An Intelligent Recommendation System Built on Emotional Analysis in a Kappa Architecture

Diana – Andreea CĂUNIAC, Adela BÂRA
Bucharest University of Economic Studies, 010374 Bucharest, Romania
diana.cauniac@csie.ase.ro, bara.adela@ie.ase.ro

Recommendation systems have become a cornerstone of modern e-commerce, directly shaping user experience and conversion rates. Yet most conventional approaches rely solely on behavioral history, clicks, views, purchases, without any awareness of how a user is feeling in the moment of decision. This paper presents an intelligent recommendation system that fills that gap by weaving emotional analysis of text reviews into a real-time data-processing pipeline. The solution is built on a Kappa architecture and leverages Apache Kafka, Google Cloud BigQuery, Bigtable, BigQuery ML, and a RoBERTa-based NLP model for emotion detection. Users are grouped into clusters through K-Means segmentation according to their emotional profiles and recommendations are then derived from well-established correlations between emotional states and product categories. In controlled evaluation, the system achieved a Precision@5 of 0.98, a Recall@5 of 0.94, and an F1-score of 0.96, confirming the strength of the proposed approach.

Keywords*: Recommendation systems, sentiment analysis, BigQuery ML, Apache Kafka, Kappa architecture, RoBERTa, natural language processing, personalized recommendations.*

1 Introduction

The rapid digital transformation of e-commerce has produced an enormous and ever-growing volume of user-generated data. Every click, page view, purchase, and review contributes to a rich behavioral profile that platforms use to personalize the consumer experience. At the center of this effort sit recommender systems, tools that have proven indispensable for improving user satisfaction and driving commercial performance [1], [2], [3].

Most widely deployed systems, however, share a fundamental blind spot. Whether they rely on collaborative filtering, popularity rankings, or content-based analysis, they treat user intent as a purely behavioral signal. They can tell you what a user did, but not why [4]. Consider two people who both purchase the same book: one is looking for a relaxing weekend read, the other for professional development. Their behavior is identical; their needs are not.

This is where affective analysis comes in. Emotions are a powerful driver of purchasing behavior, and accounting for a user's current emotional state opens the door to recommendations that feel genuinely relevant rather than merely statistically probable. The potential here is significant, not just for accuracy, but for the quality of the overall user experience [5], [6].

This paper introduces a recommender system that integrates emotional analysis of textual reviews with real-time stream processing, built on a Kappa architecture and deployed within the Google Cloud Platform ecosystem [7]. The goal is a scalable, low-latency framework that delivers recommendations sensitive to both a user's transactional history and their dynamic emotional state [8].

2 System Architecture

The system is built around two guiding principles: modularity and a microservices-based design. Each component operates independently and communicates through well-defined interfaces, which makes the architecture easy to extend without disrupting what is already running.

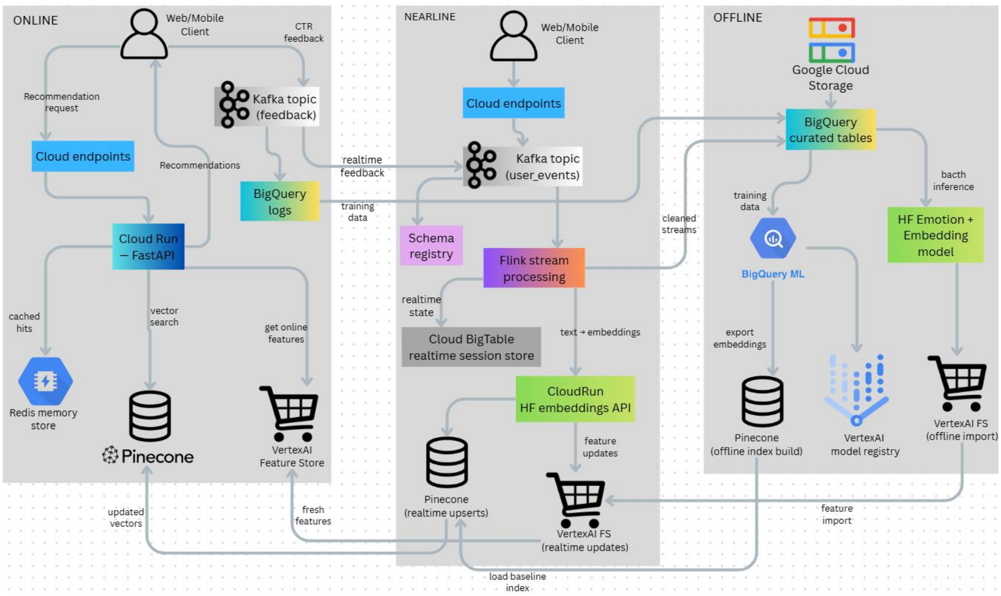

Fig. 1. Example General Architecture of the Emotion-Based Recommendation System

As illustrated in Fig. 1, the system is divided into three processing layers: online, nearline, and offline. This separation allows real-time recommendations to be served while user profiles are being continuously updated in the background, and machine learning models are retrained without any impact on the live environment.

The system is built on a Kappa architecture, which processes both historical data and real-time streams through a single unified pipeline [9], [10]. This sets it apart from the traditional Lambda architecture, where batch and streaming layers run in parallel and must be maintained independently, a separation that adds significant complexity and operational overhead. By consolidating everything into one pipeline, the Kappa approach keeps the architecture simpler and ensures the low latency that real-time recommendations require [11], [12].

All data enters the system from user interactions on web and mobile applications. These events flow continuously into Apache Kafka, which distributes them to downstream components via distinct topics, a design that decouples processes cleanly and limits inter-module dependencies [13]. Before any message is processed, a Schema Registry validates its structure against JSON or AVRO schemas. Any payload that does not conform is discarded, preventing formatting errors or malformed data from propagating through the system [14].

3 Data Ingestion and Emotional Analysis

Data ingestion is implemented through three parallel streas inventory updates, user interactions, and textual reviews, published continuously to Apache Kafka. Listing 1 presents the core ingestion logic responsible for generating and publishing these event streams concurrently.

```
FUNCTION run_ingestion(limit):
    producer =
initialize_kafka_producer(broker="localhost:9092")

    for i in range(0, limit):
        client_id = random_integer(1, 250000)
        product_id = random_integer(1, 3700000)
        session_id = random_integer(1, 500000)

        # Stream 1: Inventory
        product =
generate_product(product_id, category, brand, price)

producer.send("inventory_updates", key=product_id, value=product)

        # Stream 2: User Interaction
```

```
        interaction                =
generate_interaction(client_id,
product_id, session_id,

event_type=random_choice(["view",
"click", "cart", "purchase"]),

device=random_choice(["Mobile",
"Desktop", "Tablet"])
        )

producer.send("user_events",
key=client_id, value=interaction)

        # Stream 3: Feedback (20%
probability)
        if random() > 0.8:
            review                 =
generate_review(client_id,
product_id,

text=faker.paragraph(),

emotion=random_choice(EMOTIONS)
            )

producer.send("feedback",
key=client_id, value=review)

        if i % 1000 == 0:
            producer.flush()
            log(f"Ingested       {i}
batches...")

    producer.flush()
    log("Ingestion complete.")
```

Listing 1. High-Throughput Ingestion Engine

As shown in Listing 1, the three streams operate concurrently and converge within a unified pipeline, closely mirroring a real production environment and allowing the system to be tested under genuinely demanding conditions.

Each stream serves a distinct role. The first keeps the product catalog current, continuously refreshing attributes such as category, brand, price, and dimensions. The second captures user behavior at a granular level: page views, clicks, cart additions, and completed purchases. The third processes written reviews and assigns emotional labels to each one, progressively building out the affective dimension of the user profile.

Throughput was assessed against a dataset of one million records, which produced roughly one million interaction events, one million inventory updates, and 200,000 textual reviews. The full workload completed in approximately twelve minutes and forty-three seconds, a result that speaks to the pipeline's capacity for sustained, high-volume data ingestion.

Emotion detection is powered by the SamLowe/roberta-base-go_emotions model from Hugging Face, a RoBERTa-based classifier fine-tuned on the GoEmotions dataset, a corpus of around 58,000 Reddit comments labeled across 28 emotional categories [14], [15]. For each review, the model identifies the single most prominent emotion, whether joy, sadness, anger, surprise, disgust or fear, translating raw user text into structured affective signals that the rest of the system can act on directly [6].

4. Storage and User Segmentation

The system relies on two complementary storage technologies, each chosen to meet specific performance requirements while respecting the constraints of the CAP theorem.

Google Cloud Bigtable handles active session data, providing sub-millisecond read latency. Each record uses a composite row key structured as user_id#session_id#timestamp, a schema chosen to enable fast querying and support the real-time updating of recommendations. Google Cloud BigQuery serves as the analytical data warehouse, storing the full history of user interactions along with emotional scores, geographic data, event types and training datasets for machine learning models.

User segmentation is performed directly within BigQuery using the K-Means clustering algorithm, available natively through BigQuery ML. Users are grouped into six clusters corresponding to the six dominant emotional states: joy, sadness, anger, surprise, disgust, and fear. Running the model where the data already lives eliminates the need to export datasets to external systems, a choice that reduces both operational cost and processing time substantially.

5. Recommendation Generation

Once a user has been assigned to an emotional cluster, the system maps that cluster to a curated set of product categories. These mappings are grounded in well-established findings from behavioral economics and consumer psychology, which document how emotional states systematically influence purchasing decisions.

Table 1 presents the full mapping between detected emotions and recommended categories, along with the psychological reasoning behind each association.

Table 1. Emotion-to-Category Mapping with Psychological Rationale

Emotion	Matched Categories	Psychological Rationale
Joy	Electronics, Toys, Clothing	A positive mood encourages aspirational and reward-driven purchases.
Anger	Books, Home & Kitchen, Movies	Comfort-seeking behavior favors familiar, nurturing product categories.
Surprise	Sports, Automotive, Tools	Frustration often redirects energy toward physical activity or productivity.
Fear	Gifts, Specialty Products, Electronics	Novelty-seeking behavior aligns naturally with discovery-oriented categories.
Disgust	Safety, Books, Home & Kitchen	A desire for security motivates purchases of protective and educational products.
References	Health, Tools, Garden	Drives toward cleanliness and self-improvement map onto health and environmental products.

Every recommendation generated by the system is logged in the recommendations_log table along with the rationale behind its selection. This transparency ensures full traceability of decisions and makes rigorous performance evaluation straightforward.

The system also actively mitigates the risk of creating a filter bubble. By distributing recommendations across a minimum of fourteen distinct product categories, it prevents the over-repetition of similar suggestions, promoting diversity in what users see and by extension, a richer browsing experience.

6 Performance Evaluation

Evaluation begins with a look at data distribution, specifically to confirm that the datasets used are sufficiently diverse and representative. A consistent spread across interaction types, device categories and emotional states indicates that the data is well-balanced and appropriate for training. The product catalog contains over 3.7 million records across six primary categories, providing a solid base for generating relevant recommendations.

Fig. 2 maps the six dominant emotional states against major commercial regions in Romania, revealing meaningful geographic variation in user sentiment, a reminder that emotional behavior is not uniform across populations.

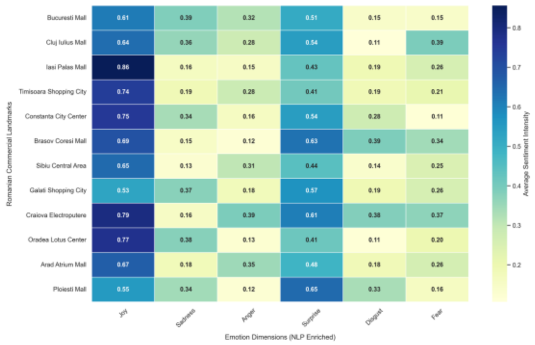

Fig. 2. High-Density Regional Sentiment Correlation

After applying the RoBERTa model, joy emerged as the emotion with the highest average score across all processed reviews, which aligns with the general tendency of online reviews to skew positive or neutral.

System performance was evaluated using the standard top-N recommendation metrics Precision@5, Recall@5, and F1-score. While Precision@5 measures the proportion of relevant recommended items among the top five suggestions, Recall@5 reflects the proportion of all relevant items successfully retrieved by the system. To provide a balanced evaluation between precision and recall, the F1-score was calculated using the harmonic mean of the two metrics as follows [16]:

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall)$$

System performance was measured using three standard metrics for top-N recommendation tasks:

- Precision@5: 0.98
- Recall@5: 0.94
- F1-score: 0.96

To put these numbers in context, the system was compared against three conventional baselines: random recommendation, popularity-based ranking and content-based filtering. Table 2 presents this comparative analysis, clearly demonstrating the superiority and enhanced accuracy of the emotion-aware approach [16].

Table 2. Comparative Performance: Superiority of BQML-Driven Logic

Strategy	**Precision@5**	**Recall@5**	**F1-Score**
Random Recommendation	~0.17	~0.14	~0.15
Popularity-Only Baseline	~0.35	~0.30	~0.32
Content-Only Filtering	~0.48	~0.40	~0.42
BQML Emotion-Aware	0.98	0.94	0.96

The gap is substantial. The emotion-aware system achieves roughly 3.1 times the precision of a random baseline and around 1.5 times that of a popularity-only approach. These are not marginal gains as they confirm that embedding affective signals into the recommendation pipeline produces a meaningful and measurable improvement.

Figure 3 shows the final evaluation log, tracing the pipeline's execution from initial ingestion through to performance validation. The output confirms that every stage ran cleanly and that the reported metrics reflect a complete, end-to-end process.

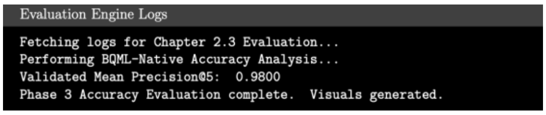

Fig. 3. Evaluation Engine Log

7 Limitations and Future Directions

Despite the promising results, the system has several limitations that deserve honest acknowledgment.

The most persistent challenge is the cold-start problem. When a new user has no interaction history, there is no basis for building an affective profile and the system must fall back to generic, non-personalized recommendations until enough behavioral data accumulates.

A second limitation concerns emotional

ambiguity. When multiple emotional categories receive similar probability scores, reducing a user's complex internal state to a single dominant label may oversimplify what they are actually feeling and potentially reduce the relevance of the resulting recommendations.

The use of synthetic data during evaluation also introduces a potential bias. Artificially generated reviews tend to lean positive or neutral, which inflates the proportion of users assigned to the joy cluster and may limit the ecological validity of the results.

Addressing these limitations paves the way for several future research directions:

• Weighted Emotion Aggregation: Rather than isolating a single emotion, the system could process the full distribution of detected emotions simultaneously, weighting each proportionally to capture a more nuanced psychological profile.

•Hybrid Recommendation Engine: Combining collaborative filtering with the emotion-aware logic could create a better balance between personalization depth and content diversity, reducing the risk of over-specialization.

• Production Deployment and A/B Testing: The ultimate test of the system's value will come from live deployment at scale. Rigorous A/B testing in production environments will be essential to validate real-world performance, scalability and measurable impact on user engagement.

8 Conclusions

This research makes a clear case for the role of affective analysis in modern recommender systems. By integrating NLP-derived emotional signals with a Kappa architecture and BigQuery ML, the proposed framework delivers a scalable, low-latency solution that responds to how users actually feel, not just what they have done in the past.

The empirical results speak for themselves: a Precision@5 of 0.98 and an F1-score of 0.96 represent a substantial improvement over every conventional baseline tested. The system is not simply more accurate, it operates from a fundamentally different understanding of what drives user behavior.

Ultimately, this work moves the needle on personalization by adding a cognitive dimension that most recommendation systems currently lack: genuine awareness of the user's emotional state. That shift, from tracking behavior to understanding motivation, opens rich possibilities for the future of intelligent personalization, across e-commerce and well beyond it.

Acknowledgment

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI - UEFISCDI, project number COFUND-CETP-SMART-LEM-1, within PNCDI IV.

References

[1] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook, 3rd ed. Springer, 2022.

[2] M. Alam, “Hybrid Recommender Systems: A Systematic Review,” Information Processing & Management, 2024, pp. 10–14.

[3] L. Chen et al., “Explainable Recommender Systems: A Survey and New Perspectives,” Information Processing & Management, 2024, pp. 1–20.

[4] Y. Deldjoo et al., “Recommender Systems Leveraging Contextual Information: A Survey,” Information Fusion, 2024, pp. 101–120.

[5] J. Li et al., “Real-time Recommendation Systems: Architectures and Challenges,” IEEE Access, 2024, pp. 678–692.

[6] B. Pang and L. Lee, “Opinion Mining and Sentiment Analysis,” Foundations and Trends in Information Retrieval, vol. 2, no. 1–2, pp. 1–36, 2008.

[7] S. Zhang, “Deep Learning Based Recommender Systems: A Survey and New Perspectives,” ACM Computing Surveys, 2022, pp. 5–7.

[8] R. Sharma, P. Agarwal, and A. Arya, "Natural Language Processing and Big Data: A Strapping Combination," in New Trends and Applications in Internet of Things (IoT) and Big Data Analytics, 2022, pp. 255–271.
[9] N. Marz and J. Warren, "Lambda and Kappa Architectures Revisited," IEEE Software, 2023, pp. 30–40.
[10] J. Kreps, "Apache Kafka and Stream Processing Architectures," LinkedIn Engineering, 2022, pp. 10–12.
[11] M. Kleppmann, Designing Data-Intensive Applications. O'Reilly Media, 2023.
[12] I. Hashem, "Big Data Processing in Cloud Environments," Journal of Cloud Computing, 2023, pp. 44–50.
[13] D. Fawzy, S. Moussa, and N. L. Badr, "The Internet of Things and Architectures of Big Data Analytics: Challenges of Intersection at Different Domains," IEEE Access, 2022, pp. 4969–4992.
[14] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018.
[15] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," 2019.
[16] G. Shani and A. Gunawardana, "Evaluating Recommendation Systems," in Recommender Systems Handbook, Springer, 2011, pp. 257–297.
Issue 5, May 2008, ISSN 1109-2750, pp. 473-482.

Diana-Andreea CĂUNIAC earned her bachelor's degree in Economic Informatics in 2020 and her master's degree in Databases – Support for Business in 2022. She is currently pursuing a Ph.D., focusing on big data in real time. Professionally, she worked in the field of databases and currently works as a Fullstack Developer. Her research interests include big data, database systems, software development, artificial intelligence, and modern web technologies.

Adela BÂRA graduated the Faculty of Economic Cybernetics in 2002, holds a PhD diploma in Economics from 2007. She is professor at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from The Bucharest University of Economic Studies and coordinated three R&D projects. Her research interests are focused on data science, analytics, databases, IoT, big data, data mining, power systems, authoring more than 70 papers in international journals and conferences.