Deep Science Publishing
https://doi.org/10.70593/978-93-49910-67-6_5

DeepScience
Open Access Books

Chapter 5: Data engineering across industries: Enabling scalable, real-time, and insightful solutions

5.1. Introduction to Data Engineering

Data engineering focuses on acquiring, organizing, and preparing raw data for analytics and other use cases and making the right data available at the right place and at the right time so that it can be consumed and processed without friction. Data engineers implement and maintain the infrastructure such as databases and data lakes pivotal in the extract, transform, load process and data pipelines. Responsible for the architecture and engineering of data systems, data engineers build and deploy data pipelines from which, in turn, data scientists, analysts, and other consumers can extract actionable insights (Napoli, 2011; Lotz, 2017; Lobato, 2019). Data engineering encompasses an array of tasks including centralizing, modeling, cleansing, and transforming data. Businesses, organizations, and institutions rely on data engineers to ensure that the volume and variety of incoming data — structured, semi-structured, unstructured, time-series, or geographical — conforms to the guidelines established by the organization and is stored in a way that optimizes computing capabilities. Big data technologies and programming languages for data engineering enable data engineers to aggregate structured and unstructured data from disparate sources, cleanse and transform it, and load it into relational and non-relational databases and data lakes for downstream reporting and machine learning. At the same time, modern analytical business intelligence and dashboarding solutions have empowered analysts to perform their own ETL without having to work with data engineers (Taneja et al., 2012; Webster et al., 2013).

5.1.1. Background and Significance

The term data engineering refers to designing, building, and maintaining the software systems and architecture that can facilitate the large scale analysis, management, storage, and retrieval of the vast amounts of data generated daily. Building a company's data architecture is a crucial yet challenging function performed by data engineers. It is challenging because of the trade-offs between data architecture performance and efficiency. Poorly designed pipelines which consume more resources to transport the data across various systems directly increase operating costs for businesses while also increasing the time developers spend managing the system, diverting them away from building value-adding applications. Data engineers leverage their data management knowledge, programming and database skills to build pipelines that keep the operations in a stable state. By abstracting the complexities of the infrastructure and designing efficient, user-friendly systems that are economical to operate, data engineers make it easy for data scientists and analysts to run experiments on the data, derive insights, and react to the trends. In addition, by ensuring that the systems are well documented and adhere to best practices, data engineers enable easy troubleshooting and recovery in the event of an incident, minimizing system outages. In a manner similar to software engineering, data engineering draws aspects of its methods and practices from diverse industry domains including database and distributed systems, data mining, and machine learning.

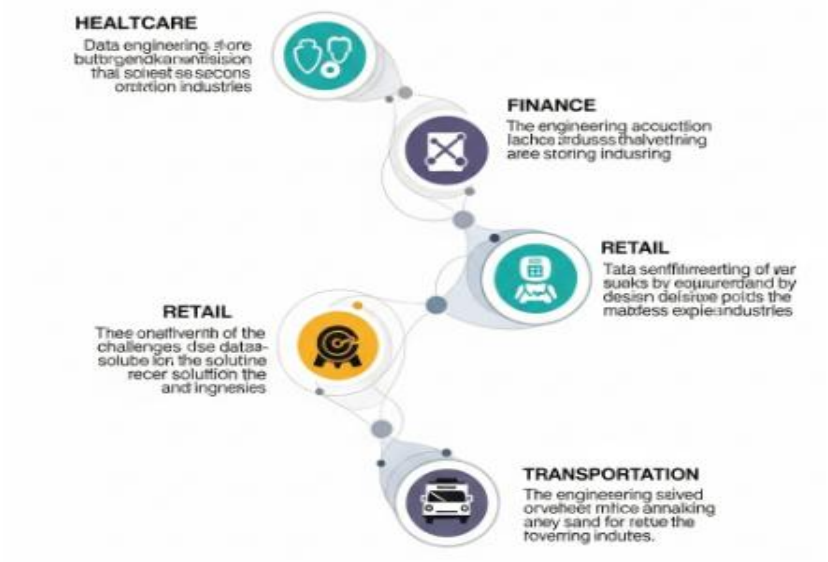

Fig 5.1: Data Engineering Across Industries.

The significance of data engineering in delivering intelligent solutions is instantiated in the number of teams that are dedicated towards performing this function. Organizations

have dedicated teams with the title “Data Engineer”, and focus primarily on building data pipelines/warehouses to empower analytics across the company. Tech companies have developed novel data architectures to enable data products across the world. These endeavors have resulted in open-source tools and warehouses.

5.2. The Role of Data Engineering in Modern Businesses

Over the last decade, data has rapidly emerged as the factor enabling many businesses and organizations to achieve a competitive advantage. Several leading ventures have become exponentially wealthy by leveraging data to build products that have created profound changes in customer habits. A common path followed by most of these organizations involves collecting, storing, and processing data at scale, in real-time, and with high availability. They have also invested heavily in researching, developing, and deploying machine learning models that can provide insightful solutions to problems in areas such as image processing, voice recognition, recommendation systems, and predictive analytics, among others. Other industries and organizations have also pursued similar paths. In an era that some refer to as the “data age”, it is increasingly common to hear that “data is the new oil” – an expression that illustrates the potential that data can deliver to organizations, modern businesses, and the economy as a whole.

To ride the new wave engendered by the data explained in the previous paragraph, and to overcome the challenges imposed by the sheer volume, velocity, and variety of data generated by modern businesses, specialized teams comprising well-trained data scientists and engineers are needed. On the one hand, data scientists and engineers must work together to liberate teams from the burdens of data collection, storage, processing, and validation, so that they can focus on data analysis and the construction of machine or deep learning models. This arises from the data engineering practice called DataOps: a methodology that extends concepts to data-related projects. On the other hand, data scientists must possess extensive domain knowledge to be able to elaborate valid and useful hypotheses, and to select salient features that allow predicting the value that the models will try to estimate and that will be optimized during model training.

5.2.1. Research Design

The research design utilized in this paper is a survey design, which is particularly valuable for exploratory research aimed at portraying a landscape, defining specific categories, and developing frameworks and typologies for the studied topic. The research question, why is data engineering important for modern businesses, is addressed utilizing an e-survey with open-ended questions and statements. The descriptive nature of the study allows for a review of the original data collection effort, as well as an

assessment of the field's state and its implications for future research directions. The portrayed landscape allows researchers, chief data officers, business leaders, data engineers, and especially businesses to have a holistic understanding of data engineering's role in modern businesses.

The responses utilized for this study were gathered in September 2020 with the help of professional networks. The calls for participation were distributed to encourage responses from data professionals worldwide. The participants included data engineers, chief data officers, business leaders, and students, allowing for the collection of either insights or observations from their indirect personal experiences. The calls for participation were coupled with professionalism reassurance so that respondents could be confident in the anonymous nature of the data gathering process and the fact that the gathered data would be used solely for research purposes. In total, 80 responses were gathered, with 3 removed due to not addressing the survey's main subject. The responses were quantitatively validated based on the participant's background. As such, since the goal of the data gathering process is to draw on personal insights or observations, the data was validated by expert judgment, namely who provided the insights. To this end, the expert judgment is directly and indirectly addressed by the professional roles of the respondents and their involvement with decisions related to data engineering and data management.

5.3. Core Principles of Data Engineering

The definition of data engineering varies among industry actors and practitioners. However, despite some variations among the definitions, most include the following core principles: data collection, data storage, data processing, and data quality management. These principles govern engineering practices within and across all major data engineering focus areas and approaches, including single and multi-company data sharing and integration efforts, ETL and DBT pipelines, enterprise data warehouses, data lakes and lakehouses, data marts, analytics workflows, and machine learning pipelines. Overall, data engineers complement data scientists and business intelligence analysts by creating and optimizing the systems that enable scalable, reliable, real-time data handling efficiencies across the rest of the data-oriented toolchain.

Data Collection. Data collection is the first principle of data engineering and encompasses the activities and systems that allow for the collection and ingestion of data from internal and external sources. Creating a reliable, performant data collection process capable of extracting data from various possible platforms, including relational, document, graph, and key-value databases, web and business APIs, external data pipelines, and IoT devices, and into various systems that are themselves co-built with business stakeholders is one of the key defining roles of a data engineer. Data ingestion

processes are the primary enabling component for business and data analysts who run reports and analyses or who build reporting and analytics tools. Data ingestion processes are also core system dependencies for training machine learning models by providing feature information for data scientists and that output predictions by making those feature and model prediction requests to data science-serving processes.

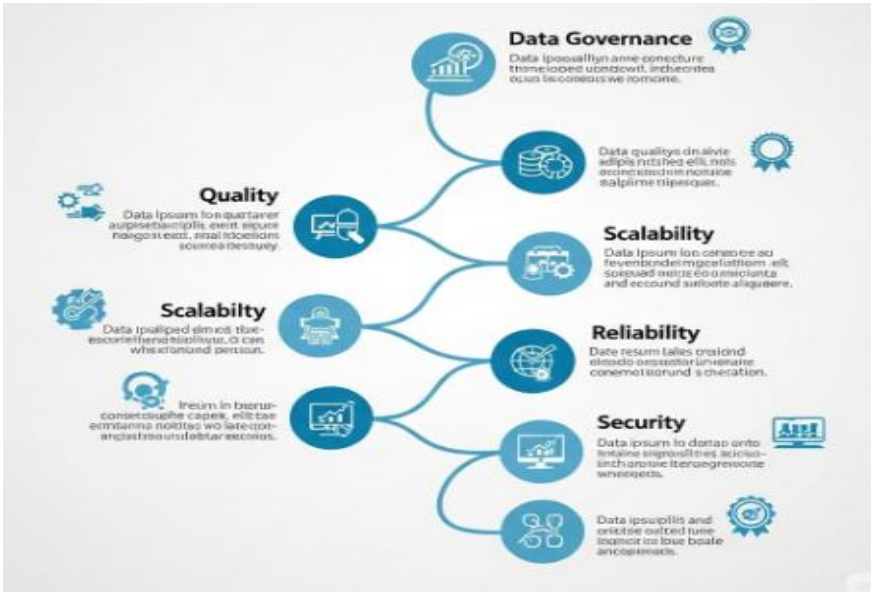

Fig 5.2: Core Principles of Data Engineering.

5.3.1. Data Collection

The speed at which an organization can collect behavior or system data is usually limited by the ability of the systems that generate the data. The social media stream, for example, generates a huge volume of data in real time but is difficult to collect without being throttled. Some data collectors are companies devoted to the collection of data. The data that search engines use for their indices or that reporters use for topics of emerging importance are collected not by the search engines or reporters but by companies that specialize in data collection.

The hardware that generates streams of data, like sensors on vehicles or floors in a hospital, communicates data in real time, as is vented gas data to identify gas emissions. The use of motion sensors to monitor movement over time generates large amounts of data. Monitoring millions of people in airports or train stations generates a massive amount of data. The speed at which data can be collected is partly determined by the protocols used for communication and the need to ensure that each packet of data sent is received correctly. Reliable networks are slow networks, as they require every packet of

data to be acknowledged before the next packet is sent. This is not an issue for streaming sensor data in which a large number of packets are lost, as the data collector is usually not concerned with monitoring every detail but only observing changes in states of the sensors.

5.3.2. Data Storage

Data Engineers enable the organization to store diverse data sources at scale and in a format that facilitates their retrieval. They store data in a way that considers how it will be used later. Data storage considerations are constantly changing, and many factors influence the storage system used, including type, size, scale, read/write tradeoff, velocity, data querying style, and cost of data. Like data pipelines, there are two types of data storage conceptual models: storage models for raw data and storage models for derived data.

There are three core principles for raw data storage: Raw data pipelines store raw data as close to the raw source data as possible. Raw source data resides in industry-standard formats that can be used by the greatest number of tools. Raw source data is compressed to reduce the cost of data storage. Tools that help to implement raw data pipelines include various data processing frameworks. While some tools also allow raw data to be stored in various backends, the most commonly associated storage is in native format compressed with standard compression methods. Columnar and row formats for data processing and serialization are used for efficient data handling. Other tools that help with making raw data available in data repositories include cloud computing services that help store the data and provide a unified namespace that multiple data tools can use.

5.3.3. Data Processing

The primary objective of data processing is to enable data analytics as well as data products that live in applications and user interfaces. It should be mentioned that users and applications can consume two types of pipelines: batch pipelines that run only on past data and streaming pipelines that run on both past and live data. The key principle is to use the least amount of infrastructure, configuration, and oversight possible.

To achieve that, we typically ask ourselves the following questions: what should be handled in the pipelines and what should be handled in the configuration system? For batch, very complex processing that has orders of magnitude more compute for a period of time, say 6 months in the past, is suitable for the pipelines. In contrast, pipelines are not suitable for tasks that are continuous—currently, say 10 minutes and longer update latency on data already seen in the past (such as a counter)—and have low compute or

are sensitive to timing issues. Streaming pipelines are not suitable for tasks that either (1) can only be done at endless latency using ways to ensure orders are later invalidated, (2) apply data dependence in some way that have long continuity periods—mere mortals—are sensitive to timing issues, (3) are extremely complex with access to external resources that vary in response time and availability, and/or (4) retraining based on the output and significant manual intervention is involved. Additionally, the batch and streaming pipelines typically differ in more than just timing.

The primary difference is their effective backend and interface to the configuration and other information systems. Batch pipelines have only an API that is either used to read data requested for processing or to query for a set of recent configuration and other data that is used for reprocessing.

5.3.4. Data Quality Management

While the focus of Data Engineering is to facilitate the visibility of data, it can be done only if the quality of data is ensured. With the volume, velocity, variety, and value of data being as high as it is today, there are many inadvertent mistakes that can affect both the quality of data and the quality of insights drawn from the data. Focusing on ensuring the quality of data should be an obvious priority for Data Engineers. Most organizations are starting to realize the importance of Data Quality. However, a fully-fledged Data Quality Management discipline is still non-existent in most organizations. Even among organizations that do have a Data Quality Management discipline, ensuring data quality is still a conundrum. This is because, unlike Data Quality at Rest, Data Quality in Motion is an aspect of data that is not completely controlled by Data Engineers. Data Quality at Rest is defined as the examination of the data itself while Data Quality in Motion examines the processes and structures, including metadata and Data Quality rules, that control the quality of moving data such as ETL or ELT pipelines.

Some key questions Data Engineers can ask themselves about Data Quality to ensure best practices for Data Quality Management are: How frequently is the data refreshed? By what methods does the data travel from the source to the target? What are the common causes of Data Quality issues for these data travel methods? What type of Data Quality control mechanisms are best suited for these particular methods? What rules should be put in place to ensure quality control? What Data Quality Management tools are best suited for these needs? What best practices can be used? How can a repeatable process be implemented? Some of these questions may already be answered for Data Quality at Rest. These same principles can sometimes be extended to Data Quality in Motion. Apart from the questions posed above, Data Engineers are advised to work with Data Scientists and Analysts to decide on specific Data Quality rules that will meet their

needs. Doing this can ensure repeatability and validity of Data Quality rule outputs compared to the needs of Data Analysts and Data Scientists.

5.4. Data Engineering Tools and Technologies

Data engineers leverage a wide range of tools and technologies to devise solutions for collecting, integrating, storing, processing, and serving data. The evolution of data engineering as a field has been facilitated by the development and advancement of specialized tools that reduce time and effort, automate tedious tasks, and improve robustness, flexibility, and scalability of solutions. The emergence of big data led to the need for novel tools that could efficiently scale to handle massive quantities of diverse and fast-flowing data, often originating from various sources. Alongside, there has also been continued use and expansion of enterprise data platforms and tools originally meant for small or medium scale data workloads.

In this section, we will review tools and technologies commonly used in data engineering along with providing a few examples, recognizing that the choice of tools and platforms is highly contextual and caters to the specific use cases, technology stack, and organizational needs. We will cover popular data extraction, transformation, and loading tools. We will also discuss data warehousing solutions, how to design and implement them, and the role of data warehouses within a larger organization, as well as outline how different data lake implementations fit into the data engineering infrastructure. We will wrap up by touching on popular data processing and integration frameworks that help with the construction of data pipelines, with a special emphasis on those that deal with real-time data processing and storage.

ETL tools help extract data from various sources, transform and combine it as needed, and load it into target data stores, such as databases, data lakes, or data warehouses. While ETL is typically understood to refer to just the data moving task, ETL tools now handle a lot of the heavy lifting in terms of schedule management, job monitoring, data quality checks and alerts, retries on failure, documentation, and more, and often incorporate best practices around data integration. These tools are particularly popular with organizations that need to operate numerous data pipelines between a large variety of data sources and targets. These pipelines may move batch or real-time data payloads and can be highly complex, necessitating reliability and maintainability features that ETL tools tend to have. Data engineers in such organizations, therefore, rely on ETL tools to implement and operate data pipelines with minimal heavy lifting.

5.4.1. ETL Tools

Large retailers, investment banks, and airlines have leveraged ETL tools for generation of reports and summaries that help in day-to-day decision-making. Many ETL tools are available today that provide a range of features. ETL tools can be classified into three categories: the first category is a set of tools that include database utilities such as SQL Plus and Bulk Copy. These traditional utilities have been providing support for ETL processes for many years. They are first-generation Active ETL tools. They provide high speed at low cost but a limited set of features to support the generation of a complete ETL solution. The second category is a set of tools designed especially for data warehousing. With the proliferation of data warehousing as a new trend in business, many companies began to develop special ETL tools. These tools are dedicated to this task. These are second-generation Active ETL tools. They include loaded formations such as Informatica PowerCenter, DataStage, and Ab Initio. These tools provide users with a larger selection of features to build a more networked process and, hence, more flexibility in the generation of ETL processes. The third category is a new class of ETL tools designed with the new business needs in mind. For instance, new web-based tools support the ETL distribution needed. Web-based ETL tools include Data Junction and Data Warehouse Builder. A new and very promising technology in the ETL tools field is the use of middleware and other brokers to facilitate data transfer between applications. However, mixing the data transfer with other applications may be cheaper and faster techniques for current data warehouse implementations. There are many steps involved in the development of the warehouse, including business requirements and business questions, verification of availability and quality of data sources, development of the ETL functions, validation of ETL results, development of the data models, design, and development of the data modeling and specific database.

5.4.2. Data Warehousing Solutions

A data warehouse is a centralized repository for structured data generated from disparate data sources from across the enterprise and provides the ability to perform easy reporting and analytics. The data warehouse acts as a single source of truth. Data is extracted from the data warehouse or source systems, transformed to conform with the warehouse structure, loaded and stored in the warehouse. Business users, analysts, and data scientists create reports and dashboards on top of the data warehouse to get insights about the business and make data driven decisions. The OLAP capability of the data warehouse allows users to aggregate data across multiple dimensions and also analyze granular level details. The data warehouse is designed for relatively simple queries that aggregate data across multiple dimensions, thus it is optimized for read as users only perform analytical

queries as compared to the OLTP systems which serve transactional queries that are read and write heavy and involve many joins.

The traditional on-premise data warehouse requires substantial resources for capacity planning, provisioning, and scaling resources for peak loads. The lack of cost-effective scalability has led customers to move to cloud-native data warehousing solutions which provide on-demand and near-infinite scalability and easy one-click provisioning. These cloud-native systems decouple storage and compute for rapid elasticity and offer pay-per-use cost models. Such systems can enable basic warehouse workloads in a matter of minutes and are built to scale up to 100 petabytes and beyond at near-zero overheads.

5.4.3. Data Lakes

Data lakes became one of the primary storage options for an increasing number of organizations, enabling them to store and analyze all types of data with any structure or schema. Unlike traditional databases and data warehouses that are designed for transactional or structured data, data lakes are intended for huge quantities of raw data, both structured and unstructured. This data is ingested in its original form without having to impose a schema a priori. Ingestion is highly parallelized, increasing velocity, scalability, and lower cost.

Within a data lake, data is kept in a distributed file system, storage provided by other cloud platforms, or in a distributed object-based system. Metadata management and automated data extraction are required to create a data catalog that can organize the data, allowing users to easily find relevant datasets in the lake. Tools to query data from the lake are also necessary, allowing end users to extract data with low latency. The products provided by the different cloud vendors are the best data lakes in terms of cost and adaptability. Data stored in a data lake can be transformed into a much more structured form in a data mart or data warehouse to be consumed by business intelligence users requiring fast and reliable queries. Data lakes became appealing as organizations collected more external data coming from the Internet of Things, surrounding a plethora of new use cases.

5.4.4. Real-Time Data Processing Frameworks

Tools that implement real-time data processing frameworks are at the forefront of data engineering. Increase in speed of data flow and the necessity to react to changes immediately are fueling the rapid evolution of innovation in data processing technologies. Companies are investing heavily in solutions that process data upon arrival and offer analytics quickly, both of the orderly batch-type and real-time type. This

growing need for speed has led to the rise of several technologies that provide extremely fast data processing that is not restricted to dead-time batch windows. Data is no longer kept in a staging area only to be transferred to storage or analytics solutions. Business model architectures have become event-driven.

Real time streaming processing can be defined as executing continuous programs on an input data stream to produce a continuous output data stream. It is continuous modification of its data. The input data stream is a sequence of a finite number of tuples. Each tuple is associated with a time stamp, integer time, or time associated with the order of arrival. The output data stream produced by a process is a sequence of tuples. Each output tuple with a time stamp represents some internal state of a processing system at the corresponding time point. The processing system works continuously over the input data stream to produce output tuples. The continuous processing can also be stateful, where the output at any moment is dependent on both the input sequence and the previous output sequences. From the point of view of data engineering, a continuous data stream is usually associated with short-duration, bursty events. Examples of event streams include sensor data from mobile devices, stock prices in a stock market, network logs of connections or transactions, GPS information from fleet tracking, user page requests on a web server.

5.5. Data Engineering in Healthcare

In the healthcare industry, data is produced constantly, via both the everyday work done by doctors and researchers and the newer tools such as wearable technology. This data, when paired with the right data engineering tools and solutions, can yield crucial insights and be invaluable in the efforts to safeguard and improve the health and well-being of both individuals and the population as a whole, as well as in innovation and cost control.

Healthcare data is, however, notoriously messy and complex. It comes in a wide variety of types and formats, from structured clinical data such as lab test results and bill codes to unstructured social determinant data such as patient discussions in social media and qualitative patient mobility data from mobility tracking apps. Healthcare data on clinical outcomes can blend and link just about any kind of structured and unstructured data imaginable, such as combining unstructured patient-generated data from social media posts with structured biometric parameters from mobile health apps, electronic health records and clinical codes, environmental data such as mobile GPS data, and structured patient characteristics such as age and proximate and cumulative social determinant information. Such complexity renders data engineering for the healthcare industry all the more difficult; it cannot be done naively with ad-hoc algorithms or a few lines of code. Rather, it requires robust, efficient systems for data ingestion, extraction, processing, linking, and aggregation. In fact, it is not uncommon in healthcare research for a data

engineering-intensive effort to take a full year before any insights are even possible to be gathered, much less then acted on.

5.5.1. Patient Data Management

Data engineering plays a pivotal role in enhancing patient data management and EHR systems by improving issues of data integrity, standardization, and accessibility during real-time interactions. In EHR data integration, a data administrator strives to combine data from different and diverse computerized systems into a single standard repository that makes it easy for clinicians and patients to find up-to-date information when and where they need it. Robust data engineering practices can also help increase data interoperability between electronic health records systems and other hospital systems. Armed with better patient data management capabilities, medical practitioners can better offer personalized patient care services and reduce human errors. Following are some of the data engineering tasks for improving patient data management supporting use cases.

Master Patient Index for HBIM: A patient management module in hospitals enables healthcare administrators to manage patient identification/ admission/ discharge/ transfer processes across multiple departments in a hospital. MPI focuses on administrative patient data management, such as generating and storing unique patient identifiers for all patients registered in the hospital, and tracking all visits to the hospital. The newly admitted patients should have unique identifiers either generated from the MPI or escorted by the staff to make sure they don't overlap with existing patients. Proper data engineering practices in hospitals can help reduce duplicate/misidentified patients in the MPI, which are a major cause of errors in inpatient service delivery and financial management.

5.5.2. Predictive Analytics

Predictive analytics is one of the areas where the use of data science in healthcare may have the largest impact. Patients respond differently to treatments, important symptoms in one patient may be negligible in others, and similar symptoms can have different underlying causes. It is necessary to analyze large amounts of data related to similar conditions, treatments, and effects in order to predict the expected outcome from a cure, understand a person's disease's particularities, and prepare a personalized disease management plan for the patient. In this context, predictive analytics can be described as a class of data analytics that uses algorithmic technique with unquantified probability to find patterns associated with conditions or treatments in data from an affected patient or group of patients that have undergone different outcomes. These identified patterns

may be used to prepare a decision tree whose leaves are the predicted outcomes for the particular patient.

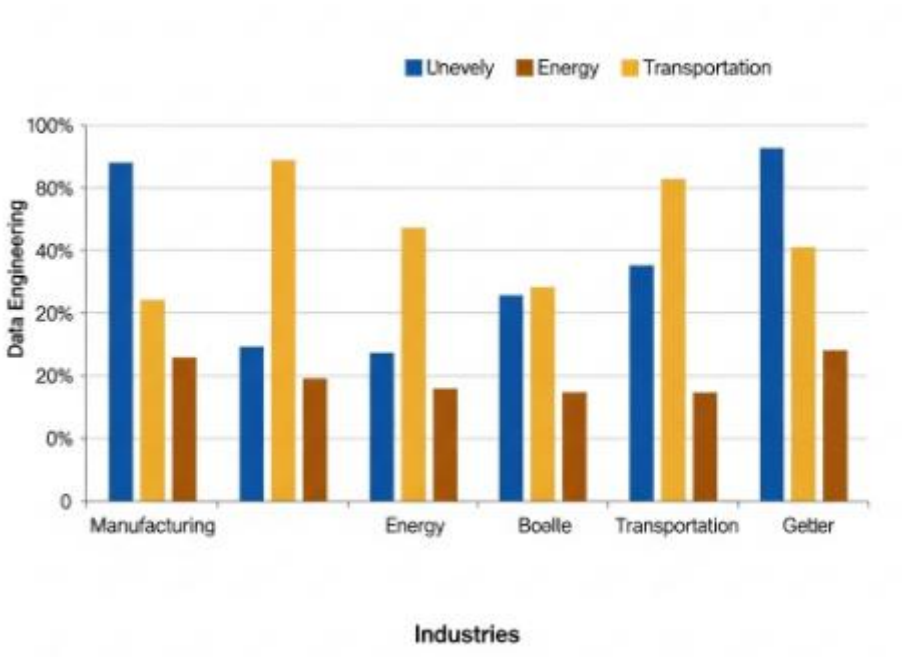

Fig : Data Engineering Across Industries: Enabling Scalable, Real-Time, and Insightful Solutions.

Predictive analytics requires data engineering to prepare the data that will be analyzed. The information collected from disparate sources should first be integrated and then worked on to eliminate noise, variations, and missing records. At the same time, feature selection and engineering techniques may need to be used in order to select or derive the right input features. This is even more critical in predictive models with a limited number of related records that aim to predict rare adverse outcomes. Finally, validation of the models used in predictive analytics is difficult, as historically the data related to cases that diverge from the expected outcome is scarce. Because of that, sensitivity and specificity of different predictive models are the metrics usually employed in healthcare, as the models are not usually employed for the very end of the decision tree.

5.6. Data Engineering in Finance

Data engineering in finance has often been an underappreciated field. It is however crucial in building systems that not only have high performance in terms of speed and volume but also have to be validated independently, with clear reporting of results. Data

engineering in finance focuses on two main areas. One is the transaction-based financial ecosystem. Payment service providers focus on building fraud detection pipelines that analyze transactions in real time. Nearly every online purchase and financial transaction is accompanied by a set of features such as the user identifier, geolocation, merchant identifier, and transaction amount. These determine the likelihood that a transaction is fraudulent. The fraud detection pipelines are often implemented as supervised learning models that are trained to assess if the target is usually related to observed transactions. This is continuously refined using active learning pipelines. The data engineering effort centers around 5 important points.

One is that validation against model errors has to be done at several levels. Second, the monitoring of model accuracy has to be implemented at every production run. Third, a set of scrapers are needed to check model accuracy on a real-time basis. Fourth, data balancing on classes is often required to minimize misclassification on important loss-inducing classes. Fifth, due to the cross-correlation between various user characteristics, various embeddings have to be built to capture latent user behavior correlations.

The other is the macroeconomic ecosystem. Investment banking and corporate finance focus on building economic forecasting models that analyze a set of economic indicators. Key economic indicators determine the health of an economy and the likelihood of a recession. The indicative videos are presented as time series that capture economic data such as employment numbers, GDP growth, and monetary policy, among others. The economic forecasting models are implemented as static red state transition models. These detect structural shifts in the underlying regimes when the state observed switching function transits from the expected regime corresponding to the chosen state variable values. The data engineering effort centers mostly on data consistency and augmentation across different economic cycles.

5.6.1. Fraud Detection Systems

Fraud detection systems aim to distinguish fraudulent events from legitimate ones based on historical transactions already classified by experts. However, the efficiency of these models suffers from many problems, including imbalanced datasets, noisy feature selection, high variance, information obsolescence, the adversarial nature of fraud makers, and dynamic patterns of fraud activities. Imbalanced classification learning aims to generate an efficient model, given the highly skewed distribution of transactions. Supervised machine learning-based fraud detection models rely on a labeled dataset representing both fraudulent and legitimate transactions. Normally, deep-neural-network-based solutions need a large number of samples for successful modeling, which is not the case in the fraud detection domain.

Imbalanced learning and meta-learning methods can be used to circumvent the data scarcity issues during model training. Moreover, label-switching issues affect certain datasets in the historical labeling phase. Reinforcement learning can take the place of supervised settings and tackle this problem. Historical data preprocessing is prone to errors and noise, since it is created from human decisions. Noisy labels lead to poor model predictions. Semi-supervised models with deep-neural-network architectures can improve model performance in noisy situations, as they learn from both labeled and unlabeled data. Limited frequency of fraud events in the real world causes overfitting and hampers the efficiency of fraud detection systems. Using ensemble approaches is common to enhance model performance and diminish overfitting, owing to the high variance of imbalanced classification tasks.

As fraud patterns dynamically change over time, model performance may degrade after a while and may require model retraining. Data virtualization and synthetic data generation from private variables have emerged as a common solution to mitigate the impact of obsolete models. Data virtualization can also lessen false negatives. To address the adversarial nature of fraud, deep reinforcement learning can be an effective solution when the agent is trained to maximize the cost of possible future fraud, which gradually minimizes the cost of past nonaction. Federated learning over synthetic data has also gained popularity.

5.6.2. Risk Management Models

The finance industry is heavily regulated and oversight bodies requiring accurate knowledge in the forefront of financial firms as they are part of the pillar that supports every economy. Risk calculations can center around credit risk, market risk, liquidity risk, operations risk, forensics risk, and/or treasury risk. Every year tons of data are accumulated by global institutions, and in some instances, sharing data for risk assessment and/or prediction purposes can be carried out. Data can be provided more efficiently in a centralized model using a private blockchain, however central banks may not have been up to speed with private blockchain construction.

Many risk assessment models are microeconomic use cases such as capital requirement decisions and pricing decisions. These tend to be forward-looking models. But as these have been an area of considerable research and development, new models tend to be cross-sectional but with these having superior predictability and calibration power. These are more reactive and can rely on deeper use of banking data. Big data models are a long-known fact in the marketing domain, but have only begun to be penetrated in the finance domain more seriously in recent years. While both micro- and macro-economic model use cases allow for cross-sectional data analytics, usually to the extent available,

the micro-economic use cases are, by nature, more granular and tend to have a shorter facing horizon.

As a sign of cautious optimism during economic highs, regulators allow continued operation of more simple models tracking only capital positions. However, as a sign of pessimism during economic lows, the models monitoring the risk of liquidity sink or market collapse are mandated to be more complex. But these are data hungry as they tend to be more forward-looking with poorer reliability, sensitivity, and/or calibration. These lead to more complex data engineering structures as automation of procedures are available.

5.7. Conclusion

Cloud-based solutions have been the technology of choice over the last decade for providing scalable applications on top of highly available infrastructure. Cloud providers have built services for authentication, resource provisioning, message queuing, data storage, etc. while letting users and businesses worry primarily about scaling their applications, and not their infrastructure. In this chapter we discuss how these existing services can be leveraged by data engineers to provide real-time data pipelines that leverage cloud services to increase throughput and decrease latencies while requiring minimal engineering effort. We believe that the open-source community, along with cloud solution providers will be crucial to accelerating rapid progress in this field in the years to come. Cloud solutions abstract away a lot of the tedious engineering that data engineers need to commonly implement—like scheduling and monitoring pipelines, ensuring high availability of data ingestion, replication of data in multiple locations, etc.—to let them focus on the most critical part of any data pipeline, the data transformation. Templates provided by cloud services not only facilitate faster deployment and experimentation by data engineers, but also enable small startups to easily build and iterate on data-centric products. We believe that a combination of data-centric startups, along with availability of open-source tools for automatic and iterative model training and development will result in a new class of intelligent, data-centric applications that seamlessly integrate machine learning and data processing.

5.7.1. Future Trends

Data is gaining much attention. New storage technologies are coming to mainstream combined with the explosion of data. This is creating unique challenges for organizations which primarily depend on actionable insights from real-time data. This new class of applications greatly affects the traditional pattern of Designing Data Architectures. Data Engineering is occupying the biggest portion in the data science product development

life cycle, even affecting the hiring pattern in organizations. In the coming years, Data Engineering teams of organizations will be very similar to Software Engineering teams because of the complexities involved in meeting deadline-driven product stories. Once the data is ingested into the Data Lake or Data Warehouse, it will become a Product. Decision makers will start tapping into Data Products to make quick decisions without constantly seeking help from the data engineering teams. Data Quality will be taken seriously and organizations will start establishing measures to periodically audit and monitor the Data Lakes/DW for accuracy. The Data visualization tools will grow in capabilities and will be comparable to BI tools. They will include version control, governance, data cataloging capabilities to allow the business users to play with the Data Products without corrupting the stored product. Collaboration in Data Engineering will help organizations. Using open-source tools to meet custom Data engineering requirements combined with Managed services in the Cloud will boost productivity and allow Data Engineers to focus on building Data Pipelines that deliver business value. With the advent of automation around Data Engineering, some of the entry-level work done primarily by fresh data engineering university recruits will be at stake. Data Engineering and Data Science areas of work in the organizations will have a much closer collaboration.

References

Napoli, P. M. (2011). Audience evolution: New technologies and the transformation of media audiences. Columbia University Press.

Webster, J. G., Phalen, P. F., & Lichty, L. W. (2013). Ratings analysis: The theory and practice of audience research (4th ed.). Routledge.

Taneja, H., Webster, J. G., Malthouse, E. C., & Ksiazek, T. B. (2012). Media consumption across platforms: Identifying user-defined repertoires. New Media & Society, 14(6), 951–968.

Lobato, R. (2019). Netflix nations: The geography of digital distribution. NYU Press.

Lotz, A. D. (2017). Portals: A treatise on Internet-distributed television. University of Michigan Press.