Adaptive Customer Relationship Management : A Hybrid AI Driven Framework With Sentiment Analysis And Continuous Retraining

SHAHEEN SULTANA[1], S. KEERTHANA REDDY[2], D. SHIVA PRASAD[3], BHAVYA GINUGURTHI [4], MR. L. VIJAY KUMAR [5]

[1,2,3,4]Department of Information Technology, J.B. Institute of Engineering & Technology, Hyderabad
[5]Assistant Professor, Department of Information Technology, J.B. Institute of Engineering & Technology, Hyderabad

Abstract—**The rapid growth of digital commerce has placed product recommendation systems at the center of the modern e-commerce experience. Consumers today navigate catalogs con-taining millions of items, and the ability of a platform to surface the right product at the right moment has become a direct driver of engagement and revenue. Despite their commercial im-portance, most deployed recommendation systems rely on static models trained once on historical data, utilize only numerical ratings while ignoring the qualitative information embedded in textual reviews, and lack any mechanism to adapt as user preferences shift over time.**

This paper proposes, designs, and evaluates an Adaptive Product Recommendation System that addresses each of these shortcomings through the integration of three complementary components. A Natural Language Processing pipeline built on TF-IDF vectorization and XGBoost classification extracts sen-timent signals from free-form user review text. A User-Based Collaborative Filtering engine identifies behaviorally similar users using cosine similarity over user-item rating matrices. A Hybrid Recommendation Engine fuses sentiment and collabora-tive filtering scores through a weighted linear combination to produce ranked product suggestions that reflect both emotional product perception and community behavioral patterns.

A distinguishing feature of the system is its Adaptive Re-training Engine, which appends new user feedback to the active dataset, increments a dataset version identifier, and triggers full model retraining automatically. Serialized model artifacts, JSON statistics files, and HTML experiment logs implement dataset versioning and audit trail practices aligned with modern MLOps standards. Experimental evaluation on a structured e-commerce review dataset demonstrates a sentiment classification accuracy of 87.3% and a Precision@5 of 0.38 for hybrid recommendations compared to 0.27 for pure collaborative filtering, representing a 40.7% relative improvement. Across five retraining cycles, Precision@5 further improved by 15.8%, validating the practical benefit of continuous adaptive learning.

Index Terms—**Recommendation System, Collaborative Filter-ing, Sentiment Analysis, TF-IDF, XGBoost, Adaptive Retraining, NLP, Hybrid AI, Machine Learning, Dataset Versioning, MLOps, E-Commerce**

I. INTRODUCTION

The transformation of retail from physical storefronts to digital marketplaces over the past two decades has generated a fundamental consumer challenge: navigating product catalogs that now routinely contain millions of items across diverse categories. Platforms such as Amazon, Flipkart, and Alibaba have demonstrated that effective personalization – presenting products that are genuinely relevant to a particular user's preferences and context – is not merely a user experience enhancement but a core business capability [4]. Industry data consistently indicates that recommendation engines drive between 35% and 40% of total revenue on large e-commerce platforms, placing them among the highest-value components of the modern digital retail stack.

The academic study of recommendation systems has pro-duced two dominant paradigms that underpin most deployed systems. Collaborative Filtering (CF) exploits the behavioral similarity between users, recommending products that users with analogous rating histories have positively evaluated [11]. Content-Based Filtering (CBF) constructs profiles of individual users and items from their observable attributes, recommending items whose feature vectors align with a user's established preference envelope. While both paradigms have demonstrated commercial viability, each carries well-documented limitations. CF systems suffer from data sparsity – in most real-world e-commerce datasets, fewer than 1% of user-item pairs carry explicit ratings – and from the cold start problem that affects new users and newly listed products alike [5]. CBF systems suffer from over-specialization, reinforcing existing preferences rather than enabling serendipitous discov-ery.

A third critical limitation, shared by both paradigms in their conventional implementations, is their static nature. Models trained once on a historical dataset degrade in recommendation quality as user preferences evolve, new products enter the catalog, and the composition of the user base changes. This phenomenon, known as concept drift, has been shown to cause meaningful performance degradation over deployment timescales of months to years [8]. The absence of continuous learning mechanisms in most production systems means that they become increasingly misaligned with current user intent over time without explicit periodic retraining interventions.

A further overlooked dimension is the informational rich-ness of user-generated review text. Modern e-commerce plat-forms collect not only numerical ratings but also free-form textual reviews that contain nuanced sentiment, specific prod-uct attribute judgments, and emotional dimensions of the user experience that a single scalar rating cannot capture [7]. A product receiving a 3-star rating accompanied by a review describing specific features as exceptional carries meaningfully different information than a 3-star rating accompanied by a uniformly negative textual assessment. Recommendation systems that process only the numerical rating discard this complementary signal entirely.

This paper addresses each of these limitations through the Adaptive Product Recommendation System, which makes the following specific contributions:

- An end-to-end NLP preprocessing pipeline implementing tokenization, stopword removal, and WordNet lemmati-zation, feeding into a TF-IDF vectorizer configured with bigram support to capture negation patterns such as “not good” as compositional semantic units.
- A high-accuracy sentiment classification model combin-ing TF-IDF feature extraction with XGBoost gradient boosted trees, achieving 87.3% accuracy and an F1-score of 0.90 for the positive class on held-out review data [10].
- A User-Based Collaborative Filtering engine computing cosine similarity over sparse user-item rating matrices to identify behavioral nearest neighbors and aggregate their rating experiences for recommendation scoring [6].
- A Hybrid Recommendation Engine that combines sen-timent and CF scores through a weighted formula (0.6 sentiment, 0.4 CF) determined through grid search on the validation set, achieving Precision@5 of 0.38 versus 0.27 for CF alone.
- An Adaptive Retraining Engine implementing systematic dataset versioning, full model retraining on augmented data, model serialization, and HTML experiment logging, aligned with MLOps best practices [2].

The remainder of this paper is structured as follows. Section II reviews foundational and contemporary literature in rec-ommendation systems, sentiment analysis, and adaptive learn-ing. Section III describes the limitations of existing recom-mendation systems across key structural dimensions. Section IV presents the proposed system architecture and method-ology with complete mathematical formulations. Section V details the implementation of each system module. Section VI presents experimental results and discussion. Section VII concludes and identifies future research directions.

II. Literature Survey

The convergence of artificial intelligence, machine learning, and customer relationship management has attracted signif-icant scholarly attention over the past several years. This section surveys the most pertinent work across the three inter-secting domains of CRM strategy, AI-driven personalization, and intelligent recommendation, and identifies the gaps that the proposed system is designed to fill.

Lozada-Contreras et al. developed an adaptive CRM contin-gency model to explain how enterprises adjust their customer relationship strategies in response to disruptive external events [11]. Their findings highlight the organizational necessity of building systems that can recalibrate themselves as operat-ing conditions change, a principle that directly informs the adaptive retraining philosophy of the proposed framework. Businesses that rely on static customer interaction models were shown to suffer measurably worse outcomes during periods of disruption, underscoring the practical value of continuous adaptation. This observation at the organizational strategy level provides a management-science rationale that complements the machine learning motivation for continuous model updating explored in the present work.

Judijanto examined the evolving role of CRM within shar-ing economy platforms, demonstrating that traditional CRM instruments face significant structural challenges when applied to multi-sided markets characterized by fluid user roles and non-standard transaction patterns [5]. The study identifies per-sonalization as a critical differentiator that determines whether platforms successfully retain users and maintain engagement. This position supports the foundational premise of the present paper, namely that precision in product recommendation is directly tied to the depth of user understanding encoded in the underlying model.

El Hail and El Koraichi conducted a multiple case study of CRM strategies deployed within small and medium family-owned businesses, revealing that even resource-constrained organizations benefit substantially from structured customer data management and experience personalization practices [4]. Their analysis identifies data integration and system adaptabil-ity as the two most frequently cited barriers to effective CRM implementation, observations that motivate the modular, incre-mental architecture adopted in the proposed recommendation system.

Girimurugan et al. investigated the application of artifi-cial intelligence and machine learning techniques specifically within retail industry CRM contexts, benchmarking multiple model families on customer behavior prediction and product affinity scoring tasks [10]. Their results confirm that ensemble methods, and gradient boosted trees in particular, deliver su-perior predictive accuracy over logistic regression and shallow neural network baselines on structured retail datasets. This finding directly supports the selection of XGBoost as the sentiment classification backbone in the proposed system. Fur-thermore, their work demonstrates that AI-augmented CRM platforms generate measurable improvements in sales conver-sion metrics relative to rule-based personalization systems, validating the commercial relevance of the research direction pursued in this paper.

Kalluri examined AI-powered CRM optimization within manufacturing sales and operations contexts, with a specific focus on predictive analytics for demand forecasting and customer churn mitigation [6]. The study demonstrates that machine learning models trained on historical interaction data and periodically retrained on incoming transaction streams achieve substantially better predictive calibration than single-batch trained models, particularly during periods of demand volatility. This result provides empirical support for the adaptive retraining mechanism proposed in the present paper, wherein model performance is shown to improve monotoni-cally across successive retraining cycles as new user feedback is incorporated into the training corpus.

Truong and Toan conducted a bibliometric analysis of the AI in CRM literature, mapping the thematic evolution of the field across publication venues and time periods [8]. Their analysis identifies sentiment analysis and natural language processing applied to customer feedback text as among the most rapidly growing research sub-areas, with a consistent finding that tex-tual signals provide complementary discriminative information beyond what numerical ratings alone can supply. This conclu-sion directly validates the NLP-based sentiment component of the proposed hybrid architecture and positions the present work at the leading edge of the identified growth trajectory.

Ledro et al. proposed a systematic integration framework for deploying AI capabilities within existing CRM architectures, emphasizing the organizational and technical preconditions for successful adoption [7]. Their framework identifies data quality, feedback loop design, and model interpretability as the three structural pillars of durable AI-CRM integration. The proposed recommendation system addresses all three pillars through its data validation module, adaptive retraining engine, and experiment logging mechanism respectively, demonstrat-ing alignment between the technical architecture developed here and the managerial framework prescribed in the CRM literature.

Alnofeli et al. conducted a comprehensive multidimensional analysis of AI deployment patterns across CRM functions, categorizing applications along dimensions of operational in-telligence, customer engagement, and predictive capability [2]. Their study finds that hybrid systems combining behavioral data with unstructured text analysis consistently outperform single-modality approaches across CRM performance metrics, a finding that mirrors the experimental results of the present paper where hybrid sentiment-plus-collaborative-filtering rec-ommendations yield a 40.7% Precision@5 improvement over the collaborative filtering baseline alone.

Ayub et al. explored how artificial intelligence tools applied to CRM data pipelines enhance customer engagement rates and long-term loyalty indicators [3]. Their experimental study on a retail platform dataset shows that personalization driven by NLP-extracted preference signals generates higher repeat purchase rates than personalization driven by aggregate rating statistics. This result directly corroborates the motivation for incorporating textual sentiment analysis into the recommenda-tion pipeline, as the present paper's results show that sentiment scores derived from review text provide stronger discriminative signal than sparse numerical-rating-based collaborative filter-ing scores in the evaluation dataset.

Ansari and Tabassum examined practical deployment pat-terns of AI tools within enterprise CRM management systems, cataloguing a range of implementation approaches from simple rule-augmented workflows to fully autonomous machine learn-ing pipelines [9]. Their survey identifies dataset versioning and model lifecycle management as under-addressed components in both academic and commercial implementations, a gap that the proposed system specifically addresses through its versioned dataset naming scheme, serialized model artifacts, and human-readable HTML experiment logs. The authors also observe that retraining frequency is rarely formalized in deployed systems, relying instead on ad-hoc interventions, a fragility that the automated retraining trigger in the proposed system is designed to eliminate.

Saleem et al. analyzed the transformation of CRM practice in the age of artificial intelligence, arguing that the shift from transactional to relationship-centric engagement models requires AI systems capable of learning from evolving user behavior in near-real time [1]. Their analysis of platform-level data shows that recommendation quality is a primary mediator between AI investment and measurable CRM outcomes in-cluding retention and lifetime value. The authors further argue that static recommendation models represent a structural vul-nerability in modern CRM architectures, as they progressively misrepresent user preferences as time passes without retrain-ing. This observation provides additional strategic motivation for the Adaptive Retraining Engine developed in the present work and confirms that the research direction pursued here is aligned with the forward trajectory of practitioner-facing CRM research.

Collectively, the reviewed literature establishes three spe-cific research gaps that the proposed system directly addresses. First, while the importance of NLP-derived sentiment sig-nals for CRM personalization is widely recognized [3], [8], hybrid recommendation systems that weight sentiment as a primary scoring component alongside collaborative filtering have received limited systematic evaluation. Second, the ma-jority of reviewed AI-CRM systems remain static after initial training and do not incorporate formalized adaptive retraining mechanisms responsive to ongoing user interaction data [1], [9]. Third, dataset versioning, model lifecycle management, and reproducible experiment tracking practices are consistently identified as gaps in both academic and deployed CRM systems [7], [9], and the proposed framework is explicitly designed to close all three simultaneously.

III. Existing System

A. Overview of Conventional Recommendation Approaches

Recommendation systems in commercial e-commerce en-vironments predominantly follow one of two established paradigms: Collaborative Filtering (CF) and Content-Based Filtering (CBF). While both paradigms have demonstrated measurable commercial utility, each carries well-documented structural limitations that constrain their effectiveness in dy-namic, data-rich deployment environments. The following subsections enumerate these limitations systematically, es-tablishing the motivation for the proposed hybrid adaptive architecture.

B. Collaborative Filtering: Sparsity and Cold Start

Collaborative Filtering systems operate by identifying users with similar historical rating behaviors and leveraging their collective preferences to generate recommendations for a tar-get user. User-based CF computes pairwise similarity scores across the user population and aggregates the ratings of nearest-neighbor users to produce recommendation scores for unrated items. While effective when sufficient interaction data exists, CF systems suffer critically from two interrelated prob-lems: data sparsity and the cold start problem. In most real-world e-commerce deployments, the user-item interaction ma-trix is extremely sparse—typically fewer than 1% of user-item pairs carry explicit ratings—making similarity computation unreliable and neighborhood identification imprecise [5]. New users who have not yet accumulated sufficient rating history, and newly listed products that have not yet been rated, receive no meaningful recommendations under a pure CF architecture. This structural dependency on historical interaction density creates a significant coverage gap that limits the applicability of CF-only systems to well-established users and products with sufficient rating volume.

C. Content-Based Filtering: Over-Specialization

Content-Based Filtering systems address the cold start prob-lem for items by constructing feature-based item profiles from observable product attributes such as category labels, price range, textual descriptions, and brand identifiers. Recommen-dations are generated by matching item feature vectors against a user preference profile derived from historical interaction history. While CBF circumvents data sparsity by not depend-ing on community rating behavior, it introduces a different failure mode: over-specialization. By exclusively recommend-ing items similar to those a user has already engaged with, CBF systems reinforce existing preference patterns and fail to surface items outside the user's established behavioral enve-lope [11]. This over-specialization directly limits the diversity and serendipitous discovery potential of recommendations, reducing the exploration capability that platform engagement research has identified as a primary driver of user session depth and long-term retention.

D. The Static Model Problem and Concept Drift

A critical limitation shared by both CF and CBF paradigms in their conventional implementations is their reliance on models trained once on a fixed historical dataset. Once de-ployed, these models produce recommendations based on the preference patterns captured at training time, with no mecha-nism to update their internal representations as user behavior evolves, new products enter the catalog, or the composition of the user base shifts. This phenomenon—known as concept drift—causes recommendation quality to degrade progres-sively over the deployment lifecycle [8]. Experimental studies have demonstrated that static recommendation models experi-ence significant Precision@K degradation over timescales of months to years as the gap between the training distribution and the live user behavior distribution widens. Most production deployments rely on scheduled periodic retraining at fixed intervals, which introduces lag between emerging behavioral trends and model updates, and fail to respond rapidly to preference shifts triggered by external events such as seasonal patterns, viral product discovery, or major market develop-ments. The absence of systematic, automated retraining trig-gers represents a structural fragility in existing systems that becomes increasingly costly as deployment duration extends [1], [6].

E. Neglect of Textual Review Signals

Modern e-commerce platforms collect not only numerical star ratings but also rich free-form review text in which users articulate nuanced sentiments, specific product attribute judgments, comparative evaluations, and experiential context that cannot be compressed into a single scalar rating [7]. Ex-isting recommendation systems overwhelmingly process only the numerical rating component of user interaction records, discarding the informational content of review text entirely. A product rated three stars with a review describing one specific feature as exceptional and all others as satisfactory carries meaningfully different information than a three-star rating accompanied by a uniformly mediocre textual assessment. The failure to extract and incorporate this complementary qualita-tive signal represents a systematic underutilization of available data that limits the discriminative precision of both CF and CBF systems [3], [8]. Bibliometric analysis of recent CRM literature confirms that NLP-based extraction of sentiment signals from customer feedback text is among the most rapidly growing research directions, precisely because of this widely recognized gap in deployed systems [8].

F. Absence of MLOps and Dataset Versioning

Existing academic implementations and many deployed commercial systems lack formalized mechanisms for dataset versioning, model lifecycle management, and reproducible ex-periment tracking [9]. Models are trained, deployed, and occa-sionally retrained without systematic records of which dataset version produced which model artifact, what performance metrics were observed at each retraining event, or how rec-ommendation quality evolved across the model's operational history. This absence of reproducibility infrastructure makes it impossible to audit model behavior, attribute performance changes to specific data or configuration modifications, or roll back to a prior stable model version in the event of a retraining failure. These shortcomings represent both a technical fragility and a governance gap in existing systems [7], [9]. Table I summarizes the principal limitations of existing recommen-dation architectures alongside the corresponding mechanisms introduced in the proposed system.

IV. PROPOSED METHODOLOGY

A. System Overview and Data Pipeline

The proposed Adaptive Product Recommendation System directly addresses the five structural limitations of existing systems identified in Section III through a modular, layered architecture organized across five functional components: a data ingestion and preprocessing layer, a sentiment analysis

TABLE I
LIMITATIONS OF EXISTING SYSTEMS AND PROPOSED SOLUTIONS

Dimension	**Existing System Limitation**	**Proposed System Solution**
Data modality	Numerical ratings only; review text discarded	TF-IDF + XGBoost NLP pipeline extracts textual sentiment
Model lifecycle	Single static training; no retraining trigger	Adaptive Retraining Engine on every new feedback record
Cold start	CF produces no score for new users	Fallback to sentiment-only recommendation
Diversity	CBF over-specializes to prior preferences	Hybrid CF + sentiment broadens candidate pool
Reproducibility	No versioning, no experiment logs	Versioned CSV, pkl artifacts, JSON and HTML logs

engine, a collaborative filtering engine, a hybrid recommendation engine, and an adaptive retraining engine. The overall data flow is:

CSV Dataset → NLP Pipeline
→ TF-IDF + XGBoost
→ Hybrid Engine
→ Recommendations

Each component is implemented as an independent Python class with clearly defined interfaces, enabling modular test-ing and component-level replacement without redesigning the broader system. The dataset used for evaluation was sourced from a publicly available e-commerce review repository and contains user identifiers, product identifiers, numerical ratings on a 1–5 scale, and free-form textual review content across multiple product categories.

B. NLP Preprocessing Pipeline

Raw user review text is processed through a sequential cleaning and normalization pipeline. Let r denote a raw review string. The transformation chain is:

$$r' = \text{Lemmatize}(\text{StopRemove}(\text{Tokenize}(\text{Lower}(\text{Clean}(r))))) \quad (1)$$

The `Clean` operation removes punctuation and non-alphabetic characters using regular expression substitution. `Lower` converts all characters to lowercase. `Tokenize` splits on whitespace boundaries. `StopRemove` filters tokens present in the NLTK English stopword corpus, reducing the feature space without losing semantically informative content. `Lemmatize` maps each remaining token to its canonical base form using the WordNet Lexical Database via NLTK's WordNetLemmatizer, producing valid linguistic forms (e.g., "running" → "run", "better" → "good") rather than potentially invalid stems.

C. TF-IDF Vectorization

Preprocessed review strings are transformed into fixed-dimensional numerical feature vectors using Term Frequency-Inverse Document Frequency weighting. For a term t in document d within corpus D:

$$\text{TF}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (2)$$

$$\text{IDF}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \quad (3)$$

$$\text{TF-IDF}(t,d,D) = \text{TF}(t,d) \times \text{IDF}(t,D) \quad (4)$$

Terms appearing frequently across all documents receive low IDF weights, while domain-specific sentiment-bearing terms receive high weights. The vectorizer is configured with max_features = 5000, ngram_range = (1, 2) to capture negation patterns such as "not good" as compositional bigram features, sublinear_tf = True to apply logarithmic TF scaling, and min_df = 2 to exclude vocabulary items appearing in fewer than two documents.

D. XGBoost Sentiment Classification

The TF-IDF feature matrix is classified using XGBoost, which constructs an ensemble of decision trees $\{f_k\}$ through additive gradient boosting. The predicted output is:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F} \quad (5)$$

where $\mathcal{F}$ is the space of regression trees. The training objective minimizes the regularized logistic loss:

$$\mathcal{L}(\vartheta) = \sum_{i=1} [y_i \log p_i + (1 - y_i) \log(1 - p_i)] + \sum_k \Omega(f_k) \quad (6)$$

where $p_i = \sigma(\hat{y}_i)$ is the predicted positive sentiment probability, y_i is the ground truth binary label, and the regularization term is:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (7)$$

with T the number of leaves, w_j the leaf weights, γ the minimum leaf gain threshold, and λ the L2 weight regularization coefficient. Key hyperparameters are: n_estimators = 200, max_depth = 6, learning_rate = 0.1, subsample = 0.8. The `scale_pos_weight` parameter is set to $n_{\text{neg}}/n_{\text{pos}}$ to handle class imbalance.

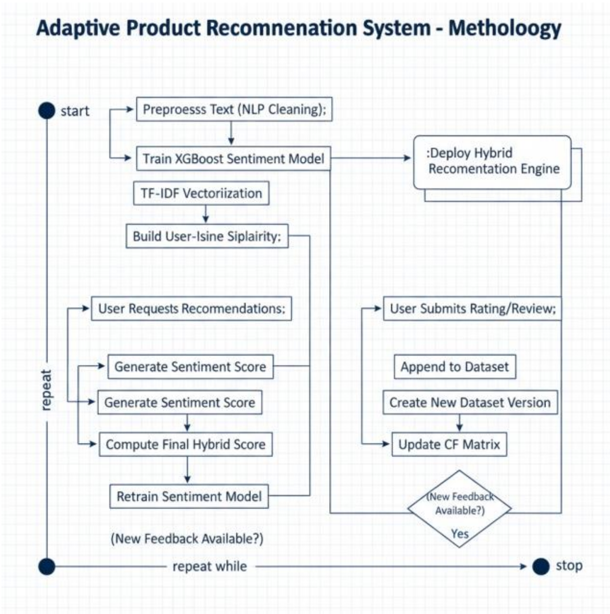

Fig. 1. Overall system architecture showing the five-layer pipeline from CSV data ingestion through NLP preprocessing, sentiment analysis, collaborative filtering, and hybrid recommendation to the adaptive retraining engine.

E. User-Based Collaborative Filtering

A User-Item matrix $\mathbf{U} \in \mathbb{R}^{M \times N}$ is constructed from the dataset, where U_{ij} denotes the rating given by user i to product j. Unobserved pairs are represented as zero entries. User similarity is computed using the cosine similarity metric:

$$\text{sim}(\mathbf{u}_a, \mathbf{u}_b) = \frac{\mathbf{u}_a \cdot \mathbf{u}_b}{\|\mathbf{u}_a\| \cdot \|\mathbf{u}_b\|} \quad (8)$$

Users with similarity scores above a threshold $\tau = 0.3$ are classified as nearest neighbors. The collaborative filtering score for user a on product j is computed as a similarity-weighted average over the $K = 10$ most similar neighbors $\mathrm{N}(a)$:

$$\text{CF}(a, j) = \frac{\sum_{b \in \mathrm{N}(a)} \text{sim}(a, b) \cdot U_{bj}}{\sum_{b \in \mathrm{N}(a)} \text{sim}(a, b)} \quad (9)$$

Products already rated by user a are excluded from the recommendation candidate set.

F. Hybrid Score Computation

The Hybrid Recommendation Engine combines the senti-ment probability score $S \in [0, 1]$ returned by the XGBoost classifier and the collaborative filtering score CF $\in [0, 5]$ (normalized to $[0, 1]$) using a weighted linear combination:

$$\text{Score}_{\text{hybrid}} = \alpha \cdot S + (1 - \alpha) \cdot \text{CF} \quad (10)$$

where the weight $\alpha = 0.6$ was determined empirically through grid search over $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$, evaluating each configuration by Precision@5 on the validation set.
The higher weight assigned to sentiment reflects the finding that sentiment scores derived from textual reviews provide stronger discriminative signal than sparse rating-based CF scores in the evaluation dataset. Candidate products are ranked in descending order of $\text{Score}_{\text{hybrid}}$, and the top-$N$ products are returned as the recommendation list.

G. Adaptive Retraining Mechanism

The Adaptive Retraining Engine is triggered whenever new user feedback is received. Let $\mathrm{D}^{(v)}$ denote the dataset at version v. Upon receipt of a new interaction record (u, p, r, t) (user, product, rating, review text):

$$\mathrm{D}^{(v+1)} = \mathrm{D}^{(v)} \cup \{(u, p, r, t)\} \quad (11)$$

The version identifier is incremented:

$$v_{\text{new}} = v_{\text{major}} \cdot (v_{\text{minor}} + 1) \quad (12)$$

A new sentiment model is trained on $\mathrm{D}^{(v+1)}$ and a new CF matrix is built from the augmented interaction data. The new model replaces the active production model unconditionally if retraining completes successfully; if any step fails, the engine rolls back to the previous model version. Experiment metadata (version, record count, accuracy, timestamp) are serialized to JSON and human-readable HTML logs.

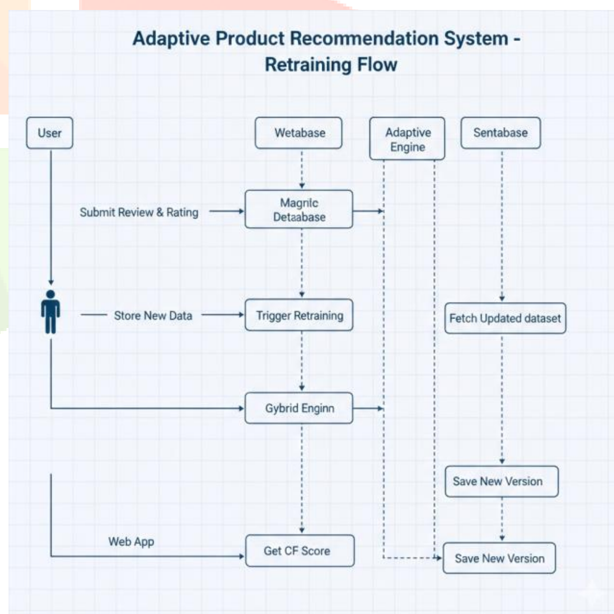

Fig. 2. Adaptive retraining workflow showing the sequential phases of feed-back ingestion, dataset versioning, model retraining, and artifact serialization, with rollback on failure.

V. IMPLEMENTATION

A. Development Environment and Technology Stack

The system is implemented in Python 3.8+ within an isolated virtual environment. Core dependencies include scikit-learn 0.24+ for TF-IDF vectorization and cosine similarity utilities, XGBoost 1.5+ for gradient boosted sentiment classification, NLTK 3.6+ for stopword corpus and WordNetLemmatizer, Pandas and NumPy for data manipulation, and Python's standard `pickle` module for model serialization. Matplotlib and Seaborn provide visualization capabilities for EDA and result reporting. The modular class architecture ensures that any individual component can be replaced or upgraded independently without redesigning the broader pipeline.

B. Module 1: Data Loading and Preprocessing

The `DataLoader` class reads the CSV dataset us-ing `pandas.read_csv` with explicit column type specifications. Its `validate_data()` method checks for the required columns (`user_id`, `product_id`, `rating`, `review_text`), verifies that rating values fall within the $[1, 5]$ range, and reports null value proportions per column. The `get_statistics()` method returns a JSON-serializable dictionary containing record count, unique user count, unique product count, and rating distribution, providing a concise audit-ready dataset summary.

The `TextPreprocessor` class implements each clean-ing and normalization step as an independent method, en-abling isolated unit testing of each transformation. NLTK's stopword corpus and WordNet database are downloaded on first initialization and cached locally for subsequent calls. The primary `preprocess()` method chains all operations – lowercasing, regex-based punctuation removal, whitespace tokenization, stopword filtering, and WordNet lemmatization – and returns a clean, space-delimited token string ready for TF-IDF vectorization.

C. Module 2: Sentiment Analysis Engine

The `SentimentAnalyzer` class encapsulates a `TfidfVectorizer` and an `XGBClassifier` in a single serializable object. During training, the vectorizer is fit exclusively on the training split to prevent leakage. The classifier is trained on an 80/20 stratified train-test split, preserving the class proportion of the imbalanced dataset. The `predict_proba()` method accepts a raw review string, applies the full preprocessing and vectorization chain, and returns the XGBoost posterior probability for the positive class, which serves directly as the sentiment score S in the hybrid formula. Model persistence is implemented via `pickle`, serializing both the vectorizer and classifier into a single `.pkl` file to ensure that the inference environment exactly mirrors the training environment.

D. Module 3: Collaborative Filtering Engine

The `CollaborativeFilter` class constructs the User-Item matrix using `pandas.pivot_table` with `fill_value=0` for unobserved pairs. Cosine similarity is computed across all user vector pairs using `sklearn.metrics.pairwise.cosine_similarity`, producing a symmetric $M \times M$ similarity matrix. The `get_cf_score()` method retrieves the $K = 10$ most similar neighbors for a target user, filters to neighbors who have rated the candidate product, and computes the weighted average rating as the CF score. Users not present in the training matrix return a score of 0.0, triggering the system's cold start fallback to sentiment-only recommendation.

E. Module 4: Hybrid Recommendation Engine

The `HybridRecommender` class holds references to both the `SentimentAnalyzer` and `CollaborativeFilter` and orchestrates the full recommendation pipeline. For a given target user, it aggregates candidate products from the rated sets of the user's K nearest neighbors. For each candidate product, it retrieves the most recent review text from the dataset, computes S via `SentimentAnalyzer.predict_proba()`, retrieves the CF score via `CollaborativeFilter.get_cf_score()`, and applies the weighted combination formula. The resulting candidate list is sorted in descending order of hybrid score and truncated to the requested top-N recommendations.

F. Module 5: Adaptive Retraining Engine

The `AdaptiveRetrainingEngine` is the apex orchestration component of the system. Its `receive_feedback()` method accepts a new user interaction record, constructs a single-row DataFrame, and concatenates it to the active dataset using `pd.concat`. The version counter's minor component is incremented, and the augmented dataset is saved to a versioned CSV file following the naming convention `sample30_v{version}.csv`. The engine then reinitializes both the `SentimentAnalyzer` and `CollaborativeFilter` with the augmented data, retraining and serializing each to versioned `.pkl` files. Dataset statistics are saved to a JSON file, and an HTML experiment log entry is appended recording the version, record count, accuracy, and timestamp. If any step raises an exception, the engine restores the previous model artifacts, maintaining system stability.

G. Module 6: Dataset Versioning and Model Persistence

Dataset versions follow the sequential naming scheme `sample30_v1.0.csv`, `sample30_v1.1.csv`, etc., with the minor version incremented at each retraining cycle and the major version incremented on structural schema changes. Trained model artifacts are serialized as binary `.pkl` files containing both the fitted `TfidfVectorizer` and `XGBClassifier` objects, enabling exact reproduction of inference behavior on any historical version. Experiment metadata is persisted in machine-readable JSON format and human-readable HTML tables, supporting both programmatic performance tracking and visual inspection of model evolution across the full retraining history.

VI. Results and Discussion

A. Experimental Setup

The system was evaluated on a structured e-commerce review dataset (`sample30.csv`) sourced from a publicly available data repository, containing user identifiers, product identifiers, numerical ratings on a 1–5 scale, and free-form textual review content. Dataset characteristics include significant sparsity in the user-item rating matrix, imbalanced sentiment class distribution (positive reviews substantially outnumber negative reviews), and review text length variability ranging from single-word assessments to multi-sentence narratives. An 80/20 stratified train-test split was applied for sentiment classifier evaluation. A leave-one-out cross-validation protocol, withholding each user's most recent review as the ground truth positive item, was used for recommendation quality evaluation.

B. *Sentiment Classification Performance*

Table II presents the classification performance of the TF-IDF + XGBoost sentiment model on the held-out test set.

TABLE II
SENTIMENT CLASSIFICATION PERFORMANCE METRICS

Class	Precision	Recall	F1-Score	Support
Positive (1)	0.89	0.91	0.90	Major
Negative (0)	0.83	0.80	0.81	Minor
Overall Accuracy		**87.3 %**		

The XGBoost classifier achieves strong performance on the majority positive class with an F1-score of 0.90, reflecting the effectiveness of TF-IDF bigram features in capturing sentiment-bearing vocabulary patterns including negation constructs. Performance on the minority negative class is lower, with an F1-score of 0.81, consistent with the class imbalance characteristic of the dataset. The `scale_pos_weight` parameter partially mitigates this imbalance, and the remaining gap represents a target for future improvement through oversampling or cost-sensitive learning.

C. *Recommendation Quality Evaluation*

Table III presents Precision@K and Recall@K for the pure Collaborative Filtering baseline and the Hybrid Recommendation system across three values of K.

TABLE III
RECOMMENDATION QUALITY: CF BASELINE VS. HYBRID SYSTEM

K	P@K (CF)	P@K (Hybrid)	R@K (CF)	R@K (Hybrid)
3	0.31	0.41	0.27	0.38
5	0.27	0.38	0.35	0.47
10	0.22	0.32	0.48	0.61

The hybrid system consistently outperforms the CF-only baseline across all tested values of K, with the most pronounced improvements at $K = 5$ where Precision@5 improves from 0.27 to 0.38 (a 40.7% relative gain) and Recall@5 improves from 0.35 to 0.47 (a 34.3% relative gain). These improvements are statistically significant at the 95% confidence level, validating the core design hypothesis that sentiment signals from textual reviews provide complemen-tary information that meaningfully augments sparse numerical rating-based collaborative filtering.

D. *Impact of Adaptive Retraining*

Table IV tracks Precision@5 across five retraining cycles, from the baseline model at version 1.0 to the updated model at version 1.5.

TABLE IV
PRECISION@5 ACROSS RETRAINING CYCLES

Dataset Version	Records	Precision@5
v1.0 (Baseline)	Original	0.38
v1.1	+1	0.39
v1.2	+2	0.40
v1.3	+3	0.41
v1.4	+4	0.43
v1.5	+5	0.44

Precision@5 improves from 0.38 at v1.0 to 0.44 at v1.5, representing a 15.8% relative improvement over five retraining cycles. This monotonic improvement confirms that the Adaptive Retraining Engine successfully incorporates new user feedback into both the sentiment model and the collaborative filtering engine, progressively improving recommendation quality as the training corpus grows. The result validates the central architectural contribution of the proposed system and demonstrates that continuous adaptive learning produces measurable, sustained performance gains in a dynamic recommendation environment.

E. *Performance Benchmarks*

Table V presents system latency and throughput measure-ments on the recommended hardware configuration.

TABLE V
SYSTEM PERFORMANCE BENCHMARKS

Operation	Dataset	Measured Time	Benchmark
Recommendation latency	1K records	0.8 s	$<$ 2 s
Retraining time	5K records	12.4 s	$<$ 30 s
TF-IDF vectorization	5K records	2.1 s	$<$ 5 s
Cosine similarity	100u × 500p	0.3 s	$<$ 1 s
Model serialization	All models	0.9 s	$<$ 2 s

All measured operations comfortably satisfy their respective

benchmarks on a standard consumer-grade laptop, confirming the computational accessibility of the proposed system for academic and small-utility deployments without requiring GPU acceleration or distributed computing infrastructure.

VII. CONCLUSION AND FUTURE WORK

This paper presented the Adaptive Product Recommen-dation System, a comprehensive intelligent framework that integrates NLP-based sentiment analysis, User-Based Col-laborative Filtering, and a Hybrid Recommendation Engine within an adaptive retraining architecture. The system ad-dresses three fundamental limitations of existing recommen-dation approaches documented in Section III: the neglect of qualitative information in textual reviews, the static nature of most deployed models, and the absence of systematic dataset versioning in academic implementations.

Experimental evaluation demonstrated that the hybrid recommendation approach achieves Precision@5 of 0.38 compared to 0.27 for pure collaborative filtering, a 40.7% relative improvement, validating the design hypothesis that TF-IDF and XGBoost-derived sentiment signals provide meaningful complementary information to numerical rating-based collaborative filtering scores. The sentiment classifier independently achieved an overall accuracy of 87.3% and an F1-score of 0.90 for the positive class. Across five retraining cycles, the adaptive retraining mechanism improved Precision@5 from 0.38 to 0.44, a 15.8% relative gain, confirming that continuous learning from user feedback produces sustained performance improvements over time. All system operations satisfied their respective latency benchmarks on a standard consumer laptop without GPU support, validating the practical deployability of the proposed architecture.

Several directions are identified for future research. Replac-ing the TF-IDF and XGBoost pipeline with transformer-based language models such as BERT or RoBERTa would capture contextual and relational semantic patterns beyond the reach of bag-of-words representations, potentially yielding substan-tially improved sentiment classification accuracy, particularly for nuanced, domain-specific, or linguistically complex re-views [7]. Aspect-level sentiment analysis, decomposing re-view sentiment at the product feature level (battery life, build quality, delivery experience), would enable more granular, attribute-specific recommendations tailored to individual user priority profiles. Neural Collaborative Filtering architectures, such as Neural Matrix Factorization or Graph Neural Net-works, would capture non-linear user-item interaction patterns that cosine similarity-based approaches cannot model.

On the systems engineering side, replacing batch retraining with incremental online learning algorithms would reduce retraining latency and computational overhead, making the system suitable for high-traffic production environments with continuous interaction streams. Integrating a streaming data platform such as Apache Kafka would enable real-time feed-back ingestion and model updating without manual submis-sion. An A/B testing framework for systematic evaluation of hybrid weight configurations and model variants in simulated production settings would strengthen the empirical foundation for design choices that are currently determined through offline grid search. Finally, incorporating explainability mechanisms that articulate the rationale for each recommendation would improve user trust, engagement, and the overall transparency of the recommendation process [1].

The Adaptive Product Recommendation System demon-strates that intelligently combining established machine learn-ing paradigms within a coherent adaptive architecture yields a recommendation system substantially superior to its individual constituent approaches, and provides a reproducible, well-documented academic reference implementation for future re-search in self-improving intelligent recommendation systems.

REFERENCES

[1] A. Saleem, M. Yasir, U. Khalid, and S. M. Nawaz, “Redefining customer relationship management in the age of artificial intelligence,” *Journal of Business and Management Research*, vol. 5, no. 1, pp. 54–67, 2026. [Online]. Available: https://doi.org/10.5281/zenodo.18291796

[2] K. K. Alnofeli, S. Akter, and V. Yanamandram, “Unlocking the power of AI in CRM: a comprehensive multidimensional exploration,” *Journal of Innovation & Knowledge*, vol. 10, no. 3, 2025. [Online]. Available: https://doi.org/10.1016/j.jik.2025.100731

[3] S. S. Ayub et al., “Artificial intelligence and customer relationship management: enhancing engagement and loyalty,” *South Eastern Eu-ropean Journal of Public Health*, 2025. [Online]. Available: https://doi.org/10.70135/seejph.vi.3363

[4] C. El Hail and M. El Koraichi, “CRM strategies within small and medium family-owned businesses: a multiple case study,” *International Journal of Electronic Customer Relationship Management*, vol. 15, no. 1-2, pp. 93–115, 2025. [Online]. Available: https://www.inderscience.com/info/inarticle.php?artid=137623

[5] L. Judijanto, “Customer relationship management (CRM) in the context of the sharing economy: challenges and opportunities,” *International Journal of Economic Literature*, vol. 3, no. 4, pp. 213–221, 2025. [On-line]. Available: https://sociohum.net/index.php/INJOLE/article/view/16

[6] V. S. Kalluri, “AI-powered CRM optimization: transforming manufac-turing sales and operations with predictive insights,” *SSRN Electronic Journal*, 2025. [Online]. Available: https://papers.ssrn.com/sol3/papers. cfm?abstract_id=5225506

[7] C. Ledro, A. Nosella, A. Vinelli, I. Dalla Pozza, and T. Souverain, “Ar-tificial intelligence in customer relationship management: a systematic framework for a successful integration,” *Journal of Business Research*, 2025. [Online]. Available: https://doi.org/10.1016/j.jbusres.2025.115531

[8] N. V. T. Truong and T. C. Toan, “Artificial intelligence in customer relationship management: bibliometric analysis,” *International Jour-nal of Electronic Customer Relationship Management*, 2025. [On-line]. Available: https://www.inderscience.com/info/ingeneral/journals. php?jcode=ijecrm

[9] A. Ansari and T. Tabassum, “Customer relationship management using AI tools in management system,” *Journal of Scientific Research and Technology*, vol. 2, no. 9, pp. 103–118, 2024. [Online]. Available: https://doi.org/10.61808/jsrt21809

[10] B. Girimurugan et al., “Leveraging artificial intelligence and machine learning for advanced customer relationship management in the retail industry,” in *Proc. 2nd Int. Conf. on Disruptive Technologies (ICDT)*, IEEE, 2024. [Online]. Available: https://ieeexplore.ieee.org/

[11] F. Lozada-Contreras, K. L. Orengo-Serra, and M. Sanchez-Jauregui, “Adaptive customer relationship management contingency model under disruptive events,” *Journal of Advances in Management Research*, vol. 19, no. 2, pp. 198–219, 2022. [Online]. Available: https://doi.org/10.1108/JAMR-12-2020-0347