (P14), while another sought the ability to apply a threshold for the relevance score (P11).

Evaluate Join Trees Stage

Here, a smaller portion (2 out of 5) indicated they were content with the existing functionalities. One participant wanted features allowing for the saving and subsequent reutilization of the tree to refine the model further, alongside the capability to employ multiple algorithms in a single execution (P14). Another participant desired a feature to export the top k most significant features (P3), and one more voiced a preference for manually inputting models rather than selecting from pre-defined options (P4).

> "Give me more control if I could just pass the classifier or the model object instead of an algorithm." (P4)

8.5.2. Assessment of Insight Depth Offered by the Tool

In our investigation of the tool's capability to deliver deep and actionable insights, participants were queried on whether the information provided by the tool was comprehensive enough for their needs. Their feedback, segmented by the tool's stages, is summarized as follows:

Input Data Stage

Most participants (3 out of 5) expressed a desire for more detailed information beyond just the names of the tables. Specifically, they were interested in understanding the size of each table and, if feasible, identifying the index columns to gain a better overview of their data at the outset.

> "Yeah. I'd like to see all the tables, definitely as you saw them here. Maybe I'd like to see some to have like a method to see some statistics about the tables as well, like how many rows that they have and things like that" (P11)

Finding Relationships Stage

During this stage, feedback highlighted a wish for enhancements in how relationships are presented. One participant (P11) was keen on identifying the top 'k' best relationships explicitly,

suggesting a preference for prioritization in relationship insights. Another (P4) wanted visibility into non-matching relationships, indicating a need for a comprehensive view of all potential data linkages, not just the successful ones.

Computing Join Trees Stage

Here, one respondent (P14) mentioned a desire for increased interactivity with the join tree visualization. This implies a need for users to manipulate or explore the join trees in more detail, suggesting that a static representation might not suffice for in-depth analysis.

Evaluate Join Trees Stage

In the final evaluation phase, participant feedback highlighted the need for more robust validation mechanisms. One participant (P11) expressed the necessity to confirm that every table was correctly joined and that the resulting dataset was accurate. Meanwhile, another (P4) found the provided information adequate but critiqued the presentation layout as confusing.

8.5.3. Identification of Emerging Issues

During the evaluation interviews, we also noticed that some issues occurred that did not fall under any specific stage of functionality.

Connect to an SQL Database

Two participants (P14 and P17) expressed a need for the tool to integrate directly with SQL databases. This feature would enable them to import data directly from their databases without the intermediary step of converting it to a CSV format. Furthermore, they envisioned a capability where, upon completion of data augmentation, the enhanced table could be exported back into their database as a view. Such automation would facilitate the recurrent use of the tool with dynamically updated database content, allowing for the seamless incorporation of the refreshed views into their ML models.

> ”The tool and then go from there, right? It’s going to be the same thing. It will be good if you have a connection. You can kind of give the URL of the database and then the database name so you can connect to the database and run this tool.” (P17)

Semantic Similarity

One participant (P4) desired the tool to compute semantic relationships, explaining that equi-joins would not suffice for their daily tasks.

Computational worries

Four out of five participants expressed concerns about the tool’s performance, particularly its runtime. They noted that the stage dedicated to finding relationships among 17 tables required 3 minutes. This observation led to apprehensions about the scalability of the process, especially when considering scenarios involving a larger scale, such as working with around a thousand tables, which they feared could significantly increase the computation time.

> ”Performance about these two because what you’re doing is joining a lot of stuff trying to find similarities between the tables so.□This is something I would be mostly concerned before I start using this tool basically.” (P11)

GUI/Notebook

We also wanted to know if the participants would prefer a GUI or a notebook version of the tool. All the participants said they would like to use the tool in a notebook. A notebook would be better for automating the process.

> ”As long as you have the Python library and then you have the data frame at the end, you can play with it as much as you want, right?” (P17)

8.5.4. Overall Effectiveness Evaluation

With each stage, we inquired with participants about the ease of using the tool compared to performing the process manually. Unanimously, all responses (18/18) indicated that utilizing the tool was easier than the manual approach.

Additionally, when questioned about their willingness to use and trust this tool for their Tabular Data Augmentation (TDA) tasks, all participants affirmed their trust. They expressed a commitment to review and validate the tool's output personally. In instances of inaccuracies, they planned to refine their workflow by adjusting the tool's functionalities, continuing this process until reaching a satisfactory level of confidence in the results.

9

Part 2 Conclusions

In the second stage of our study, we aimed to identify a novel tool capable of addressing the gaps in TDA, ensuring it would be practical for use by data specialists. The first stage of our research highlighted a significant preference among data specialists for transparency over using opaque, ”black box” algorithms in their TDA processes. Leveraging this insight, we developed Human in the Loop AutoTDA, a tool designed to automate the TDA process while granting users greater insight and control over their data. To evaluate Human in the Loop AutoTDA’s effectiveness, we conducted five evaluative interviews, during which participants assessed the tool’s control, insight provision, and overall effectiveness.

Feedback from these interviews indicated that Human in the Loop AutoTDA afforded users adequate control and delivered valuable insights across most stages of the TDA process. Participants identified certain functionalities that were absent, suggesting areas for enhancement in subsequent tool versions. Crucially, when asked whether they would use and trust Human in the Loop AutoTDA for their TDA workflows, all participants (5/5) responded affirmatively. This unanimous endorsement underscores the tool’s potential to aid data specialists in their TDA tasks significantly

10

Discussion, Limitations and Future Work

10.1. Discussion

The voiced preference for an insightful tool underscores a broader implication for developing TDA technologies: the necessity of transparency and user empowerment in the design of automated tools. This preference signals a shift towards tools that not only automate processes but also provide meaningful insights and explanations, thereby respecting and enhancing the user's expertise and understanding of their data. This requirement marks a critical consideration for future TDA tool development, pointing towards a paradigm where user engagement and algorithmic transparency are not just preferred but essential.

10.2. Limitations

10.2.1. Assumption Limitations

Not all assumptions collected

The assumptions in the literature are collected by a snowballing method. However, it may be possible that not all assumptions are collected. With possible assumptions still in the literature, it could be that there are still incorrect statements in the state-of-the-art literature.

10.2.2. Interview Limitations

Participants may not be representative of the exact TDA problem

Our study involved interviews with nineteen participants, a sample size sufficient for analysis. However, it's important to acknowledge that this group may not fully represent the broader population of data specialists who utilize Tabular Data Augmentation (TDA) in their work.

Participants are data specialists, not TDA experts

Finding participants who engage with Tabular Data Augmentation (TDA) as part of their daily tasks proved challenging, leading us to conduct interviews with data specialists. While these participants possess a solid understanding of data integration and machine learning processes, they might not have specialized expertise in the TDA workflow. Therefore, their perspectives may not fully capture the nuanced views of dedicated TDA specialists.

10.2.3. Tool Limitations

Memory issues when storing data

When constructing join trees, columns within these trees are selected through sampling to mitigate excessive memory usage. Nevertheless, in scenarios involving a large number of tables, the size of the join trees can expand significantly. Consequently, even with column sampling, the resultant tables may become so extensive that they pose memory consumption concerns.

Performance issues when using many files

During our tool evaluation, we utilized a dataset comprising 17 tables, which required several minutes to process. The execution time for the find_relationships() function significantly increases with the number of tables involved, potentially extending to hours or even days for larger datasets. We recommend executing this function as a background task or overnight to mitigate this issue. However, the feasibility of this approach for users remains uncertain.

10.2.4. Evaluation Limitations

Participants may give biased answers to please

In the evaluation phase, all five participants indicated trust in the tool and expressed willingness to incorporate it into their TDA workflow. However, this feedback might be influenced by a desire to provide positive responses, aiming to be courteous or to avoid disappointing us, which could introduce a bias into their responses.

Evaluation was in a controlled environment

During the evaluation, we presented only the standard approach, excluding the complete documentation and range of functions, thus showcasing a basic workflow. Consequently, we're uncertain about the participants' perspectives had they utilized the tool within their unique workflows and applied functions in a different sequence. This limited exposure may lead to discoveries or insights that could potentially fall short of user expectations.

10.3. Future work

This section contains some ideas for future work on this project.

10.3.1. Fuzzy and Semantic Joins

During the evaluation, participants inquired about the tool's capability to handle fuzzy or semantic joins, highlighting an area of interest in enhancing its functionality. We noted that some researchers, as referenced in works by Fan et al.[19] and Malysiak-Mrozek et al. [36], have developed tools that cater to these specific needs. An intriguing possibility for future development involves the integration of Large Language Models (LLMs) to facilitate decision-making processes in joining operations. Such advancements could significantly broaden Human in the Loop AutoTDA's utility and effectiveness, presenting a promising avenue for further research and implementation.

10.3.2. Union tables

Within the confines of this project, our discussion primarily focused on enhancing machine learning model outcomes through the horizontal expansion of datasets by joining tables. However, an alternative strategy involves vertical dataset augmentation, whereby tables with similar schemas and meanings are combined using union operations. This approach, as highlighted in the work by Khatiwada [24], presents another viable method for improving model performance. Such vertical integration could offer complementary benefits to the horizontal methods discussed, warranting further exploration in future research endeavours.

10.3.3. Do queries from and to a Database

Among the participants of the first interviews, five mentioned utilizing relational databases as their data sources. Specifically, one participant (P14) expressed interest in a feature allowing direct connectivity with these databases. Such functionality would not only facilitate the swift import of data into the tool but also enable the efficient export of augmented tables back into the database as views. This automation would streamline the process, allowing users to employ the algorithm to identify optimal parameters and construct the join tree, subsequently generating a database view automatically. This view could be directly utilized for machine learning models, enhancing workflow efficiency and integration.

10.3.4. Implement the missing control and insights

To enhance the tool's effectiveness further, implementing the functionalities related to control and insights identified as missing during the evaluation phase could be highly beneficial. This improvement would address specific user needs and preferences, making the tool more versatile and valuable for its intended audience.

References

[1] Ziawasch Abedjan et al. *DataXFormer: A robust transformation discovery system*. May 2016. DOI: 10.1109/ICDE.2016.7498319.

[2] A. E. Blandford. “Semi-structured qualitative studies”. eng. In: *In: Soegaard, M and Dam, R, (eds.) The Encyclopedia of Human-Computer Interaction. Interaction Design Foundation: Denmark. (2013)*. Ed. by M. Soegaard and R. Dam. Denmark: Interaction Design Foundation, 2013. URL: http://www.interaction-design.org/encyclopedia/semi-structured_qualitative_studies.html (visited on 03/03/2024).

[3] Alex Bogatu et al. “Dataset Discovery in Data Lakes”. In: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. Apr. 2020, pp. 709–720. DOI: 10.1109/ICDE48307.2020.00067. URL: https://ieeexplore.ieee.org/document/9101607 (visited on 10/17/2023).

[4] Michael J. Cafarella, Alon Halevy, and Nodira Khoussainova. “Data integration for the relational web”. In: *Proceedings of the VLDB Endowment* 2.1 (Aug. 2009), pp. 1090–1101. ISSN: 2150-8097. DOI: 10.14778/1687627.1687750. URL: https://dl.acm.org/doi/10.14778/1687627.1687750 (visited on 11/23/2023).

[5] Raul Castro Fernandez et al. “Aurum: A Data Discovery System”. In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. ISSN: 2375-026X. Apr. 2018, pp. 1001–1012. DOI: 10.1109/ICDE.2018.00094. URL: https://ieeexplore.ieee.org/abstract/document/8509315?casa_token=qt7WxWc3jucAAAAA:RyYJ0EMFvqxy8a5v8ZFu4gfbDZDy8U9unP4uUSFG47nBgvJLLqVbcfkbDUajlVC0v9EnwWNw (visited on 11/20/2023).

[6] Adriane Chapman et al. “Dataset search: a survey”. In: *The VLDB Journal* 29.1 (Jan. 2020). arXiv:1901.00735 [cs], pp. 251–272. ISSN: 1066-8888, 0949-877X. DOI: 10.1007/s00778-019-00564-x. URL: http://arxiv.org/abs/1901.00735 (visited on 03/19/2024).

[7] Nadiia Chepurko et al. *ARDA: Automatic Relational Data Augmentation for Machine Learning*. arXiv:2003.09758 [cs, stat]. Mar. 2020. DOI: 10.48550/arXiv.2003.09758. URL: http://arxiv.org/abs/2003.09758 (visited on 10/17/2023).

[8] Tianji Cong et al. *WarpGate: A Semantic Join Discovery System for Cloud Data Warehouses*. Jan. 2023. DOI: 10.48550/arXiv.2212.14155. URL: http://arxiv.org/abs/2212.14155 (visited on 10/17/2023).

[9] Reidar Conradi, ed. *Empirical methods and studies in software engineering: experiences from ESERNET*. en. Lecture notes in computer science 2765. Berlin Heidelberg: Springer, 2003. ISBN: 978-3-540-40672-3.

[10] Anish Das Sarma et al. “Finding related tables”. en. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. Scottsdale Arizona USA: ACM, May 2012, pp. 817–828. ISBN: 978-1-4503-1247-9. DOI: 10.1145/2213836.2213962. URL: https://dl.acm.org/doi/10.1145/2213836.2213962 (visited on 10/17/2023).

[11] Hong-Hai Do and Erhard Rahm. “Chapter 53 - COMA — A system for flexible combination of schema matching approaches”. In: *VLDB ’02: Proceedings of the 28th International Conference on Very Large Databases*. Ed. by Philip A. Bernstein et al. San Francisco: Morgan Kaufmann, Jan. 2002, pp. 610–621. ISBN: 978-1-55860-869-6. DOI: 10.1016/B978-155860869-6/50060-3. URL: https://www.sciencedirect.com/science/article/pii/B9781558608696500603 (visited on 02/20/2024).

[12] Hong-Hai Do and Erhard Rahm. “COMA — A system for flexible combination of schema matching approaches”. en. In: *VLDB ’02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 2002, pp. 610–621. ISBN: 978-1-55860-869-6. DOI: 10.1016/B978-155860869-6/50060-3. URL: https://linkinghub.elsevier.com/retrieve/pii/B9781558608696500603 (visited on 03/03/2024).

[13] Owen Doody and Maria Noonan. “Preparing and conducting interviews to collect data”. eng. In: *Nurse Researcher* 20.5 (May 2013), pp. 28–32. ISSN: 1351-5578. DOI: 10.7748/nr2013.05.20.5.28.e327.

[14] Steve Easterbrook et al. “Selecting Empirical Methods for Software Engineering Research”. en. In: *Guide to Advanced Empirical Software Engineering*. Ed. by Forrest Shull, Janice Singer, and Dag I. K. Sjøberg. London: Springer London, 2008, pp. 285–311. ISBN: 978-1-84800-043-8 978-1-84800-044-5. DOI: 10.1007/978-1-84800-044-5_11. URL: http://link.springer.com/10.1007/978-1-84800-044-5_11 (visited on 03/02/2024).

[15] Julian Eberius et al. “Top-k entity augmentation using consistent set covering”. In: *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*. SSDBM ’15. New York, NY, USA: Association for Computing Machinery, June 2015, pp. 1–12. ISBN: 978-1-4503-3709-0. DOI: 10.1145/2791347.2791353. URL: https://doi.org/10.1145/2791347.2791353 (visited on 10/17/2023).

[16] Nova Eka Diana and Ikrima Hanana Ulfa. “Measuring Performance of N-Gram and Jaccard-Similarity Metrics in Document Plagiarism Application”. en. In: *Journal of Physics: Conference Series* 1196 (Mar. 2019), p. 012069. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1196/1/012069. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1196/1/012069 (visited on 02/23/2024).

[17] Nick Erickson et al. *AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data*. Mar. 2020. DOI: 10.48550/arXiv.2003.06505. URL: http://arxiv.org/abs/2003.06505 (visited on 03/04/2024).

[18] Mahdi Esmailoghli, Jorge-Arnulfo Quiané-Ruiz, and Ziawasch Abedjan. “COCOA: COrrelation COefficient-Aware Data Augmentation”. en. In: ().

[19] Grace Fan et al. *Semantics-aware Dataset Discovery from Data Lakes with Contextualized Column-based Representation Learning*. Jan. 2023. DOI: 10.48550/arXiv.2210.01922. URL: http://arxiv.org/abs/2210.01922 (visited on 10/17/2023).

[20] David Hadwick and Shimeng Lan. *Lessons to Be Learned from the Dutch Childcare Allowance Scandal: A Comparative Review of Algorithmic Governance by Tax Administrations in the Netherlands, France and Germany*. en. Place: Rochester, NY Type: SSRN Scholarly Paper. Oct. 2021. URL: https://papers.ssrn.com/abstract=4282704 (visited on 02/21/2024).

[21] R. W. Hamming. “Error detecting and error correcting codes”. In: *The Bell System Technical Journal* 29.2 (Apr. 1950), pp. 147–160. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1950.tb00463.x. URL: https://ieeexplore.ieee.org/abstract/document/6772729 (visited on 02/23/2024).

[22] Andra Ionescu et al. “Join Path-Based Data Augmentation for Decision Trees”. In: *2022 IEEE 38th International Conference on Data Engineering Workshops (ICDEW)*. May 2022, pp. 84–88. DOI: 10.1109/ICDEW55742.2022.00018. URL: https://ieeexplore.ieee.org/document/9814493 (visited on 11/20/2023).

[23] Aamod Khatiwada et al. “Integrating Data Lake Tables”. In: *Proceedings of the VLDB Endowment* 16.4 (Dec. 2022), pp. 932–945. ISSN: 2150-8097. DOI: 10.14778/3574245.3574274. URL: https://dl.acm.org/doi/10.14778/3574245.3574274 (visited on 11/22/2023).

[24] Aamod Khatiwada et al. “SANTOS: Relationship-based Semantic Table Union Search”. In: *Proceedings of the ACM on Management of Data* 1.1 (2023), 9:1–9:25. DOI: 10.1145/3588689. URL: https://dl.acm.org/doi/10.1145/3588689 (visited on 10/17/2023).

[25] B. Kitchenham, L. Pickard, and S.L. Pfleeger. “Case studies for method and tool evaluation”. In: *IEEE Software* 12.4 (July 1995), pp. 52–62. ISSN: 07407459. DOI: 10.1109/52.391832. URL: http://ieeexplore.ieee.org/document/391832/ (visited on 03/26/2024).

[26] Barbara Ann Kitchenham. “Evaluating software engineering methods and tool part 1: The evaluation context and evaluation methods”. en. In: *ACM SIGSOFT Software Engineering Notes* 21.1 (Jan. 1996), pp. 11–14. ISSN: 0163-5948. DOI: 10.1145/381790.381795. URL: https://dl.acm.org/doi/10.1145/381790.381795 (visited on 02/19/2024).

[27] Aleksander Ko, Abdur Chowdhury, and Joshua Alspector. “Data duplication: an imbalance problem ?” en. In: ().

[28] Christos Koutras et al. “Valentine: Evaluating Matching Techniques for Dataset Discovery”. In: *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. Apr. 2021, pp. 468–479. DOI: 10.1109/ICDE51399.2021.00047. URL: https://ieeexplore.ieee.org/abstract/document/9458921 (visited on 03/03/2024).

[29] Arun Kumar, Jeffrey Naughton, and Jignesh M. Patel. “Learning Generalized Linear Models Over Normalized Data”. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 1969–1984. ISBN: 978-1-4503-2758-9. DOI: 10.1145/2723372.2723713. URL: https://dl.acm.org/doi/10.1145/2723372.2723713 (visited on 11/16/2023).

[30] Arun Kumar, Feng Niu, and Christopher Ré. “Hazy: making it easier to build and maintain big-data analytics”. In: *Communications of the ACM* 56.3 (2013), pp. 40–49. ISSN: 0001-0782. DOI: 10.1145/2428556.2428570. URL: https://dl.acm.org/doi/10.1145/2428556.2428570 (visited on 10/17/2023).

[31] Arun Kumar et al. “To Join or Not to Join?: Thinking Twice about Joins before Feature Selection”. en. In: *Proceedings of the 2016 International Conference on Management of Data*. San Francisco California USA: ACM, June 2016, pp. 19–34. ISBN: 978-1-4503-3531-7. DOI: 10.1145/2882903.2882952. URL: https://dl.acm.org/doi/10.1145/2882903.2882952 (visited on 11/16/2023).

[32] Jundong Li et al. “Feature Selection: A Data Perspective”. en. In: *ACM Computing Surveys* 50.6 (Nov. 2018), pp. 1–45. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3136625. URL: https://dl.acm.org/doi/10.1145/3136625 (visited on 02/29/2024).

[33] Yuliang Li et al. “Data augmentation for ML-driven data preparation and integration”. en. In: *Proceedings of the VLDB Endowment* 14.12 (July 2021), pp. 3182–3185. ISSN: 2150-8097. DOI: 10.14778/3476311.3476403. URL: https://dl.acm.org/doi/10.14778/3476311.3476403 (visited on 11/16/2023).

[34] Jiabin Liu et al. “Feature Augmentation with Reinforcement Learning”. en. In: *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. Kuala Lumpur, Malaysia: IEEE, May 2022, pp. 3360–3372. ISBN: 978-1-66540-883-7. DOI: 10.1109/ICDE53745.2022.00317. URL: https://ieeexplore.ieee.org/document/9835530/ (visited on 10/27/2023).

[35] Jayant Madhavan, Philip A Bernstein, and Erhard Rahm. “Generic Schema Matching with Cupid”. en. In: ().

[36] Bozena Malysiak-Mrozek, Anna Lipinska, and Dariusz Mrozek. “Fuzzy Join for Flexible Combining Big Data Lakes in Cyber-Physical Systems”. en. In: *IEEE Access* 6 (2018), pp. 69545–69558. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2879829. URL: https://ieeexplore.ieee.org/document/8525267/ (visited on 03/27/2024).

[37] S. Melnik, H. Garcia-Molina, and E. Rahm. “Similarity flooding: a versatile graph matching algorithm and its application to schema matching”. In: *Proceedings 18th International Conference on Data Engineering*. Feb. 2002, pp. 117–128. DOI: 10.1109/ICDE.2002.994702. URL: https://ieeexplore.ieee.org/document/994702 (visited on 03/03/2024).

[38] Fatemeh Nargesian et al. “Table union search on open data”. en. In: *Proceedings of the VLDB Endowment* 11.7 (Mar. 2018), pp. 813–825. ISSN: 2150-8097. DOI: 10.14778/3192965.3192973. URL: https://dl.acm.org/doi/10.14778/3192965.3192973 (visited on 10/23/2023).

[39] Redactie. *'Belastingdienst gebruikte algoritme dat lage inkomens selecteerde voor extra fraudecontroles'*. nl-NL. Publication Title: de Volkskrant. Nov. 2021. URL: https://www.volkskrant.nl/nieuws-achtergrond/belastingdienst-gebruikte-algoritme-dat-lage-inkomens-selecteerde-voor-extra-fraudecontroles~bac84336/ (visited on 02/25/2024).

[40] Jennifer Rowley. "Conducting research interviews". In: *Management Research Review* 35.3/4 (Jan. 2012), pp. 260–271. ISSN: 2040-8269. DOI: 10.1108/01409171211210154. URL: https://doi.org/10.1108/01409171211210154 (visited on 11/23/2023).

[41] C.B. Seaman. "Qualitative methods in empirical studies of software engineering". en. In: *IEEE Transactions on Software Engineering* 25.4 (Aug. 1999), pp. 557–572. ISSN: 00985589. DOI: 10.1109/32.799955. URL: http://ieeexplore.ieee.org/document/799955/ (visited on 03/02/2024).

[42] Vraj Shah, Arun Kumar, and Xiaojin Zhu. *Are Key-Foreign Key Joins Safe to Avoid when Learning High-Capacity Classifiers?* June 2017. DOI: 10.48550/arXiv.1704.00485. URL: http://arxiv.org/abs/1704.00485 (visited on 10/17/2023).

[43] *Simplifying the Difference: Machine Learning vs Deep Learning - Singapore Computer Society*. URL: https://www.scs.org.sg/articles/machine-learning-vs-deep-learning (visited on 02/22/2024).

[44] Carol A. B Warren. "Qualitative Interviewing". In: *Handbook of Interview Research*. SAGE Publications, Inc., 2011, pp. 83–102. ISBN: 978-0-7619-1951-3. URL: http://dx.doi.org/10.4135/9781412973588.

[45] *What Is Reinforcement Learning?* en. URL: https://nl.mathworks.com/discovery/reinforcement-learning.html (visited on 02/24/2024).

[46] Joost C. F. de Winter, Samuel D. Gosling, and Jeff Potter. "Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data". In: *Psychological Methods* 21.3 (2016), pp. 273–290. ISSN: 1939-1463. DOI: 10.1037/met0000079.

[47] Lei Yu and Huan Liu. "Efficient Feature Selection via Analysis of Relevance and Redundancy". en. In: ().

[48] Meihui Zhang et al. "Automatic discovery of attributes in relational databases". en. In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. Athens Greece: ACM, June 2011, pp. 109–120. ISBN: 978-1-4503-0661-4. DOI: 10.1145/1989323.1989336. URL: https://dl.acm.org/doi/10.1145/1989323.1989336 (visited on 03/03/2024).

[49] Yi Zhang and Zachary G. Ives. "Juneau: data lake management for Jupyter". en. In: *Proceedings of the VLDB Endowment* 12.12 (Aug. 2019), pp. 1902–1905. ISSN: 2150-8097. DOI: 10.14778/3352063.3352095. URL: https://dl.acm.org/doi/10.14778/3352063.3352095 (visited on 11/21/2023).

[50] Erkang Zhu et al. "JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes". In: *Proceedings of the 2019 International Conference on Management of Data*. SIGMOD '19. New York, NY, USA: Association for Computing Machinery, June 2019, pp. 847–864. ISBN: 978-1-4503-5643-5. DOI: 10.1145/3299869.3300065. URL: https://dl.acm.org/doi/10.1145/3299869.3300065 (visited on 10/17/2023).

A

Assumptions in the state-of-the-art TDA literature

A.1. Tools & Methods

If users are unaware of what is available, or unable to trust what they find, they end up reinventing their own schemas, import processes and cleaning process. [49]

For another, the proliferation of ever-evolving algorithms to gain insights from data (in the name of machine learning, data mining, statistical analysis, and so on) can often be daunting to a developer with a particular dataset and specific goals: the developer not only has to keep up with the state of the art, but also must expend significant development effort in experimenting with different algorithms. [30]

Juneau assumes workflows are specified as sequences of cells within computational notebooks, hosted for multiple users on the cloud, such as Jupyter Notebook/JupyterLab, Apache Zeppelin, or RStudio. [49]

A.2. Aquisition / Collection: data catalogue, external data, searching

For example, assume that a data scientist from the payment team in Ant Group wants to predict whether a user is a scammer or not, although she might be able to quickly collect useful information (features) from her domain (such as basic user profile, payment transaction history, etc.), there is plenty valuable information outside of her domain (possibly even out of her awareness) that is also useful (such as credit data or LBS data). [34]

We hypothesize that during interactive data science tasks, users often want to search the data lake, not only by keyword, but using a table, to find other related tables. [49]

Data publishers often do not provide search functionality beyond simple keyword search on the metadata. This metadata varies greatly in quality across different datasets and publishers. The lack of sophisticated search functionality creates barriers for data scientists who want to use Open Data for their research. [38]

However, for the problem of searching data lakes, using a threshold may confuse users, who have no knowledge of what data exists in the lake and therefore do not know what is a good threshold that will retrieve some, but not too many answers [50]

A search engine for joinable tables asks a user to input a table and specify a column, and returns tables that can be joined with the input table on the specified column. [50]

However, when dealing with enterprise data or open data, keyword search via textual queries or over metadata is challenging because metadata may be missing, inconsistent, or incomplete [24]

Their tables are often assumed to be “entity tables” (i.e., all attributes are properties of a known common entity represented by a single attribute) or they use a subject column and a single relationship from that column to search for tables [24]

A.3. Define needs: column names, ID names, input type (files, dbs)

Analysts want to find datasets according to different criteria, e.g., datasets with a specific schema, similar datasets of one of reference, joinable datasets [5]

Analyst who is building the stock change prediction model may start with a search for tables that include metrics of relevance, such as stock prices and mentions in social media (schema similarity). [5]

analyst may be interested in finding similar datasets to the ones found so far to make sure no information is missing (content similarity). [5]

Firstly, it promotes issues of consistency, as the sources may be of high quality when considered on their own, but still do not form a consistent answer. [15]

A.4. Diverge pipeline: input size

Almost all ML toolkits assume that the input is a single table, but many datasets are not stored as single tables due to normalization [31]

in some applications, analysts have dozens of tables in the input and prefer to join only a few “most helpful” tables (colloquially called source selection). Answering our question might help analysts assess which tables matter less for accuracy. [31]

Moreover, there might be hundreds of related tables, creating a large amount of work for the user. [7]

Doing this manually is extremely labour intensive, as the analyst has to browse many files and tables to find those that might match, and then try to find key attributes that can be used to join these tables together. [5]

A.5. Preparation: profiling, wrangling, creatiion, cleaning etc

However, since web tables are noisy in nature, the requirements here are more flexible: for example, overlap between related tables or renaming of attributes should be allowed. [10]

When integrating data from multiple sources, there is often a need to perform transformations. [19]

However, it is clearly very tedious for a human to construct such transformations manually, which is one of the reasons why data analysts spend the overwhelming majority of their time “massaging” their data into usable form. [19]

Analysts sometimes judge foreign keys as being too “uninterpretable” and simply drop them. [31]

While existing data management systems predominantly assume that data has rigid, precise semantics, increasingly more data (albeit valuable) contains imprecision or inconsistency. [30]

A.6. Exploration: understanding NEW data, time spent

We thus assume (query column) q is selected explicitly by the user. [18]

Traditionally, a user has had to invest significant effort in deciding what data can usefully augment a classification problem. [7]

Doing this manually is extremely labour intensive, as the analyst has to browse many files and tables to find those that might match, and then try to find key attributes that can be used to join these tables together. [5]

While the entity-table assumption may be mostly true in web tables, data lake tables (in both open data and enterprise lakes) tend to be much wider and may describe the interplay between more than one entity. [24]

few people have a good understanding of data and their relationships [8]

data is added in a much less modeled state as cloud computing power and storage become easily accessible; [8]

business users who do not have comprehensive knowledge of underlying schema designs, are usually unaware of relationships between datasets [8]

Analysts spend more time finding relevant data to answer the questions at hand than analyzing it [5]

[the feature selection process] can be both painful and wasteful because the increase in the number of features might make it harder for analysts to explore the data and also increases the runtime of ML and feature selection methods. [31]

A.7. Relatedness: common knowledge, similarity

Users often have a hard time distinguishing a semantically meaningful join from an meaningless join if they don't know details about the join itself; [7]

The correlation coefficient is one of the extensively used statistical measures in data science. Data scientists use the correlation coefficient to find dependencies in the data and identify possible causal relationships [18]

in some applications, analysts have dozens of tables in the input and prefer to join only a few "most helpful" tables (colloquially called source selection). Answering our question might help analysts assess which tables matter less for accuracy. [31]

First, analysts often join all tables almost by instinct. Our work shows that this might lead to much poorer performance without much accuracy gain. [31]

A.8. Integration: join type, join problems, pk-fk, pipeline

To join the taxi dataset with the NYC weather dataset, one needs to define an appropriate join key and potentially pre-aggregate and interpolate the weather dataset, as well as deal with null values. [7]

Thus, analysts typically perform KFK joins before ML to construct a single table that collects features from all base tables and then apply a feature selection method (either explicitly or implicitly [17]) over the entire set of features. [31]

Or, having already found a handful of relevant datasets, the analyst may want to find a join path to join them together (a primary-key/foreign-key (PK/FK) candidate) [5]

But from conversations with data scientists at many enterprise and Web companies, we learned that even this simple process of procuring features by joining tables could be painful in practice because different tables could be “owned” by different teams with different access restrictions. This slows down the ML analytics lifecycle [42]

When coercing multiple data source’s values into one result, properties important for data analysis such as transparency, lineage and trustworthiness of the result are lost. [15]

This is because in many domains the user can not blindly trust an automatic integration result, no matter how sophisticated the method used to create it. [15]

After discovery, data scientists would often integrate the discovered tables before analyzing and applying statistical tools [23]

A.9. Feature selection: feature engineering, manual / automatic selection

“[the feature selection process] can be both painful and wasteful because the increase in the number of features might make it harder for analysts to explore the data and also increases the runtime of ML and feature selection methods. [31]

Thus, analysts typically perform KFK joins before ML to construct a single table that collects features from all base tables and then apply a feature selection method (either explicitly or implicitly [17]) over the entire set of features. [31]

in applications in which analysts explore features by being in the loop [27, 46], having fewer tables and features might make exploration easier. [31]

Finally, most of the burden of feature engineering (designing and choosing features) for ML today falls on analysts [31]

Feature selection algorithms can broadly fall into the filter model or the wrapper model [47]

A.10. Modelling: baseline, training, hyper-param tuning

So, MovieID can help predict future ratings. Note that closed domain does not mean new MovieID values can never occur! It means that analysts build models using only the movies seen so far but revise their feature domains and update ML models periodically (say, monthly) to absorb movies added recently [31]

In this case, the initial training data that she prepared
is sub-optimal, and it is desirable to augment [34]

B

An Overview of the Created Labels

Category	Subcategory
collection	data catalogue external data someone else tools
data preparation	preparation tools
define needs	column names define problem file names id column id column name input type tools
diverge pipeline	1K features 1K tables 1M rows data size
establish goals	data catalogue domain knowledge someone else
exploration	time spent tools understanding new data
feat selection	auto selection feat engineering manual selection
follow-up	autofeat company size data understanding degree ideal world industry sector last time DI task problems tools working with data
integration	join column

	join problems join type pipeline pk-fk tools
modelling	baseline other training
relatedness	business knowledge common knowledge similarity
roles	data analyst data engineer data scientist ML engineer other
stories	

Table B.1: Categories and subcategories of labels for the interviews

C

Evaluation Notebook

Recap

Last interview, we gave you a dataset (school) of 17 tables. Your task was to augment the base table by integrating tables and selecting features which would increase the accuracy of a tree-based ML model. The target column was "class" in the base table "base.csv"

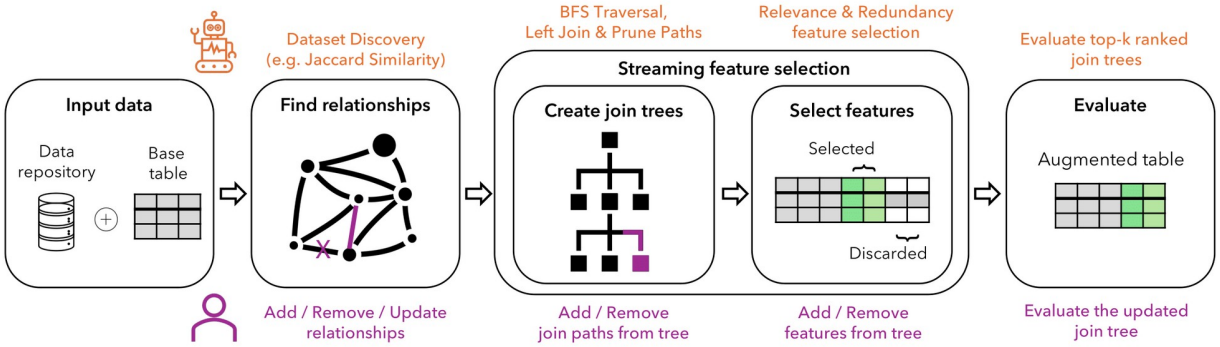
Now during last task, you had to do the dat aumgentation manually. This process takes a lot of time and effort when doing manually. Therefore, we asked you the question: Would you rely on a tool that does this manually? Your answers were yes, but or no because, I would like to have more control and insights during the workflow. This resulted in creating a tool that does the autoTDA, but also gives you control and insights during the workflow. We would like to demonstrate the tool for you and ask for your opinions.

Introduction

AutoTDA (Automatic Tabluar Data Augmentation)

AutoTDA is a Python-based, user-friendly library that incor- porates our human-in-the-loop methodology for feature discovery, designed for seamless integration in any notebook environment. AutoTDA streamlines the process of selecting and integrating relevant tables from a dataset collection into the base table, thus creating an augmented table. Additionally, AutoTDA employs heuristic-based feature selection strategies to eliminate redundant or irrelevant features from this augmented table. By doing so, AutoTDA notably enhances the efficiency and accuracy of subsequent machine learning operations.

An over of the AutoTDA pipeline is shown below.

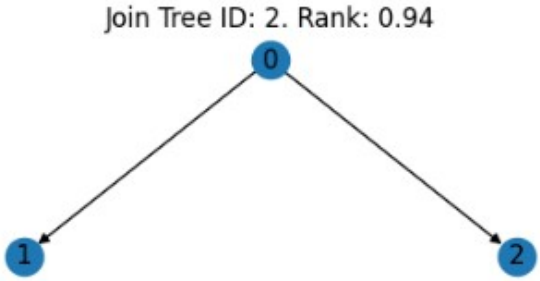


Relationships

The connections between tables are called Relationships. These are calculated in the first step. These show the similarities between columns in different tables.

Join Trees

Join trees are a combination of multiple integrated tables. Every node is a table and every edge is a connection between 2 columns between the tables. An exmple of a join tree is shown below.



Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/2013_NYC_School_Survey.csv

Intialisation

- First, we start by importing the package and creating an instance of Tabular Data Augmentation
- Then we set up the base table (school/base.csv) and select the target_column (class)
- We also set up the repository which is school for now. You can select multiple repositories

```
from autotda.autofeat_class import TDA
autoTDA = TDA(repository_location="../data/benchmark")
autoTDA.set_base_table(base_table="school/base.csv",
target_column="class")
autoTDA.set_dataset_repository(dataset_repository=["school"])
```

```
/home/zeger/hci-auto-feat/.eval_venv/lib/python3.10/site-packages/
tqdm/auto.py:21: TqdmWarning: IProgress not found. Please update
jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html

  from .autonotebook import tqdm as notebook_tqdm
```

Automatic Tabular Data Augmentation

To aid by the Tabular Data Augmentation, there is a function augment_dataset(). This function does the whole proces automatically without any interference. This function returns the best created augmented table. However, the function is a black box. You can set the explain=True flag, but that only explains the end result.

```
augmented_table = autoTDA.augment_dataset(explain=False)
```

```
Calculating relationships...
 100%|██████████| 136/136.0 [02:52<00:00,  1.48s/it]
```

Control and Insights

The function augment_dataset() does everything, however. You exlained to us during the interveiws that you would like more control and insights in the algorithm. Therefore we created 4 stages in the algorithm where the user can have more control while showing the intermediate results. For every stage we show how you can adjust the algorithm and how to show the intermediate results.

Stage 0: Input Data

Control

In the first stage, you can change the data repositories by adding and removing tables. There are 2 functions called remove_table and add_tables. These functions removes tables from the existing data repostiory or adds new tables to the data repository.

```
autoTDA.remove_table(table="school/oss.csv")
# autoTDA.add_table(table="credit/table_0_0.csv")
```

Insights

To show what tables are in the current repository, you can call the get_tables_repository() function

```
autoTDA.get_tables_repository()
```

```
['school/pe.csv',
 'school/crime.csv',
 'school/s2tr.csv',
 'school/math.csv',
 'school/yabc.csv',
 'school/2013_NYC_School_Survey.csv',
 'school/disc.csv',
 'school/sat.csv',
 'school/2010_Gen_Ed_Survey_Data.csv',
 'school/gender.csv',
 'school/esl.csv',
 'school/ap.csv',
 'school/transfer.csv',
 'school/qr.csv',
 'school/Schools_Progress_Report_2012-2013.csv',
 'school/base.csv']
```

Stage 1: Find Relationships

For every table in the repository, we want to calculate the relationships between the tables. The function find_relationships() calculates the similarity score between all columns in the

repository. Setting a relationship_threshold will only include the relationships in the workflow that are higher than that score.

```
autoTDA.find_relationships(explain=True, relationship_threshold=0.6,
matcher="coma")
```

```
 1. AutoFeat computes the relationships between 16 tables from the
datasets: ['school'] repository using coma similarity score with a
threshold of 0.6 (i.e., all the relationships with a similarity < 0.6
will be discarded) and excludes tables: ['school/oss.csv']

Calculating relationships...
```

Control

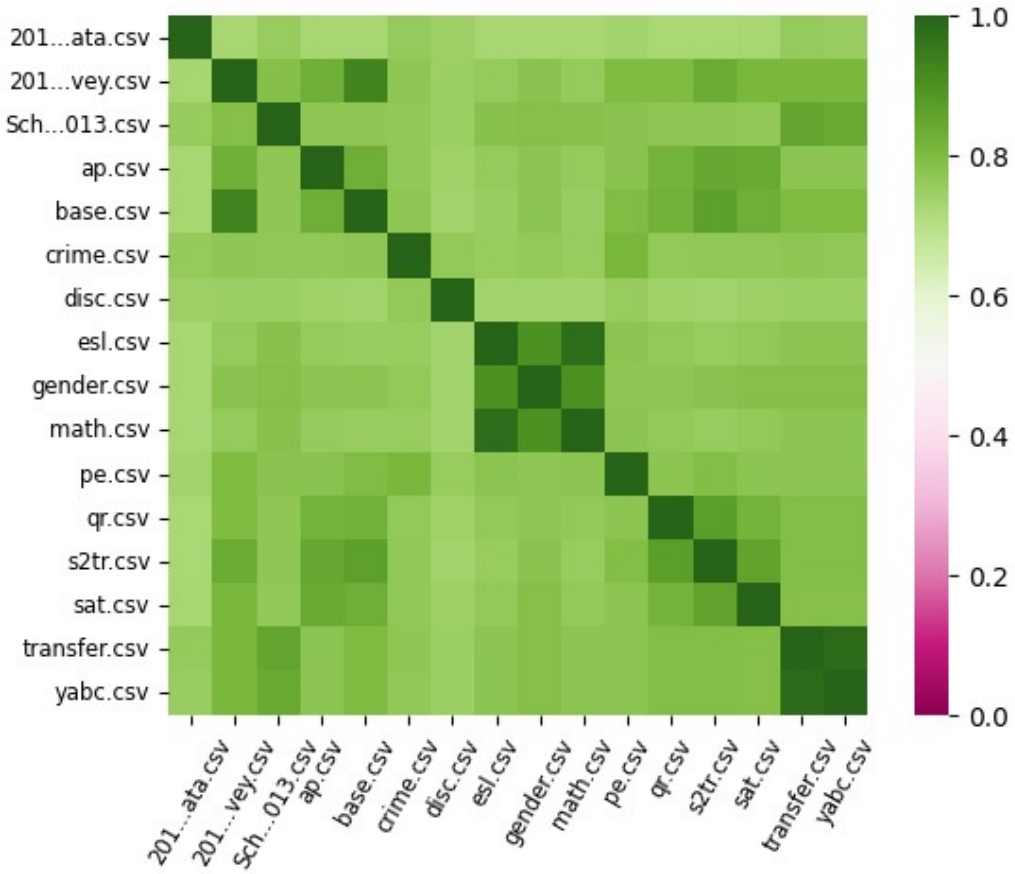
If you are sure that tables are related. Or you are sure that tables are not related. Then, you can adjust the relationships with remove_relationships(), update_relationships() and add_relationships()

```
autoTDA.remove_relationship(table1="school/ap.csv", col1="DBN",
table2="credit/table_0_0.csv", col2="DBN")
autoTDA.update_relationship(table1="school/ap.csv", col1="DBN",
table2="school/qr.csv", col2="DBN Name", weight=1)
autoTDA.add_relationship(table1="school/ap.csv", col1="SchoolName",
table2="school/qr.csv", col2="School Name", weight=0.2)
```

Insights

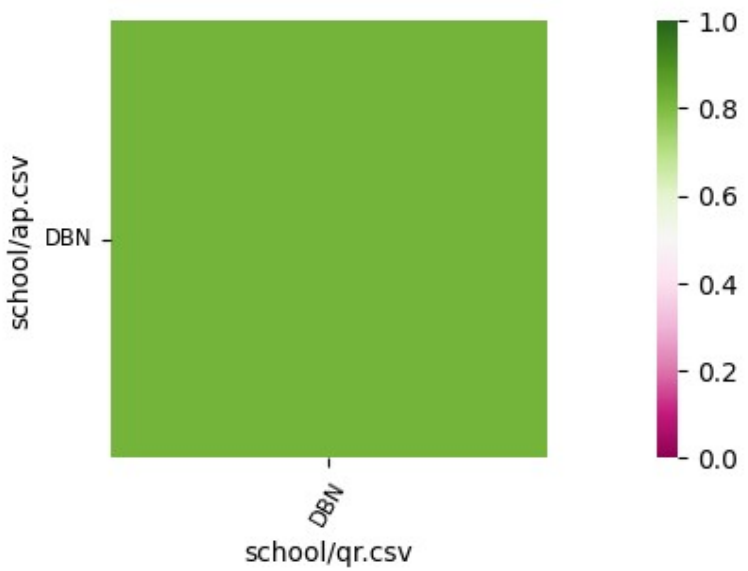
If you want to see the relationships between the all the tables. You can call the display_best-relationships() function. This will display the best relationship score between all tables. Every relationship score is the best score between all columns of those tables.

```
autoTDA.display_best_relationships()
```



If you want to see what the relationships scores are between the columns of 2 tables. you can call the display_table_relationship() function.

```
autoTDA.display_table_relationship(table1="school/ap.csv",
table2="school/qr.csv")
```



You can also let the tool explain the relationship between 2 tables for you.

```
autoTDA.explain_relationship(table1="school/ap.csv",
table2="school/qr.csv")
```

```
Relationships between school/ap.csv and school/qr.csv:

from_table     to_table         weight

-------------  -------------  --------

school/ap.csv  school/qr.csv  0.816558

school/qr.csv  school/ap.csv  0.816558
```

Stage 2: Computing Join Trees

Now that we have the relationships, we can integrate the tables. This function creates all possible join trees.

```
autoTDA.compute_join_trees()
```

```
Calculating join trees...

/home/zeger/hci-auto-feat/.eval_venv/lib/python3.10/site-packages/
pandas/core/arrays/base.py:513: RuntimeWarning: invalid value
encountered in cast

  result = np.asarray(self, dtype=dtype)
```

Insights

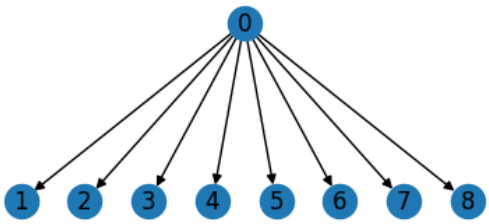
With the display_join_trees(), you can see all the generated join trees. Every node is table and every edge is a join between those 2 tables.

```
autoTDA.display_join_trees()
```

Join Tree ID: 1. Rank: 1.25

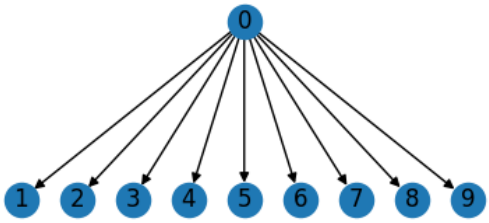
Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv

Join Tree ID: 9. Rank: 1.10



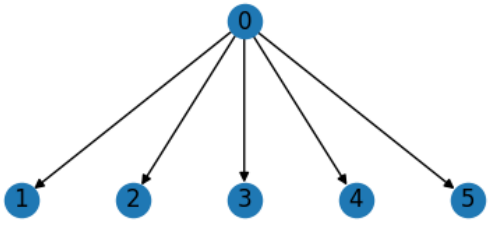
Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/Schools_Progress_Report_2012-2013.csv
3	school/disc.csv
4	school/esl.csv
5	school/gender.csv
6	school/math.csv
7	school/oss.csv
8	school/pe.csv

Join Tree ID: 10. Rank: 1.09



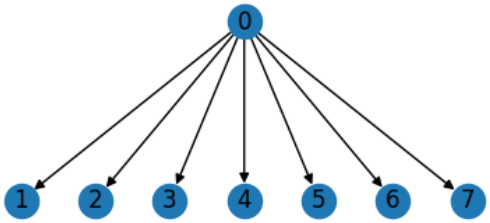
Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/Schools_Progress_Report_2012-2013.csv
3	school/disc.csv
4	school/esl.csv
5	school/gender.csv
6	school/math.csv
7	school/oss.csv
8	school/pe.csv
9	school/qr.csv

Join Tree ID: 6. Rank: 1.06



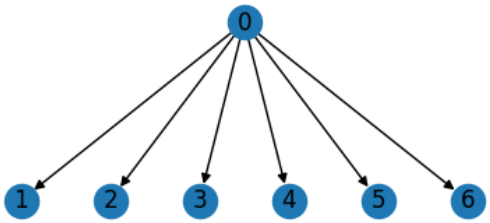
Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/Schools_Progress_Report_2012-2013.csv
3	school/disc.csv
4	school/esl.csv
5	school/gender.csv

Join Tree ID: 8. Rank: 1.05



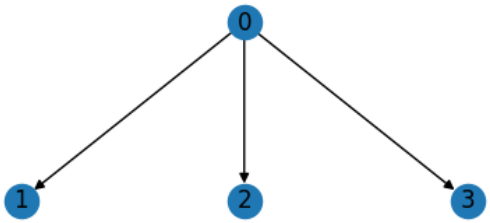
Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/Schools_Progress_Report_2012-2013.csv
3	school/disc.csv
4	school/esl.csv
5	school/gender.csv
6	school/math.csv
7	school/oss.csv

Join Tree ID: 7. Rank: 1.03



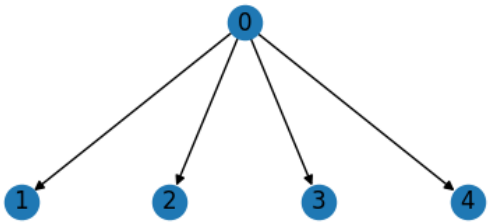
Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/Schools_Progress_Report_2012-2013.csv
3	school/disc.csv
4	school/esl.csv
5	school/gender.csv
6	school/math.csv

Join Tree ID: 4. Rank: 1.03



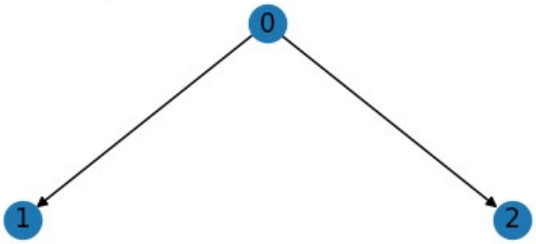
Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/Schools_Progress_Report_2012-2013.csv
3	school/disc.csv

Join Tree ID: 5. Rank: 1.03

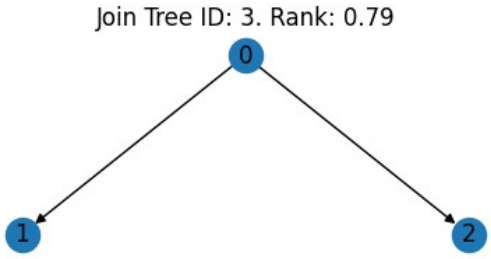


Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/Schools_Progress_Report_2012-2013.csv
3	school/disc.csv
4	school/esl.csv

Join Tree ID: 2. Rank: 0.94



Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/2013_NYC_School_Survey.csv



Node ID	Table Name
0	school/base.csv
1	school/2010_Gen_Ed_Survey_Data.csv
2	school/Schools_Progress_Report_2012-2013.csv

Furthermore the tool can explain a certain join tree for you.

```
autoTDA.explain_tree(tree_id=5)

The join tree starts at the table school/base.csv.

The join tree has the following joins:

      from (table.column) school/base.csv.DBN to (table.column)
school/2010_Gen_Ed_Survey_Data.csv.DBN with Non-Null ratio 0.87.

      from (table.column) school/base.csv.DBN to (table.column)
school/Schools_Progress_Report_2012-2013.csv.DBN with Non-Null ratio
0.96.

      from (table.column) school/base.csv.DBN to (table.column)
school/disc.csv.DBN with Non-Null ratio 0.86.

      from (table.column) school/base.csv.DBN to (table.column)
school/esl.csv.DBN with Non-Null ratio 0.63.

The rank of the tree is 1.03.
```

If you want to see the features of a join tree, you can call show_features() and decide if you also want to see the show the discarded features.

```
autoTDA.show_features(tree_id=5, show_discarded_features=False)


     Selected features:


+-------------------------------------------------------------------------
----------------------------------------+------------------
+-------------+--------------+

| Key
|   Non-Null Ratio |   Relevance |   Redundancy |

+=========================================================================
```

```
=======================================+=================+==========
==+==============+

| school/2010_Gen_Ed_Survey_Data.csv.studentsurvey
|         0.873168 |   0.270114  |     1         |

+-----------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/Schools_Progress_Report_2012-2013.csv.2012-2013 ENVIRONMENT
GRADE                                    |         0.959977 |
0.446768  |     1         |

+-----------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/Schools_Progress_Report_2012-2013.csv.2011-12 PROGRESS REPORT
GRADE                                   |         0.959977 |   0.221042
|     0.614142 |

+-----------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/Schools_Progress_Report_2012-2013.csv.2012-2013 OVERALL GRADE
|         0.959977 |   0.206767  |     0.592583 |

+-----------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/Schools_Progress_Report_2012-2013.csv.2010-11 PROGRESS REPORT
GRADE                                   |         0.959977 |   0.202085
|     0.517268 |

+-----------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/Schools_Progress_Report_2012-2013.csv.2012-2013 COLLEGE AND
CAREER READINESS GRADE                  |         0.959977 |
0.200034  |     0.270331 |

+-----------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/Schools_Progress_Report_2012-2013.csv.2009-10 PROGRESS REPORT
```

```
GRADE                                    |         0.959977 |   0.197591
|     0.503428 |

+-------------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/Schools_Progress_Report_2012-2013.csv.2012-2013PERFORMANCE
GRADE                                      |         0.959977 |
0.18178   |     0.56969  |

+-------------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/Schools_Progress_Report_2012-2013.csv.2012-2013 PROGRESS
GRADE                                       |         0.959977 |
0.102839  |     0.519741 |

+-------------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/disc.csv.B40 Intimidating and Bullying Behavior S
|         0.863021 |   0.0228692 |     0.99721  |

+-------------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/disc.csv.B22 Trespassing R
|         0.863021 |   0.0228692 |     0.996215 |

+-------------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/disc.csv.A34 Coercion/Threats P
|         0.863021 |   0.0200047 |     0.994153 |

+-------------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+

| school/disc.csv.B23 Using Slurs (Bias) R
|         0.863021 |   0.0200047 |     0.993962 |

+-------------------------------------------------------------------------
---------------------------------------+------------------
+-------------+--------------+
```

```
| school/disc.csv.B49 Using Controlled Substances w/o Authorization,
Illegal Drugs or Alcohol S              |         0.863021 |
0.0200047 |     0.992854 |

+-------------------------------------------------------------------
-------------------------------------+------------------
+-------------+--------------+

| school/disc.csv.A57 Using Weapon (Category II) to Attempt Injury
upon School Personnel, Students, Others R |         0.863021 |
0.0195479 |     1         |

+-------------------------------------------------------------------
-------------------------------------+------------------
+-------------+--------------+

| school/disc.csv.B53 Using Extreme Force Against/Inflicting
to/Inflicting Serious Injury to Students P      |         0.863021 |
0.0195479 |     1         |

+-------------------------------------------------------------------
-------------------------------------+------------------
+-------------+--------------+

| school/disc.csv.A27 Tampering/Altering Records or Documents R
|         0.863021 |   0.0195479 |     1         |

+-------------------------------------------------------------------
-------------------------------------+------------------
+-------------+--------------+

| school/disc.csv.A23 Using Slurs (Bias) R
|         0.863021 |   0.0195479 |     1         |

+-------------------------------------------------------------------
-------------------------------------+------------------
+-------------+--------------+

| school/esl.csv.Category
|         0.629651 |   0.0480833 |     1         |

+-------------------------------------------------------------------
-------------------------------------+------------------
+-------------+--------------+
```

Control

If you want to adjust a join tree, you can remove a join. Adding is not possible since this has to include calculations.

```
autoTDA.remove_join_path_from_tree(tree_id=5, table="school/disc.csv")
```

```
Path removed
```

If you still want to select a feature that the tool discarded, you can call move_feature_to_selected()

```
autoTDA.move_features_to_selected(tree_id=5,
features=["school/esl.csv.Grade"])
```

Stage 3: Evaluating the Join Trees

Now that we have join_trees, we want to see how well they perform in the ML Model. We do this by calling the evalaute_trees() function

```
autoTDA.evaluate_trees(algorithm='GBM', top_k_trees= 3, verbose=True)
```

```
Evaluating join trees...
```

```
100%|██████████| 3/3 [00:08<00:00,  2.87s/it]
```

Insights

To see how the join trees performed, you can call the get_all_results() function. This returns tuples of the tree id and the accuracy of each tree that is evaluated.

```
results = autoTDA.get_all_results()
print(results)
```

```
[(1, 0.6845070422535211), (9, 0.8056338028169014), (10,
0.8140845070422535)]
```

If you only need the best result, you can call get_best_result(). This returns the best result. If you want to see the details, like the joins, you can print it.

```
best_result = autoTDA.get_best_result()
print(best_result)
```

```
Result with model LightGBM

       rank:1.09

       with tree:

Rank: 1.09

From base table: school/base.csv
```

```
Join Trees:

+---------------------
+-------------------------------------------------+------------------
+

| From Table.Column   | To Table.Column
|   Non-Null Ratio |

+====================+================================================
===+==================+

| school/base.csv.DBN | school/2010_Gen_Ed_Survey_Data.csv.DBN
|         0.873168 |

+---------------------
+-------------------------------------------------+------------------
+

| school/base.csv.DBN | school/Schools_Progress_Report_2012-
2013.csv.DBN |         0.959977 |

+---------------------
+-------------------------------------------------+------------------
+

| school/base.csv.DBN | school/disc.csv.DBN
|         0.863021 |

+---------------------
+-------------------------------------------------+------------------
+

| school/base.csv.DBN | school/esl.csv.DBN
|         0.629651 |

+---------------------
+-------------------------------------------------+------------------
+

| school/base.csv.DBN | school/gender.csv.DBN
|         0.617249 |

+---------------------
+-------------------------------------------------+------------------
+

| school/base.csv.DBN | school/math.csv.DBN
|         0.629651 |
```

```
+---------------------
+-------------------------------------------------+------------------
+

| school/base.csv.DBN | school/oss.csv.DBN
|         0.880496 |

+---------------------
+-------------------------------------------------+------------------
+

| school/base.csv.DBN | school/pe.csv.DBN
|         0.850056 |

+---------------------
+-------------------------------------------------+------------------
+

| school/base.csv.DBN | school/qr.csv.DBN
|         0.844419 |

+---------------------
+-------------------------------------------------+------------------
+

       Accuracy:0.81

       Feature importance:

            school/Schools_Progress_Report_2012-2013.csv.2012-2013
ENVIRONMENT GRADE : 0.20563380281690144

            school/base.csv.Total Parent Response Rate (%) :
0.034929577464788794

            school/2010_Gen_Ed_Survey_Data.csv.studentsurvey :
0.034929577464788794

            school/base.csv.School Type : 0.02704225352112679

            school/base.csv.Total Student Response Rate (%) :
0.026478873239436672

            school/Schools_Progress_Report_2012-2013.csv.2010-11
PROGRESS REPORT GRADE : 0.014084507042253547

            school/qr.csv.School Year : 0.010704225352112661

            school/pe.csv.Borough : 0.006760563380281681

            school/Schools_Progress_Report_2012-2013.csv.2011-12
```

```
PROGRESS REPORT GRADE : 0.006197183098591541

           school/esl.csv.Category : 0.005633802816901401

           school/Schools_Progress_Report_2012-2013.csv.2012-2013
PROGRESS GRADE : 0.004507042253521121

           school/Schools_Progress_Report_2012-2013.csv.2012-
2013PERFORMANCE GRADE : 0.004507042253521121

           school/qr.csv.Overall Rating : 0.0039436619718309805

           school/Schools_Progress_Report_2012-2013.csv.2012-2013
COLLEGE AND CAREER READINESS GRADE : 0.0033802816901408405

           school/Schools_Progress_Report_2012-2013.csv.2012-2013
OVERALL GRADE : 0.0028169014084507005

           school/disc.csv.A23 Using Slurs (Bias) R :
0.0028169014084507005

           school/gender.csv.Category : 0.0016901408450704202

           school/disc.csv.B40 Intimidating and Bullying Behavior S :
0.0011267605633802802

           school/oss.csv.AMERICAN INDIAN/ALASKAN NATIVE STUDENTS With
2 or More Suspensions or Removals : 0.0005633802816901401

           school/Schools_Progress_Report_2012-2013.csv.2009-10
PROGRESS REPORT GRADE : 0.0005633802816901401

           school/disc.csv.B49 Using Controlled Substances w/o
Authorization, Illegal Drugs or Alcohol S : 0.0005633802816901401

           school/disc.csv.A27 Tampering/Altering Records or Documents
R : 0.0

           school/disc.csv.A57 Using Weapon (Category II) to Attempt
Injury upon School Personnel, Students, Others R : 0.0

           school/disc.csv.B53 Using Extreme Force Against/Inflicting
to/Inflicting Serious Injury to Students P : 0.0

           school/base.csv.School Name : 0.0

           school/disc.csv.B22 Trespassing R : 0.0

           school/disc.csv.A34 Coercion/Threats P : 0.0

           school/disc.csv.B23 Using Slurs (Bias) R : 0.0
```

```
           school/oss.csv.UNKNOWN STUDENTS With 2 or More Suspensions
or Removals : 0.0

           school/oss.csv.MULTI-RACIAL STUDENTS With 2 or More
Suspensions or Removals : 0.0

           school/math.csv.Category : 0.0
```

If you are happy with the final result, you can export it to a dataframe.

```
dataframe = autoTDA.materialise_join_tree(1)
dataframe.head()
```

```
  school/base.csv.DBN     school/base.csv.School Name  \
0              01M015       P.S. 015 Roberto Clemente
1              01M019             P.S. 019 Asher Levy
2              01M020            P.S. 020 Anna Silver
3              01M034  P.S. 034 Franklin D. Roosevelt
4              01M063       The STAR Academy – P.S.63

  school/base.csv.School Type  school/base.csv.Total Parent Response
Rate (%)  \
0                          ES
66
1                          ES
96
2                          ES
71
3                       ES/MS
40
4                          ES
70

   school/base.csv.Total Teacher Response Rate (%)  \
0                                              100
1                                               97
2                                               79
3                                               88
4                                              100

   school/base.csv.Total Student Response Rate (%)  \
0                                              NaN
1                                              NaN
2                                              NaN
3                                             94.0
4                                              NaN

  school/2010_Gen_Ed_Survey_Data.csv.studentsurvey
0                                               No
```

```
1                                                   No
2                                                   No
3                                                  Yes
4                                                   No
```