NextGen Scientific Publication
American International Journal of Computer Science and Technology

Volume 7 Issue 4, Pg. No. 129-141, AIJCST-V7I4P112, 2025
https://doi.org/10.63282/3117-5481/AIJCST-V7I4P112

Original Article

Data-Centric AI in the Era of Large Volumes: Improving Model Outcomes through Data Quality Engineering

***Sivadeep Katangoori[1], Anudeep Katangoori[2]**

[1,2]*Individual Contributor, USA.*

Abstract:

Nowadays, the whole AI system' success is determined not only by algorithms but also, and even more importantly, by the data quality that the systems use. The rise in the amount, speed, and diversity of data has reached the point where it is no longer possible to rely only on a model-centric approach. This article tracks the extent of the growth of data-centric AI which addresses the improvement of the data itself as the main lever for better model outcomes. At the core of this transformation lies data quality engineering—a highly disciplined and oftentimes neglected profession that goes way beyond performing the research on datasets and running formal verification tests, but it is basically the very essence of completeness, consistency, and context appropriateness of the datasets in question prior to their being used in any modeling work. In high-volume environments that are prone to falling, minor inconsistencies may become a big headache and using the preventative data hygiene approach in this context becomes indispensable. The paper looks into ways of how implementing data quality pipelines, anomaly detection, labeling integrity checks, and feedback loops into AI workflows not only makes models more reliable but also minimizes retraining costs and shortens deployment cycles. The embedding of data quality engineering into the development lifecycle has facilitated organizations™ transition from being reactive to the failures to a culture of continuous improvement driven by trusted data. This transformation allows teams to focus more on innovation and less on debugging. In the final analysis, this mind shift paper argues that in the world of such a massive increase of data, a data-centric approach—rooted in engineering discipline—is the most effective way of achieving scalable and robust AI.

Keywords:

Data-Centric AI, Data Quality Engineering, Big Data, AI Model Performance, Data Labeling, Data Cleaning, Feature Engineering, Data Governance, Model Retraining, Data Validation, Data Pipelines, Machine Learning, AI Reliability.

Article History:

Received: 07.06.2025

Revised: 10.07.2025

Accepted: 21.07.2025

Published: 01.08.2025

1. Introduction

AI has learned over the past few years that improved models can't make bad data better. For a long time, AI got better because algorithms, computers, and model structures all got better. The way of doing things that focused on the model won. It had transformers-based tools and frameworks for deep learning. But when businesses started using AI a lot in other areas, it became evident that this way of thinking was very wrong. The usefulness of AI systems, especially in the real world, started to level off. The models weren't too hard to understand; the problem was that the data they were provided was inaccurate, noisy, or not a big enough sample. A lot has changed since this new information came out. Now, data-centric AI is becoming more popular, which focuses on enhancing data instead of models.

This change isn't only a thought; it's also useful. Companies are collecting and analyzing data on a scale that has never been seen before, which is causing new concerns. A lot of people think that big datasets are helpful, yet their scale makes them challenging to work with in a lot of ways. A model might not operate as well as it should if it has errors in labeling, missing values, bias, redundancy, or not enough context coverage. Even slight faults in quality can have a major impact on the whole system as a dataset goes from a few dozen records to millions. We also need full answers for pipelines, storage, governance, and validation because these kinds of datasets are becoming more and more difficult to work with. Self-driving cars, fraud detection, and natural language processing are all examples of things that deal with a lot of data. A little bit of mislabeling may put the whole enterprise in jeopardy.

It is more crucial than ever to do things the right way. More and more, AI is being employed in the public sector, healthcare, banking, and the courts. The choices it makes have real-world repercussions. Data needs to be safe, fair, and trustworthy in addition to working well. Negative data not only gets predictions wrong, but it also keeps negative biases running, makes businesses do the wrong things, and leads to false diagnoses. Getting good data is hard for the backend, and it's also a big aspect of employing AI in a smart and ethical way.

In this case, it is very necessary to learn about data quality engineering. AI systems today shouldn't merely prepare data once. Instead, kids should think of it as something that happens all the time. Automated data profiling, anomaly detection, data labeling checks, and feedback-driven refinement cycles can help teams keep an eye on and enhance the quality of their data throughout the model's life. Making sure that data is of good quality is very important for the growth of AI. It helps businesses generate better models, lower the risks of running their businesses, speed up the process of making changes, and make systems more open.

Figure 1. Data-Centric in the Era of Large Volumes

This paper talks about how strict data quality engineering is making data-centric AI the new standard for designing AI systems that can change and are reliable. We want to help AI experts, engineers, and people who make decisions deal with the problems that come up when they have to work with a lot of essential data. These problems span a lot of ground, from the limits of reasoning based on models to the specific things that could be done to make data more reliable.

2. Foundations of Data-Centric AI

2.1. Data-Centric vs. Model-Centric Paradigms

The primary difference between model-centric and data-centric AI is what they want to improve. When developers use a model-centric approach, they try to improve algorithms by changing the topologies and hyperparameters and adding new layers or optimizers. These modifications don't do anything to help. People see the data as a fixed input after it has been collected and organized. People usually change the algorithms to correct flaws with how the model operates.

Data-centric AI, on the other hand, believes that the dataset is the most important element that determines how effectively a model works. Now, instead of changing the model all the time, the goal is to improve the data's quality, organization, and representativeness. This means changing wrong labels, adding more variety to categories that don't have enough representation, getting rid of duplicates, and making sure that the data accurately shows how things are distributed out in the real world. This is a simple but crucial point: a simpler model that is trained on cleaner, more relevant data frequently works better than a more complicated model that is trained on data that isn't appropriate or doesn't fit.

This way of thinking about powerful models doesn't make them less valuable; it only changes how we think about them. When there isn't a lot of labeled data, edge cases have a big effect on the findings, or data collection is still going on, making the data better, which can aid in the short and long term, even with simple model topologies.

2.2. Historical Evolution of Data Practices in AI

The first several decades of AI development had a very strong connection with the computational and algorithmic limitations. During the 1980s and 1990s, not much data was available and researchers concentrated on symbolic reasoning and rule-based systems. After machine learning became popular in the 2000s, the focus was turned to the likes of support vector machines and ensemble methods.

Deep learning suddenly changed the picture in the 2010s. Models gained new strength through datasets on a large scale like ImageNet, OpenSubtitles, and Common Crawl. But this also created a kind of defect: with the increasing complexity of the models, less and less attention was given to the data they were based on.

Broader acknowledgement of the importance of data quality has only come about recently, notably with the democratization of AI and its rollout in sensitive areas. Organizations have come to the realization that in order to have AI at scale, they need sustainable and trustworthy data engineering practices.

2.3. Real-World Implications of Poor Data Quality

Poor data quality is not only a source of failure in the technical part of a system, but it also brings concrete negative impacts. In the medical sector, both mistakes in the labeling of radiology images and training a model with data from a non-diverse population are factors that might lead to wrong medical diagnoses for those groups that are less represented. In the financial sphere, incorrect transaction datasets can cause fraud detection systems to raise false alarms. In such cases, users may be locked out of their accounts or the fraudulent activities may be missed.

Malfunctioning data, even in simpler cases such as recommendation engines and chatbots, will lead to lower user satisfaction. Users will in turn interact less with the platform and will have less or even no trust in the company. The expenses for the reestablishment of models, re-annotation of datasets, and managing the users' reactions can be very high.

Data-centric AI is aware of the danger and it does not shy from presenting it at the forefront. Data-centric AI supports the use of thorough data profiling, ethical data sourcing, clearly visible data documentation, and feedback loops that not only depend on model behavior but also user interactions. The latter is not only an evolution of techniques or tools but a new cultural AI team/workflow paradigm.

3. Principles of Data Quality Engineering

With AI systems progressively being integrated into essential business functions and public services, the data that drives these systems has become very crucial. Data Quality Engineering (DQE) offers a formalized framework to verify that data sets are not only clean and usable but also meaningful in the given context and in line with the goals of the models that use them. This practice centers around a set of core attributes: accuracy, completeness, consistency, timeliness, and validity. In combination, these aspects determine the quality, trustworthiness, and usefulness of AI training and inference data.

3.1. Accuracy

Accuracy is the measure of how close the data is to the actual values in the real world. In the case of supervised learning, this means that the data should have correct labels, be measured precisely, and be a true representation of the process. The inaccurate data might be the result of the failure of the sensors, the mistake made by the person during labeling, or the defective extraction process.

Increasing accuracy is a part of the effort that requires validation of inputs on a regular basis, checking the labels by the hands of a person or partly automated, comparing with the source systems, and the use of annotation confidence scores. Besides this, it is helpful to have the instruments that highlight the irregularity of the entries at an early stage so that the team members can locate the problematic records and correct them before they become a source of error for the models.

3.2. Completeness

Completeness depicts whether or not all the necessary information is available in the dataset. The absence of values in the metadata fields, key attributes, or labels can lower the working efficiency of the model, and if the missingness is systematic (e.g., certain demographic groups being underrepresented), the results can also be distorted. Quality engineers usually set up the limits of how much missing data is acceptable, they also proceed with the filling of gaps using appropriate strategies when this is the case, and they mark the incomplete records. Even tiny holes in data in areas like healthcare or finance, where the work is critical, can lead large problems; thus, the need for a watchful eye always on the lookout.

3.3. Consistency

Consistency guarantees that information follows the same formatting and meaning for all records and datasets. It also involves the use of uniform naming conventions, standard units of measurement, and the use of harmonized schema definitions. In large data pipelines that have different input sources, inconsistency is the most frequent problem—dates might be in different formats, categories might be labeled in a slightly different way ("NY" vs. "New York"), and even the same concepts might have different names. First of all, consistency can be a major headache and require centrally managed data schemas, very thoroughly documented ontologies, and transformation logic that would actually convert the data into a standard before further processing.

3.4. Timeliness

Timeliness refer to the fact that data is up-to-date and relevant. In fast-moving situations, for example, stock trading, logistics, or cybersecurity, data that is even a few minutes old might make models turn ineffective or wrong. A timely data is the one that allows AI systems to make decisions based on the current reality. Data quality engineering commits to timeliness by means of the real-time ingestion pipelines, periodic batch updates, and freshness indicators showing how recent the data is. This aspect also covers latency awareness, i.e., giving assurance that models are not depending on old data due to a delay in the downstream data systems.

3.5. Validity

Validity is the extent to which data adhere to the expected rules and constraints. Such rules may be permissible ranges of values for numeric fields (for instance, temperature readings between -50°C and 60°C), categorical constraints (allowed country codes), and even relational logic (a child's birthdate cannot be before the parent's). Most times, invalid data can be a sign of either data collection processes going wrong or the presence of some attackers. Validation rules, unit tests for the data pipelines, as well as schema enforcement frameworks such as Great Expectations or Deequ, facilitate the maintenance of data validity throughout the data lifecycle. These instruments form an essential part in establishing "contractual" expectations between the upstream data producers and the downstream consumers.

3.6. Role of Domain Knowledge

Quality data cannot be produced just by following some rules. Moreover, a great deal of it depends on specialized knowledge of the domain. This is because what may be a simple, obvious true fact, or even a suspicious issue in a given particular area, may often be a set of intricacies that only a few domain experts can understand. As a result, embedding domain knowledge into data quality engineering will mean the involvement of experts for annotation and audits of the said domain. It will also mean that contextual checks that are more advanced than just syntax are incorporated.

3.7. Importance of Data Documentation and Standards

Quality cannot persist without documentation. This also is in line with data documentation such as data dictionaries, schema definitions, annotation guidelines, and lineage reports that act as the main support structure for the understanding, sharing, and storing of high-quality datasets. Teams, as a matter of fact, can implement standards such as Datasheets for Datasets, Model Cards, or Data Statements, which make data origin, intended use, potential biases, and known limitations more transparent. Moreover, a commitment to such standards allows better collaboration across teams, easier compliance, and greater trust in AI systems within the organization.

4. Key Techniques in Data Quality Engineering

Data quality assurance is definitely not a single event—it is a recurring, interactive process in which various supportive techniques are used. The successful data quality management essentially involves the use of automation, the input of the human mind, statistical information, and domain knowledge. The part here is the most significant step to the great performance of AI systems, as it ushers in the basic yet reliable, scalable, and meaningful data preprocessing techniques.

4.1. Data Cleaning & Preprocessing

The primary way of safeguarding against a model's poor performance is thorough data cleaning and preprocessing. This procedure means treating the data in a very methodical way where you not only identify but also fix (or get rid of) those parts that are inaccurate, incomplete, or irrelevant. It is virtually impossible to have perfectly clean data.

Noise are data points that are irrelevant or contain errors and therefore they distort the statistical patterns in the data. For instance, random characters that are typed in when the text dataset is being created or zero values that are incorrectly set in financial transactions are just a few of the many things that can make existing statistical models get perplexed.

Missing values can be just as problematic as noise and they are a frequent occurrence. They may come about due to human error, issues in the transmission, or partially filled-out questionnaires. Here are some methods to deal with missing data:

- Deletion, in case the missing rate is low and the missingness is random.
- Imputation, using techniques such as mean substitution, K-nearest neighbors (KNN), or regression models.
- Defaults from the domain, in which a few fields are able to be deduced from business rules or with the help of experts.

Outliers are unusually high or low values that are very different from the expected range of values in question. They may be a sign of an anomaly or the wrong data entry. Depending on the situation, they can be removed or managed employing a method such as winsorization, scaling, or logarithmic compression.

4.2. Data Labeling and Annotation at Scale

Labeling data of high quality is the root that supports supervised learning. Nevertheless, the creation of labeled datasets is a major issue that is extremely laborious and prone to errors, especially when trying to scale it up. Even the most developed models will pick up the incorrect patterns if the labels are not accurately done.

Human-in-the-loop (HITL) labeling is one method to achieve the goal. Experts, usually helped by simplified user interfaces, manually annotate the data. Although this guarantees the accuracy of the data, it is costly and time-consuming. To be able to scale more efficiently, a combination of HITL and label review could be used. The crowd-sourced labels are validated by the trusted experts in such a scenario.

Weak supervision, though, is a more scalable solution over the long term. It depends on the use of imperfect but scalable labeling strategies, such as the use of heuristic rules, pattern matching, external knowledge bases, or even pre-trained models, for the quick assignment of noisy labels. Then these labels are denoised by some probabilistic models like Snorkel or generative label models, which help in the rapid expansion of the training datasets.

Active learning turns even more the labeling effort to the best by selecting the most "informative" examples, i.e., those about which the present model is most uncertain, for the human review. This, thus, drastically decreases the number of examples that require manual annotation and at the same time, it benefits the maximum model.

When these strategies are combined, it is possible to create a feedback loop through which not only the quantity of data but also the quality and source as well improve over time as the model learns from more and more relevant and reliable examples.

4.3. Feature Engineering & Signal Extraction

While it is believed that more data is always better, AI systems are not completely data-dependent; rather, they are largely dependent on quality features. Feature engineering is a method that allows machines to recognize the same patterns in raw data that humans might not be able to see.

For example, lag variables and rolling averages in time-series forecasting might show the temporal trend that raw timestamps cannot reveal. In NLP, as extensible as Word2Vec, TF-IDF, or contextual embeddings from transformer models can be, they can still do little without a simple text input to extract the semantic information from. In structured data, predictive performance of a model can be significantly improved with simple transformations like binning, scaling, or combining fields (e.g., "age" and "employment status" into a "risk profile").

The essence, however, is in signal extraction—identifying the relevant information in a sea of noise. Good features also make models more interpretable, have a lesser tendency to overfitting, and, as a result, most of the time they do not require the use of very complex architectures.

Some feature engineering work can be done by machine with the help of AutoML tools. The knowledge of the domain, however, is still very important for you to be able to come up with features that make sense and then prove them right especially in sectors like healthcare, law, and supply chain, where the contextual nuances are the deciding factor.

5. Managing Data at Scale

With the integration of AI technologies into everyday business processes and services that operate in real time, handling large-scale data has become the main issue, not only from a technical perspective but also from an operational one. As the amount, speed, and variety of data continue to grow exponentially, companies must put in place a scalable infrastructure, dependable validation measures, and automated MLOps workflows to maintain the quality of the data over time. Lack of such functionalities will result in the rapid decline of the efficiency of well-designed models, thus leading to trust and performance deficits.

5.1. Infrastructure: Data Lakes, Warehouses, and Versioning

If you want to have proper data management that can be scaled up, you will first need a properly working and strong infrastructure. Most of the big companies at present are relying on a combination of data lakes and data warehouses to store and organize their data in different ways.

- Data lakes are designed to be more adaptable, which means they allow users to store raw, semi-structured, or unstructured data without putting any limits on the volume of data. The implementation of AI in audio, video, sensor data, and logs has made the training pipeline more diversified, and data lakes can be a viable solution to cope with this kind of data.
- Data warehouses usually provide high performance when it comes to querying and are structured. In general, they are great for business intelligence, creating metrics reports, and delivering ML models with the necessary structured features for training.

In order to keep these environments from falling into disarray, the data versioning aspect is very important. Technologies such as DVC (Data Version Control) or LakeFS allow for Git-like functionalities for datasets, which means groups of people can monitor the changes made, revert back to previous instances, and carry out identical experiments. Versioning is, furthermore, a step in preventing the occurrence of one of the most frequent model failure reasons—schema drift that happens silently or changes in the data sources that result in models gradually becoming less effective.

5.2. Scalable Validation Mechanisms

Once data pipelines are large enough, manual quality checks are no longer an option. The organizations require validation mechanisms that are automated, scalable, and can work without causing latency or bottlenecks with data volumes in terabytes and records in thousands per second.

Such mechanisms generally consist of:

- Statistical profiling: The automated tools produce the statistical data of the features (mean, distribution, cardinality, and null values) and, based on this, identify the anomalies or unexpected patterns.
- Schema enforcement: Systems of data validation check the incoming data against the given expected schemas. They report the results of missing fields, data of wrong types, or unexpected zero values.
- Drift detection: The technique of comparing the distributions of live data with the historical baselines enables the systems to find the distributional drift that is a common occurrence of retraining sources.

Great Expectations, TFDV, and WhyLogs are some of the frameworks that facilitate these validations without losing scalability. One can deploy them in ETL pipelines, training pipelines, or production workflows, thus providing a real-time view of whether the input data is trustworthy or not.

Moreover, to enhance the scalability even more, the validation chores can be assigned in distributed processing systems, such as Apache Spark, Databricks, or Flink. It makes it possible to do the checks on the large datasets without losing the speed.

5.3. Automation Using AI and MLOps Practices

Strong MLOps (Machine Learning Operations) practices are a big factor in successful data quality at scale. MLOps basically takes the DevOps ideas and applies them in the machine learning cycle, thus making the processes of training, deploying, monitoring, and retraining automated.

Regarding data quality, MLOps provides automation in different critical areas such as

- Data ingestion and preprocessing pipelines are created by Apache Airflow, Kubeflow Pipelines, or Dagster, which enables teams to set workflows that are modular and can be reused in the process of changing and validating data.
- Automated retraining triggers: In case of data drift or concept drift, models can be automatically retrained with the most recent validated datasets by pipelines.
- CI/CD for data and models: Simultaneously with software code, any modification in the dataset or in the data pipeline should be passed through the automated testing along with staging environments for validation and deployment.

Artificial Intelligence (AI) solutions are being progressively incorporated in the cycle described above. For instance, an anomaly detection model can observe the data and in case of an unexpected pattern, alerts can be generated at the same time generative models or LLMs can provide help in semi-automated labeling and documentation. Companies can lessen the operational pressure on their data engineers and at the same time have a more viable, scalable AI development process by automating those tasks that are repeatable and incorporating quality checks in each stage.

6. The Impact of Data Quality on Model Outcomes

In many cases, the significance of superior data is referred to in theory, but its effect on actual model performance is increasingly recognized through numerous empirical studies, benchmarks, and deployment situations. Practically, the main takeaway that is common across all industries and applications is the following: enhancing data quality may lead to better results than upgrading models or give results at par with them.

6.1. Empirical Studies Demonstrating Performance Improvements

Multiple experiments, research studies and the internal experiments of companies like Google, Microsoft and Airbnb have consistently shown the significant influence on the performance of AI systems when the data is clean, well-labeled and representative.

The example that comes quite close to Andrew Ng's Landing AI is the work on small datasets for industrial visual inspection, which was used to create defect detection models. To their surprise, the improvement of label quality and coverage and the re-labeling of just 10–15% of the data resulted in increases of F1 scores by 20–40%, and this happened without any changes to the model. Analogous to that, a research study at Google Research found that for certain NLP tasks, the improvements in performance were more significant from refinements in annotation quality and the clarity of class labels than from increases in model size. Moreover, in the case of industry deployments, the teams often acknowledge that the resources devoted to data audits, the setting up of annotation guidelines and the creation of validation pipelines not only lead to better generalization but also fewer failures of unexpected edge cases and an easier transition from prototype to production.

6.2. Data-Centric Benchmarks

Several benchmarking initiatives have emerged to measure the effects of data quality on model outcomes after understanding the growing relevance of data-centric development.

- DAWNBench is a benchmark created by Stanford that was initially very much focused on the time-to-accuracy for the training of the deep learning models. But later DAWNBench evaluations revealed that data preprocessing and augmentation pipelines

were often the main contributors to performance improvements—much more than the model architecture—especially when efficient datasets and curated subsets were used.

- Feature quality, data normalization, and interaction modeling have been the focus of experiments enabled by the Criteo Terabyte Click Logs dataset, which is extensively used in advertising ML research.
- WILDS is a collection of datasets and a benchmark aimed at evaluating the robustness of models subject to distribution shifts. The implicit emphasis on data quality is conveyed through the benchmark, which addresses the issue of performance degradation of "clean but narrow" models under real-world conditions; hence, the need for diverse, well-balanced, and representative training data.

These benchmarks demonstrate that model performance cannot be done to the fullest extent or even facilitated without factoring in the quality and structure of the underlying data.

7. Integrating Data Quality Engineering in AI Workflows

The creation and upkeep of efficient AI systems require that data quality be an integral part of the origination, application, and control process. It implies going further than data cleaning as a single event and integrating data quality engineering (DQE) activities directly into the machine learning (ML) lifecycle. Here, we consider the organizational ways of incorporating quality checks in CI/CD pipelines, setting up feedback mechanisms for continuous improvement and creating a cooperative relationship between data engineers and ML practitioners.

7.1. Embedding Quality Checks in CI/CD Pipelines

Modern software engineering has undergone significant changes, and with the help of CI/CD (Continuous Integration/Continuous Deployment) pipelines, the process of testing, building, and deployment has been automated. The same idea is being utilized in ML workflows—MLOps is the term used to refer to this new concept.

Some of the main ways of implementing quality checks in a system are the following:

- Schema validation: When data is either newly ingested or changed, schema validation guarantees that field types, value ranges, and mandatory fields agree with expectations. Great Expectations, TFDV (TensorFlow Data Validation), and Deequ are some of the tools that can automatically carry out these validations.
- Distribution monitoring: New data is checked against the training distributions, thus making it possible to find data drift or other abnormal situations. Instance, when "transaction amount" is the feature and values that have never been seen before in the range are suddenly found, an alert will be activated in the system. This will stop the process of silent degradation.
- Automated data profiling: Data samples are profiled for completeness, nulls, duplicates, and changes in cardinality at every CI stage. These metrics can be stored and drawn for different periods, which can be used to set standards of what "qualified" data should be.
- Testing datasets as assets: Similarly to application code, before datasets are released, they are subjected to pre-deployment testing. In summary, it involves validating row-level constraints, checking for label leakage, and making sure that sensitive data is masked or encrypted.

Teams can find errors in data by implementing these checks within CI/CD processes even before the data is used to train models. Such precautions not only improve the reliability of models but also make the transition from development to production more efficient.

7.2. Feedback Loops for Continuous Improvement

Data is always evolving—users behave differently, new edge cases pop up, and models can come across patterns that they haven't seen before. Over time, if there aren't any continuous feedback loops, the quality of data will almost certainly decrease. Feedback loops allow AI systems to self-correct and grow by bringing the real-world signals back into the training and labeling process.

Some of the effective feedback mechanisms are

- Error analysis: Upon deployment, mispredictions of the model or high-confidence failures can be inspected in order to comprehend whether they were caused by poor or outdated data. These examples are very powerful for retraining and improving data coverage.

- User signals: In the case of chatbots, recommendation engines, or fraud detection, user actions (e.g., corrections, complaints, engagement drop-offs) can be a form of implicit feedback on data/model performance. The signals can help to identify the parts of the data that need more attention.
- Retraining triggers: Retraining pipelines can be made to function automatically when certain limits, such as data drift, concept drift, or model accuracy thresholds, are surpassed. Most importantly, retraining should bring in better data—new labels, enhanced features, and even more samples from rare classes.
- Annotation refinement: Feedback loops might be a reason for changes in labeling. For example, if human reviewers keep disagreeing with the original labels, the annotation guidelines can be modified to provide more examples along with ambiguities.

Continuous improvement changes AI from a non-living deployment into a living system that can adjust to new data surroundings. Additionally, it lessens the technical debt by preemptively dealing with issues before they become large performance failures.

7.3. Collaboration between Data Engineers and ML Teams

Cross-functional collaboration is one of the least acknowledged, but it is still one of the most powerful contributors to high data quality. Data quality engineering is situated at the juncture between data engineering, machine learning, and domain expertise—thus it is necessary that there is coordination among the staff who work in different roles and have been historically separated.

The major ways to encourage such teamwork comprise:

- Shared ownership of data quality: Rather than designating the responsibility solely to the data engineers, organizations need to endorse a practice of shared accountability. ML teams, analysts, and even product managers ought to take part in specifying quality metrics as well as data expectations.
- Unified tooling and observability: The use of common platforms (for example, dashboards, data catalogs, and version control) is one of the easiest and shortest ways to get a good level of transparency among different teams. If every person can see what the state of the data is and how it can influence model outcomes, the collaboration becomes a proactive one instead of being reactive.
- Collaborative debugging and triage: In the case that a model underperforms, the process of troubleshooting should be done by joint root-cause analysis. Is it the feature logic? A bad data batch? A change in user behavior? Cross-team retrospectives engage not only data pipelines but also model design.
- Embedding domain experts: In some very delicate or specialized areas such as healthcare, legal tech, or fintech, subject-matter experts can be in charge of checking the correctness of the labels, reporting (flagging) of the anomalies, and also the contextual accuracy. Their participation reduces the distance between statistical regularities and real-world implications.

By bringing open communication, joint review processes, and collaborative tooling into your daily routines, you can transform DQE from a bottleneck into a reliable and impactful AI catalyst.

8. Case Study: Retail AI Platform Using Data Quality Engineering

8.1. Background and Problem Statement

A leading online retailer working majorly in both North America and Europe decided to apply the latest AI technology to help not only better product discovery but also to increase sales with a personalized recommendation system. So basically, this machine learning engine gathered long-term customer behavioral data, including clickstreams, cart activity, search queries, and purchase history over the web and mobile platforms.

In spite of employing a strong matrix factorization model supported by neural collaborative filtering, the output of the system in production was not up to the expectations. The main symptoms of the problem were that conversion rates for recommended items were stagnant, and customer engagement with the “Recommended for You” section was declining. Our internal evaluation showed that the performance of the offline model was not consistent with the real-world results—a frequent but expensive problem in AI deployments.

The fundamental cause: while the model was technically competent, automatically generated recommendations were absurd or irrelevant, such as suggesting winter coats in the middle of summer, recommending products that you could already have, and failing

most of the time to even recommend accessories or complementary products. The company had reached the stage where it was no longer able to ask the question, is it the fault of the model or is it a data problem? The answer was the data quality team.

8.2. Data Issues Identified

The DQE team thoroughly examined the entire data pipeline, which is the flow of data from the source to the end consumer and each step they took led to discoveries of several key issues:

- Inconsistent Schemas across Sources: The behavior data was collected from different places that include platforms of a website, iOS, Android, and some others that were not mentioned. Data fields like item_id, session_id, and timestamp used to be defined differently in each source; thus, data mismatches were a result.
- Missing Transactions: Purchase confirmations along with cart-add events were absent in certain segments of mobile app users because of the occurrences of API failures in one region. User conversion profiles were skewed by this in addition to low 'turned on' product categories being falsely rewarded.
- Label Noise: The data used to train the recommender model, which indicated product engagement as positive labels, was extracted from "clicks followed by purchase within 24 hours." However, to talk about labeling noise in data, pure bots were found to generate completely automated traffic, and duplicate purchase events were the reason for some of the false positives.

The issues mentioned above were not only some standalone problems but also had a domino effect. Because of the flawed input data, the model's comprehension of user desires was incorrect. What made some products appear to be popular or relevant where they were not, and at the same time actual trends of interest were weakened or even falsely represented?

8.3. Remediation via Data Quality Engineering

In order to solve the problems that were deeply rooted in the system, the group decided to apply a structured data quality engineering plan, which would consist of automated tools, standardization protocols, and MLOps integration.

8.3.1. Schema Standardization

The very first step was the implementation of a schema registry that was centrally managed and that all the sources for data ingestion had to be compatible with. This involved:

- Core field (user_id, product_id, event_type) definitions that were uniform
- Timestamps that were both standardized and normalized with respect timezones
- Platform-specific identifiers that were encoded in a consistent manner

Schemas were version-controlled using LakeFS, and schema evolution rules were established to prevent breaking changes. Downstream pipelines were equipped to auto-detect and flag deviations in real time.

8.3.2. Label Verification and Deduplication

The automated validation rules were included in the redesigned label generation process to facilitate seamless operations. By applying heuristics and bot-detection algorithms, non-human traffic was separated from the rest of the data. Part of the logic for labeling is return purchase data; thus, only purchases that have been kept will be the source of positive labels. Reduced duplicate transactions were achieved through the use of hash-based deduplication for records, while temporal filters were employed to prevent delayed events from causing false correlations.

8.3.3. Real-Time Data Validation

To perform lightweight, real-time statistical profiling, the team implemented WhyLogs. Every batch of behavioral events was validated automatically against:

- New fields or null value increase
- Large changes in feature distributions (e.g., decrease in the number of clicks for a highly used category)
- Unexpected class that are more predominant than usual

Such examinations were integrated into the CI/CD pipeline with the help of Airflow, while any warnings that were generated could be found in the Slack channels, where the data engineering and ML teams monitor them. Customer support and product

managers' input was also used by the system. For instance, if customers complained about the recommendations that were irrelevant, sessions affected were marked and reviewed to find out if there were any data anomalies.

8.4. Outcomes and Improvements

- 18% Improvement of Recommendation Accuracy: Offline A/B tests demonstrated to the team better top-K precision and recall metrics, while live experiments showed an 18% rise in the click-through rate (CTR) for recommended products, thus verifying the offline results.
- 25% Decrease of Model Drift: The team, by introducing real-time validation and feedback, found a significant reduction of 25% in data and concept drift over a three-month rolling window. As a result, the model exhibited more stable behavior and thus fewer emergency retraining cycles were needed.
- Operational Resilience: Apart from the schema violation and data gap alerts, the team was able to detect and fix the problem within a few hours and not wait for days. Therefore, the debugging process was accelerated along with the ML pipeline's uptime that was enhanced considerably and also the stakeholders' trust in the models' outputs increased.
- Enhanced Collaboration: As a result of this work, the relationships between data engineers, ML practitioners, and product owners have improved significantly. The shared dashboards and post-mortem reviews have fostered a culture of joint accountability for the outcomes of the models and not just for the code.
- User Experience Gains: Several user surveys conducted after implementation revealed the platform's recommendation features had greatly improved in terms of user-friendliness. The percentage of users who returned to the site only slightly decreased; however, the basket sizes of those who interacted with recommendations grew.

9. Conclusion

Quality of data has become the determining factor of system performance in AI systems that are deployed in real-time, high-stake, and data-rich environments. AI has gone through a rapid evolution owing to the advancements in algorithms, computing power, and architectures. Still, the industry is at a point where it is not the next bigger model that will bring further improvements but rather better data that will feed the existing ones. One of the major sources of the results of the model's accuracy, fairness and robustness has been found to be the quality, completeness and contextual relevance of the underlying data, according to the research, benchmarks, and real-world deployments.

This realization indicates a major paradigm shift from a model-centric to a data-centric one. The latter puts data at the forefront as the most dynamic, first-class asset—audited, engineered, versioned, and validated with the same rigor as code. The Data Quality Engineering (DQE) ecosystem is a core discipline; its components are the tools, standards, and collaborative practices that bring data fidelity checks directly into AI development and operations. Data-centric investing in AI systems is increasingly popular and is being credited for safer and more effective deployment of intelligent systems, from automated validation pipelines to feedback-driven improvements and cross-functional collaboration.

The future of AI looking beyond is not just about systems that can learn but also about those that can self-assess and self-correct their data environments. When combining the concepts of AI-assisted labeling, anomaly detection, and data governance automation, the result is a model that can easily pinpoint the noisy inputs, bring to the surface the edge cases, and suggest the refinement of the training data. In essence, AI will be the first to help it remove weaker links in its data chain—ushering in more agile, durable, and self-reliant systems.

References

[1] Whang, S. E., Roh, Y., Song, H., & Lee, J. G. (2021). Data collection and quality challenges in deep learning: A data-centric ai perspective. *arXiv preprint arXiv:2112.06409*.

[2] Jarrahi, M. H., Memariani, A., & Guha, S. (2022). The principles of data-centric AI (DCAI). *arXiv preprint arXiv:2211.14611*.

[3] Srigadde, B. R. (2021). When Rounding Up Matters: Working with Decimals in Apex. International Journal of AI, BigData, Computational and Management Studies, 2(1), 122-131. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V2I1P113

[4] Abedjan, Z. (2022). Enabling data-centric AI through data quality management and data literacy. *it-Information Technology*, *64*(1-2), 67-70.

[5] Suryadevara, S. S. K. (2021). AI-Driven Multi-Cloud Orchestration System for Enterprise Digital Experience Delivery. American International Journal of Computer Science and Technology, 3(1), 21-34. https://doi.org/10.63282/3117-5481/AIJCST-V3I1P103

[6] Polyzotis, N., & Zaharia, M. (2021). What can data-centric AI learn from data and ML engineering?. *arXiv preprint arXiv:2112.06439*.

[7] Shiramalla, R., & Guntupalli, B. . (2021). Cost-Effective Softphone Integration in CRM Platforms Using RESTful APIs: A Salesforce Case Study for Voice-to-Text Sales Enablement. International Journal of Emerging Trends in Computer Science and Information Technology, 2(1), 101-114. https://doi.org/10.63282/3050-9246.IJETCSIT-V2I1P112
[8] Vppalapati, M., & Talasila, P. K. (2022). Correlated Independence: Why Redundant Storage Systems Share the Same Fate. International Journal of Emerging Trends in Computer Science and Information Technology, 3(1), 169-179. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I1P119
[9] Pentyala, D. K. (2020). Enhancing the reliability of data pipelines in cloud infrastructures through AI-driven solutions. *The Computertech*, 30-49.
[10] Allenki, S. S., & Lee, N. (2022). Performance Tuning Cloud-Hosted Databases: Resource Allocation & Query Optimization. International Journal of AI, BigData, Computational and Management Studies, 3(4), 152-163. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I4P116
[11] Aversa, M., Malik, Z., Geier, P., Droz, F., Upegui, A., Murray-Smith, R., ... & Sanguinetti, B. (2022). Data-centric AI workflow based on compressed raw images. *Proceedings of the OBPDC2022-8th Internationl Worshop on Onboard payload data compression, 28-30 September 2022, Athens, Greece*.
[12] Kumar Doodala, A. N., & Thatraju, S. (2022). NLP-Driven Benefits Interpretation Engine for Personalized Member Communication. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 3(1), 173-183. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I1P118
[13] . Diop, F. (2019). Data-Centric Refactoring: Techniques for Improving Model Quality via Codebase Changes. *International Journal of Artificial Intelligence & Digital Transformation*, *2*(1), 01-15.
[14] Muppaneni, K. (2021). Cross-Browser Debugging Strategies. American International Journal of Computer Science and Technology, 3(5), 25-36. https://doi.org/10.63282/3117-5481/AIJCST-V3I5P103
[15] Chuprina, T., Mendez, D., & Wnuk, K. (2021). Towards artefact-based requirements engineering for data-centric systems. *arXiv preprint arXiv:2103.05233*.
[16] Suryadevara, S. S. K. (2021). Generative AI–Powered Authoring Assistant for Enterprise Content Management. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 2(2), 103-113. https://doi.org/10.63282/3050-9262.IJAIDSML-V2I2P111
[17] Broo, D. G., & Schooling, J. (2020). Towards data-centric decision making for smart infrastructure: Data and its challenges. *IFAC-PapersOnLine*, *53*(3), 90-94.
[18] Muppaneni, R. K. (2021). How Enterprises are Achieving 360° Customer Views with Dynamics 365. International Journal of AI, BigData, Computational and Management Studies, 2(2), 129-138. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V2I2P114
[19] Maimun, A., Siow, C. L., Khairuddin, J., & Hiekata, K. (2022). Digital Transformation Through Data-driven and Data-centric Approaches with Artificial Intelligence. In *International Conference on Computer Application in Shipbuilding, Yokohama, Japan. https://doi. org/10.3940/rina. iccas*.
[20] Vppalapati, M. (2022). The Storage Stack Nobody Draws: Cabling, Panels, and the Illusion of Isolation. International Journal of Emerging Research in Engineering and Technology, 3(2), 211-220. https://doi.org/10.63282/3050-922X.IJERET-V3I2P121
[21] Srigadde, B. R. (2021). Future Methods, Most Underrated Apex Features. American International Journal of Computer Science and Technology, 3(1), 35-45. https://doi.org/10.63282/3117-5481/AIJCST-V3I1P104
[22] Carey, M. J., Ceri, S., Bernstein, P., Dayal, U., Faloutsos, C., Freytag, J. C., ... & Whang, K. Y. (2008). Data-Centric Systems and Applications.
[23] Kumar Doodala, A. N. (2022). Strategic Migration for JBoss to IIBM WAS: A Framework for Enterprise-Grade Modernization. International Journal of Emerging Research in Engineering and Technology, 3(2), 161-170. https://doi.org/10.63282/3050-922X.IJERET-V3I2P117
[24] Allenki, S. S. (2022). Securing Databases in the Cloud with RBAC and Encryption Best Practices. International Journal of Emerging Research in Engineering and Technology, 3(3), 173-182. https://doi.org/10.63282/3050-922X.IJERET-V3I3P117
[25] Bhat, J. (2022). The Role of Intelligent Data Engineering in Enterprise Digital Transformation. International Journal of AI, BigData, Computational and Management Studies, 3(4), 106-114. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I4P111
[26] Shiramalla, R., & Guntupalli, B. . (2021). Cost-Effective Softphone Integration in CRM Platforms Using RESTful APIs: A Salesforce Case Study for Voice-to-Text Sales Enablement. International Journal of Emerging Trends in Computer Science and Information Technology, 2(1), 101-114. https://doi.org/10.63282/3050-9246.IJETCSIT-V2I1P112
[27] BALAHUR-DOBRESCU, A., JENET, A., HUPONT, T. I., CHARISI, V., GANESH, A., GRIESINGER, C., ... & TOLAN, S. (2022). Data quality requirements for inclusive, non-biased and trustworthy AI.
[28] Muppaneni, K. (2021). HTTP/3 & REST Latency Improvement. International Journal of Emerging Research in Engineering and Technology, 2(1), 122-132. https://doi.org/10.63282/3050-922X.IJERET-V2I1P113
[29] Reis, M. S., & Saraiva, P. M. (2019). Data-Centric Process Systems Engineering for the Chemical Industry 4.0. *Systems Engineering in the Fourth Industrial Revolution*, 137-159.
[30] Muppaneni, R. K. (2021). Securing the Enterprise: How Dynamics 365 Meets Global Compliance Standards. International Journal of Emerging Research in Engineering and Technology, 2(1), 133-143. https://doi.org/10.63282/3050-922X.IJERET-V2I1P114
[31] Azimi, S., & Pahl, C. (2021, April). AI quality engineering for machine learning based IoT data processing. In *International Conference on Cloud Computing and Services Science* (pp. 69-87). Cham: Springer International Publishing. 2022, pp. 1-27
[32] Suresh, A. (2024). Auto-BI Frameworks Powered by Generative Artificial Intelligence for Scalable Self-Service Data Analytics in Large Organizations. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 5(3), 191-201. https://doi.org/10.63282/3050-9262.IJAIDSML-V5I3P119

[33] Taluri, R. (2024). A Hybrid Data Engineering and Generative AI Architecture for Intelligent Data Governance, Metadata Management, and Automated Data Quality Assessment on AWS. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 5(2), 230-240. https://doi.org/10.63282/3050-9262.IJAIDSML-V5I2P126