**Scalable Data Strategy for Enterprise-Grade Data Preparation Pipeline**

A Scalability Optimization Study

Sadig Akhund

The ADA University, Azerbaijan

The George Washington University, USA

sadigaxund@hotmail.com

Although there is no distinctive header, this is the abstract. This submission template allows authors to submit their papers for review to an ACM Conference or Journal without any output design specifications incorporated at this point in the process. The ACM manuscript template is a single column document that allows authors to type their content into the pre-existing set of paragraph formatting styles applied to the sample placeholder text here. Throughout the document you will find further instructions on how to format your text. If your conference’s review process will be double-blind: The submitted document should not include author information and should not include acknowledgments, citations or discussion of related work that would make the authorship apparent. Submissions containing author identifying information may be subject to rejection without review. Upon acceptance, the author and affiliation information must be added to your paper.

CCS CONCEPTS • Insert your first CCS term here • Insert your second CCS term here • Insert your third CCS term here

**Additional Keywords and Phrases:** Insert comma delimited author-supplied keyword list, Keyword number 2, Keyword number 3, Keyword number 4

ACM Reference Format:

First Author’s Name, Initials, and Last Name, Second Author’s Name, Initials, and Last Name, and Third Author’s Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock ’18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1. Introduction

a. Definition of the Problem

b. Objective of the Study

c. Significance of the Problem

d. Review of Significant Research

e. Assumptions and Limitations

1. Model Design
   1. Acquisition

When it comes to acquisition of data for ML models, it is important to consider not only the quality of the data but also its representativeness. In other words, the data should accurately reflect the characteristics of the population or system being studied, in order to produce valid and useful insights. One challenge in data acquisition is dealing with data that is subject to noise or bias, which can lead to inaccurate or misleading results. To address this challenge, data cleaning, and preprocessing techniques can be applied to adjust the data and remove any errors or outliers. These techniques may involve the use of statistical algorithms or ML models, which can be run on distributed computing frameworks like Apache Hadoop or Apache Spark to handle large datasets. In addition to data cleaning and preprocessing, organizations can improve the scalability of their data acquisition pipeline by carefully selecting and evaluating data sources. This involves assessing the reliability, scalability, and data quality of potential sources, and using optimized tools and techniques for efficient and scalable data extraction, ingestion, and storage. Therefore, the acquisition stage for most data pipelines can be broken down into the following steps: *defining data requirements*, *selecting data sources*, *data extraction*, *data ingestion*, *data storage*, and *versioning*. Cloud-based platforms like Amazon Web Services (AWS) and Google Cloud Platform (GCP) provide scalable storage and computing resources that can help address these challenges. Additionally, proper labeling and annotation of data is another important consideration in data acquisition, as it can significantly impacts the model performance. Organizations may use crowdsourcing, collaborative approaches, or automated labeling and annotation tools to improve the quality of labels and annotations. Finally, ethical considerations are important in data acquisition and use, particularly concerning data privacy, security, and ownership. Organizations can implement data governance frameworks, policies, and procedures for data collection, storage, processing, and sharing, as well as technical solutions like encryption and access controls to protect data from unauthorized access or disclosure. Overall, addressing scalability challenges in the data acquisition stage is critical to ensuring that models produce valid, reliable, and useful insights. Organizations can develop an efficient and effective data acquisition pipeline by carefully evaluating data sources, applying data cleaning and preprocessing techniques, and considering ethical considerations.

* + 1. Defining Data Requirements

In today's world, managing big and varied data sets can be a daunting task for organizations. It's not just about getting hold of the data, but also maintaining its quality, consistency, and accessibility across the organization. Defining data requirements is a critical step in the data acquisition stage, as it serves as the foundation for the entire pipeline. It entails specification of what data is needed, how it should be collected, processed, and stored, and the desired outcomes. The first step in defining data requirements is to identify the stakeholders who will use the data and what their data needs are. This can help to identify any potential conflicts or gaps in data requirements and ensure that all stakeholders are in agreement on what data is needed and why. It is also important to keep in mind that different stakeholders may have different data requirements based on their roles, responsibilities, and objectives. To tackle this issue, new approaches such as **data governance**, **data lineage**, and **data mesh** have been developed. These approaches have become popular because they help organizations define their data requirements, improve scalability, and guarantee the reliability and trustworthiness of the data. Therefore, in this discussion, we will delve into these concepts in detail and highlight their importance in enhancing the defining data requirements step of data acquisition. By the end of this discussion, you will have a clear understanding of how data governance, data lineage, and data mesh can help organizations acquire and manage their data more efficiently.

* + - 1. Data Mesh

One relatively new concept called data mesh comes in handy when considering decentralization, autonomy, and domain-driven design. As defined by [1], The concept of Data Mesh, which was introduced by Zhamak Dehghani, proposes a distributed and decentralized approach to managing enterprise data. This approach considers each dataset as a distributed product, which is focused on specific domains. The aim is to establish a sense of data ownership and responsibility by assigning embedded engineers and product owners to manage the data and its accessibility to other teams. This differs from current data platforms, which are often centralized, monolithic, and rely on complex pipelines that lack ownership and responsibility at the data level. In essence, Data Mesh is a holistic approach that prioritizes domain-specific data management to enhance data ownership, responsibility, and accessibility across the organization. Figure 1 provides a visual representation of how data mesh architecture is different from traditional, centralized architecture.

Diagram

Description automatically generated

Figure 1: Decentralized vs Mech Architecture by C. Grande [9].

In the context of building a robust pipeline, it is evident that the collected and stored data meets the needs of stakeholders and by implementing this approach, organizations can empower their data product teams to make decisions on how data is used, collected, and stored within their domains, therefore improving the quality of data and enabling more efficient data sharing and collaboration among teams. At its core, data mesh involves creating small, self-organizing teams, known as data product teams, that are responsible for specific data domains. These teams are empowered to make decisions about how data is collected, stored, and used within their domain, based on the needs of their stakeholders. Data mesh also involves a number of technical practices, such as data ownership, data as a product, and federated data governance, that are designed to support the autonomy and flexibility of the data product teams. From a technical standpoint, data mesh is a technical approach that promotes autonomy and flexibility in data product teams. The approach involves practices like domain-driven design, APIs, and event-driven architectures to facilitate data sharing and collaboration among teams. Additionally, data observability plays an essential role in creating a shared understanding of data quality, lineage, and usage across the organization. Moreover, challenges in implementing a data mesh approach include ensuring team alignment with organizational goals and having technical infrastructure that supports autonomy and flexibility while also maintaining data security and compliance. To avoid scalability issues, organizations need to create well-defined and well-bounded data domains with clear ownership boundaries, data quality requirements, and effective communication channels. Furthermore, to optimize or automate this step, organizations can leverage automated data profiling and cataloging tools to streamline the process of data discovery and documentation. They can also use machine learning and other AI technologies to identify patterns and insights within the data, which can inform the development of more effective data requirements. Collaborative platforms and tools like data wikis and governance dashboards can facilitate communication and collaboration between data product teams.

According to ‘Data Mesh’ book by O’Reilly [2], data mesh is based on *four* basic principles that form the foundation of its logical structure and operational model. These principles are intended to help organizations achieve the goals of data mesh, which include enhancing the value derived from data at scale, maintaining agility as the organization expands, and adapting to change in a constantly evolving and unpredictable business environment. Those principles are:

1. **Principle of Domain Ownership –** Idea of distributing ownership of analytical data to the business areas that are closest to the data, whether they are the data source or the primary users [2].
2. **Principle of Data as a Product –** By following this principle, domain-based data is shared directly with data users such as data analysts and scientists, as a product with a set of usability characteristics, including being discoverable, understandable, trustworthy, and natively accessible [2, 3]. This data product is defined by a set of data sharing contracts, is autonomous with its own lifecycle and model, and is encapsulated in a data quantum that includes all necessary structural components like metadata, code, policy, and infrastructure dependencies [2].
3. **Principle of the Self-Serve Data Platform –** The Data Mesh addresses the limitation of the organization's ability to grow and expand its operations in utilizing data effectively, and as such, the platform must facilitate data sharing both within and outside the organization while also ensuring secure interoperability across multiple platforms. Without such interoperability, a single monolithic solution will limit scalability. Since data has limited value if it is not shared, the Data Mesh platform must be designed accordingly [3].
4. **Principle of Federated Computational Governance –** As defined by Tartow, C., & Mott, A in their article at starburst, data mesh emphasizes a move away from centralized data teams and architectures towards a decentralized model, which raises questions about governance functionality that remains under central IT organization and the responsibilities of domains. Although Data Mesh entails a shift from centralized to decentralized architecture, there are certain aspects of data governance that need to remain under a central group to maintain balance and ensure comprehensive governance [1]. The principle involves establishing a data governance operating model that comprises a team consisting of domain representatives, data platform experts, and subject matter experts like legal, compliance, and security personnel. This model allows for decentralized decision-making and accountability, while promoting autonomy and agility within domains, as well as global interoperability of the Data Mesh. This incentivizes effective management of data governance, while balancing the autonomy of different domains with the need for effective collaboration across the organization [2].

In conclusion, data meshes have emerged as a powerful solution for managing data at scale. They provide a way to handle the complexity and heterogeneity of data in modern distributed systems by enabling a more decentralized approach to data management. Data meshes have a significant impact on scalability by enabling teams to manage data more efficiently and reducing the costs associated with centralization. Furthermore, data meshes indirectly improve machine learning performance by facilitating access to more diverse and high-quality data, which is essential for training more accurate and robust models. With the continued growth in data volume and complexity, data meshes are likely to play an increasingly important role in data management and machine learning in the future.

* + - 1. Data Governance

Data governance in the data acquisition stage involves setting up policies, procedures, and standards for how data is collected, managed, and utilized within an organization. These policies aim to ensure that the data is accurate, complete, reliable, and compliant with relevant regulations and privacy laws. Organizations must define data quality standards, data storage and retrieval processes, and ensure ethical data collection with appropriate consent. Roles and responsibilities for data ownership, stewardship, and access must also be established. Looking at the technical dimension, it requires implementing various mechanisms to ensure data quality, such as data validation rules, automated data cleansing processes, and data profiling tools. The quality and integrity of the data used to train machine learning models have a significant impact on the accuracy and effectiveness of the models.

Castra et al. present a comprehensive analysis of the role that data governance plays in simplifying the complexity of data management processes as it is responsible for controlling the decision-making and accountability for all aspects of data management [5]. However, this step presents its own set of challenges, including managing large and complex datasets, maintaining data accuracy and completeness, ensuring data privacy and security, and handling data from diverse sources and formats. Castro and et. al. in their writing proposed **ontology-based** models that will provide significant benefits for knowledge representation, particularly due to their formalization and extensive expressive capabilities, in various fields, including data governance [5]. Ontology-based reasoning is an approach to managing data that utilizes ontologies to represent knowledge and perform automated reasoning on data [5,6,7]. This method is implemented in data governance to simplify the complexity of managing large amounts of data by controlling decision-making and responsibilities related to data management [5,6]. The proposed data governance system in the paper is based on such reasoning by using semantic techniques and a shared knowledge plane to represent all data management processes. The system was tested in Telefonica's global video service, showing its feasibility in reducing the complexity of managing big data environments [5].

To summarize the main findings of this study, ontologies enable the representation of behavior through the definition of rules, restrictions, and policies. As a result, they provide a powerful means of structuring and organizing data, enhancing its usability, and facilitating decision-making processes. Furthermore, in addition to the aforementioned points, it should also be noted that, ontologies can be used to establish data governance policies that can guide data management activities, ensuring that data is managed in a consistent and structured manner. Overall, ontologies offer a robust and flexible framework for knowledge representation, which can be applied to various fields, including *data governance*, to improve data management practices and decision-making processes.

* + - 1. Data Lineage

Data lineage is a critical aspect of the data pipeline, as it tracks the origin, transformation, and movement of data throughout various stages of the process. Many people confuse data lineage with **data provenance**. For instance, Beneventano et al. defines both terms, rightfully so, as “…where data came from, how it was derived and how it was modified over time”. Although similar, data lineage and provenance are related concepts but have different meanings. Data lineage refers to the process of identifying the origins, transformations, and movements of data as it passes through various stages of a data pipeline. It involves understanding the source of the data, how it's collected, and how it's processed before being ingested into the pipeline. Data lineage is concerned with tracking the flow of data from its source to its destination and ensuring that the data is accurate, reliable, and aligned with the needs of stakeholders. Provenance, on the other hand, refers to the history of the data itself, including information about its origin, ownership, and any changes or modifications that have been made to it over time. Provenance is concerned with the metadata associated with the data and provides information about how the data was created, who created it, and how it has been used. Provenance is important for data governance, as it helps ensure that data is used appropriately and in compliance with organizational policies. In other words, data lineage is focused on the flow of data through a pipeline, while provenance is focused on the metadata associated with the data itself.

At the acquisition step, understanding the source of the data and how it is collected and processed before ingestion into the pipeline is vital. In order to realize this, it is imperative that metadata is captured, including *source*, *format*, and *transformation history*. Although crucial, data lineage presents a few challenges, including handling large volumes of data, data security and privacy, and maintaining lineage as the pipeline evolves. However, well-defined data domains, clear ownership boundaries, data quality requirements, and standard definitions can mitigate these challenges. The graphical illustration in Figure 2 showcases the intricate data flow within a real-life data warehouse system, specifically highlighting the movement of information from tables and views towards the end goal of report generation, providing valuable insight into the underlying mechanics and processes of the system.

Background pattern

Description automatically generated

Figure 2: Lineage of Data Visualized. The movement of information from tables and views are depicted in blue on the left and middle and towards the end goal of report generation are represented in red [10].

Devoting significant effort to this stage is of utmost importance when building a robust system as it ensures the *data's accuracy*, *reliability*, and *alignment with stakeholder needs*, leading to fewer errors, inconsistencies, and increased transparency and accountability. Organizations can automate the process by using data profiling and cataloging tools to streamline data discovery and documentation. Furthermore, machine learning and AI technologies can help identify patterns and insights to develop more effective data requirements. Finally, collaborative platforms and tools, such as data wikis and governance dashboards, can facilitate communication and collaboration between data product teams. The landscape of metadata management in ETL systems is constantly evolving, with a plethora of commercial tools available in the market. On one hand, ETL software products such as Informatica PowerCenter, IBM DataStage, and Microsoft SQL Server Integration Services include metadata repositories to store metadata generated and managed by the tool [10, 11]. Moreover, vendors now offer interfaces to import metadata from other tools, including databases and OLAP/reporting systems, into their repositories. Notably, the IBM Metadata Workbench, a component of the Information Server product family, permits lineage and impact analyses of transformations implemented in the IBM DataStage tool [10]. However, lineage and impact analyses remain challenging to implement in Microsoft SQL Server Data Transformation Services [10, 12]. The metadata model that was once used to maintain transformations based on the Open Information Model (OIM) has been abandoned, and SQL Server now uses a similar approach to store transformation metadata, which can be accessed via an object-oriented API. Nonetheless, graph analysis methods are not provided out-of-the-box, making lineage and impact analyses difficult to implement. On the other hand, specialized metadata management tools such as ASG Rochade and Adaptive Metadata Manager, are not tied to any specific ETL tool, and can import metadata using standard CWM or specialized tool-specific interfaces [10]. This approach offers more flexibility in specifying a customized metadata model that caters to a company's specific requirements. However, these tools often use proprietary storage and querying approaches: one such example presented by Tomingas is that ASG Rochade uses a proprietary file-based storage queried internally using the Rochade [10] Procedural Language (RPL), while Adaptive Metadata Manager uses an Oracle database queried using SQL. Nevertheless, these commercial metadata management tools offer advanced mechanisms for data lineage and impact analyses, highlighting the potential for metadata management in the ETL field. Despite the limitations of commercial metadata management tools, they continue to advance with sophisticated mechanisms for data lineage and impact analyses. As mentioned in the introduction, the storage and querying approaches of these tools are often proprietary, making cross-repository queries from different vendors unfeasible. However, these tools offer promising opportunities for companies to efficiently manage their metadata and enhance their ETL systems.

From a technical perspective, data lineage involves capturing metadata about the data, including its source, format, and transformation history. This metadata can be stored in a metadata repository or data catalog, which can be queried to retrieve information about the data lineage. It may also involve the use of data profiling tools to identify any quality issues with the data and data governance tools to ensure compliance with organizational policies. One such ecosystem of tools can be identified as the **managed metadata environment** (MME). The MME denotes the structural constituents and procedures that are necessary for the efficient and methodical accumulation, preservation, and distribution of metadata across the entire organization [13]. It encompasses the ideas of metadata repositories, catalogs, data dictionaries, and any other relevant term used for referring to the organized handling of metadata.

Diagram

Description automatically generated

Figure 3: Managed Metadata Environment [13].

In conclusion, defining data requirements and understanding the data lineage is a critical step in building a robust system because it ensures that the data being collected and used is accurate, reliable, and aligned with the needs of the stakeholders. Therefore, focusing on this step helps to improve the quality of the data pipeline, reduces errors and inconsistencies, and increases the transparency and accountability of the data pipeline. Although, some of the challenges of data lineage include dealing with complex data transformations, managing large volumes of data, and ensuring data security and privacy, it can also be difficult to maintain data lineage over time as the data pipeline evolves and new sources of data are added. Moreover, to avoid scalability problems, it's essential to create well-defined and well-bounded data domains that can be easily managed by the data product teams including creating clear ownership boundaries, defining data quality requirements, and establishing effective communication channels between the teams. It's also important to establish standard data definitions and ensure that the data is standardized and normalized across the organization. Essentially, if you were to optimize or automate this step a number of tools and technologies, such as data profiling and data cataloging tools, can help streamline the process of data discovery and documentation. Additionally, they can also use machine learning and other AI technologies to help identify patterns and insights within the data, which can inform the development of more effective data requirements. Finally, organizations can use collaborative platforms and tools, such as data wikis and data governance dashboards, to help facilitate communication and collaboration between the data product teams. In summary, the research presented here underscores the importance of addressing the obstacles that arise at this stage of the pipeline. To mitigate or deal with such challenges, organizations should consider implementing infrastructure such as cloud-based solutions or distributed databases, as well as utilizing data processing pipelines that can handle large volumes of data efficiently and leveraging automation tools to reduce the need for manual intervention, thereby increasing efficiency.

* + 1. Selecting Data Sources

Selecting data sources is an important part of preparing data for building a scalable data pipeline. It basically means figuring out which data to use in the model and where to find it. This can include data from different sources like databases, data warehouses, APIs, or third-party providers, both inside and outside the organization. The data sources should be selected based on how useful, good, and available they are, and how well they fit with the model's needs. Sometimes, the data may have to be changed or pre-processed before it can be used in the model. From a technical perspective, selecting data sources involves identifying what types of data the model needs, like *structured* or *unstructured* data, and getting that data from different sources like databases, web services, or files. After that, the data has to be transformed into a format that can be used by the model, which includes cleaning, filtering, and normalizing it. Finally, the data must be checked for quality to make sure it's accurate, complete, and consistent. Data versioning can be used to keep track of changes to the data over time, so the model always uses the most recent and relevant data.

With the increasing volume and complexity of data available today, it is more important than ever to carefully consider the selection of data sources to avoid biased or inaccurate results. It is a critical step in the pipeline's data preparation stage since it ensures the model is built on accurate, high-quality, and relevant data. However, not carefully considering and selecting data sources can lead to costly mistakes and poor decision-making due to biased or inaccurate results. To avoid such situations, various techniques and methods can be employed, including **data exploration** and **profiling**, which involves performing exploratory data analysis to understand the data's characteristics, detect potential issues, and filter out irrelevant or low-quality data sources.

* + - 1. Data Exploration and Profiling

Data profiling and data exploration are two important techniques used in the data preparation stage of the data pipeline. Both techniques involve analyzing and understanding the data before it is used in the model, but they differ in their focus and purpose. On one hand, data profiling is a process that involves analyzing the data to identify its *characteristics*, *structure*, and *quality*. It aims to create a summary of the data, such as its size, distribution, completeness, and consistency. Data profiling helps in identifying potential data quality issues and provides insight into the data's suitability for the model. On the other hand, data exploration involves digging deeper into the data to understand its *patterns*, *relationships*, and *correlations*. It aims to gain insights into the data and to identify any trends or patterns that can be used in the model. Data exploration may involve visualizing the data through charts, graphs, and histograms to identify patterns and relationships.

Having a comprehensive understanding of data is crucial before using or processing it and data profiling is a technique that enables users to find metadata related to a dataset, providing basic information to aid in understanding it [14, 15]. It is a significant area of research for IT experts and scholars with many use cases, such as *data integration*, *data quality*, *data cleansing*, *big data analysis*, *database management*, and *query optimization* [14]. According to Liu et al. [14] some papers primarily focus on data profiling for relational data. Not only that, but also that other types of databases such as *time-series data*, *graph data*, and *heterogeneous data* also require data profiling. Moreover, as stated by Liu et al., data profiling tasks are classified into single column data profiling, multiple columns data profiling, and dependencies. However, Naumann [16], on the other hand, classifies data profiling from single to multiple data sources. See Figure 4:

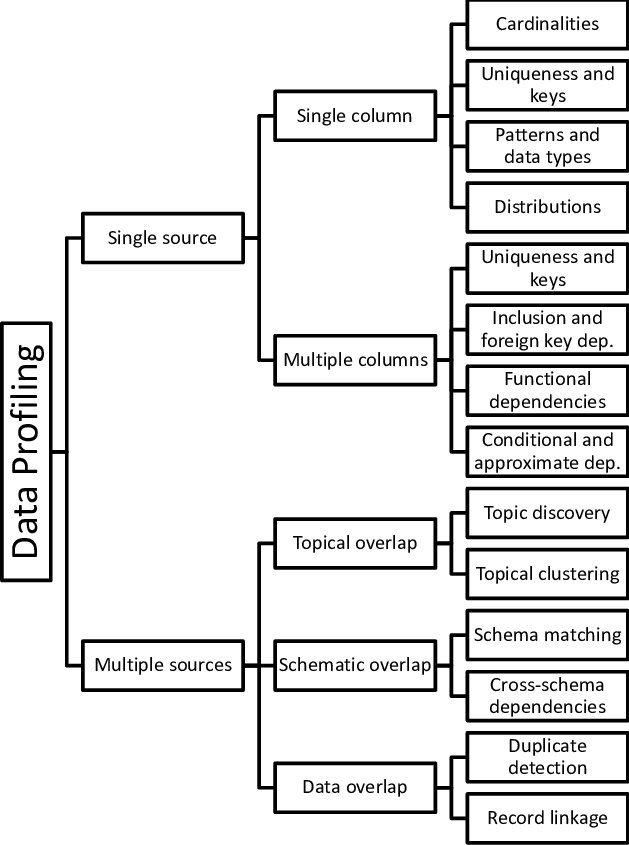


Figure 4: Data profiling tasks classified [16].

According to Figure 1, data profiling tasks are classified into two categories: "single sources" and "multiple sources" [16]. As indicated by Naumann, the tasks for "single sources" are well-established in tooling and research, while the tasks for "multiple sources" represent new research directions for data profiling. In order to evaluate different methods and architectures for data profiling, a data profiling benchmark is necessary to define a set of tasks, data to be used, and efficiency measures. Creating realistic data is the most difficult part, and measures should go beyond traditional response time and cost efficiency evaluations to include approximations. The benchmark should evaluate both entire profiling systems and individual tasks. The computational complexity of data profiling presents three main challenges [14] – [16]:

1. **Managing Input**: The results are computationally complex to discover and often require sorting steps.
2. **Performing Computation**: The verification of complex constraints on all columns and combinations of columns in a database is necessary, which is often an exponential process.
3. **Managing Output**: The datasets may be too large to fit into main memory.

It is difficult to handle a large amount of data in traditional data mining algorithms, machine learning algorithms, and data profiling tasks, as it demands significant hardware resources and is time-consuming. To tackle these challenges, various tools and algorithms have been developed, such as relying on the capabilities of a DBMS, using innovative ways to handle the challenges, delivering only approximate results, and narrowing down the discovery process to certain columns or tables. The article by Liu [14] discusses two main strategies for data mining and analysis: **sampling** and **distributed systems**. In this paragraph, sampling will be the main focus. Sampling methods are an effective approach to reduce data volume and expedite data processing. In other words, it is a way to select a representative sample of data from a larger set to reduce the amount of data that needs to be processed. Sampling is particularly useful when computing power is limited, and approximate results are acceptable. Researchers must choose appropriate sampling methods and consider the biases that can result from sampling. Sampling involves probability and non-probability sampling, with probability sampling having every unit in a finite population having a certain probability of being selected. Moreover, it is used in various data applications, such as data profiling, data analysis, data mining, data visualization, and machine learning. There are different sampling techniques, including *simple random sampling*, *stratified sampling*, *systematic sampling*, *cluster sampling*, *oversampling*, and *under sampling*, and *reservoir sampling*. In big data contexts, sampling can be performed with the help of big data computing frameworks such as MapReduce. Determining an effective sample size is crucial, since a sample size that is too small may result in incorrect conclusions, while a sample size that is too large may take too long to compute. Traditional methods for determining sample size have been summarized, and the most appropriate number of samples is when the accuracy rate reaches the maximum value. Additionally, **sampling error** occurs when a randomly chosen sample does not reflect the underlying population purely by chance, while **sampling bias** occurs when the sample is not randomly chosen at all. Sampling bias is caused by the failure of the sampling design to truly extract the sample randomly from the population.

* + - 1. Single-Column Data Profiling

Single column data profiling is the process of analyzing and understanding the characteristics and properties of individual columns or fields within a dataset. This type of data profiling focuses on a single attribute of the data and aims to identify patterns, anomalies, and statistics related to that attribute. Single column data profiling can involve various techniques, such as summary statistics, histograms, frequency distributions, and data quality checks. The goal is to gain insights into the data and identify any potential issues or discrepancies that may affect the quality or reliability of the data. For example, in a dataset containing customer information, single column data profiling may involve analyzing the "age" column to understand the age distribution of the customers, identify any outliers or missing values, and determine whether the data is consistent and accurate. This type of analysis can help data engineers and data scientists make informed decisions about the suitability of the data for their purposes and inform any necessary data cleaning or preprocessing steps. According to Liu et al. [14], single-column data profiling tasks are classified into different categories such as *cardinalities*, *value distributions*, *patterns*, *data types*, and *domains*. Typical metadata resulting from these tasks is listed in Table 1. Moreover, for some tasks such as calculating the maximum number of decimals in numeric values, simple sampling methods may not be reliable. Additionally, identifying the domain of one column is often challenging and cannot be fully automated [14]. Among the different categories, cardinality, histograms, and quantiles are frequently used for query optimizers, and thus, sampling techniques are commonly used in these tasks.

Table 1: Single-Column Profiling Tasks. Source: [14]

| Category | Task | Definition |
| --- | --- | --- |
| Cardinalities | num-rows | Number of rows |
|  | value length | Measurements of value lengths (minimum, maximum, median, and average) |
|  | null values | Number or percentage of null values |
|  | distinct | Number of distinct values; sometimes called "cardinality" |
| Value Distributions | uniqueness  histogram | Number of distinct values divided by the number of rows  Frequency histograms (equi-width, equi-depth, etc.) |
|  | constancy | Frequency of most frequent value divided by number of rows |
|  | first digit | Distribution of first digit in numeric values |
| Patterns, Data Types | quartiles  basic type | Three points that divide the (numeric) values into four equal groups  Generic data type, such as numeric, alphabetic, alphanumeric, date, time |
| and Domains | data type | Concrete DBMS-specific data type, such as varchar, timestamp. |
|  | size | Maximum number of digits in numeric values |
|  | patterns | Histogram of value patterns |
|  | domain | Classification of semantic domain, such as credit card, first name, city, phenotype |
|  | decimals | Maximum number of decimals in numeric values |
|  | data class | Semantic, generic data type, such as code, indicator, text, date/time, quantity, identifier |
|  |  |  |

a Typical metadata resulting from tasks.

In summary, selecting the right data sources can minimize the risk of overfitting or underfitting the model, increasing its ability to generalize to new data. Prioritizing the selection of data sources can also help build a scalable and flexible model that can handle changing data sources or requirements. This can future-proof the system, making it easier to maintain and update. Furthermore, data profiling focuses on creating a summary of the data, while data exploration focuses on discovering insights and patterns in the data. Both techniques are important in the data preparation stage and can help in selecting appropriate data sources, identifying potential data quality issues, and gaining insights into the data that can be used in the model. Moreover, big data is susceptible to selection bias, and many scholars’ study how to solve selection bias in the sampling process. Additionally, there are several challenges associated with selecting data sources for machine learning models. Some of these challenges include identifying the most relevant and useful data sources, ensuring the quality and accuracy of the data, managing data from multiple sources, and dealing with data that may be incomplete or inconsistent. Also, selecting data sources can be time-consuming and may require specialized knowledge or expertise to identify and extract the necessary data. The challenges of selecting data sources can vary depending on the specific project or use case, but some common challenges include:

1. **Finding relevant data sources**: It can be difficult to locate and identify data sources that are relevant to the problem or use case at hand. This may involve searching for data across multiple systems or platforms or working with third-party providers to obtain the necessary data.
2. **Ensuring data quality**: The quality of data can have a significant impact on the performance of the machine learning model. It is important to ensure that the data is accurate, complete, and consistent, and that any missing or erroneous data is appropriately handled.
3. **Managing data compatibility**: Different data sources may have varying formats or structures, which can make it difficult to integrate them into a single model. It is important to consider how the data will be transformed, cleaned, and processed to ensure compatibility with the model.
4. **Addressing data privacy concerns**: Some data sources may contain sensitive or personally identifiable information, which can raise privacy concerns. It is important to ensure that the data is collected, stored, and processed in compliance with applicable privacy regulations.

To avoid problems with scalability when selecting data sources, there are several key points that should be considered. First, it is important to identify data sources that are easily scalable and can handle large volumes of data. This may involve selecting cloud-based data sources or using distributed storage systems like Hadoop or Spark. Second, data should be organized and stored in a way that allows for efficient processing and retrieval. This may involve using data partitioning or indexing techniques to optimize data access. Finally, it is important to consider the computational resources required to process the data and ensure that there is enough capacity to handle growing data volumes over time. This may involve using parallel processing techniques or scaling up the computational resources as needed. In conclusion, selecting data sources is crucial to building a robust machine learning system that produces accurate and reliable results, minimizing errors or biases and future-proofing the system for scalability and flexibility. By carefully considering and selecting data sources, we can improve data quality and relevance, reduce risk, and build a system that can adapt to changing data sources and requirements.

* + 1. Data Extraction

Data extraction is a crucial step in the pipeline's data preparation stage, involving the collection and extraction of raw data from various sources. To extract data, relevant sources are identified, and a connection to the source is established. To establish a connection to the data source, appropriate methods such as ODBC, JDBC, or REST API are used. Depending on the data source, the extraction process may involve querying a database, parsing a file, or sending a request to an API endpoint. For file-based data sources, extraction may involve parsing text files or using specialized libraries for specific file formats. Once the data is extracted, it can be transformed and pre-processed to meet the data requirements of the model. The data extraction process involves several substeps, which are unique to this stage. The first substep involves defining the data sources and the specific data that needs to be extracted. The appropriate method for extracting the data is then determined, and the structure and format of the extracted data is defined. Finally, the data is extracted using the chosen method. Next sub-chapters discuss the aforementioned steps in more detail.

* + - 1. Determining the Data Sources

According to Talend [24], the term "data source" can refer to the original location where data is created or where physical information is first converted into a digital format. However, even highly refined data can serve as a source as long as it is accessed and used by another process. Examples of data sources include *databases*, *flat files*, *live measurements* from physical devices (IOT devices), *scraped web data* (web crawlers), and *various static and streaming data services* (APIs) available on the internet. That said, databases are the most commonly used data sources and are found in relational database management systems (RDBMS). In this context, the Data Source Name (DSN) is an essential concept [24]. A DSN is a pointer within a destination database or application that indicates the actual data location, whether it is local or remote, and whether it is in a physical or virtual location. The DSN is not necessarily the same as the relevant database or file name, but it is a label or address used to access the data. Furthermore, the interpretation of the term "data source" is differentiated by Talend depending on the systems used to ingest data as following examples [24]. For instance, in the Java software platform, a "Datasource" is a specific object that represents a connection to a database, while in newer platforms, "DataSource" refers to any collection of data that provides a standardized way of access. It is essential to understand the context of the systems used to interpret the meaning of data source terminology. Furthermore, data sources play a significant role in the transportation and management of data. Different network protocols, such as FTP, HTTP, and APIs, are used to move data from various services and applications [25]. For instance, Adobe Analytics uses a file data source uploaded through FTP to process data automatically. When it comes to security, SFTP and FTPS with added TLS encryption are employed to protect sensitive information [24]. In addition to network protocols, APIs also provide more customized access methods for managing data sources. Spark, for example, offers an API with different implementations for generic relational sources and detailed JDBC connections. Other protocols, such as NFS, SMB, SOAP, REST, and WebDAV, are also used to transfer data within APIs or as standalone processes [24]. The primary objective of data sources is to make it easier for users and applications to connect and transport data to its intended destination. They collate essential technical information and present it in a user-friendly manner, saving stakeholders from dealing with low-level connection details. Moreover, data sources provide consistent storage of connection information, which makes processes such as system migrations or structural changes much smoother. Overall, data sources are critical for seamlessly integrating disparate systems and ensuring that data consumers can focus on processing and utilizing data to its fullest potential.

When determining data sources, it is also important to consider **data ownership** because it affects the accessibility and availability of the data. The owner of the data may have specific requirements for how the data should be handled, including how it should be extracted, where it can be stored, and who can access it. Therefore, understanding data ownership is an important aspect of data extraction, as it helps ensure that data is being extracted in a way that aligns with the owner's policies and requirements. In the world of data management, it is important to determine who is responsible for keeping data accurate and up-to-date [22]. This is what is meant by data ownership. During the data design phase, it is crucial to identify who has the ability to write data to the directory. There are various approaches to determining data ownership, such as allowing only a small group of directory content managers to have read-write access to the directory while the rest of the users have read-only access. Alternatively, individual users can be given the authority to manage certain subsets of information, such as their passwords or personal information. A person's manager can also be given the authority to write to specific subsets of their information, such as contact information or job title. Additionally, organization administrators can create and manage entries for their organization, becoming the directory content managers. To provide access privileges to specific groups of people, roles can be created for each group, such as human resources, finance, or accounting. These roles can be given read or write access to the data needed by their respective groups, such as salary information, government identification number, and home phone numbers and address. Overall, determining data ownership is essential in ensuring the accuracy and reliability of data in the directory.

Data ownership is a critical aspect of managing data in any organization. It ensures that data is well-governed, protected, and effectively utilized [23]. Assigning ownership to data assets is important for a variety of reasons. Firstly, it provides a clear line of responsibility for protecting the data and ensuring its quality [23]. By designating an owner for a particular data asset, it becomes the responsibility of that person to ensure that the asset is secure, reliable, and accessible to the appropriate stakeholders. This includes ensuring that the data is properly structured, stored, and managed according to industry best practices and regulatory requirements. Another key benefit of data ownership is that it promotes easier collaboration across teams [23]. Having a designated owner for a data asset eliminates the need for time-consuming and often confusing communication channels to track down the person with the most domain knowledge of an asset. The asset owner is identified explicitly, making it easy for other team members to know who to reach out to for questions, updates, or any issues related to the data asset. This creates an instant path of communication between team members and the asset owner, improving data quality and ensuring data is being used effectively across the organization. Finally, assigning data ownership promotes accountability within the organization [23]. When assets don't have owners, data quality issues such as outdated data pipelines, irrelevant logic, and duplicated data can pile up, and no one is held responsible for them. By assigning an owner, the organization ensures that one person is held accountable for maintaining the quality of the asset. This person will be responsible for ensuring that the data is refreshed regularly, cleaning up any duplicates, and ensuring that the data asset remains relevant and accurate. This provides a clear chain of responsibility, and eliminates the guesswork involved in determining who should be maintaining what. In conclusion, data ownership is crucial in promoting effective data governance, collaboration, and accountability in any organization. By assigning ownership to data assets, organizations can ensure that their data is secure, reliable, and accessible to the appropriate stakeholders. Data ownership also promotes easier collaboration among team members and ensures that data quality issues are promptly addressed by assigning clear responsibility. As data continues to become an increasingly valuable resource for organizations, assigning data ownership will become more important in ensuring that organizations are effectively managing their data assets.

* + - 1. Determining the data extraction method

Determining the data extraction method is a crucial step in the data extraction process. It involves selecting the most appropriate method to extract data from the source systems while considering the data sources and the data extraction requirements. The data extraction method can vary depending on the type of source system and the type of data being extracted. For example, if the source system is a relational database, the extraction method may involve running SQL queries to retrieve data from tables. On the other hand, if the source system is a web-based application, the extraction method may involve using web scraping tools to extract data from HTML pages. One way for organizations to collect and store information from different sources is by using a cloud warehouse as a central data store. To achieve this, they typically use the Extract, Transform, and Load (ETL) process, which involves several steps. The first step is data extraction, where data is obtained from various sources through APIs or webhooks and then staged into files or relational databases. The second step is data transformation, where the data is converted into a format that's appropriate for reporting and analytics. This step involves enriching and validating the data, applying business rules, and ensuring consistency across all data fields. Finally, the high-quality, transformed data is loaded into a target data store such as a data lake or a data warehouse, where it can be accessed by other stakeholders for analysis and reporting purposes. ETL is a widely used approach across different industries and organizational sizes to integrate various applications, data services, and unstructured files. A well-engineered ETL pipeline with an appropriate data extraction process can provide valuable insights and ensure that stakeholders have access to complete information, enabling them to make informed decisions and eliminating ambiguity resulting from incomplete or uncertain data.

While data extraction is a crucial component of data analysis, it comes with its own set of challenges. These challenges include *managing data volume*, *considering data source* and *API limitations*, *data complexity*, *security and privacy*, and *monitoring data extraction processes* [25]. Firstly, data volume management can be challenging if data extraction tools are designed for smaller amounts of data, and parallel extraction solutions may be necessary when it comes to handling larger quantities. However, designing and maintaining these solutions can be complex. Consequently, for large volumes of data, it may be necessary to use a distributed data processing system such as Apache Spark or Apache Hadoop to extract data in parallel. Next up on the list is data source or API constraints must also be considered, as data sources vary and have different extractable fields [25]. APIs and webhooks may have restrictions on the amount of data that can be extracted at once. Another factor to consider is the data complexity and structure. If the data is structured and well-defined, then a straightforward extraction method such as SQL queries may suffice. However, if the data is unstructured or semi-structured, then more advanced extraction methods such as web scraping or natural language processing may be required. Moreover, data security and compliance requirements are also essential factors to consider when selecting the data extraction method. It is important to ensure that the method does not violate any data privacy regulations or compromise the security of the source systems. To provide a concise summary of the aforementioned points, it can be concluded that, determining the data extraction method involves selecting the most appropriate method to extract data from the source systems while considering the data sources and the data extraction requirements and by selecting the right data extraction method, organizations can ensure that they can extract the necessary data accurately and efficiently while adhering to regulatory and security standards.

* + - 1. Defining the data schema

Defining the data schema is an important step in the data extraction process, as it establishes the structure and format of the data that will be extracted. A data schema is a blueprint or plan that defines the data types, attributes, relationships, and constraints that govern the data. In order to define the data schema, it is necessary to have a clear understanding of the data sources and the requirements of the target system or application. This involves identifying the data entities and their attributes, as well as any relationships or constraints between them. The schema can be defined using a variety of tools and techniques, including data modeling languages such as Entity-Relationship (ER) diagrams or Unified Modeling Language (UML) diagrams. These tools can help visualize the structure of the data and the relationships between different data entities. The data schema should be designed to meet the specific needs of the target system or application, while also ensuring data integrity and consistency. This involves defining data types and constraints that reflect the semantics of the data and ensure that the data is valid and consistent. Once the data schema has been defined, it can be used to guide the data extraction process, ensuring that the extracted data is in the expected format and structure. It can also be used to validate the extracted data and ensure that it meets the requirements of the target system or application. In summary, defining the data schema is an important step in the data extraction process, as it establishes the structure and format of the data that will be extracted. This involves identifying the data entities and their attributes, as well as any relationships or constraints between them, and designing the schema to meet the specific needs of the target system or application. The data schema can be defined using a variety of tools and techniques and can be used to guide the data extraction process and validate the extracted data.

* + - 1. Scalability Challenge

Data scalability is a critical consideration in the design and implementation of machine learning systems. As data volumes increase, the processing and storage requirements can quickly become a bottleneck, limiting the ability of the system to handle large datasets and complex models. This can result in longer processing times, increased costs, and decreased performance, ultimately impacting the accuracy and reliability of the system. One of the main challenges of scaling data is ensuring that the system can handle large volumes of data without sacrificing performance or accuracy [19, 20]. This requires careful consideration of the data storage and processing architecture, including factors such as data partitioning, load balancing, and parallel processing [20]. Additionally, the system must be able to adapt to changing data volumes and requirements over time, without requiring significant modifications or downtime. Another challenge is ensuring that the system can handle a variety of data types and formats, including structured, semi-structured, and unstructured data [21]. This requires flexible data processing pipelines and the ability to handle a range of data inputs and sources, from databases and files to streaming data and IoT devices. To avoid these problems, it is essential to consider several key points. First, it is crucial to plan for scalability from the beginning of the project, rather than as an afterthought. This involves selecting the right *data storage* and *processing technologies*, designing the system *architecture for scalability*, and establishing clear goals and metrics. Second, it is important to prioritize data quality and data governance to ensure that the data used in the system is accurate, complete, and consistent. This can help avoid issues with *data inconsistency* or *duplication*, which can impact the accuracy and reliability of the system. Finally, it is essential to monitor and optimize the system performance over time, using metrics such as *data throughput*, *processing time*, and *accuracy*. This can help identify and address bottlenecks in the system, optimize resource allocation, and ensure that the system is meeting scalability goals and requirements.

There are several data integration and preprocessing tools available, ranging from open-source to commercial products. Popular open-source tools include Apache Spark, Apache Kafka, and Apache NiFi, which offer features such as *data ingestion*, *processing*, and *transformation*. These tools can also be integrated with other machine learning tools and frameworks, making them a flexible and powerful option. Moreover, commercial data integration and preprocessing tools, such as Informatica, Talend, and IBM InfoSphere, provide additional features such as *data quality*, *governance*, and *lineage*. They can be used to manage complex data pipelines and workflows, ensuring that data is consistent and reliable throughout the pipeline.

In the paper by Imawan et al. [26], two approaches for scalable extraction of timeline information from road traffic data were introduced: (1) an *iterative MapReduce approach* and (2) a *single MapReduce approach*. The iterative MapReduce approach utilizes multiple iterations of MapReduce, whereas the single approach involves only one iteration of MapReduce. First off, the authors described how they divided iterative MapReduce approach into two parts: (1) *detecting congestion events* and (2) *updating the dependency of each event*. They explained the details of how to extract the timeline information using one machine in their previous work. In the paper, they propose MapReduce-based algorithms for scalable extraction [26]. The basic iterative MapReduce approach involves running two iterations of MapReduce jobs. In the first iteration, congestion events are detected from the given daily speed log data. The output size is much smaller than the input size after this step. The second iteration updates the dependency of each event generated from the first iteration. The final output is a set of congestion events without the inter-date events. To start, the mapper of the first iteration takes the raw traffic data for each date, which is represented as a list of records containing a date, time, road link ID, and speed value. The mapper produces a key-value pair consisting of a date and a road link ID as the key and a pair of time and speed as the value. This process is described in more detail in the paper [26]. In the basic version of this approach, the entire process of extracting timeline information is carried out at the mapper. The input of this mapper is the daily speed log data and its output is daily congestion events. These inputs and outputs are the same as those of the first mapper and the output of the second reducer of the iterative approach. To illustrate an example of the single MapReduce approach for the basic application, the red box in Figure 6 is used. With the same input as the iterative approach, the mapper detects the congestion event and directly updates its dependency. As there is no additional task for the reducer, the mapper uses date D as key, and there is no explicit reducer needed for the single approach. Similarly, the mapper of the single approach is explained in the paper [26].

Timeline

Description automatically generated

Figure 5: Iterative MapReduce Approach [26].

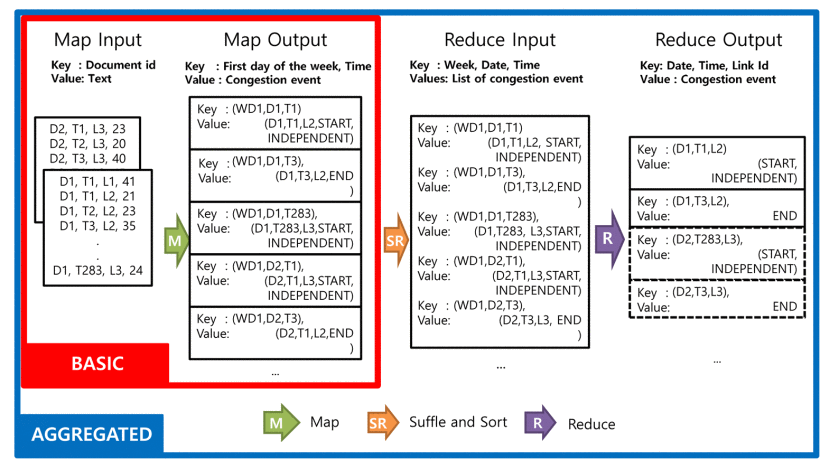


Figure 6: Single MapReduce Approach [26].

* + - 1. Conclusion

As a concluding remark, it is important to highlight that prioritizing this step is also important for building a scalable and flexible system that can handle changing data sources or requirements. Subsequently, this future-proofs the system and allows for easier maintenance and updates as needed. It is necessary to retrieve relevant data from various sources and format it appropriately for ingestion into a machine learning model. Inaccurate or biased results can occur if the data extraction process is not carefully considered and executed, which can ultimately impact the effectiveness of the model and lead to poor decision-making. By selecting the right data sources and extracting data properly, the risk of overfitting or underfitting the model can be minimized, leading to better generalization to new data. In summary, carefully considering and executing the data extraction process is essential to building a robust machine learning system that produces accurate and reliable results, reduces the risk of costly mistakes, and future-proofs the system for scalability and flexibility.

* + 1. Data Ingestion

Data Ingestion is a critical step in the process of acquiring data from various sources and loading it into a data storage system for further processing. This step involves identifying the data sources and their formats, connecting to them, extracting the data, and transforming it into a format suitable for storage and processing. Downstream applications can then access the extracted data that has been loaded into a data storage system such as a database, data warehouse, or data lake. To achieve this, Data Ingestion requires a range of tasks such as defining data sources and formats, establishing connectivity to the sources, implementing data extraction logic, and transforming data into a standardized format. The use of tools and technologies such as ETL (Extract-Transform-Load) frameworks can automate these tasks. The data can be ingested in batches or streams, depending on the data source and processing requirements. Data Ingestion plays a crucial role in building a scalable and robust machine learning system by ensuring reliable data collection, transformation into a consistent format, and availability for processing. By focusing on this step, data quality issues such as missing or inconsistent data can be identified and addressed. Moreover, proper labeling and tagging of data for downstream processing can be ensured. A well-designed data ingestion process can minimize data loss and enable timely and efficient data processing. Moreover, this step can be a complex and challenging process, particularly when dealing with large and diverse data sources; data sources may have different formats, structures, and quality levels, leading to inconsistencies and errors in the ingestion process. Additionally, connectivity and access issues can also arise, especially when dealing with secure or remote data sources. Furthermore, the high volume of data involved can lead to scalability issues if not handled properly. There are two common methods of data ingestion: real-time and batch-based. Following subchapters delves into more details about these methods, as well as the most common ingestion processes like extract-transform-load (ETL), extract-load-transform (ELT) and Reverse ETL.

* + - 1. Real-Time or Stream Ingestion

According to the definition given by Airops team [27], real-time data ingestion (a.k.a. streaming or stream processing) involves *collecting*, *manipulating*, and *loading* *data* as soon as it is generated to create a continuous output. This approach is ideal for organizations that analyze data from web, mobile, and server events, and those that require immediate, reactive actions. Real-time data ingestion is also useful for monitoring IT systems, manufacturing equipment, and Internet of Things (IoT) devices. For instance, a SaaS business that provides a tracking app for on-demand delivery would require real-time data ingestion. Top streaming analytics technologies include Apache Kafka, Amazon Kinesis, and Confluent.

* + - 1. Batch-Based Ingestion

On the other hand, batch processing is the most widely used form of data ingestion and is preferred when real-time data is not required [27]. This approach enables organizations to collect data in large quantities at specific intervals or during a scheduled event. Batch processing is more affordable and is generally the preferred option for most businesses. Batch-based data ingestion is commonly used in data analytics work such as business intelligence, data science, and machine learning. However, it's not uncommon for organizations to use both real-time and batch-based data ingestion. When deciding which method to use, it's crucial to establish your organization's standard of what "real-time" means, as most companies that require real-time ingestion actually want "near real-time" ingestion, which is a batch-based process. Using real-time data ingestion sparingly is recommended as it is costlier and more complex. Remember that infrastructure costs can vary greatly depending on the difference between processing times, and batch processing is generally recommended for anything over five minutes.

* + - 1. ETL, ELT, Reverse ETL
      2. Scalability Challenge

To ensure the scalability of a data ingestion process, it is crucial to consider various factors such as data volume, data velocity, and data variety. These factors play a crucial role in determining the most suitable data ingestion approach and the ideal data storage system, which could be either a data warehouse or a data lake. Additionally, factors such as data quality, consistency, access, and security must also be taken into account. When it comes to determining the right data ingestion approach, a number of factors must be carefully considered, such as the data volume, latency requirements, processing complexity, and the need for real-time insights. Next steps may assists in avoiding discrepancies while designing. First, when identifying the sources of data, it is important to consider their reliability and accuracy. Additionally, examining the source systems and applications that produce the data can provide insight into the data's context and meaning. Then, analyzing the processing requirements should also include an assessment of the tools and technologies required for data processing and analysis, as well as the level of expertise and resources available for implementation [25, 27]. This can help identify potential constraints and limitations in the data processing approach. Third, in determining the latency requirements, it is important to consider the timeliness of insights and whether real-time analysis is necessary [26, 27]. Depending on the application, real-time analysis may be critical for identifying trends and patterns that could otherwise be missed. Subsequently, evaluating the trade-offs between batch processing and streaming should also take into account factors such as *data complexity*, *processing needs*, and *infrastructure constraints*. For example, batch processing may be more appropriate for large, complex data sets that require significant processing power, while streaming may be better suited for real-time, event-driven data processing. Finally, choosing the appropriate data processing approach is critical for ensuring the *accuracy*, *reliability*, and *timeliness of insights* [27]. It is important to regularly monitor and evaluate the data processing approach to ensure it continues to meet the changing needs and requirements of the organization. After considering all these factors, the appropriate approach can be chosen that best suits the processing needs and constraints. By taking a thoughtful and thorough approach to data ingestion, organizations can build an enterprise-level data pipeline that is scalable, efficient, and effective.

* + - 1. Conclusion

As a final note, it is imperative to highlight the difference between *data ingestion* and *data integration.* They are two distinct processes in the data pipeline, but they are closely related and often used together. Data ingestion is the process of acquiring data from various sources and loading it into a data storage system for further processing. It involves identifying the data sources and their formats, connecting to them, extracting the data, and transforming it into a format suitable for storage and processing. The focus is on moving the data from the source to the storage system in a timely and efficient manner. Data integration, on the other hand, is the process of combining data from multiple sources and presenting it as a unified view. It involves mapping and transforming data from different sources to a common format and schema, and then loading it into a central repository or data warehouse. The focus is on making the data usable and accessible for analysis and decision-making. While data ingestion and data integration have different goals, they often go hand-in-hand. Data ingestion is the first step in the data pipeline, and it lays the foundation for data integration. Without reliable and consistent data ingestion, data integration becomes difficult or impossible. Conversely, data integration often requires data from multiple sources, and data ingestion is the means by which that data is acquired. There are several open source tools that can help with data ingestion and integration, each with their own strengths and capabilities. Here are some examples. **Apache Kafka**, it is a distributed streaming platform that is designed to handle high volumes of data in real-time. It can be used for data ingestion, data processing, and data integration across multiple systems and applications. Kafka supports message queuing and pub-sub messaging models and provides strong durability and fault-tolerance guarantees. Some advantages of Kafka include its scalability, high throughput, and flexibility in supporting a wide range of use cases. Many companies, such as LinkedIn, Netflix, and Airbnb, use Kafka for their real-time data needs. Moreover, **Apache Nifi** is a data integration and processing tool that is designed to automate data flows between systems and applications. It supports a variety of data sources and destinations, and provides a graphical interface for building and managing data flows. Nifi supports data transformation, routing, and enrichment, and includes built-in security and governance features. Some advantages of Nifi include its ease of use, extensibility, and support for real-time and batch data processing. Companies such as NASA, Cisco, and Symantec use Nifi for their data integration needs. One last software which is worth to note is **Apache Airflow**. It is a workflow automation tool that is designed to manage and orchestrate data pipelines. It supports a variety of data sources and processing frameworks, and provides a powerful workflow definition and scheduling engine. Airflow supports data validation, transformation, and quality checks, and includes built-in monitoring and alerting capabilities. Some advantages of Airflow include its flexibility, extensibility, and ease of deployment. Companies such as Airbnb, Twitter, and Lyft use Airflow for their data pipeline orchestration needs. In summary, data ingestion is focused on acquiring and loading data into a storage system, while data integration is focused on combining and transforming data from multiple sources into a unified view. Both are important steps in the data pipeline and are often used together to achieve the ultimate goal of making data usable for analysis and decision-making.

* + 1. Data Storage
    2. Data Versioning
  1. Preprocessing
     1. Data Formatting
     2. Data Cleaning
     3. Data Transformation
     4. Data Normalization
     5. Data Reduction
     6. Data Augmentation
     7. Data Sampling
     8. Data Enrichment
     9. Data Aggregation
  2. Validation
     1. Validation Criterias
     2. Data Quality

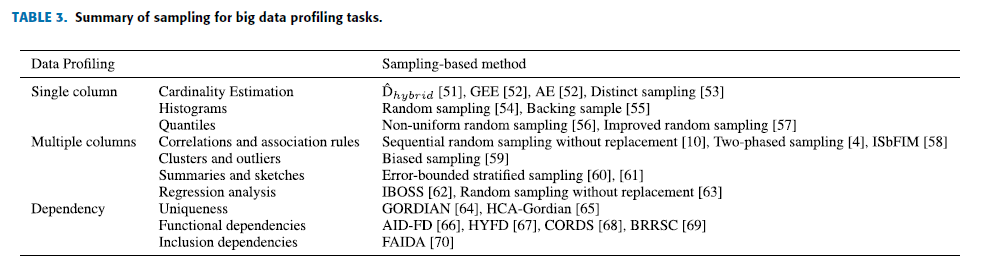
data, including its completeness, accuracy, consistency, validity, currency, and latency.

* + 1. Data Reconciliation
    2. Detecting Anomalies
    3. Error Handling
    4. Documentation

1. Literature Review

[5] Data Governance, Onthologies- The article describes an ontology and a reasoning system for data governance processes, using a Shared Knowledge Plane and a set of rules to control actions. The system was evaluated through two use cases focusing on ETL and security processes, with positive results showing the feasibility of using this technology to simplify the management of big data environments. In the article, the authors present an ontology-based reasoning system for data governance, utilizing a Shared Knowledge Plane and a set of rules to control data processes. The system is evaluated through two use cases involving ETL and security processes and is found to be feasible in simplifying the management of large data environments [5]. In the paper, as a proof-of-concept Telefonica's global video service is used in multiple countries and its service platform produces data required to fuel a big data system, and both use cases have been created based on this service. The focus of one of these use cases is on an environment where data is extracted, transformed, and loaded to obtain information that serves as the basis for subsequent data science tasks. The aim is to show that the implemented system can effectively perform data governance tasks related to the extraction, transformation, and loading of this information. As a result, the paper discusses a testing process to review the system's response to new elements and the generation of new data files. Different JSON files are generated to simulate the arrival of file generation events, and the system detects events and creates corresponding instances of each type with their respective properties. The loading time of events is calculated, and a set of tests is conducted to determine how this time measure varies with the number of instances of the ontology. The loading time remained below 20 seconds for a number of events close to 800, and when the number of events grows, the function can be approximated by a polynomial function. The maximum value obtained for the event loading time was 111301 ms, corresponding to 1760 events.

[14] Data Profiling - The paper discusses the role of sampling techniques in big data profiling. It begins by defining data profiling and sampling technology separately. Data profiling has been studied in depth in previous research, particularly in relation to relational databases. The classification of data profiling is introduced, and the paper investigates the sampling techniques used for data profiling tasks in single column, multiple columns, and dependency scenarios. Traditional sampling methods are introduced, but the paper emphasizes the need for additional sampling applications and information in the context of big data, such as block-based sampling. To ensure a comprehensive survey, the paper follows the systematic search method outlined in previous research. The paper categorizes data profiling approaches into three aspects: data profiling for single columns, data profiling for multiple columns, and data profiling for dependencies. While sampling techniques are not emphasized in previous research, the paper extensively selects studies on sampling for data profiling in each of these categories. The typical methods selected in each category are summarized in Table 3.



The remainder of the paper is organized into different sections. Section II introduces the relevant knowledge of data profiling and sampling techniques, along with important factors to consider in sampling. Section III, IV, and V introduce the application of sampling for single-column data profiling tasks, multi-column data profiling tasks, and dependencies, respectively, based on the classification of data profiling tasks introduced earlier. Finally, in Section VI, the paper summarizes its content and proposes future works. The article's organizational structure is shown in Figure 2. Overall, the paper aims to provide a comprehensive understanding of the sampling techniques used for big data profiling tasks. It emphasizes the need for additional sampling applications and information in the context of big data and offers a systematic approach to select studies on sampling for data profiling tasks in different scenarios.

1. Proposed Solution
   1. Approach & Methodology

To illustrate my point, I would like to begin by presenting a real-life company as an example. However, in the interest of preserving confidentiality and protecting sensitive information, it will be necessary to anonymize or substitute certain details and datasets. By doing so, I hope to ensure that the company's identity and proprietary information are safeguarded while still providing a relevant and informative case study. Moreover, I will implement the techniques and method discussed in this paper and will compare several metrics that are used to evaluate the performance of a data pipeline. The following discourse aims to present a case study of a medium-sized retail company (say company A), which seeks to enhance its customer experience by offering personalized recommendations and augmenting the accuracy of its demand forecasting. In pursuit of this objective, the organization identifies two fundamental business capabilities that will play a pivotal role in its overarching goal, namely, (1) Personalized recommendation engine and (2) Accurate demand forecasting. It should be noted that the company in question, Company A, has an existing data stack, hereafter referred to as Pipeline A, that is intended for similar purposes as the proposed pipeline, hereafter referred to as Pipeline B. As part of the evaluation process, Pipeline B is being subjected to rigorous testing and benchmarking against Pipeline A to determine its effectiveness and efficiency in meeting the company's data needs.

The utilization of company data is expected to provide benefits to multiple teams, among which are the following teams that are anticipated to be significant beneficiaries:

| Team Name | Responsibility | Data Needs | Usage |
| --- | --- | --- | --- |
| Data Science Team | Conducting data analysis, modeling, and experimentation to generate insights and improve data products | Raw data, large amounts of data, historical data, and real-time data, and metadata about the data | Modelling and Analysis |
| Marketing Team | Developing and executing marketing strategies, campaigns, and initiatives to promote products or services, attract and retain customers, and achieve business objectives | Customer data, user behavior data, market data, and competitive data | Conducting market research and analysis, analyzing marketing channel performance, developing targeted marketing campaigns, and measuring the success of marketing initiatives |
| Sales Team | Gather and analyze data related to customer interactions and behavior | Customer behavior and preferences, demographics, and purchasing patterns, sales data and market trends | Identify potential customers and clients, develop sales strategies |
| BI Team | Creating and managing business intelligence tools and dashboards that enable business stakeholders to access and visualize data | Customer data, sales data, financial data, and marketing data | Ensure that data is accessible, accurate, and relevant, and that stakeholders can easily interpret and act upon the insights provided by the data |

Table 2: Mapping of the Data Domains

a This table provides a mapping of various data domains to the teams responsible for managing them, their corresponding data needs, and how the data is utilized by each team.

Table 2 provides a comprehensive overview of the data generated by the company, including the types of data, sources, and intended usage. This essential information lays the foundation for the subsequent steps involved in building a data pipeline. Specifically, the initial step of defining the data requirements has been successfully accomplished through the identification of the specific data needed to achieve specific business objectives or resolve issues, as detailed in Chapter 2.1.1. This step ensures that the pipeline is tailored to the company's specific needs, avoiding unnecessary data collection and processing that can slow down the process and inflate costs. By providing a clear picture of the required data, Table 2 facilitates the design of a more efficient and effective data pipeline.

The subsequent step in the data pipeline development process involves the meticulous identification of data sources. In compliance with privacy regulations, a thorough examination of the data collection methods employed by the company will be elucidated. Furthermore, the nature of the data, including its characteristics, format, and structure, will be expounded upon, along with a comprehensive overview of the specific data entries that are earmarked for ingestion into the pipeline. This step serves a crucial role in establishing a robust and secure data pipeline by carefully scrutinizing the origin, quality, and relevance of the data that will be fed into the pipeline. This meticulous approach ensures that only pertinent and authorized data is integrated into the pipeline, safeguarding against potential data breaches, inaccuracies, and inconsistencies that can undermine the integrity and reliability of the pipeline's outputs.

In this research study, the data sets and data sources to be used for the purpose of the experiment will be introduced. The selection of appropriate data sets and sources is crucial for designing an effective data pipeline and conducting thorough data profiling. This section will provide an overview of the important metrics and characteristics that need to be considered during the data collection process. However, for a more comprehensive and detailed description of each data set/source, reference to Appendix A will be made. The selection of appropriate data sets and data sources is a critical step in any data-driven research or experiment. For this study, multiple data sets and sources have been carefully chosen to ensure the availability of relevant and reliable data for analysis. These data sets and sources are essential for constructing a robust data pipeline and conducting thorough data profiling, which are essential steps in the data analysis process.

When designing a data pipeline and conducting data profiling, several important metrics and characteristics need to be considered. These metrics and characteristics play a crucial role in ensuring the quality and integrity of the data used for analysis.

The collection of these metrics serves as a crucial step in establishing a robust foundation for effective data governance, comprehensive data profiling, and comprehensive data lineage analysis. These metrics not only facilitate the accurate characterization and classification of data, but also provide invaluable insights into the quality, accuracy, and timeliness of the data. This, in turn, enables organizations to establish data management best practices and ensure the integrity and reliability of the data used for decision-making processes. Furthermore, the systematic collection and analysis of these metrics empower organizations to identify data quality issues, identify potential data lineage gaps, and establish data profiling protocols to mitigate risks associated with data inconsistency, inaccuracy, or incompleteness. Therefore, the meticulous collection of these metrics serves as a cornerstone in the development of a robust data governance framework, a comprehensive data profiling strategy, and a reliable data lineage analysis approach. In the data collection process, several key metrics and characteristics play a crucial role in ensuring the quality and reliability of the collected data. These metrics provide insights into various aspects of the data, including its completeness, accuracy, consistency, validity, currency, and latency. In order to gain a comprehensive understanding of the key metrics and characteristics that are important during the data collection process, it is essential to delve deeper into the details of these terms. A thorough examination of these metrics can provide valuable insights into the quality and reliability of the collected data, which in turn can significantly impact the outcomes and conclusions of a research study. For further elucidation on these terms, Chapter 2.3.2 of the research report serves as a rich source of information. This chapter delves into a detailed discussion of each of these metrics, providing a comprehensive overview of their significance in the context of data collection. It sheds light on the nuances, implications, and practical considerations associated with these metrics, providing a robust foundation for understanding their role in ensuring the quality of the collected data.

Furthermore, Appendix A of the research report serves as a valuable resource for gaining insights into the characteristics of each dataset and data source used in the research. This appendix provides detailed information on various aspects of the datasets, including their quality, collection method, schema (i.e., entries of the dataset), and what each entry represents. This information can be crucial in understanding the context and reliability of the data used in the research, as it provides a comprehensive overview of the data sources, their characteristics, and their relevance to the research study. The quality of the data used in a research study is of utmost importance, as it directly impacts the validity and reliability of the research findings. Appendix A provides a detailed assessment of the quality of each dataset, including measures such as data accuracy, completeness, consistency, and validity. This assessment provides a comprehensive understanding of the data quality and its implications for the research study, allowing researchers to make informed decisions and interpretations based on the quality of the data. In addition to the quality of the data, the collection method of the data is another critical aspect that can significantly impact the reliability of the research findings. Appendix A provides in-depth information on the collection method of each dataset, including details on the data collection process, data sources, and data collection techniques employed. This information allows researchers to evaluate the rigor and reliability of the data collection process, and to make informed judgments about the potential biases, limitations, or strengths of the data. Furthermore, Appendix A also provides insights into the schema of each dataset, which includes the entries or fields of the dataset and what they represent. Understanding the schema of the dataset is crucial for interpreting the data correctly and extracting meaningful insights. The schema information provided in Appendix A enables researchers to understand the structure and organization of the data, and to identify the specific data elements that are relevant to their research objectives.

In conclusion, the selection of appropriate data sets and data sources, as well as consideration of important metrics and characteristics during the data collection process, is crucial for ensuring the quality and integrity of the data used for analysis. The data capture frequency, data completeness, data accuracy, and data latency are important aspects to be considered in the data collection process, as they directly impact the reliability and validity of the analysis results. By carefully considering these metrics and characteristics, a robust data pipeline can be designed, and thorough data profiling can be conducted, laying the foundation for accurate and insightful data analysis in this research study. For a more detailed description of each data set/source, reference to Appendix A is recommended.

* 1. Technical Details & Tools
     1. Experimental Setup
     2. Test Bed
  2. Results & Analysis
  3. Future Work

**Data Mesh Steps (Apache Kafka):**

**Define the domain boundaries**: To effectively manage the data paradigm, a segmentation approach has been implemented, dividing the data into distinct domains that can be assigned to specific cross-functional teams. This ensures clear ownership and accountability for each domain, allowing for more efficient and effective management of the data as a whole.

**Data Lineage(Apache Atlas)**: metadata is captured, including *source*, *format*, and *transformation history*.

Data lineage is the process of tracking and recording the flow of data through the various stages of a system or process. Here are the steps to set up data lineage:

1. **Identify the data sources**: The first step in setting up data lineage is to identify the sources of data that will be tracked. This can include databases, files, applications, and APIs.
2. **Determine the tracking scope**: Decide which data elements and processes you want to track. This could include specific data fields, entire data sets, or specific processes within your data pipeline.
3. **Choose a lineage tracking tool**: There are many tools available to help you track data lineage, such as Apache Atlas, IBM InfoSphere, and Collibra. Choose the one that best fits your needs and budget.
4. **Establish data standards**: Establish data standards to ensure consistency across data sources and to make it easier to track data lineage.
5. **Define the data flow**: Define the data flow by mapping the source and target systems, identifying the transformations, and recording the metadata.

<https://docs.cloudera.com/HDPDocuments/HDF3/HDF-3.5.2/installing-hdf/content/configure_nifi_for_atlas_integration.html>

<https://dev.to/tspannhw/connecting-apache-nifi-to-apache-atlas-for-data-governance-at-scale-in-streaming-f5n>

**Data Profiling**: exploring a data source may involve analyzing the data schema, reviewing data types, and assessing the data distribution. Profiling a data source may involve assessing the quality of the data by checking for missing values, outliers, and inconsistencies. These steps help to ensure that the data sources are appropriate for the model and that the data used to train the model is accurate and reliable.

1. Define the scope: The first step is to define the scope of the data profiling exercise. This involves identifying the data sources, tables, and columns that need to be profiled.
2. Select the profiling tool: There are several data profiling tools available in the market, such as Talend, IBM InfoSphere Information Analyzer, and Trifacta. Choose a tool that suits your requirements.
3. Define the profiling rules: The next step is to define the profiling rules that will be used to analyze the data. These rules can include data type, range, completeness, uniqueness, and consistency.
4. Execute the profiling: Once the rules have been defined, the data profiling tool can be used to execute the profiling. The tool will scan the data sources and generate a report on the quality and structure of the data.
5. Analyze the results: The results of the data profiling exercise need to be analyzed to identify any data quality issues or anomalies. This will help in identifying the areas that need improvement.
6. Resolve data quality issues: Once the data quality issues have been identified, appropriate actions can be taken to resolve them. This may involve data cleansing, data enrichment, or data transformation.
7. Maintain the profiling: Finally, it is important to maintain the data profiling exercise on an ongoing basis. This ensures that any changes to the data sources are captured and the data quality is continuously monitored.

Identify Sources – Data Ownership: <https://docs.oracle.com/cd/E19424-01/820-4806/fpdep/index.html>

1. Summary And Future Work
2. Bibliography

References:

[1] Tartow, C., & Mott, A. (2022, October 4). Data Mesh and starburst: Federated computational governance. Starburst. Retrieved April 2, 2023, from <https://www.starburst.io/blog/data-mesh-and-starburst-federated-computational-governance/>

[2] Zhamak Dehghani. (2022, March). Data Mesh. O'Reilly Media, Inc. <https://learning.oreilly.com/library/view/data-mesh/9781492092384/>

[3] Tao, M., &amp; Gates, S. (2022, August 30). How to treat your data as a product. Monte Carlo Data. Retrieved April 2, 2023, from <https://www.montecarlodata.com/blog-how-to-treat-your-data-as-a-product/#:~:text=%E2%80%9CData%20as%20a%20Product%20(DaaP,personalized%20products%2C%20or%20detecting%20fraud>.

[4] Anderson, R. (2022, July 7). Data Mesh Book Bulletin: Principle of self-service data platform. Starburst. Retrieved April 2, 2023, from <https://www.starburst.io/blog/data-mesh-book-bulletin-principle-of-self-service-data-platform/>

[5] A. Castro, V. A. Villagrá, P. García, D. Rivera and D. Toledo, "An Ontological-Based Model to Data Governance for Big Data," in IEEE Access, vol. 9, pp. 109943-109959, 2021, doi: 10.1109/ACCESS.2021.3101938.

[6] Robert Stevens, Carole A. Goble, Sean Bechhofer, Ontology-based knowledge representation for bioinformatics, Briefings in Bioinformatics, Volume 1, Issue 4, 1 November 2000, Pages 398–414, <https://doi.org/10.1093/bib/1.4.398>

[7] Chen, Y.-J. (2010). Development of a method for ontology-based empirical knowledge representation and reasoning. Decision Support Systems, 50(1), 1-20. <https://doi.org/10.1016/j.dss.2010.02.010>

[8] D. Beneventano, Abdul Rahman Dannoui and A. Sala, "Data lineage in the MOMIS data fusion system," 2011 IEEE 27th International Conference on Data Engineering Workshops, Hannover, Germany, 2011, pp. 53-58, doi: 10.1109/ICDEW.2011.5767645.

[9] Carlos Grande. (2022, July 5). My Data Mesh Thesis. Carlos Grande. Retrieved April 3, 2023, from <https://carlosgrande.me/my-data-mesh-thesis/>

[10] Tomingas, Kalle. (2018). Semantic Data Lineage and Impact Analysis of Data Warehouse Workflows. 10.13140/RG.2.2.30860.03204.

[11] J. Scherbaum, M. Novotny and O. Vayda, "Spline: Spark Lineage, not only for the Banking Industry," 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 2018, pp. 495-498, doi: 10.1109/BigComp.2018.00080.

[12] M. Tang et al., "SAC: A System for Big Data Lineage Tracking," 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 2019, pp. 1964-1967, doi: 10.1109/ICDE.2019.00215.

[13] Marco, D. P. (2021, April 14). Metadata management fundamentals. EWSolutions. Retrieved April 3, 2023, from <https://www.ewsolutions.com/metadata-management-fundamentals/>

[14] Z. Liu and A. Zhang, "Sampling for Big Data Profiling: A Survey," in IEEE Access, vol. 8, pp. 72713-72726, 2020, doi: 10.1109/ACCESS.2020.2988120.

[15] Abedjan, Ziawasch & Golab, Lukasz & Naumann, Felix. (2015). Profiling relational data: a survey. The VLDB Journal. 24. 10.1007/s00778-015-0389-y.

[16] Naumann, Felix. (2014). Data Profiling Revisited. ACM SIGMOD Record. 42. 40-49. 10.1145/2590989.2590995.

[17] S. Song, B. Liu, H. Cheng, J. X. Yu, and L. Chen, ``Graph repairing under neighborhood constraints,'' VLDB J., vol. 26, no. 5, pp. 611-635, Oct. 2017.

[18] A. Bifet, ``Mining big data in real time,'' Informatica, vol. 37, no. 1, 2013.

[19] Gorton, I. (2020). Foundations of Scalable Data Science. O'Reilly Media. Chapter 1, can be accessed at <https://learning.oreilly.com/library/view/foundations-of-scalable/9781098106058/ch01.html#examples_of_system_scale_in_the_early_t>.

[20] Lawton, G. (2022, September 9). 7 data quality best practices to Improve Data Performance: TechTarget. Data Management. Retrieved April 6, 2023, from <https://www.techtarget.com/searchdatamanagement/tip/Data-quality-best-practices-to-improve-data-performance>

[21] A. Q. Mahlawi and S. Sasi, "Structured data extraction from emails," 2017 International Conference on Networks & Advances in Computational Technologies (NetACT), Thiruvananthapuram, India, 2017, pp. 323-328, doi: 10.1109/NETACT.2017.8076789.

[22] Oracle. (n.d.). Determining data sources and ownership. Determining Data Sources and Ownership (Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide). Retrieved April 9, 2023, from https://docs.oracle.com/cd/E19424-01/820-4806/fpdep/index.html

[23] Schott, M. (2022, November 30). What is data ownership and why is it important? Retrieved April 9, 2023, from https://www.y42.com/blog/data-ownership/

[24] What is a data source? definitions and examples. Talend. (n.d.). Retrieved April 9, 2023, from <https://www.talend.com/resources/data-source/>

[25] Eteng, O. (2023, January 27). What is data extraction ? everything you need to know. Hevo. Retrieved January 9, 2023, from https://hevodata.com/learn/data-extraction/

[26] A. Imawan, F. K. Putri, S. An, H. -Y. Jeong and J. Kwon, "Scalable extraction of timeline information from road traffic data using MapReduce," 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, France, 2015, pp. 1-8, doi: 10.1109/DSAA.2015.7344850.

[27] AirOps Team. (2023, April 2). Data ingestion 101: What it is &amp; how to choose the right tool. AirOps. Retrieved April 9, 2023, from <https://www.airops.com/blog/data-ingestion-101>

[28]

[29]

[30]

[31]

[32]

[33]

APPENDIX

Appendix A: Datasets & Data Sources Used in the Study

1. **Transaction Data:**

This dataset comprehensive sales transaction data for one year obtained from an e-commerce shop. The dataset encompasses essential transaction details, including transaction number, date, product number, product name, price, quantity, customer number, and country of origin. Notably, cancelled transactions are denoted by the inclusion of the letter "C" in the transaction number code. Moreover, the data in the dataset is captured on a daily basis, with new transactions recorded and added to the dataset at the end of each day. This frequent capture of data ensures that the dataset is updated regularly with the latest sales data, allowing for timely analysis and decision-making based on up-to-date information. The daily capture frequency also helps in identifying any trends or patterns in sales data in near real-time, facilitating agile and informed decision-making. Furthermore, the dataset exhibits a high level of data completeness, with minimal missing values in key columns such as TransactionNo, Date, ProductNo, Product, Price, Quantity, CustomerNo, and Country. This indicates that the dataset contains comprehensive information for most of the transactions captured, providing a reliable and robust foundation for data analysis. However, it is worth noting that there may be some missing values in the Quantity column for canceled transactions, which can be addressed during the data cleaning process to ensure a more complete and accurate dataset. Subsequently, data accuracy is a critical aspect of data quality, and the captured data is thoroughly checked for accuracy during the data capture process. Validation checks are in place to ensure that the TransactionNo, ProductNo, and CustomerNo columns contain unique and valid values, eliminating any duplication or inconsistency in the data. Additionally, the Price and Quantity columns are validated to ensure that they contain valid numeric values and are consistent with the nature of the sales transactions. This meticulous validation process ensures that the data in the dataset is reliable and accurate, minimizing the risk of erroneous or misleading analysis results. Finally, the data in the dataset is captured with minimal latency, as new transactions are recorded and added to the dataset on a daily basis. This means that the dataset is updated regularly and reflects the most recent sales data available, allowing for near real-time analysis and decision-making. The minimal latency in data capture ensures that the dataset remains current and relevant, enabling timely identification of sales trends, patterns, and anomalies for effective business decision-making.

Table 3: Transaction Data

| Column Name | Description | Collection Method | Data Quality |
| --- | --- | --- | --- |
|  |  |  |  |
| TransactionID | Categorical column representing a unique six-digit transaction number. "C" in the code indicates a cancellation. | Captured from online sales platform and POS system | Transaction number is a primary key, no missing values expected |
| Date | Numeric column representing the date when each transaction was generated. | Captured from online sales platform or POS system | No missing values expected |
| ProductID | Categorical column representing a unique five or six-digit product number used to identify a specific product. | Captured from online sales platform, POS system, or sales receipts/invoices | Some missing values expected, as not all products may have a product number |
| ProductName | Categorical column representing the product/item name. | Captured from online sales platform, POS system, or sales receipts/invoices | Some missing values expected, as not all products may have a product name |
| Quantity | Numeric column representing the quantity of each product per transaction. Negative values related to cancelled transactions. | Captured from online sales platform, POS system, or sales receipts/invoices | No missing values expected |
| Price | Numeric column representing the price of each product per unit in Euros (€). | Captured from online sales platform, POS system, or sales receipts/invoices | No missing values expected |
| CustomerID | Categorical column representing a unique five-digit customer number. | Captured from online sales platform, POS system, or customer data collection methods such as customer registration or surveys | Some missing values expected, as not all customers may have a customer number |
| Country | Categorical column representing the name of the country where the customer resides. | Captured from online sales platform, POS system, or customer data collection methods such as customer registration or surveys | Some missing values expected, as not all customers may have a country of residence |
|  |  |  |  |
|  |  |  |  |

a This table shows columns that are in this dataset, brief description what are they about, as well as, how this data is collected and the quality of it.

This table presents comprehensive sales transaction data for one year obtained from a e-commerce shop. The dataset encompasses essential transaction details, including transaction number, date, product number, product name, price, quantity, customer number, and country of origin. Notably, cancelled transactions are denoted by the inclusion of the letter "C" in the transaction number code. The data contained in this table is obtained through diverse data sources, encompassing the online sales platform, point-of-sale (POS) system, sales receipts/invoices, and customer data collection techniques such as customer registration or surveys. During the sales process or through recording from sales receipts/invoices, transaction details, including transaction number, date, product number, product name, price, quantity, customer number, and country, are captured and documented. Following data collection, the acquired data is stored in a digital database or spreadsheet, depending on the data management system employed by the shop. To enable efficient data analysis and retrieval, the data may be organized in a structured format with appropriate data types assigned to each column. The dataset encompasses a significant volume of 500K rows, signifying 500,000 sales transactions recorded over the course of one year from the UK-based e-commerce shop. However, the actual size of the dataset may vary depending on the level of data captured for each transaction and the level of detail recorded in the sales transaction data. It should be noted that the data in this table may contain missing values, as not all products may have a product number or name, customer records may lack customer numbers or country of residence, and some transactions may be cancelled, denoted by the "C" in the transaction number. Notably, the transaction number is expected to serve as a primary key and, thus, should not contain any missing values. As such, data cleaning and validation techniques, such as data imputation or removal of incomplete records, may be necessary to ensure data quality and accuracy in subsequent data analysis or modeling tasks. Furthermore, implementing data validation checks during data collection can help minimize missing values and ensure data completeness.

1. **Web-Analytics Data**

The dataset being described is a web-analytics dataset obtained from an ecommerce store, namely the one owned by the abovementioned company. The data in this dataset is collected from Google Analytics 360, a popular web analytics tool that provides comprehensive insights into website performance and user behavior. This dataset contains valuable information about website visitors, their interactions with the website, and transactional data related to purchases made on the website. The dataset under consideration is representative of a typical ecommerce website, containing valuable information related to various aspects of website performance. This includes comprehensive traffic source data, providing insights into the origins of website visitors, such as organic traffic from search engines, paid search traffic from online advertising campaigns, and display traffic from banner ads or other display media. Additionally, the dataset includes content data, which sheds light on user behavior on the website, including the URLs of pages visited, how visitors interact with content, and other relevant metrics. Furthermore, the dataset encompasses crucial transactional data, which provides detailed information about the transactions that occur on the Google Merchandise Store website. This includes data related to purchases made by visitors, such as product details, transaction amounts, payment methods, and other transactional attributes. This transactional data can provide valuable insights into customer behavior, purchase patterns, and revenue generation, aiding in understanding the effectiveness of the ecommerce website's sales performance. By analyzing and leveraging this dataset, various data-driven strategies can be formulated to optimize the website's performance. For instance, insights gained from traffic source data can be used to refine marketing strategies and allocate resources effectively to drive traffic to the website. Content data can provide valuable feedback on website usability, user engagement, and content performance, leading to website optimizations and improvements. Transactional data can be used to analyze sales patterns, identify popular products, understand customer preferences, and refine pricing and product strategies. The columns in the dataset are:

* **TimePeriod**: the time period for which the data is recorded, that is daily, weekly, or monthly time intervals.
* **Page**: the specific pages on the website that are being analyzed, that is the URLs or titles of the web pages.
* **Title**: the title of the web pages being analyzed, which could provide additional context about the content of the pages.
* **Visits**: the total number of visits or sessions to the website during the specified time period. A visit or session refers to a single user engaging with the website within a given time frame.
* **UniqueVisits**: the number of unique visitors to the website during the specified time period. Unique visitors represent the number of individual users who have visited the website, regardless of the number of visits or sessions they have made.
* **Views**: the total number of page views or the number of times the pages on the website have been loaded or viewed by visitors during the specified time period.
* **AverageTimeViewedSeconds**: the average time, in seconds, that visitors have spent viewing the pages on the website during the specified time period.
* **ConversionRate**: the conversion rate, which is the percentage of visitors who have completed a specific goal or conversion action on the website, such as making a purchase or filling out a form.
* **Transactions**: the total number of transactions or purchases made on the website during the specified time period.
* **Revenue**: the total revenue or sales generated from the transactions on the website during the specified time period.
* **QuantitySold**: the total quantity of products or items sold on the website during the specified time period, providing information about the volume of sales.

Additionally, this dataset can be utilized for data-driven decision-making and strategic planning across multiple teams within an ecommerce organization, including marketing, sales, product management, and business intelligence. For instance, the marketing team can use this data to identify effective marketing channels, optimize advertising campaigns, and tailor marketing messages based on user behavior data. The sales team can analyze transactional data to understand sales trends, customer preferences, and opportunities for upselling or cross-selling. The product management team can leverage the data to identify popular products, optimize pricing, and make data-driven decisions about product development and inventory management. The business intelligence team can utilize this dataset to generate insights, create reports, and inform strategic decision-making at the organizational level. In summary, this dataset provides a rich and diverse set of data points that are essential for understanding the performance of an ecommerce website. It encompasses traffic source data, content data, and transactional data, which can be analyzed to extract meaningful insights, optimize website performance, and inform data-driven strategies across various teams within an ecommerce organization.

1. **Social Media Data - Twitter**

The social media data in this dataset is collected using the Twitter API, which allows for the retrieval of publicly available tweets based on certain search criteria or user profiles. The Twitter API provides access to real-time and historical tweets and allows for the collection of large volumes of data for analysis purposes. However, there are several data quality considerations to take into account. First, the dataset may contain incomplete tweets, with truncated or missing information such as hashtags, mentions, or URLs. Second, the accuracy of tweet content, including potential misspellings, grammatical errors, or misinformation, should be validated. Third, noise in the form of irrelevant tweets not related to customer churn or the product/service being analyzed should be filtered out. Fourth, data consistency should be ensured by cleaning and transforming the data to a standardized format. Fifth, data integrity should be checked by identifying and handling duplicate tweets, removing potential spam or bot-generated tweets, and ensuring the data is authentic. Lastly, privacy and ethical considerations, such as adhering to data protection regulations, anonymizing PII, and following ethical guidelines, should be taken into account when dealing with social media data.

The social media dataset under consideration is intended for analyzing customer churn, and it comprises data extracted from tweets posted by customers. Such data can provide valuable insights into customer behavior and sentiment towards a particular product or service. The dataset is expected to contain multiple columns, including but not limited to the following:

* **ID**: This column is likely to contain a unique identifier or ID assigned to each tweet in the dataset. Such identifiers can be used for reference and analysis purposes, allowing for the identification and tracking of individual tweets throughout the analysis.
* **Tweet\_Posted\_Time**: This column is expected to contain the timestamp or date and time in Coordinated Universal Time (UTC) when each tweet was posted by the customer. This temporal information can be analyzed to understand the timing and frequency of tweets, and how it may correlate with customer churn. For instance, it may be possible to identify patterns in the timing of tweets, such as increased activity during specific time periods, which could provide insights into customer behavior and potential triggers for churn.
* **Tweet**: This column is likely to contain the text or content of the tweets posted by customers. The tweets may encompass a wide range of content, including comments, feedback, complaints, or other messages related to the product or service being analyzed. This text data can be subjected to various natural language processing techniques, such as sentiment analysis, keyword extraction, and topic modeling, to derive meaningful insights. For example, sentiment analysis can help identify the sentiment expressed in the tweets (e.g., positive, negative, neutral), which can provide insights into customer sentiment towards the product or service and its potential impact on customer churn.

In addition to these columns, the dataset may also contain other relevant data, such as user profile information, tweet engagement metrics (e.g., retweets, likes), location data, or hashtags used in the tweets. Such additional data can further enrich the analysis and provide additional context for understanding customer churn behavior. However, it is important to note that the quality of the social media dataset can significantly impact the accuracy and reliability of the analysis results. Factors such as data completeness, accuracy, noise, data consistency, data integrity, and privacy and ethical considerations should be carefully considered and addressed to ensure the integrity of the findings obtained from the dataset. For instance, incomplete or inaccurate data may lead to biased or unreliable analysis results, and noise in the form of irrelevant tweets or spam tweets can distort the findings. Therefore, thorough data cleaning, validation, and transformation procedures should be employed to ensure data quality and reliability. Moreover, privacy and ethical considerations should be taken into account, such as adhering to relevant data protection regulations, anonymizing any personally identifiable information (PII), and conducting the analysis in compliance with ethical guidelines and best practices. In conclusion, the social media dataset being analyzed for customer churn is expected to contain various columns, including *ID*, *Tweet\_Posted\_Time*, and *Tweet*, which can provide valuable insights into customer behavior and sentiment. However, it is crucial to carefully consider and address data quality, privacy, and ethical considerations to ensure the reliability and validity of the analysis results.

1. **Sales Data**

The dataset represents a comprehensive collection of historical sales data from three distinct sources, providing a wide range of insights into consumer behavior and market trends. The data is sourced from a physical store, an online retail shop, and a thrid party data provider, covering a period of six years, from 2014 to 2019. With a diverse set of columns including OrderID, CustomerName, Category, SubCategory, City, OrderDate, Region, Sales, Discount, Profit, State, and PurchaseAddress, the dataset offers a holistic view of various aspects of sales transactions. Also, we should keep in my that, columns such as PurchaseAddress, CustomerName should be anonymized before processing it. This comprehensive dataset offers a wealth of information for sales analytics, market analysis, and predictive data analytics. It can be utilized to gain insights into consumer preferences, market trends, and sales performance, allowing businesses to set targets, forecast future sales, and make data-driven decisions. Whether it's analyzing the sales performance of different product categories, understanding the impact of discounts on sales, identifying top-performing products or underperforming segments, or evaluating customer satisfaction through ratings, this dataset provides a robust foundation for in-depth analysis and actionable insights. The dataset is a valuable resource for businesses seeking to optimize their sales strategies, enhance customer experiences, and drive revenue growth in today's highly competitive market landscape.

Description of columns:

* **OrderID**: This column represents the unique identifier for each sales order. It is generated by the respective data sources and serves as a primary key for the dataset. The OrderID is collected through automated systems for online sales and point-of-sale (POS) systems for in-store sales.
* **CustomerName**: This column captures the name of the customer who made the purchase. It is collected through customer registration or account information, and in the case of the pharmacy data, it may also include prescription details. The data collection method varies depending on the source, with online sales capturing customer names through online registrations, while in-store sales may require manual data entry or scanning of customer loyalty cards.
* **Category**: This column represents the broad category or type of product sold in the sales transaction. It includes categories such as Electronic accessories, Fashion accessories, Food and beverages, Health and beauty, Home and lifestyle, Sports and travel, and Anatomical Therapeutic Chemical (ATC) Classification System for drugs in the pharmacy data. The category information is typically collected through product codes or descriptions provided by the retailer or pharmacy.
* **SubCategory**: This column provides further granularity within the broader category, capturing the specific sub-category or type of product sold. For example, in the supermarket data, it may include sub-categories such as Mobile phones, Laptops, Clothing, Groceries, etc. The sub-category information is collected in a similar manner as the category, through product codes or descriptions provided by the retailer or pharmacy.
* **City**: This column captures the city or location where the sales transaction occurred. It is typically obtained from the billing or shipping address provided by the customer during the purchase process for online sales or through the point-of-sale (POS) system for in-store sales.
* **OrderDate**: This column represents the date of the sales order, capturing the day, month, and year of the transaction. It is collected through the respective data sources' transactional systems, such as online sales platforms or point-of-sale (POS) systems in the case of in-store sales.
* **Region**: This column captures the region or geographic location where the sales transaction occurred. It may include information such as country, state, or province, depending on the data source. The region information is obtained from the billing or shipping address provided by the customer for online sales or through the point-of-sale (POS) system for in-store sales.
* **Sales**: This column represents the total sales amount for the transaction, including any taxes or fees. It is collected as the gross revenue from the sales transaction and may be recorded in different currencies, depending on the data source. The sales data is captured automatically from the respective data sources' transactional systems.
* **Discount**: This column captures any discount applied to the sales transaction, expressed as a percentage or a fixed amount. It may be provided by the retailer or pharmacy as part of promotional offers or customer-specific discounts. The discount information is collected from the respective data sources' transactional systems or promotional data.
* **Profit**: This column represents the profit or margin earned from the sales transaction, calculated as the difference between the sales amount and the cost of goods sold (COGS). It is typically obtained from the respective data sources' financial systems, which track the cost of goods sold and other relevant expenses.
* **State**: This column captures the state or province where the sales transaction occurred, providing more specific geographic information. It is obtained from the billing or shipping address provided by the customer for online sales or through the point-of-sale (POS) system for in-store sales.
* **Purchase Address**: This column captures the complete address where the sales transaction occurred, including the street address, city, state/province, and postal code. It is obtained from the billing or shipping address provided by the customer for online sales or through the point-of-sale (POS) system for in-store sales.

The dataset is carefully curated to ensure high data quality. However, like any real-world dataset, it may contain some imperfections. Data quality measures include identifying and addressing missing values, inconsistencies, and errors in data entry. For example, missing values in customer names, product categories, sub-categories, or other columns may occur due to incomplete or inaccurate data entry during the data collection process. Data quality checks and data cleaning techniques are applied to minimize such issues and ensure the accuracy and reliability of the dataset. In conclusion, this comprehensive dataset includes historical sales data from multiple sources, capturing various aspects of sales transactions, customer information, and product details. The dataset is collected through automated systems for online sales and point-of-sale (POS) systems for in-store sales, and includes data on customer names, product categories, sub-categories, city, order dates, region, sales, discounts, profits, state, and purchase addresses. Data quality measures are implemented to ensure the accuracy and reliability of the dataset for analysis and insights into consumer behavior and market trends.

1. **Product Review Data**

The dataset provided appears to be a comprehensive collection of customer reviews for product X on the e-commerce website of Company A. The dataset includes various fields, such as *review title*, *review content*, *reviewer information* (such as name and location), *rating*, *likes*, *dislikes*, *average rating*, *category*, *crawled date*, *product description*, *images*, *price*, *URL*, and *whether the purchase was* *verified*. This dataset is likely utilized for sentiment analysis, product evaluation, and customer feedback analysis by Company A. It can provide valuable insights into overall customer satisfaction with product A, including its features, performance, and areas for improvement. The method of data collection for this dataset was not explicitly mentioned by the company, but it appears to be obtained from customer reviews submitted on the Company A website. It is possible that Company A uses web scraping techniques to collect reviews from their product pages or they have internal tools that make this process easier and more efficient.

In terms of data quality, several observations can be made based on the given entry. Firstly, about data completeness, the entry contains various fields, such as *review title*, *review content*, *reviewer information*, *rating*, *likes*, *dislikes*, etc. However, some fields are missing or incomplete, such as the "reviewed\_at" field, which is empty, and fields like "review\_uniq\_id" and "crawled\_at" that have incomplete values. Moreover, data accuracy of the data cannot be fully verified as there is no information on how the data was collected and processed. Furthermore, data consistency, the data is generally consistent, with fields like "rating" and "likes" containing numeric values and fields like "verified\_purchase" containing boolean values. However, there are discrepancies in the format of some fields, such as "price" which is represented with a currency symbol and may require further processing to extract meaningful numerical values. Next up on the list, data validity, the dataset contains subjective information such as review content, which may be biased or opinionated. The "verified\_purchase" field indicates whether the review was submitted by a verified buyer, but it is not clear how this verification is done and if it is reliable. Finally, data currency: The "crawled\_at" field indicates the date and time when the data was crawled, which is mentioned as "02/09/2021, 01:37:30". It should be noted that the dataset may not contain the most up-to-date information as it was crawled at a specific point in time.

It is important to note that the quality and reliability of the dataset provided can significantly impact the accuracy and validity of any analysis or insights derived from it. Therefore, careful consideration of data quality and potential biases should be taken into account when utilizing this dataset for any analysis or decision-making purposes in an academic or research context.

Keys:

* **\_id**: This column captures the unique identifier for each review entry in the dataset. It is generated by the system and serves as a primary key for data management and retrieval.
* **average\_rating**: This column captures the average rating given by customers for a specific product. It is a numerical value representing the average of all the ratings provided by customers. It is obtained from customer reviews and ratings submitted on an online retail platform.
* **category**: This column captures the category of the product being reviewed.
* **crawled\_at**: This column captures the timestamp when the review data was crawled or extracted from the online retail platform. It provides information about the time and date when the data was collected for analysis.
* **description**: This column captures the detailed description provided by the reviewer about their experience with the product. It includes information about various aspects of the product. The description is obtained from the review text submitted by customers on the online retail platform.
* **dislikes**: This column captures the number of dislikes received by the review entry. It is a numerical value indicating the count of customers who expressed dissatisfaction with the review. It is obtained from customer interactions, such as dislikes or thumbs-down ratings, on the online retail platform.
* **images**: This column captures the URLs of the images associated with the product being reviewed. It includes multiple URLs pointing to different product images. The URLs are obtained from the image attachments submitted by customers along with their reviews on the online retail platform.
* **likes**: This column captures the number of likes received by the review entry. It is a numerical value indicating the count of customers who expressed satisfaction with the review. It is obtained from customer interactions, such as likes or thumbs-up ratings, on the online retail platform.
* **location**: This column captures the location of the reviewer who provided the review. The location is obtained from the reviewer's profile information on the online retail platform.
* **price**: This column captures the price of the product being reviewed. It is a numerical value with currency information, indicating the cost of the product at the time of the review. The price is obtained from the product information available on the online retail platform.
* **product**: This column captures the name or title of the product being reviewed. The product name is obtained from the product information available on the online retail platform.
* **rating**: This column captures the overall rating given by the reviewer for the product. It is a numerical value ranging from 1 to 5, indicating the reviewer's overall satisfaction level with the product. The rating is obtained from the customer's review and rating submission on the online retail platform.
* **review\_title**: This column captures the title or summary of the review provided by the reviewer. It is a concise representation of the main point or sentiment expressed in the review text. The review title is obtained from the review text submitted by customers on the online retail platform.
* **review\_uniq\_id**: This column captures a unique identifier for each review entry in the dataset. It is generated by the system and serves as a secondary key for data management and retrieval.
* **reviewed\_at**: This column captures the timestamp when the review was submitted by the