Deep Science Publishing
https://doi.org/10.70593/978-93-49910-26-3_6

Chapter 6: Developing robust data engineering pipelines for complex retail analytics and manufacturing intelligence

6.1. Introduction

What is Data Engineering? Why is data engineering crucial for analytics in retail and manufacturing? Researchers in the fields of marketing and operations management have emphasized the importance of making analytical insights readily available for front-line users. Largely ignored in this literature is the necessity of creating a data engineering platform that will make the availability of data possible. Such a data engineering effort not only facilitates the smooth functioning of analytics in organizations but also leads to democratization of analytics by making it available for use by front line management and, even by customers. In this paper, we define, delineate, and describe the extensive and complex processes and activities involved in constructing the foundations of a robust data engineering platform (Grolinger et al., 2014; Choi et al., 2018; da Costa et al., 2022).

Data engineering data pipelines transform, consolidate, and make available data in a timely, reliable, and usable manner to front-line management analytics tools, such as dashboards, risk management tools, and report generators. A data engineering pipeline has three key objectives: addressing all reporting needs of front-line users, not just those known at the design stage, at low cost and on a timely basis; ensuring the correctness and reliability of all the data used for reporting, and being flexible enough to adapt to the rapid evolution of front-line management needs. Front-line management typically prefers simple but timely reports that leverage previously developed and reliable data. This goal can be realized by providing schedules for the computations of metrics that will hold true through such schedules. The typical user prefers tools similar to traditional spreadsheet applications rather than analytic applications using data presented to the user in abstract database format. A robust data engineering pipeline is a prerequisite for analytics and business intelligence (Kreps et al., 2011; Singh et al., 2021).

6.1.1. Significance of Data Engineering in Retail and Manufacturing

Data engineering is a vital evolution from traditional business intelligence (BI) in its focus on delivering a scalable, reliable, and flexible data architecture that meets organization-wide data needs and acts as a foundation for data analysis, machine learning and reporting. Retail and Manufacturing organizations generate a lot of operational data, and our experience in building both small and enterprise scale data pipelines in both these industries highlight the importance of data engineering in enabling effective analytics. These industries have their own complexities in terms of business logic and processes which drive the data feeds from various enterprise application modules, scheduling and orchestration of multiple jobs to manage key time-based and transactional events and finally designing the pipelines to be highly scalable, robust, and able to deal with the data velocity and volume that such events generate. Complex business processes, data volume, operational imperatives and technology heterogeneity call for robust, scalable and enterprise-grade data pipelines to be built using data engineering, as compared to small scale efforts usually seen in smaller organizations across less complex sectors.

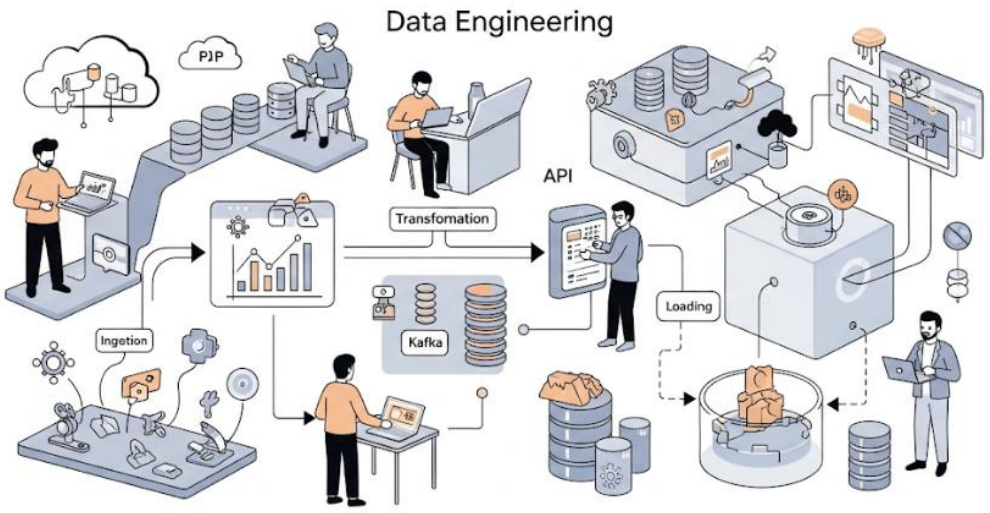

Fig 6 . 1 : Data Engineering

Significant amounts of data are generated by business functions and enterprise applications over the course of day to day operations. While the data is available in these systems, it is difficult to access it, integrate it and deliver it in an easy to consume nature. Data engineering solves this business problem across verticals and functions and across both decision support and operational systems. The organizational quest to gain insights and analytics on business performance, on a weekly and daily basis, combined with the specialized technology stack in place further enhances the importance of good data engineering. The pipelines need to be designed to handle multiple analytics, support batch and real-time use-cases and support complex and ever-changing business needs.

6.2. The Importance of Data Engineering in Retail and Manufacturing

Over the last decades, increasingly data-driven decision making has become critical for any business aiming to remain competitive and make the most out of its resources during both turbulent and stable phases. Making the right data-driven business decisions depends on various factors, e.g., high quality data being collected, stored, processed, transformed, integrated and presented in user-friendly dashboards or reports. Critical and inclusive among these factors is a step, which is often wrongfully undermined, overlooked or simply neglected: the step of developing all the necessary procedures and tools for conducting these operations in an efficient, trustworthy, and scalable manner, that is, data engineering.

Data engineering can be defined as the missing link between data collection and data analytics, whose purpose is to create the foundation that allows for the impactful utilization of the constantly increasing amounts of data that are generated within and across organizations alike. Data engineering activities encompass methods and tools for the impactful collection, storage and retrieval, modeling, processing, transformation, integration and delivery of data in order to enable, support and enhance data analytics activities. Hence, data engineering plays a critical role in any data analytics project, as the results produced by data analytics initiatives ultimately depend on the way that the underlying data have been handled, that is, data engineering for data analytics. In particular, the data pipeline is at the very core of data engineering. Data pipelines connect the disparate systems utilized for storing the data resources used within and across data analytics projects.

6.2.1. The Strategic Value of Data Engineering for Business Success

Whether in the retail sector or manufacturing industry, data are increasingly recognized as essential strategic assets for business success, especially as it relates to the effective delivery of core operational as well as customer value propositions. In large measure because of the widespread adoption of eCommerce and social media by consumers, and the ubiquity of sensors in manufacturing facilities, the volume of data generated and collected is growing exponentially, for some companies increasing at an ever-accelerating pace. Yet, simply collecting these data is not in itself creating value, especially for the larger corporations that have distribution networks and installed bases with billions of touchpoints. In order to augment revenues and profitability, and do so sustainably, these businesses have to analyze these data, derive insights, and act upon these in near real-time. The larger and more complex the business's operations, the more difficult providing these high value-added analytics services becomes. As more and more analytics use cases are deployed across any large enterprise's manufacturing and distribution ecosystems, the more critical the underlying data engineering pipelines become. A well-designed, robust, flexible, automated, scalable, and efficient data engineering pipeline enables the organization to rapidly, accurately, and timely derive business-critical insights from its data. Conversely, a poorly-engineered pipeline increases the burden for analytic teams, decreases the speed with which analytics can be delivered, and limits the number of analytic applications that can be put into production.

6.3. Understanding Data Pipelines

Data pipelines are the engines that power data-driven decisions. Regardless of how advanced an organization’s machine learning models, predictive engines or dashboards are, data pipelines are required to efficiently ingest, model, and curate the data they rely on. High quality data is paramount for all downstream use-cases that rely on it. This data must go through rigorous data engineering processes to enable the correct, timely and accurate usage of information for intelligence. The objectives for a data pipeline includes the collection, organization, and transformation of data, its storage in a suitable data warehouse solution, and the maintenance of the pipeline itself such that the processes are automatically scheduled, monitored and retried when failure is detected.

The architecture of a typical data pipeline comprises of multiple modules, which can be broadly classified into five steps: Data Ingestion, Data Extraction, Data Transformation, Data Load, Data Pipeline Maintenance. Data ingestion, as the name suggests, refers to the collection of data from different sources and the initial store of this information in a staging zone, which is usually a data lake or raw table in a database. Data extraction is the step of extracting the relevant pieces of information from the ingested data. This is usually done after the discovery of which information is relevant to the downstream business processes. Data Transformation refers to the steps of transforming the extracted data - either in a series of transformations that lead to a modeling template being created, or modeling templates being created for common use-cases. Data Load further organizes the data into warehouses and storage courses that ease the consumption of data or efficient model training for process automation, or both, included in a same step of the data pipeline. In the final step, monitoring and maintenance of the pipeline ensures that the pipeline continues to function even as changes are introduced in data structures.

6.3.1. Fundamentals of Data Pipeline Architecture

Data pipelines have evolved from hard-wired data extraction, transfer, and loading (ETL) scripts, often executed in batch at night, to highly configurable core services that move high volumes of data in real-time. A major reason for this evolution is the explosion of internal and external data available to business and operational functions. Data integration and movement are no longer simple background jobs that flow additional data to the business intelligence systems. Instead, virtually every business function generating high volumes of event-based transactions relies on real-time event data moving in and out of core systems, into repositories for offline analysis, and feeding applied machine learning system inputs for prediction and recommendation of immediate actions. Various data reveal some surprising statistics about the structure and maintenance of modern data pipelines. They ingest over 1 petabyte of user event data every single day, in addition to user data, groups, company events, groups, messages, and many other tables. Each event message may modify more than 3,000 different back-end tables. Batch ETL jobs generally move more than 10 petabytes monthly to their OLAP data warehouse.

Consequently, companies should not think of system communication solely from the perspective of data transport and format. Data pipelines are critical core services requiring distributed system architecture patterns that support real-time movement of data, high throughput, and low latency. Supporting criteria include scalability for astoundingly large volumes of data generated every minute, from hundreds of asynchronous sources; micro-batching or streaming capability; guaranteed delivery; configurable or programmable sequence of transformation processes; reliable monitoring and alerting; and fault tolerance. Data movement pipelines should support reliable, lossless storage of potentially vast amounts of event data in common formats for ad hoc analysis during initial investigations of system events or as long-term archival for compliance.

6.4. Key Components of Data Engineering Pipelines

Understanding of data engineering techniques would be incomplete without consideration of various components of data engineering pipelines. Based on our experience, a data engineering pipeline is generally composed of the following components:

Data ingestion involves the movement of data from source systems to staging areas. Data can be ingested in real-time or batch mode. Batch data ingestion is still the most common approach to ingest data from source systems. However, it is observed that more and more organizations are adopting stream-based or change data capture methods of data ingestion. This is especially true for organizations operating in areas of e-commerce, fintech, online gaming or other areas of low latency. Using tools, organizations are capable of ingesting terabytes of data, commensurate with the underlying database transaction volume, with minimum latency.

Data storage components refer to specialized storage structures that allow optimized storage of raw and transformed data at multiple levels based on predefined service level requirement policies. These policies can be image, low level storage with high time required to access, frequency of data change, and query access frequency. Most data engineering pipelines stage raw data in data lakes. However, purely raw data storage may not be suitable for analytics. Therefore, it is recommended that data lakes be curated through the creation of one or more formats optimized for analytical query access or a specialized storage structure suitable for query execution such as data warehouses or data marts.

Data processing, the next component of data pipelines, involves the preparation of data for real-time or near real-time analytics. This step typically involves the execution of a combination of multiple algorithms from the domain of enterprise analytics such as data exploration, data engineering, machine learning, deep learning or business rule execution either in batch or real-time mode. These algorithms typically involve mapping of records through mapping functions, extraction of specific fields through data scrubbing, enrichment of records through association with other data sources, matching of records using algorithms from the domain of probability and statistics, execution of business rule actions using algorithms from AI and recreation of data tables and schemas using microservices.

6.4.1. Data Ingestion

The data ingestion layer is the main entry point for data into a data engineering pipeline. It is responsible for ingesting data coming from multiple producers, integrating the data coming from multiple sources, and forwarding the transformed data to downstream layers of the data pipeline for storage and processing. In a typical real-world data pipeline for a retail application, a variety of data sources need to be ingested including various enterprise applications, data lakes, database systems and cloud services, Internet of Things devices, web services, flat files, and other legacy systems.

Enterprises use multiple tools and systems for executing their business processes and thus creating important data that logs business transactions. Some of these applications are developed and maintained in-house while other enterprise applications are provided by vendors. Various enterprise applications will have their own proprietary storage systems, data formats, and authentication authorization interfaces. We see a need for building an automated, plugin-capable, flexible data ingestion system for enterprise applications, which is capable of handling diverse types of enterprise applications, is easily extensible, can be integrated with cloud and on-premises infrastructure, and is resilient, reliable, and fault-tolerant. Additional complexities arise in the case of applications that do not expose APIs and only store data in some proprietary format. An even greater challenge is posed by legacy applications that do not offer any means for integrating with external systems, but often need to be analyzed alongside the data from existing systems.

Once the data is transported to the data ingestion layer of the retail data pipeline, the data ingestion layer needs to mediate interfaces between the various data sources and the underlying data structure, enabling uniform data storage and querying interfaces. The objective of the mediation agent is to provide a uniform viewpoint of all data from multiple sources to the data storage and processing systems.

6.4.2. Data Storage

A data storage system stores the ingested data for access, at rest, by other components of the data engineering pipeline. In the case of batch processing, the system must be capable of holding the data until processing is complete. In scenarios where data is transformed and subsequently added to the dataset to enable additional insights about the collective dataset, such as a common use case in business analytics, the data must also be able to persist through data transformations. The data storage system must also align with the volume of incoming traffic and the intended analytics workloads accessing the data in terms of storage type and how the storage is deployed. Finally, data access times may also factor into the choice of data storage utilized, especially when considering real-time processing use cases where time is a critical factor in enabling the decision to take action on the analytics insights.

We generally distinguish between two categories of data stores: data warehouses and data lakes. Data warehouses are typically configured to accommodate analytical workloads that access small volumes of the total dataset at a time and involve expensive, multi-table joins. Because of that, a data warehouse generally requires the processed data, such as aggregated data or fact tables and dimension tables, to be persisted to enable efficient access times, especially when considering low-latency requirements. In scenarios where analytics needs require all of the underlying data to be dynamically assembled with each query execution - such as in web analytics and where the primary table is queried often and joins with other small tables are common - a traditional relational database may be more efficient for storage and access times, especially when considering that the data is persisted and may also have transactional processing applied to it.

Data lakes, on the other hand, are better configured to handle the sheer volume rather than the access, in terms of counts and complexity of query operations, required to complete analytics workloads. Therefore, data lakes may be better suited to hold raw, unprocessed, and unstructured datasets, especially when considering the continued growth of Internet and remote devices used for sensor detection and real-time location data. The analytics workloads for these types of data are often run against the raw data itself and extract datasets for downstream consumption or serve reports for the business rather than return analytics insights at application runtime, so these workloads may be lower on average in terms of availability requirements.

6.4.3. Data Processing

Data processing is the act of transforming data from one form to another. We use data processing to explain a comprehensive and exhaustively detailed set of operations that are specified and developed to transform raw data into refined data so that it is amenable for end-user consumption such as visualization or model experimentation. These data operations may include information processing, production planning, interactive demand forecasting, item categorization, and so on for the retail and manufacturing industries.

The retail and manufacturing industries generate enormous quantities of raw data from wide and diverse heterogeneous sources such as sales transactions, product catalog, promotional calendars, marketing budgets, product supply structure, supplier reliability, and customer service systems in an unrelenting and continuous pace. This raw data has little reusability or consumption potential. It needs to undergo a series of varied levels of pre-processing, manipulation and processing to convert it into multi-purpose unique historical data products; rich and data resourceful that capture important trends and patterns, not only are they directly usable for business operations and high-level decisions but also are they indispensable for developing and modeling any data levels, from product to channel to wide food market at both national and local levels; and used for creating and calibration of demand models for forecast and execution at all product levels, across all channels, and at all points in time; as well as for execution and deployment of multi-level demand component models with specific validations, they are key inputs for measurement of uncompensated elasticities and for assessments of short-term and long-term feedback responses to price, coupon, premiums, display, and item-specific advertising.

6.4.4. Data Transformation

The data transformation component of a data engineering pipeline is meant to apply transformations on the cleaned and processed data into interpretable and analytic-specific data. This is also the final stage in the data engineering process before the data is made available to the business analytics and intelligence teams for actual business reporting and analysis. Because of the wide spectrum of divergent analytical, reporting, and machine learning needs, the requirement for transforming data is dynamic. Our work largely supports queries submitted by the BI team as well as other business functional teams across different flavors of data e.g. flat-columnar reshaped tables meant for dimensional modeling and efficient querying, advanced data models that aggregate metrics over customer, time, and product hierarchies, as well as datasets for predictive modeling and bottoms-up calculations used by data science. A separate analytics layer is built using query optimization techniques as part of the transformation pipeline, to support the wide variety of functional and analytical stakeholder teams across business functions.

Given the diverse team objectives, the complexity in their transformation requirements, and the dynamic business needs, it is vital to build a flexible transformation engine that abstracts the undifferentiated heavy lifting for common use cases. A small subset of common transformations consumed over time by these users hold the key to reusability and efficiency. Optimization and automation should be the key mantras while building such a transformation engine. Providing a central repository for these templates, automating their orchestration, and enabling rapid change management form the key pillars in building a performant data transformations suite. In our use case, we augment these use cases with business metadata embedded in query templates to optimize for runtime performance and data freshness with intelligent scheduling that dynamically throttles based on performance impact.

6.4.5. Data Quality Management

The astuteness of any data pipeline is hinged on its ability to efficiently reinforce its underlying intelligence and to allow effective remedial measures to be devised towards maintaining a consistent quality of data during the effective lifecycle of the pipeline. In our pipeline design, we make use of an additional layer - Data Quality Manager (DQM) - which is a service layer that encapsulates functionalities that allow the data architects to define Data Quality Rules (DQRs) for different data sources across the data pipeline processes. Additionally, it allows the user to set the frequency and conditions under which these DQRs require to be validated and the route for further remedial actions when the data quality rules get violated. DQRs may include checks for validation of business rules on the incoming and output of data sources, intrinsic checks for missing values, null values, outliers and threshold checks for aggregates and ratio calculations. The DQM repository hooks into the various data sources or components of the pipeline where users have defined the DQRs. The DQM service exposes a REST API that is triggered by an external orchestrator for validating predefined DQRs before the commencement or during the execution of various pipeline jobs.

Having a dedicated DQM wrapper over source transformation jobs achieves the advantages of processing DQRs at appropriate times and locations, reduces duplicity of QA code, and centralizes enforcement of business/quality rules, defect remediation, and notification alerts. The DQM also allows for optimizing the re-checking exercise by deciding the various sources of concern based on which DQRs require obtaining only those DQRs which haven't passed previously for the current data jobs for checking rather than checking for all predetermined DQRs. Data quality automation in pipeline architectures is a critical cornerstone for establishing and maintaining trust in the datasets generated and for an organization's eventual success in achieving their different data-driven objectives.

6.5. Technologies and Tools for Data Engineering

The primary purpose of this chapter is to present the key methodologies, processes, technologies, tools, and services used in Data Engineering. The chapter is a review of such methodologies and of the best-of-breed tools capable of serving the needs of complex Data Engineering Pipelines in Big Data technologies and services. Because Data Engineering is an ever-evolving field, this chapter can't be prescriptive as to which tools and services should be utilized in which situations. If the expansion of Digital Data and Data Services has democratized Machine Learning and Data Usage, it has also provided a rich marketplace of ETL, Data Preparation, Data orchestrators, Data Warehouses, Data Lakehouses, Data Lakes, Batch and Stream Processing Engines, Data Catalogs, Cloud Infrastructure, and Vendors to Data Engineers that aim to build, manage and monitor complex Data Engineering Pipelines. This broad availability of such services can only speed up and simplify the work of Data Engineering teams around the globe and allow them to focus on ensuring that it provides business value in a timely and error-free manner. Particularly in large organizations, there tends to be hundreds if not thousands of Data Analysts and Data Scientists generating digital artifacts every day but only a few dozen Data Engineers who are responsible for designing, developing and monitoring the Data Pipelines that allow for the proper and safe flow of data. However, to ensure the success of the company efforts around Data, Machine Learning, Business Intelligence and Digital Data Presentation, the work of Data Engineers have to enable at scale in a timely and error-free manner the work of Data Analysts and Data Scientists.

6.5.1. ETL Tools

Introduction to ETL Tools

A vital component of developing modern pipelines for decision support in retail analytics and manufacturing intelligence is to read, transform, and write large stream, structured datasets. These datasets are usually transactional and initialization files. Major Data Engineering processes such as data extraction, data cleansing, data validation, and data integration are executed at scale over these datasets, so correct design of data pipelines has huge impact on business results. The tools that provide such Pipeline as a Service solutions are known as ETL Tools.

ETL tools move and manipulate data into meaningful data warehouses, the hub for business intelligence to achieve business insights. ETL has two major operations, one of which is responsible for data integration. Data is collected from a variety of sources; such sources include database systems, text files, XML files, online services, as well as other ETL sources. Strictly speaking, the graphic representation of ETL focuses on the movement of data to the warehouse where the data is integrated, summarized, filtered, and stored in a predictable and scalable fashion. Integration may also include metadata lookups in data catalogs. The other major operation is the movement of the integrated data to the warehouse. Movement involves efforts to summarize and stage the data and transfer it on a schedule that fits the needs of the organizations. Data movement means dealing with decision-timed data; timely data is moved from staging and summary tables to the decision support environment. When the data cubes are small enough, movement for present period data is accomplished by memory copy commands. If data cubes are larger, only a few bytes of summary data are copied to cube summary tables for the present period, and database commands redistribute the other data.

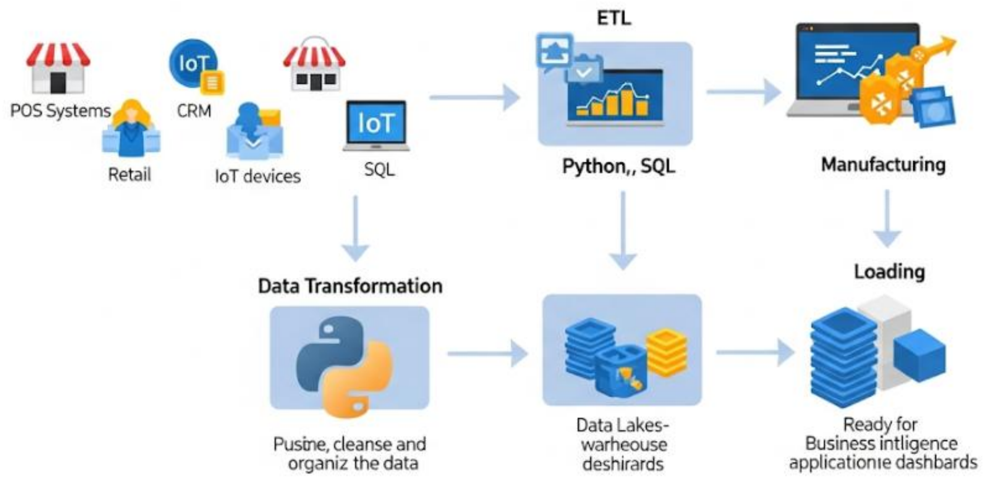
POS Systems
IoT
CRM
Retail
IoT devices
IoT
SQL
ETL
Python,, SQL
Manufacturing
Data Transformation
Loading
Pusine, cleanse and organiz the data
Data Lakes- warheouse deshirards
Ready for Business intligence applicationie dashbards

Fig 6 . 2 : Retail manufacturing Intelligence

6.5.2. Data Warehousing Solutions

The data we have discussed so far is mostly transient in nature. After an e-commerce promotion is over, the user behavior data is removed from the source. However, if you think from a data analyst point of view, this is the data that she would like to run queries on. If the data is continuously flowing in, and you want to query on the historical data along with the current one, you need to save the queue in some location. Another important thing about most of these source systems, they are not designed for querying and analytics. You want a fast solution where you can run your SQL queries without compromising performance. Coming to another point, there can be multiple source systems. You may need to regularly extract data from each of them and load it into a single system, from where you can easily query the data. For example, if you want to see how many users are clicking on ads posted on social media, or how many clicks on paid ads are resulting in sales on the website, or how many users are following the posts. You need to have the data from all the systems in one location. A Data Warehouse is an optimal solution to this, which is designed for query and analytics. And though not single handedly, you may use Data Warehouse as a foundational piece and may build similar systems to resolve your analytics needs given that Data Warehouses come at a cost.

6.5.3. Big Data Technologies

Big data technologies, frameworks, and tools are key infrastructure components of a modern data engineering stack that can scale to diverse requirements of sourcing, processing, packaging, curating, and delivering consumed-ready data and insights. In principle, traditional technology stacks featuring relational backends and batch-based ETL can be expanded to satisfy these needs by scaling vertically or horizontally by deploying more powerful machines or more lower-cost machines. We briefly summarize key considerations when evaluating what stack and tools to choose based on use case requirements since the number of available solutions is dizzying.

Key capabilities that are novel in big data stacks compared to traditional stacks include (a) functioning efficiently without relying on SQL to describe data sources and transformation intents, (b) multiple data input and output options that can include files, databases, realtime and batch message protocols, and structured, semi-structured, and unstructured data of different velocity, volume, variety, and value characteristics, (c) effectively managing evolving and heterogeneous schemas across different sources, especially data lakes, (d) combining batch, near-realtime streaming, and point transformation of data on a source, (e) support multiple incompatible data processing engines including SQL-like engines, graph engines, MapReduce, Dataflow, and Flink Streaming, Python, and machine learning SDKs, without needing to redefine workflows for each engine, (f) Python, SQL, Scala, Java, and R APIs to allow developers to work in their language of choice for every step, (g) simplifying managing infrastructure such as provisioning, monitoring, and managing costs without incurring platform lock-in.

6.5.4. Cloud Services

Cloud computing has transformed technological innovation, allowing companies to outsource their data processing needs and focus on their core competencies. Cloud computing allows organizations to decrease their investments in physical infrastructure, while flexibly adapting their operations to changing demand. Major cloud computing companies provide a comprehensive catalogue of products and services that allow users to implement a complete Data Engineering stack using cost-effective cloud solutions. This versatility has been successfully leveraged for a wide range of Data Engineering problems across industries.

To cite a few examples based on our experience, we have used associated cloud offerings from major cloud service providers to develop Data Engineering pipelines for: data harnessing, data storage, data processing, and analytics deployment. All these were done in some cases as hybrid Data Engineering implementations that involved cost-effective integration of cloud and on-premises assets, based on performance, security and/or privacy considerations. Hence, to enable organizations to adapt their additional Data Engineering needs, irrespective of their architectonic preferences, we describe next the available cloud services across the different Data Engineering pipeline stages.

6.6. Challenges in Building Data Pipelines

Building data pipelines capable of processing the volume and variety of data generated by retail organizations today is not a straight-forward task. The data generated by such businesses is not only huge in size, but also increasingly diversified in source and structure, and is generated continuously. Processing such diverse data sets present a number of challenges to data architects: data pipelines should be capable of scaling both in terms of the size of data and in terms of throughput; the development process should be efficient in terms of time, and not give rise to costly delays in data availability; enterprise data should not be scattered across data silos, and mechanisms need to be developed for ETL/ELT processes that cover data products besides centralized data lakes, making data available in a consistent manner; and finally, organizations should have a mechanism of efficiently processing the real-time streams of events generated by customers and in-store devices, and reconcile it with the batched data for historical analytics.

With recent advances in cloud computing, scalability has become less of a challenge. However, data pipelines typically created in ad-hoc manner during the initial phase of development, become more elaborate as organizations recognize the impacts of the transformations performed in the data pipeline stages. Heuristic and code-driven approaches to integration become production versions requiring careful review to ensure that the transformation and enrichment process is accurate, consistent and efficient. In large organizations, different teams define their own versions of the schemas for data products, leading to divergence and inconsistency. Continuous change in data schema should be exposed to the consumers of the data, and if the data products are to be used by different teams, the access should be appropriately controlled. Data pipelines should also allow for audit and tracking of data statistics to trace anomalies and holes in the data.

6.6.1. Scalability Issues

Herculean efforts are mainstream today in the pursuit of building a single source of truth information system for retail and specific verticals of rapid delivery, for instance, Q-Commerce or quick commerce. Such efforts eventually jeopardize timely, new, fact-based critical decision-making processes for retail businesses, elevating their failure risk – slow businesses die. The challenge is two-fold, first that of managing massive amounts of data ingested daily for hundreds of thousands of products sold across physical and/or digital channels, often referred as long tail problem, across many regions and countries. Far more than selling specialized and/or niche portfolio products, large-scale retailers typically conduct mass consumption and/or fast-moving product inventories – FMCG sector for instance – research and marketing work. This output sales volume from hundreds of borders with an intensive transaction criteria involves such risky factors as hastening amounts of working capital that are easily blocked for weeks, months, even years; unpredictable product-toward consumer-direct behaviors; unrealistic and incorrect supply chain delivery dates. The second challenge arises from panicking remote or virtual critical decision periods required by business management when sudden lifting or reduction of sensitive volumes imposed operational and marketing disturbances for retail companies or e-commerce organizations are experienced. During sets of notably fluctuated date ranges, key and principal macros must be sensed, tracked along their historic timeline; timely repositories, in the sense of effective process, efficient data transformation should be organized to consolidate wholes or large components of specific product family groups data accompanying temporary seasonal breakdown duration.

6.6.2. Data Silos

Data entered by an organization gradually gets distributed as pockets across the enterprise. This is especially true for the analytics domain, where the data gets processed, ingested, and copied several times for different departments and groups. For example, marketing, sales, finance, and logistics all need the information related to orders. Hence, data related to orders is stored on multiple systems for different purposes, like order monitoring, order analysis, reporting, etc. The customer, product, and order data pipelines often have been running in silos over the years. Each of the data pipelines is developed and operated by a specific team and hence has become a black box for everyone else. The internal product structure in the data pipelines has become implicit for the consumer teams after a couple of data transformations. It is rare, if ever, that data vocabulary is shared across teams.

Security and validation often act as useful guards for these silos. The data stewards keep a watchful eye on the changes to the data pipelines from other consumer teams or source teams. The risk of fire-fighting an issue at the last moment also helps. The increasing need for agility in product development, use of cloud technologies, data-sharing integrations, among other reasons, is sapping the security and validation barriers. The groups of analysts and developers supporting the data pipelines are similar in several requests. Having near real-time access to fresh data is crucial for the success of an organization. New cloud technologies let groups quickly prototype and share data pipelines. This leads to similar data getting processed by multiple teams, vacationing the data silos.

6.6.3. Real-time Processing

With the rise in the volume of embedded devices, the number of systems producing data has exploded. In retail analytics, there is almost an order of magnitude of difference between traditional structured data and systems, and continuous unstructured data from the systems in the form of sound, video, and pictures coordinated with geo-positioning capabilities. The former is routinely analyzed to provide intelligence in bulk, while the latter is utilized for capturing micro-moments of the retail experience in as close to real time as possible, with various hues in data engineering skills, technology, and implementation complexity levels. In the world of Smart manufacturing, embedded devices communicate continuously during the entire life cycle of the product and its ancestors. For defending against adversarial learning, real-time systems and skills are becoming common. These use motion detection and social networking capabilities for tagging, approval, or disapproval for the actions, modeled mostly with unsupervised learning techniques.

Emerging software, hardware, and technology stacks are capable of identifying and processing such data in near real time. The business use case and nature of the data determine the right solution stack, as hybrid solutions tend to work better in aligning with business goals while providing feasible decisions. While the design, integration, and functioning of data flows seem to be easier for such capabilities, there are still gaps and issues in the actual day-to-day functioning from engineering and design perspectives. Data engineers are needed, with a mix of technology and operational skills, so that flow processes, periodically coarse-filtered or notified, produce actionable streaming data products, which can then be used for fast aggregation, inference, or analytical products aligned with the business goals. The potential of technology is enormous, and encompassing its entire dimensions of synergy would require transformative thinking.

6.7. Best Practices for Designing Data Pipelines

Data is omnipresent in today's world. As everything around us becomes only more complex and entwined with technology, optimally storing, processing, analyzing, and utilizing data becomes a matter of importance. Oftentimes, this problem is delicately handled by data scientists, who also address the corresponding business problems. While the output and the corresponding models are of utmost importance, the manner in which the data was processed and turned into a consumable product is often not given enough attention. However, the design of the data pipeline holds equal or often more importance than the data models. This is for several reasons. Firstly, in order for data predictions to be actionable, decision makers require usable reports which present the model outputs in an understandable manner, that is updated frequently and automatically. Secondly, data models are constantly in flux, whether because the underlying business dynamics change or because of updates in the underlying data. Therefore, the data pipelines have to be modified frequently and quickly. However, lack of attention to the design of data pipelines leads to these updates being extremely challenging. Extending the functionality of poorly designed pipelines is extremely cumbersome and can consume time and resources.

If data pipelines are more modular and documented, incorporating changes to pipeline with additional data sources, updated data structures, and altered model designs is significantly easier, ensuring that the time and resources saved can be utilized in driving business impact in other ways. Furthermore, documentation of data pipelines further enhances the quality of the data pipeline because it reduces the burden of comprehension of other data scientists and analysts who are utilizing it building off it.

6.7.1. Modular Design

Modularizing tasks within large analytic pipelines is one of the better practices for engineers to follow, as it leads to better-maintained pipelines, quicker debugging, reduces overall development and QA cycles, and often helps in reusability of similar components for future projects. So, what does module design look like? Each module should do only one thing and be responsible for that one thing. In the world of data engineering, this unit of work could be an individual DAG or a set of closely associated functions and classes that have the expected module interface. In quite a few data workflows, task modularization is naturally available. However, this is not the case for all analytic pipelines. In projects with scarce documentation or components put together on an ad-hoc basis, understanding and splitting an analytic pipeline into reusable modules can be an arduous task.

The above ad-hoc construction might not expose the logical flow of the functional parts of the pipeline clearly, and during a proper refactoring, a more interface-driven design should be used. The input-output contracts of the module should be clearly defined upfront, and data types should be validated and error-handled before the internal logic of the module is executed to ensure that there are no unintended behaviors. Data type and null-safe validation become even more important while developing generic modules where the input and output contracts should cover the major use cases. If the module in question interacts with third-party proprietary software, then such checks should be built in also. Single responsibility keeps the modules clean and relatively straightforward to figure out. Consequently, with input and output contracts clear, using the modules should reduce the development overhead on the engineer using it.

6.7.2. Version Control

Data pipelines are complex software systems and should be reviewed and updated on a regular basis. Creating a modular versioned package to allow better collaboration among data engineers and data scientists working together on projects will be useful as it will help identify causes of changes, and help support and respond to business users. Data pipelines with long lifecycles are often refactored and optimized. Forgetting the original design decisions can lead to inhibiting future changes or introducing bugs in programs that contain historical analyses or data. The same is true for virtually any computer code, not just pipeline and analysis code, but it is important to mention here because of the longer lifecycles and serious effects when data pipelines for automated reporting or indirect or direct decisioning of core business objectives break due to incorrect data, logistics or cost overruns.

Version control systems track changes to code, allowing programmers to see changes in functionality, speed, the use of memory or storage, and any new features, along with the dates of those changes and the initials of the person(s) who made the changes. Version control at a high abstract level is done using a library to package collections of data functions and data classes. High-level version control does not allow you to specify changes at a low level such as the values of each variable or constant. In practice, source code is modified more frequently than compiled libraries and images and therefore is usually what is stored in a version control system.

Using version control systems and libraries is useful for groups of developers and whole companies. Using them to manage data pipelines and libraries helps data companies develop and manage code more efficiently and improve code quality. Automated testing, code review, and branching features of version control systems help ensure that the code entering production gives correct results and can be maintained efficiently.

6.7.3. Documentation

In any pipeline implementation, whether from scratch or using a pipeline framework, I cannot overstate the importance of documentation. During the pipeline design, implementation, QA, deployment, and execution, there needs to be documentation for the design and implementation decisions and the outcomes of each of the above stages. Obviously, the above documentation will not be perfect or sufficiently influential the first time it is made. More likely than not, the first draft of any documentation made while coding along will not be great. That is fine. However, as time goes by and there are numerous iterations and versions of the data pipeline documented earlier in its lifetime, one should continue to improve on the earlier drafts to the best of one's ability.

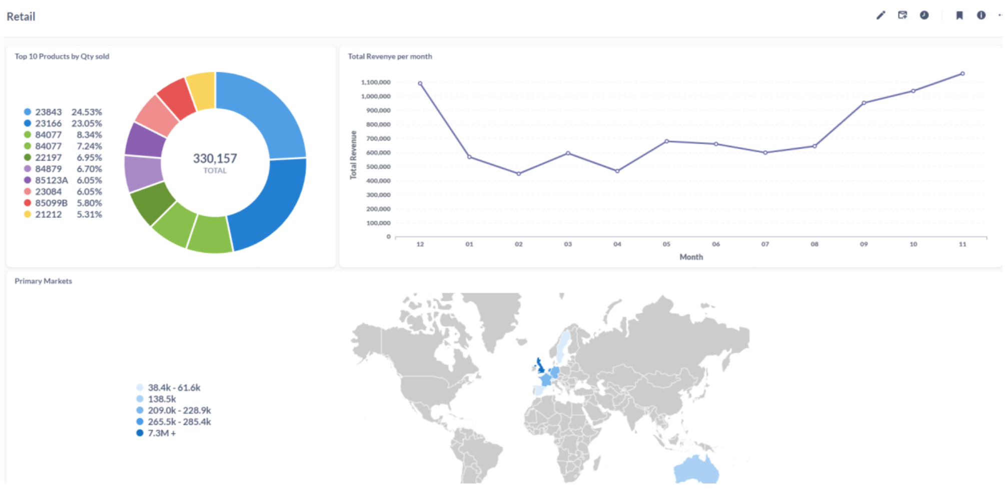

Fig 6 . 3 : Building a Robust Retail Data Pipeline

Why, one may ask, should documentation be continuously improved? For several reasons. One should remember that each pipeline has a lifetime that will span years, at least several months when built on a commercial system. During these months and years, organizations will invariably experience personnel reassignments, turnover, or retirements. In addition, these pipelines perform business-critical functions that ensure the pipeline owners, business analysts, and data scientists get timely and accurate results in a form that is usable for their analytical functions. Therefore, the quality of the documentation is critical to the success of these data pipelines. Documents such as business requirements, design documents, and functional and data element specifications provide essential background for both new personnel and existing personnel to understand what the pipeline was designed to accomplish, the relevant data elements that are used, who were the relevant stakeholders that either influence the pipeline or use its results, and the tradeoffs used during the design and implementation stages, as well as the latter stages of testing and deployment.

6.8. Conclusion

In the digital age where real-time data reigned supreme, the writing team pioneered the discipline of data engineering. With a wealth of experience, having designed, implemented, and operated numerous analytical pipelines in rich collaboration with operational experts, they documented the knowledge to create a streamlined methodology tailored to not only build pipelines that produce trusted, robust data but also teach others to do the same. The objective was to eliminate the bottlenecks associated with data trust and usability and lower the barrier to entry to the world of advanced analytics in retail and manufacturing settings. The discipline of data engineering is not just about how to extract data from source systems and generate a report on it. What is needed is a fusion of expertise and experience in advanced data manipulation and industry knowledge of the many potential drivers, curvatures, and anomalies happening in every retail season. It is how to structure, mend, and collate data across systems, timelines, and business units into a usable form in which the value of advanced analytics truly becomes realized and then leveraged into increased sales and profit through better decision-making. Data engineering capabilities allow organizations to escape from spreadsheets and into automated decision-making tools that reduce time spent on decision support and increase time spent making decisions. The time saved reduces human dimensionality and increases the accuracy of topical decisions being made. It’s about tools that can shape what would otherwise be brutalized under the sizing of innocently automated uploads of transactional detail into caches for administrative burden stretching back homogenized historic reports.

6.8.1. Key Takeaways and Future Directions in Data Engineering for Retail and Manufacturing

Emerging advanced computing and machine learning technologies hold the promise of novel retail analytics and manufacturing intelligence applications to improve the decision making capabilities of the workforce and executives in retail and manufacturing businesses. However, it is equally important to invest in the right foundational solutions to collect, integrate, and structure the right data for analysis. While the period of hype about big data is over, the complexity and scale of the data being generated and used in these applications are. These applications are common to many other domains such as healthcare and financial services. However, what makes retail analytics and manufacturing intelligence data engineering research different and important is the complexity unique to this domain. For retail analytics to enable the relevant corporation customer execution models for CEOs of retail corporations to teach the science and art of execution to the floor teams, it's important to collect the right set of simple and advanced analytics so that they can be done without excessive processing and data engineering time. Specific algorithms and their resource loading and throughput dependencies allow business analysts to feed predictions to the planning dimensions so that these predictions can be used in assortment planning and pricing actions at the relevant forecast and allocation time granularity.

There is further complexity when looking at value stream based analytics for manufacturing and design-to-analytics for microelectronics. Future work can extend the paradigm of data mesh both vertically and horizontally for more domains, and the domain modeling approach can be extended to more data use cases within retail and manufacturing domains.

References

Grolinger K., Higashino W.A., Tiwari A., Capretz M.A.M. (2014). Data Management in Cloud Environments: NoSQL and NewSQL Data Stores. Journal of Cloud Computing, 2(1), 1–24. https://doi.org/10.1186/2192-113X-2-22

Singh A., Singh M., Chatterjee K. (2021). Big Data Analytics in Manufacturing and Retail: A Review and Future Research Directions. Journal of Intelligent Manufacturing, 32(6), 1509–1532. https://doi.org/10.1007/s10845-020-01571-3

Choi T.M., Wallace S.W., Wang Y. (2018). Big Data Analytics in Operations Management. Production and Operations Management, 27(10), 1868–1889. https://doi.org/10.1111/poms.12838

da Costa C.A., Papa J.P., Lisboa C.O., Munoz R., Albuquerque V.H.C. (2022). Internet of Things: A Survey on Machine Learning-Based Edge-IoT Architectures for Industrial Applications. Sensors, 22(5), 2020. https://doi.org/10.3390/s22052020

Kreps J., Narkhede N., Rao J. (2011). Kafka: A Distributed Messaging System for Log Processing. Proceedings of the NetDB, 1–7. https://doi.org/10.1145/2000000/1990000