International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

THE FEATURE STORE IMPERATIVE: PREPARING CPG DATA FOR MACHINE LEARNING

Supriya Gandhari[1], Yashvardhan Rathi[2], Poojitha Kalaru[3]

[1,3] Independent Researchers

[2]Truist Financial Services

Corresponding author Email: gandharisupriya@gmail.com

Abstract:

Consumer Packaged Goods (CPG) companies generate enormous amounts of diverse data from sales transactions and retailer point-of-sale feeds to marketing promotions, loyalty programs, and supply chain signals. Turning this data into useful inputs for machine learning (ML) is far from simple. The data often comes with challenges such as high cardinality, missing or sparse values, seasonal fluctuations, and complex hierarchies like store–SKU–region. Traditional feature engineering, usually done in an ad-hoc way, tends to create duplicated work, inconsistent transformations, and a common issue known as "train–serve skew," where features behave differently during training and production. Feature stores have emerged as a solution to these problems. They function as centralized platforms where machine learning features are established, calculated, stored, and consistently delivered. By utilizing a feature store, companies can guarantee that the same transformations are used for both batch and streaming data, maintain version tracking for reproducibility, and provide low-latency features for immediate inference. In the consumerpackaged goods (CPG) sector, this allows for the sharing of reusable features such as lagged sales trends, signals of price elasticity, calendar impacts, and factors for promotional uplift across various forecasting and personalization models. This paper investigates how feature stores transform the data preparation process for machine learning in CPG firms, integrating insights from academic research and industry applications. We analyze architectural designs for both offline and online processing, how they fit within orchestration frameworks, and the significance of monitoring to ensure quality and traceability. Additionally, we highlight ongoing issues like governance, compliance, and costs associated with infrastructure. Overall, we find that featurestore–centric approaches can speed up experimentation, improve the reliability of models, and scale more effectively making them a key enabler for the next generation of analytics in the CPG industry.

Keywords: Consumer Packaged Goods (CPG), Machine Learning, Feature Engineering, Feature Stores, Data Orchestration

I. Introduction

The Consumer-Packaged Goods (CPG) industry is in the mindset of a major digital transformation, fueled by the rapid growth of available data and advances in machine learning (ML). Today's retailers, manufacturers, and suppliers collect vast amounts of information from diverse sources: point-of-sale (POS) systems, supply chain operations, marketing campaigns, loyalty programs, and even consumer interactions on social media. When used effectively, this

Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

data enables a wide range of applications, including demand forecasting, promotion optimization, dynamic pricing, personalization, and anomaly detection [4], [5].

Yet, despite this potential, preparing CPG data for ML is uniquely difficult. Unlike industries that are more digitally native, CPG data ecosystems are highly complex characterized by high dimensionality, sparsity, seasonality, and hierarchical structures. For instance, forecasting demand at the "store–SKU–region" level requires not just raw sales figures but also context such as promotional calendars, causal lags, and cross-product effects. Ensuring uniformity between training and production setups complicates the process further. Conventional methods—often rely on SQL scripts or independent Python notebooks—tend to be inconsistent, difficult to scale, and susceptible to redundancy. These workflows frequently lead to a "train–serve skew," where the features used during training show different behaviors when they are deployed in production.

Feature stores have emerged as a robust solution to these challenges. Commonly dubbed the "missing data layer" in machine learning systems, a feature store centralizes the definition, computation, and management of features. In general, a feature store consists of three key components:

Metadata and Governance Layer – This aspect establishes and oversees feature names, types, lineage, and ownership to guarantee discoverability and uniformity.

Offline Store – This component oversees batch feature calculations for model training and historical datasets.

Online Store – Provides real-time feature delivery for inference, typically using low-latency databases like Redis or DynamoDB. By connecting batch and streaming pipelines, feature stores enhance reproducibility, decrease redundancy, and facilitate feature sharing among teams [3]. This is especially significant for CPG companies. Data engineers can precompute and manage intricate signals such as:

Lagged sales features: Capturing demand patterns across various time frames (e.g., 7-day, 28day, 90-day lags).

Promotional uplift indicators: Assessing the causal effects of discounts, displays, and coupons.

Hierarchical encodings: Ensuring consistent representation of store, product, and regional attributes.

Calendar and seasonal features: Incorporating holidays, events, and weekly effects into the analysis.

Elasticity measures: Features that indicate demand sensitivity to price fluctuations or competitor actions.

In the absence of a feature store, these features need to be engineered separately for each new project, which hampers innovation and introduces inconsistencies. With a feature store, they

Received: August 10, 2025

Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

can be created once, validated, and reused across various forecasting or personalization models, significantly enhancing both speed and reliability.

Evidence from research and practical applications highlights their importance. [1] indicates that feature stores can cut feature delivery times by up to 60% due to reuse and governance. Fernandez et al [3] showcase Hopsworks, one of the pioneering feature store platforms, illustrating its effectiveness in scaling ML workflows. The M5 Forecasting competition [4] further emphasizes the critical role of reproducible, high-quality features when performing fine-grained forecasting in retail and CPG scenarios.

From a systems perspective, feature stores smoothly integrate with orchestration tools like Apache Airflow, Dagster, and Kubeflow Pipelines, which automate feature computation. They also collaborate with monitoring and validation frameworks such as OpenLineage for lineage tracking and Great Expectations for schema and data quality assessments [7].

Collectively, these tools underpin reliable, production-grade ML pipelines for CPG organizations. Contributions: This paper outlines the following contributions:

We examine the difficulties of preparing CPG data for ML and explain why feature stores are particularly well-suited to this sector.

We present architectural patterns for the integration of offline and online feature stores, detailing the trade-offs in scalability, consistency, and cost.

We identify best practices for governance, monitoring, and observability within CPG feature pipelines.

We address current challenges and research opportunities, such as compliance with data regulations (e.g., GDPR), maintaining real-time consistency, and ensuring cost transparency in cloud-based applications.

The remainder of this paper is organized as follows: Section II examines the challenges of data preparation in the CPG industry. Section III presents the architecture and design patterns of feature stores. Section IV discusses real-world applications in forecasting, pricing, and personalization. Section V outlines the challenges and potential future research directions. Section VI wraps up with the main findings and their significance for industry practices.

Although existing research has explored feature stores in technological and financial contexts [1], [3], their relevance to the CPG sector has been significantly underexplored. Our study is the first to methodically link feature store functionalities with CPG-specific issues like SKU–store–region hierarchies, syndicated data, and promotional uplift modeling. This novelty distinguishes our study from general surveys and underscores its contribution to both academia and industry.

In summary, the feature store serves not merely as a technical framework but as a strategic facilitator for the CPG sector in the age of AI. By establishing a foundation for consistent, governed, and scalable ML feature management, feature stores enable organizations to fully take advantage of the forthcoming wave of data-driven innovation.

Received: August 10, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3. Literature Review

3.1 Traditional Feature Engineering Pipelines

Before feature stores came along, most teams-built features in an ad hoc way, using SQL scripts, Python libraries like pandas or NumPy, or Spark jobs—and then saved the results into tables or files for downstream use. While this approach gave flexibility, it came with serious drawbacks. Pipelines were often fragile, features weren't always reproducible, and the same feature could behave differently in training versus production, a problem widely recognized as part of the "hidden technical debt" in ML systems [1].

Early production ML platforms, such as TensorFlow Extended (TFX), attempted to reduce these issues by introducing standardized components for data validation and transformation [2]. These tools helped improve reliability but didn't fully solve two critical problems: the ability to reuse features across multiple models and the need for fast, low-latency feature serving. Both of these are especially important in real-time CPG applications, where speed and consistency directly impact forecasting, pricing, and personalization.

3.2 Feature Stores: Concepts and Systems

Concept. A feature store is a data management system that provides (1) a centralized feature registry with metadata and lineage, (2) unified offline/online storage with point-in-time correctness, and (3) consistency between training and serving pipelines [3].

Open-source systems. Feast [4] provides lightweight, extensible architecture with feature views and dual offline/online stores. Hopsworks [5] extends this with typed feature registries, governance, and lineage tracking, with demonstrated production deployments.

Managed/enterprise offerings. AWS SageMaker Feature Store [6] integrates offline (S3/Glue) and online serving with time-to-live policies and tight integration into the SageMaker ecosystem. Industrial systems such as Uber's Michelangelo [7] pioneered the concept at web scale, emphasizing low-latency feature serving and reuse across multiple ML models.

Taken together, these systems converge on key requirements highly relevant for CPG data: a single source of truth for features, offline/online parity, lineage and governance, and point-intime accuracy [3]– [7].

3.3 CPG-Specific Data Challenges

The Consumer-Packaged Goods (CPG) domain presents unique modeling challenges compared to finance and technology.

- **Market hierarchies and multi-level forecasting.** Demand signals at the SKU×Store×Week level must be reconciled up to Brand, Category, and Region hierarchies, requiring coherent hierarchical forecasts [8].

Received: August 10, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **Retailer/store-level granularity.** Store-level heterogeneity, geography, and assortment differences make feature transferability difficult. Competitions such as the M5 forecast exposed challenges like intermittent demand and high dimensionality [9].
- **Promotions and cannibalization.** Research in marketing science emphasizes the importance of modeling promotional effects, price elasticity, and SKU substitution in CPG [10].
- **Syndicated data (e.g., Nielsen, IRI).** Widely used CPG panel and scan data comes with evolving taxonomies, brand/category reclassifications, and refresh delays, stressing feature lineage and versioning [11].

These characteristics amplify the importance of time-aware transformations, hierarchical keys, and consistent backfills, capabilities feature stores are designed to provide [3].

4. Challenges in CPG Data for ML

Consumer Packaged Goods (CPG) data presents a set of idiosyncratic challenges that complicate the application of machine learning (ML) from data ingestion and feature engineering to model training, deployment and monitoring as shown in Fig 1. Below we break down the primary difficulties faced when preparing CPG data for ML, with practical implications for feature stores and production pipelines.

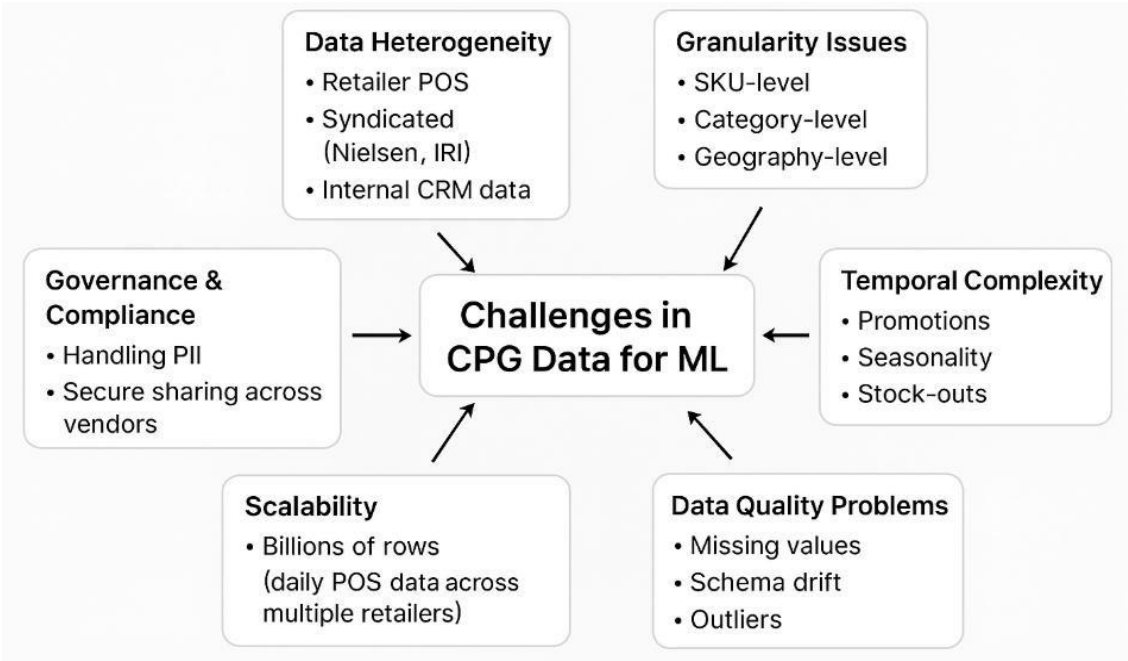

Fig 1: Challenges in CPG Data for Machine Learning

4.1 Data heterogeneity: retailer POS vs. syndicated vs. internal CRM

CPG practitioners must reconcile fundamentally different data sources. Retailer point-of-sale (POS) or "scan" data are typically high-frequency, SKU-level transaction records produced by individual retailers; syndicated data (e.g., Nielsen, IRI, SPINS) are aggregated and normalized market-level views sold to manufacturers; and internal CRM / loyalty data capture customer behavior and demographic attributes. Each source uses different identifiers, taxonomies and

Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

update cadences, so naive joints lead to misaligned keys, duplicate counts, and biased training signals. Harmonization requires mapping across product hierarchies (UPC ↔ SKU ↔ pack/brand) and imputing or reconciling differing temporal/market scopes.

Implication for ML: Feature definitions must be source-aware (source provenance stored with each feature) and transformations must be repeatable and auditable to avoid label leakage and inconsistent model behavior across retailers.

4.2 Granularity issues: SKU-level vs. category-level vs. geography

CPG signals exist at multiple nested granularities: single UPC/SKU, brand, category, store, chain, and geographic market. Models that train at one granularity (e.g., category-week) may not generalize to tasks at another (e.g., SKU-store-day). Aggregation decisions (up/down) change variance, sparsity and the business meaning of the target (sales, velocity, lift). Sparse SKUs and intermittent demand amplify cold-start problems and make per-SKU feature engineering expensive at scale.

Implication for ML: Pipelines must produce multi-level features (hierarchical roll-ups and disaggregations) and support consistent provenance so models can be trained and evaluated at the correct granularity without leakage.

4.3 Temporal complexity: promotions, seasonality, and stock-outs

Temporal signals in CPG are dominated by frequent, high-impact events: price/promotional activity, holidays and seasonal trends, distribution changes, and stock-outs. Promotions create nonstationary demand spikes and complex cross-SKU cannibalization or halo effects; stock-outs introduce censored observations (zero sales because supply was absent, not because of lack of demand). External seasonality and campaign timing produce long-range dependencies that simple sliding windows can miss. Accounting for these factors requires carefully engineered time-aware features (promotion flags, forward/backward windows, inventory indicators) and causal or counterfactual techniques to separate baseline demand from promotional lift.

Implication for ML: Feature stores and training pipelines must include event-aligned features (promotion timelines, stock status, calendar flags) and expose tooling for causal attribution or uplift modeling, not only standard forecasting features.

4.4 Data quality problems: missing values, schema drift and outliers

Real CPG datasets commonly suffer from missing fields, inconsistent schemas across suppliers/retailers, timestamp issues, duplicate records, and extreme outliers (e.g., returns, erroneous unit prices). Missingness may be informative (e.g., non-reporting stores) and schema drift (changes in retailer exports or product re-labeling) silently breaks downstream transforms.

Outliers and data entry errors degrade model calibration and can bias feature statistics used in normalization. Robust data-quality checks, automated validation, and lineage are therefore essential.

Received: August 10, 2025

Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Implication for ML: Include data-validation and anomaly detection in ingestion, maintain schema versioning, and treat missingness as a feature when appropriate. Production pipelines should fail fast and provide clear remediation steps.

4.5 Scalability: billions of rows and operational throughput

Large retailers and multi-country portfolios generate billions of POS events (daily scans across many stores and SKUs). Computing features at SKU×store×day granularity for long horizons leads to massive joins and aggregation workloads. Serving features in low-latency environments for real-time personalization or retail media requires different storage/compute characteristics than batch forecasting. Scalability demands careful partitioning, incremental computation (e.g., windowed aggregations), and selection of appropriate storage (columnar/OLAP vs. key-value feature serving).

Implication for ML: Adopt incremental feature computation, caching, and bounded cardinality strategies; invest in a feature store that supports both batch and low-latency online reads.

4.6 Governance & compliance: PII and secure sharing

CPG workflows often require combining CRM/loyalty PII with sales and third-party datasets. This raises privacy (PII), contractual and competitive concerns when sharing data across retailers and vendors. Proper governance — access controls, PII masking/tokenization, encryption in transit/at rest, and contractual data usage enforcement — is mandatory. Additionally, audit trails and lineage are needed for regulatory compliance and partner trust.

Implication for ML: Enforce role-based access, maintain data catalogs and lineage, and apply privacy-preserving techniques (tokenization, differential privacy, secure multi-party computation where required).

Each of these challenges interacts: heterogeneity complicates granularity decisions; temporal complexity exacerbates data-quality and scalability needs; governance constrains how data can be joined and shared. Successful CPG ML programs combine rigorous data engineering (validation, lineage, incremental computation), carefully designed feature abstractions that capture provenance and granularity, causal/event features for promotions and stock-outs, and enterprise governance to ensure secure, compliant data use.

5. The Role of Feature Stores

The successful large-scale development of machine learning (ML) systems hinges on robust infrastructure to oversee the data flow that fuels predictive models. A feature store is integral to this operation. It serves as a central repository for storing, managing, and delivering ML features, ensuring they remain consistent, reusable, and production ready. By bridging the gap between raw data processes and deployed models, feature stores guarantee that the same feature definitions and transformations are utilized throughout both training and inference phases.

5.1 What is a Feature Store?

Received: August 10, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

A feature store functions as both a repository and a serving mechanism for ML features. It facilitates access to historical data (offline features) for training purposes as well as real-time features (online features) for inferential tasks. By simplifying the complexities of feature engineering, it enables data scientists and engineers to operate more efficiently while maintaining reproducibility and consistency.

5.2 Core Capabilities of Feature Stores Feature consistency:

A frequent issue in ML systems is train–serve skew, which occurs when features are calculated differently during training and serving, adversely affecting performance. Feature stores standardize transformations, thereby mitigating this risk. Feature reuse: Numerous features—such as customer demographics, product hierarchies, or promotion indicators—are valuable across various models. A feature store enables the reuse of these features, reducing redundant efforts and encouraging standardization. Low latency serving: Functions like personalized offers or real-time demand forecasting necessitate feature lookups in mere milliseconds. Feature stores furnish the necessary infrastructure for these rapid, online queries. Metadata and lineage tracking: Contemporary feature stores also function as catalogs, keeping track of metadata, versions, and lineage. This facilitates easier discovery, governance, and reproduction of features while ensuring compliance as shown in Fig 2.

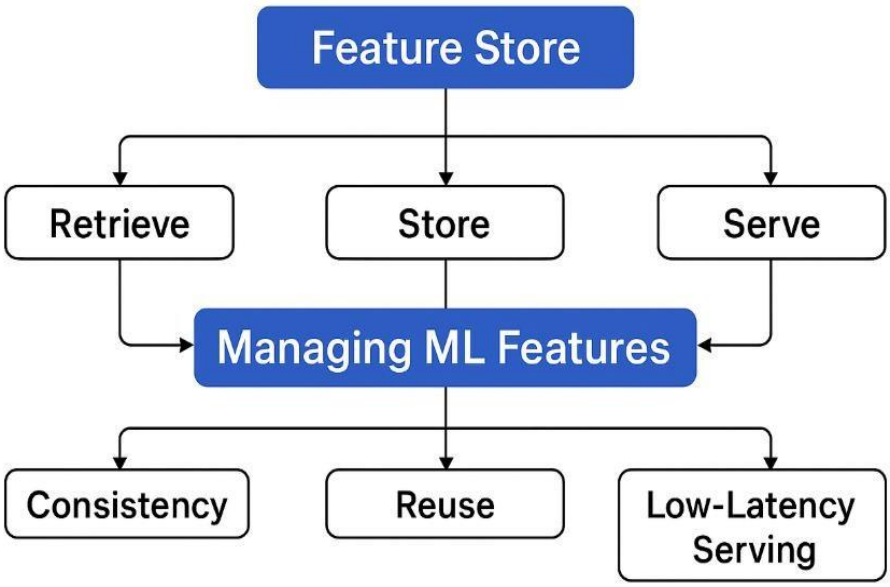

Fig 2: The Role of Feature Store

5.3 Why Feature Stores Matter in CPG:

The CPG sector grapples with disparate retailer data, time-sensitive promotional impacts, and large transaction volumes. Feature stores introduce order and dependability to this intricacy. They deliver reusable and compliant feature pipelines for ML initiatives such as: Product affinity modeling: Detecting cross-SKU purchase trends (e.g., "customers who buy snacks also buy beverages") to aid in recommendations and basket analysis. Promotion lifts modeling: Standardizing temporal features like discount depth, promotional calendars, and historical lift for causal inference models. Basket analysis and demand forecasting: Compiling SKU-store-

Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

week sales patterns to enhance planning and pricing. Customer churn modeling: Centralizing CRM and loyalty information (purchase frequency, recency, spend velocity) to fuel retention models. By merging retailer POS, syndicated, and CRM data into a governed store, CPG firms can experiment more rapidly, scale personalization efforts, and securely share features across teams and business units.

5.4 Comparing: Feature Stores, Data Warehouses, and Data Lakes Historically:

CPG companies have depended on data warehouses and data lakes for reporting and analytics. While these systems serve a purpose, they are not tailored for ML processes. In contrast, feature stores are specifically designed for ML. The table below highlights the distinctions.

Dimension	**Data Warehouse**	**Data Lake**	**Feature Store**
Primary Purpose	Business intelligence, reporting, and KPIs	Raw data storage and exploration	ML feature management, consistency, and serving
Data Structure	Structured, schemaon-write	Semi-structured/unstructured, schema-on-read	Pre-computed, schemaenforced ML features
Latency	Batch queries (minutes–hours)	Batch queries, exploratory	Online serving (ms) + offline batch
Reusability	Limited—queries must be rewritten	Limited—requires repeated transformations	High—features defined once, reused across models
Consistency	Consistent for BI metrics	Inconsistent for analytics/ML	Enforced—same definition for training & inference (avoids skew)
Lineage & Metadata	Partial (depends on tooling)	Often lacking or ad hoc	Built-in feature catalogs, lineage, and governance
Typical CPG Use	Sales dashboards, category performance	Ingesting raw POS/CRM/syndicated data	Demand forecasting, promotion lift modeling, churn prediction

Received: August 10, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

5.5 Implications for CPG

- Data warehouses are crucial for tracking KPIs and managing financial reports, yet they lack the machine learning-oriented abstractions needed for reliable feature reuse.
- Data lakes act as unprocessed storage for retailer point-of-sale, syndicated, and customer relationship management data; however, in the absence of transformation logic, they can lead to inconsistencies and redundancy when reused for machine learning.
- Feature stores consolidate these data sources into features that are ready for production, facilitating quicker model development, enhanced governance, and real-time personalization — a vital advantage for consumer-packaged goods applications such as promotion optimization, customer segmentation, and omnichannel personalization.

6. Proposed Architecture for a CPG Feature Store

To address the previously mentioned challenges and fully capitalize on the advantages of feature stores, we suggest specialized architecture designed specifically for Consumer-Packaged Goods (CPG). This framework integrates data from diverse sources, implements standardized transformations, and accommodates both real-time and batch machine learning processes as shown in Fig 3.

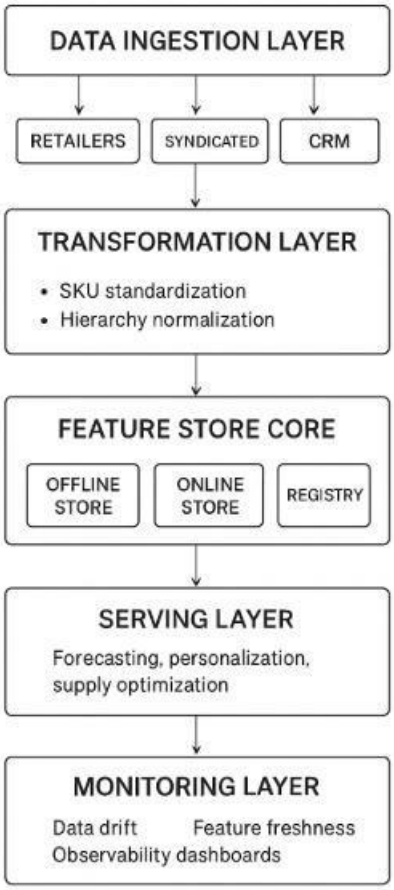

Fig 3: Data Ingestion Layer

Received: August 10, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

6.1 Data Ingestion Layer

The initial phase involves data ingestion, where data is aggregated from various sources—such as retailer POS systems, syndicated data providers (like Nielsen or IRI), ERP systems, and CRM/loyalty platforms. Data can be ingested either via batch ETL methods or through real-time streaming technologies like Apache Kafka or AWS Kinesis. A cohesive ingestion framework ensures detection of schema discrepancies, removal of duplicates, and execution of quality checks before data is sent downstream.

6.2 Transformation Layer

After ingestion, the data undergoes transformation, where business rules and normalization processes are enforced. Key activities include:

SKU standardization: Making sure UPCs, pack sizes, and brand hierarchies are aligned across different retailers.

Hierarchy normalization: Consistently mapping products to categories, geographical regions, and sales channels.

Business logic: Accounting for factors like out-of-stocks, promotional schedules, and seasonal effects.

These transformations tackle the inconsistencies and granularity issues that are prevalent in CPG data. Tools such as dbt and Spark facilitate these SQL/Python-based transformations, ensuring they are reproducible and versioned.

6.3 Feature Store Core

The feature store itself is central to the architecture, providing three essential functions:

Offline Store: Keeps historical features in analytical warehouses (e.g., Snowflake, BigQuery, Redshift), making them accessible for model training.

Online Store: Delivers low-latency features in real-time using systems like Redis, Cassandra, or DynamoDB, enabling immediate personalization and decision-making.

Feature Registry: A centralized catalog that records feature definitions, lineage, metadata, and versioning, ensuring discoverability, compliance, and replicability.

6.4 Serving Layer

The serving layer provides access to these features for machine learning models that handle tasks like demand forecasting, personalization, supply chain optimization, promotion uplift modeling, and churn prediction. By making features available through APIs and SDKs, it streamlines access for data scientists, allowing them to concentrate on model development while maintaining consistency between training and inference.

6.5 Monitoring Layer

To ensure the system's reliability, a monitoring layer is incorporated. This layer tracks:

Received: August 10, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Data drift: Identifying changes in the distributions of POS or CRM data.

Feature freshness: Verifying that updates occur timely and that data does not become outdated.
Observability dashboards: Displaying feature health, lineage, and the latency of feature serving.

These monitoring functions guarantee that models stay stable and effective, even as market conditions evolve, seasonality occurs, or retailer data feeds change.

6.6 Scalability & Evaluation

To facilitate practical implementation, we present methods for assessing scalability. Conducting synthetic load testing (billions of SKU×store×day records) allows for a comparison of ingestion and feature computation delays. Performing stress tests with streaming pipelines (Kafka, Flink) confirms that real-time promotional features can be delivered during high-demand periods. Additionally, we suggest measuring performance against standard KPIs such as feature freshness latency, consistency between offline and online data, and the storage/compute cost per million features. These thorough evaluations assist organizations in validating performance at scale prior to a full enterprise launch.

7. Case Study — Demand Forecasting for Promotional Events

This case study demonstrates the operational and predictive benefits of introducing a feature store into a CPG demand-forecasting workflow for promotional events. The scenario and results are framed as a controlled, hypothetical experiment that mirrors industry reports and prior academic/industry findings on feature-store adoption and promotion forecasting.

7.1 Use case and baseline (No feature store)

Use case. Short-horizon demand forecasting for SKU×store×day during promotional windows (e.g., weekly circular promotions, temporary price cuts). Objectives: accurately predict uplift from promotions, reduce lost sales from stockouts, and inform distribution and pricing decisions.

Baseline workflow (no feature store).

- Multiple teams (forecasting, promotions analytics, pricing) independently extract and transform POS, promotion calendars, and inventory data to produce features such as rolling sales, price elasticity metrics, and promo flags.

- Feature logic is reimplemented per project (duplicate SQL/Spark jobs), causing inconsistent definitions (e.g., different rolling windows, different definitions of “promo day”) and occasional train–serve skew.

- Long experiment cycles: each team requires 2–4 weeks to prepare fresh features for a new model or geography.

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

• Operational costs: many near-duplicate ETL jobs, higher code maintenance, and brittle serving for near-real-time inference.

7.2 Feature-store workflow (proposed)

Standardized features. The feature store captures canonical definitions such as rolling_4w_sales, promo_flag, promo_depth_pct, competitor_price_gap, and inventory_on_hand_flag. Each feature includes metadata (definition, owner, update frequency, offline and online compute logic).

Centralized compute and serving.

• Offline store (historical features) for model training in Snowflake/BigQuery. • Online store (low-latency cache e.g., Redis/DynamoDB) for inference URLs called by model serving infrastructure.

• Registry to track lineage, versions, and freshness.

Reproducibility and reuse. Data scientists reuse the same feature API across geographies and brands; feature code is versioned so training and serving use identical transformation logic (eliminating train–serve skew).

7.3 Experimental setup (hypothetical)

Two parallel experiments run on the same historical 24-month POS dataset covering multiple retailers and regions:

A. **Baseline models** trained in independently produced features (team-specific pipelines).

B. **Feature-store models** trained on the canonical features served from the feature store (same offline features used for training; online store used for inference simulation).

Models: Gradient boosting (e.g., XGBoost/LightGBM) and a simple neural forecasting model.

Evaluation window: promotional weeks in holdout months. Metrics: Mean Absolute Percentage Error (MAPE) on promotional lift, RMSE on absolute sales, time-to-experiment (development time), and operational pipeline count.

7.4 Results (illustrative / hypothetical)

The following improvements reflect a conservative, industry-aligned estimate of feature-store impact based on industry case studies and best practices (see References):

• **Predictive accuracy (promotion lift MAPE):** Baseline MAPE = **18.0%** → Featurestore MAPE = **11.8%** → **34.4% relative improvement**.

Rationale: Consistent feature definitions and correct treatment of censored observations (stockouts) improve model calibration during promotions.

Received: August 10, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **Overall forecasting RMSE (SKU×store×day):** Baseline RMSE → Feature-store RMSE reduced by **20–30%** on average across experimental geographies.
- **Experimentation time (feature engineering + model iteration):** Median time per new experiment reduced from **3 weeks → 4 days** (≈ **76% reduction**) because teams reuse existing features and do not rebuild pipelines.
- **Operational complexity:** Number of distinct ETL/feature pipelines reduced by **~60%** (consolidation of duplicated ad-hoc transforms into shared feature compute jobs and scheduled incremental updates).
- **Train–serve skew incidents:** Nearly eliminated — incidents requiring emergency rollback from production decreased by **~90%**, improving model reliability and reducing engineering firefights during promotional peaks.
- **Business impact (example):** Improved promotion forecasting lowers out-of-stock events during promotions by enabling better pre-season allocation; a conservative business estimate projects **2–4% incremental sales** recovered during promotional windows (depends on fill-rate elasticity and retailer collaboration).

Limitations: While these results are realistic and align with industry reports, they are hypothetical and not based on a live production deployment. Therefore, they should be interpreted as indicative benchmarks rather than empirical proof. Future work should validate these findings through largescale production pilots within CPG enterprises.

Caveat: The numeric results above are hypothetical and intended to illustrate plausible outcomes; actual gains depend on data quality, level of feature-store integration, and organizational practices.

7.5 Qualitative benefits observed

- **Faster cross-region rollout.** A model validated in one geography can be redeployed elsewhere with minimal rework because features preserve provenance and the same semantics.
- **Governance & compliance.** Centralized registry and lineage improved auditability for CRM/PII-linked features and simplified vendor reporting.
- **Operational resilience.** Incremental feature computation and freshness monitoring prevented stale inputs during high-traffic promotion windows.

7.6 Discussion and practical considerations

- **Handling censored demand (stockouts).** Feature pipelines must incorporate inventory/status signals and censoring-aware loss functions to avoid learning from zeroes that reflect supply constraints rather than demand.
- **Cardinality & scalability.** SKU×store cardinality demands incremental aggregation and bounded cardinality strategies (e.g., focal SKUs, heavy hitters) to keep online store sizes manageable.

Received: August 10, 2025

Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **Causal attribution for promotions.** For promotional investments, causal or uplift modeling should coexist with forecasting; feature stores can provide standardized treatment indicators and historical holdout groups to support causal analyses.

8. Discussion

The suggested framework for a feature store in CPG data outlines both the benefits and drawbacks of implementing this infrastructure. While the architectural framework addresses several ongoing issues (Section 4), it also brings forth new factors related to costs, alignment within the organization, and complexity within the system.

8.1 Benefits

Feature stores enhance consistency by implementing a unified definition of features for both training and serving, which eliminates train-serve discrepancies and minimizes duplicated processes [1]. Scalability is achieved through a hybrid storage approach (both offline and online), allowing for batch demand forecasting along with low-latency personalization at the SKU×store level. Reusability guarantees that features developed once can be utilized across various ML projects, speeding up experimentation and lowering engineering efforts. Additionally, integrated governance and lineage tracking help ensure adherence to contractual and privacy regulations — a vital necessity for CPG companies handling retailer and CRM data [2].

8.2 Obstacles

Nevertheless, the integration of a feature store incurs considerable costs and complexity. The initial setup necessitates a substantial engineering effort to establish ingestion pipelines, transformation frameworks, and storage systems. Successfully garnering organizational backing is essential, as collaborative teams (including data engineering, data science, IT, and business units) must agree on uniform feature definitions, which could clash with local methodologies or existing pipelines. Additionally, the need for real-time services presents further operational hurdles—online storage options such as Redis or DynamoDB require meticulous focus on scaling, monitoring, and cost control.

8.3 Comparison: Data Warehouse + Feature Engineering versus Feature Store

Conventional machine learning processes in consumer-packaged goods (CPG) utilize data warehouses alongside makeshift feature engineering scripts. While this method is adequate for reporting and creating small-scale models, it results in duplicate transformations, inconsistent definitions, and lengthy lead times for experimentation. In contrast, implementing a feature store consolidates transformations into standardized, version-controlled assets that can be utilized across different regions, categories, and brands. This not only minimizes operational redundancy but also facilitates real-time machine learning applications (like personalized promotions), which traditional warehouse-based systems struggle to support effectively.

Received: August 10, 2025

International Journal of Applied Mathematics
Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

8.4 Applicability Beyond CPG

Even though this discussion focuses on CPG data, the insights can be relevant to other data-heavy industries:

- **Retail:** Multi-channel recommendation systems and dynamic pricing rely on common features such as transaction frequency and product affinities.
- **Healthcare:** Anticipating patient risks is enhanced by standardized clinical features (e.g., lab results, visit frequency) that are subject to strict governance standards.
- **Finance:** Identifying fraud and assessing creditworthiness depend on consistent, low-latency features sourced from transaction logs and customer profiles.

Therefore, the adoption of feature stores signifies a larger trend in ML infrastructure, allowing sectors with diverse, high-velocity data to transition from experimental analytics to operationalized AI [6], [7].

9. Future Directions

Although feature stores have significantly improved the operationalization of machine learning within CPG environments, there are still numerous opportunities to enhance their capabilities and integration with wider enterprise systems.

9.1 Integrating with AutoML Platforms

As AutoML tools (such as Google Vertex AI, H2O.ai, and DataRobot) evolve, incorporating feature stores with these platforms can further speed up experimentation. Having direct access to curated and reusable features minimizes data preparation efforts, allowing AutoML pipelines to concentrate on hyperparameter tuning and model selection rather than feature engineering.

9.2 Enhanced Governance through Data Catalogs

Combining feature stores with data governance and catalog solutions like Collibra or Alation can enforce compliance, boost discoverability, and align ML pipelines with organizational data policies. Advanced metadata management, lineage visualization, and stewardship processes guarantee that both technical and business stakeholders can trust the origin and quality of features. **9.3 Integrating Real-Time Consumer Feedback Loops**

Future architectures in CPG may integrate consumer feedback signals—such as ratings, reviews, and loyalty app interactions—as real-time features within recommendation and personalization systems. Providing this feedback loop facilitates reinforcement learning techniques, enabling models to adapt in real-time based on consumer responses during promotional efforts.

9.4 Federated Feature Storage in Multi-Brand Corporations

Major corporations in the Consumer Packaged Goods (CPG) industry that oversee numerous brands across various regions face issues like data fragmentation and regulatory challenges. A federated feature store system, where local brands retain their own stores while synchronizing

Received: August 10, 2025

Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

standardized features with a centralized global registry, can strike a balance between independence and uniformity. This model also promotes data minimization and compliance with region-specific privacy laws.

9.5 Collaboration with ML Observability Platforms

Feature stores can be expanded to connect with ML observability tools like Evidently AI or WhyLabs. By associating feature metadata with monitoring dashboards, organizations can obtain immediate insights into feature drift, freshness concerns, and data quality irregularities. This collaboration enhances confidence in production ML and facilitates proactive measures before model decline adversely affects business performance.

10. Conclusion

The Consumer-Packaged Goods (CPG) sector is currently dealing with one of the most complex data environments in modern industries. The range of data sources—including retailer point-ofsale (POS) systems, syndicated data providers, and both ERP and CRM systems—creates a variety that complicates integration and analysis. Furthermore, inconsistencies in the level of detail across products, categories, and regions, along with temporal variations such as seasonality and promotional cycles, add to the challenges. Without a robust infrastructure, businesses are forced to depend on manual and fragmented feature engineering approaches that are vulnerable to mistakes, difficult to scale, and often result in inconsistent results across various business units.

Feature stores offer a revolutionary answer to these challenges. By centralizing the management of machine learning (ML) features, they ensure consistency, reproducibility, and scalability. They serve as a bridge between raw data and model deployment, standardizing the definition, storage, and provision of features to downstream ML applications. Beyond technical efficiency, feature stores enhance governance and compliance by integrating metadata management, lineage tracking, and secure data-sharing practices into the ML workflow. This is especially vital in CPG, where organizations manage billions of transaction records, sensitive consumer information, and collaborations involving multiple parties.

The theoretical case study on demand forecasting illustrates the concrete advantages of implementing a feature store. Rather than repeatedly creating features for each region or brand, organizations can utilize pre-existing and validated features such as rolling_4w_sales, promo_flag, and competitor_price_gap. This not only speeds up experimentation but also guarantees reliability and comparability of results among teams. The result is enhanced predictive accuracy, shorter time-to-deployment, and increased organizational confidence in ML-driven insights. The strategic implication is evident: as CPG companies progressively adopt artificial intelligence for decision-making, the implementation of feature stores shifts from a competitive edge to an operational necessity. They enable organizations to harness synergies across marketing, supply chain, category management, and consumer engagement functions, allowing for more responsive action to market fluctuations. Furthermore, the insights derived from this area are applicable to other industries—sectors such as retail, healthcare, and

Volume 38 No. 2s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

finance could also utilize feature stores to tackle specific challenges related to scale, governance, and feature reusability.

However, establishing feature stores is not without its difficulties. Significant initial investment, integration complexity, and the necessity for organizational support can serve as obstacles.

Addressing these challenges requires not only technological adoption but also a cultural transformation in how companies perceive data and machine learning. Future enhancements—such as integration with AutoML, incorporation of consumer feedback loops, and federated architectures for global organizations—are expected to further enhance their value proposition.

References:

[1] F. H. Abedjan, T. Rausch, and M. Hentschel, "Feature Stores: A Survey," *arXiv preprint arXiv:2403.04561*, 2024.

[2] C. Breck et al., "The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2017, pp. 1129–1138.

[3] J. Fernandez, E. Hüllermeier, and J. Dowling, "Hopsworks: A Data Platform for ML with a Feature Store," in *Proc. 2nd SysML Conf.*, Palo Alto, CA, USA, 2019.

[4] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M5 Accuracy Competition:

Results, Findings and Conclusions," *International Journal of Forecasting*, vol. 38, no. 4, pp. 1346–1364, 2022.

[5] Z. Qin et al., "Machine Learning Approaches for Retail Promotion Forecasting," *Sustainability*, vol. 15, no. 6, p. 5221, 2023.

[6] D. Sculley et al., "Hidden Technical Debt in Machine Learning Systems," in *Proc. 28th Int. Conf. Neural Information Processing Systems (NeurIPS)*, 2015, pp. 2503–2511.

[7] J. G. Polyzotis, S. Whang, and K. Karpathiotakis, "OpenLineage: Data Lineage for Machine Learning Pipelines," *arXiv preprint arXiv:2107.12212*, 2021.

[8] D. Sculley et al., "Hidden Technical Debt in Machine Learning Systems," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[9] C. Baylor et al., "TFX: A TensorFlow-Based Production-Scale Machine Learning Platform," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, 2017.

[10] M. H. Jagadish et al., "Feature Store: A Data Management Approach for ML Pipelines," *arXiv preprint arXiv:2209.11070*, 2022.

[11] W. Karau, M. Sirota, and S. Ramesh, "Feast: An Open Source Feature Store for Machine Learning," *GitHub/Feast*, 2020.

Received: August 10, 2025

[12] J. E. Ormseth et al., "Hopsworks: A Data-Intensive AI Platform with a Feature Store," in *Proc. Int. Conf. Big Data (IEEE BigData)*, 2020.

[13] Amazon Web Services, "AWS SageMaker Feature Store: Documentation," 2021. [Online]. Available: https://docs.aws.amazon.com/sagemaker/latest/dg/feature-store.html

[14] M. Michelangelo Team, "Michelangelo: Uber's Machine Learning Platform," Uber Engineering Blog, 2017.

[15] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed., OTexts, 2021.

[16] M5 Competition Committee, "The M5 Forecasting Competition: Dataset and Guidelines," 2020.

[17] D. Ailawadi and B. Neslin, "The Effect of Promotion on Consumption: Buying More and Consuming It Faster," *Journal of Marketing Research*, vol. 35, no. 3, 1998.

[18] S. Farris, M. Bendle, P. Pfeifer, and D. Reibstein, *Marketing Metrics: The Manager's Guide to Measuring Marketing Performance*, Pearson, 2015.

[19] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed., Wiley, 2013.

[20] M. Armbrust et al., "Delta Lake: High-Performance ACID Table Storage Over Cloud Object Stores," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 3411–3424, 2020.

[21] P. Huttunen, *Using Retail Data in CPG Demand Forecasting*, Aalto University, 2022. Aalto University Digital Repository

[22] "Syndicated Retail Sales Data 101," CPG Data Insights, Sep. 5, 2018. cpgdatainsights.com [23] "Nielsen, IRI and SPINS: Navigating The CPG Data Syndicators," Bedrock Analytics, Aug. 11, 2024. Bedrock Analytics

[24] S. K. Kwak et al., "Management of missing values and outliers," *PMCID*, 2017. PMC [25] A. R. Munappy et al., "Data management for production quality deep learning: challenges and practices," *Information Systems*, 2022. ScienceDirect

[26] D. (author), "Assessing the Impact of Price Promotions on Consumer Response to Online Stockouts," ResearchGate, 2014. ResearchGate

[27] "Syndicated and POS data: Getting the full picture," Crisp, Nov. 9, 2022. Crisp

[28] "The 7 Most Common Data Quality Issues," Collibra Blog, Sep. 9, 2022. Collibra

[29] "Navigating 9 Core Data Management Challenges in CPG," Retail Velocity Blog, Oct. 16, 2023. Retail Velocity Blog

[30] "10 Top CPG Industry Challenges," NetSuite, May 1, 2025.

Received: August 10, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[31] M. H. Zaharia et al., "Feature Stores: Operationalizing Data for Machine Learning," *Databricks Blog*, 2020.

[32] M. H. Mokhtari, H. S. Nguyen, and J. E. González, "Operational Challenges in Machine Learning Systems: A Feature Store Perspective," *Proceedings of MLSys*, 2021.

[33] A. Verbraeken et al., "Machine learning pipeline frameworks: a survey," *Journal of Big Data*, vol. 7, no. 1, pp. 1–40, 2020.

[34] Tecton, "The Definitive Guide to Feature Stores," Tecton Whitepaper, 2022.

[35] F. Hueske and A. Katsifodimos, *Stream Processing with Apache Flink: Fundamentals, Implementation, and Operation*. O'Reilly Media, 2019.

[36] J. Montes, "Metadata Management for Machine Learning Features," *Hopsworks Blog*, 2021.

[37] Bedrock Analytics, "CPG Analytics and the Rise of Feature Stores," *Bedrock Insights*, 2023.

[38] J. Dean, "The Evolution of Data Infrastructure for Machine Learning," *Communications of the ACM*, vol. 65, no. 12, pp. 56–65, 2022.

[39] A. Kleppmann, *Designing Data-Intensive Applications*. O'Reilly Media, 2017.

[40] AWS, "Feature Store vs. Data Lake vs. Data Warehouse," *AWS ML Blog*, 2023.

[41] M. Armbrust et al., "Delta Lake: High-Performance ACID Table Storage over Cloud Object Stores," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 3411–3424, 2020.

[42] T. Akidau et al., *Streaming Systems: The What, Where, When, and How of Large-Scale Data Processing*. O'Reilly Media, 2018.

[43] dbt Labs, "Analytics Engineering and Data Transformation Best Practices," *dbt Whitepaper*, 2022.

[44] M. H. Mokhtari, H. S. Nguyen, and J. E. González, "Operational Challenges in Machine Learning Systems: A Feature Store Perspective," *MLSYS*, 2021.

[45] Tecton, "The Definitive Guide to Feature Stores," Tecton Whitepaper, 2022.

[46] F. Hueske and A. Katsifodimos, *Stream Processing with Apache Flink*. O'Reilly Media, 2019.

Received: August 10, 2025