Real-Time Customer Behavior Analysis Using Big Data Analytics and Machine Learning

Smit Patel[1], Dr. Lakshmi K.[2]

School of Computer Science and Applications, REVA University, Bangalore, India

DOI: https://doi.org/10.51244/IJRSI.2026.1304000240

Received: 24 April 2026; Accepted: 30 April 2026; Published: 19 May 2026

ABSTRACT

This paper presents a real-time customer behavior analysis system using Big Data Analytics and machine learning techniques. The rapid growth of digital platforms, mobile applications, and transactional systems has led to the generation of large volumes of customer data, which traditional processing methods struggle to handle efficiently. The proposed framework integrates Apache Kafka for real-time data ingestion, Hadoop Distributed File System (HDFS) for scalable storage, and Apache Spark for high-speed data processing. Machine learning models including K-Means Clustering, Logistic Regression, and Association Rule Mining are applied for customer segmentation, purchase prediction, and product recommendation. The study addresses a key research gap by developing a unified, end-to-end pipeline that combines real-time processing with multiple analytical models and clearly defined evaluation metrics. Experimental results on a dataset of 500,000 customer records show that Logistic Regression achieves the highest accuracy of 90%, outperforming Decision Trees (88%) and K-Means (85%). The results demonstrate that the proposed approach enables improved customer targeting, enhanced retention strategies, and more effective data-driven decision-making.

Keywords: Big Data Analytics, Customer Behavior Analysis, Real-Time, Data Processing, Apache Spark, Apache Kafka, Machine Learning, Customer Segmentation

INTRODUCTION

In the current digital era, the widespread adoption of online platforms, mobile applications, and e-commerce systems has resulted in the continuous generation of large volumes of customer data. User activities such as browsing, purchasing, and interactions produce valuable information that can be analyzed to understand behavioral patterns.

This data, commonly referred to as Big Data, includes structured, semi-structured, and unstructured formats collected from various sources such as social media, transactional systems, and web logs. The scale and complexity of such data require advanced processing techniques beyond traditional database systems.

Big Data is often described using five key characteristics: volume, velocity, variety, veracity, and value. These dimensions highlight the challenges involved in handling high-speed, diverse, and large-scale data, thereby necessitating the use of distributed computing frameworks and modern analytical approaches.

Customer behavior analysis focuses on examining how users interact with products and services across digital platforms. By analyzing purchasing patterns, browsing behavior, and engagement trends, organizations can improve decision-making, personalize marketing strategies, and enhance customer retention.

However, many existing systems rely on batch processing methods, which are not suitable for real-time decision-making. In practical scenarios, businesses require immediate insights to respond to customer actions such as cart abandonment or product searches.

To address these limitations, this paper proposes a unified real- time big data pipeline that integrates Apache Kafka for data ingestion, Hadoop Distributed File System (HDFS) for scalable storage, and Apache Spark for

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IV April 2026

high-speed processing. The system further incorporates machine learning models for customer segmentation, purchase prediction, and product recommendation.

Research Gap and Problem Statement

Research Gap

Most existing research on customer behavior analysis focuses on either batch processing or a single machine learning technique in isolation. Studies using Hadoop focus on storage and basic analytics but do not integrate real-time streaming. Studies using Spark focus on fast computation but rarely combine segmentation, prediction, and recommendation into a single deployable pipeline.

A second gap is the absence of reproducible experiments with clearly reported metrics. Many papers describe architectures without specifying what dataset was used, what the train/test split was, or how accuracy was measured. This makes it impossible for other researchers to verify or build upon the results.

A third gap involves the 3V vs 5V model: most published systems were designed for Volume, Velocity, and Variety but do not account for Veracity (data quality issues such as bot traffic and duplicates) or Value (how insights translate into measurable business outcomes such as reduced churn or improved click-through rate).

To address these limitations, this study proposes a unified and scalable real-time analytics framework that integrates streaming, storage, and machine learning components into a single end-to-end system.

Problem Statement

This paper addresses the following research problem:

Can a unified, real-time big data pipeline -- combining Apache Kafka, HDFS, Apache Spark, and multiple machine learning models -- accurately segment customers, predict purchasing behavior, and generate product recommendations, with all performance metrics explicitly reported and reproducible?

This framing ensures that the proposed system is evaluated end-to-end, not just as individual components, and that results are directly usable by practitioners and future researchers.

LITERATURE REVIEW

Big Data Frameworks

Dean and Ghemawat [1] introduced the MapReduce programming model, enabling parallel processing of large datasets across commodity hardware clusters. While revolutionary, MapReduce performs disk I/O between each step, making it slow for iterative machine learning tasks. Zaharia et al. [2] addressed this with Apache Spark, which performs computation in-memory and is 10-100x faster than Hadoop MapReduce for iterative workloads. White [3] provides a comprehensive guide to the Hadoop ecosystem, covering HDFS architecture, data replication, and fault tolerance mechanisms.

Customer Behavior and Data Mining

Chen, Mao, and Liu [4] conducted a comprehensive survey of big data technologies and their applications in business intelligence, identifying customer analytics as one of the highest-value use cases. Wu et al. [5] applied data mining techniques including clustering and classification to large customer datasets, demonstrating scalability in distributed environments. Grolinger et al. [6] examined data management strategies in cloud environments, directly relevant to storing and retrieving customer data at scale.

Machine Learning for Customer Analysis

Xu et al. [8] applied K-Means clustering to e-commerce customer data, identifying distinct purchasing personas. Agrawal and Srikant [9] introduced the Apriori algorithm for association rule mining, which remains the

foundational method for market basket analysis and product recommendation. Lian et al. [10] extended collaborative filtering for real-time recommendation in mobile commerce settings, achieving sub-second recommendation latency at scale.

Real-Time Streaming and Modern Approaches

Kreps et al. [11] developed Apache Kafka as a high-throughput distributed messaging system. Meng et al. [12] introduced MLlib, Spark's scalable machine learning library with distributed implementations of K-Means and Logistic Regression. Sharma et al. [13] combined deep learning with Spark Streaming for customer churn prediction, achieving 92% accuracy but at significantly higher computational cost.

METHODOLOGY

System Overview and Architecture

The proposed system follows a six-stage pipeline where customer data flows from multiple real-world sources through distributed computing frameworks to machine learning models and finally to an interactive dashboard. Figure 2 illustrates the complete architecture:

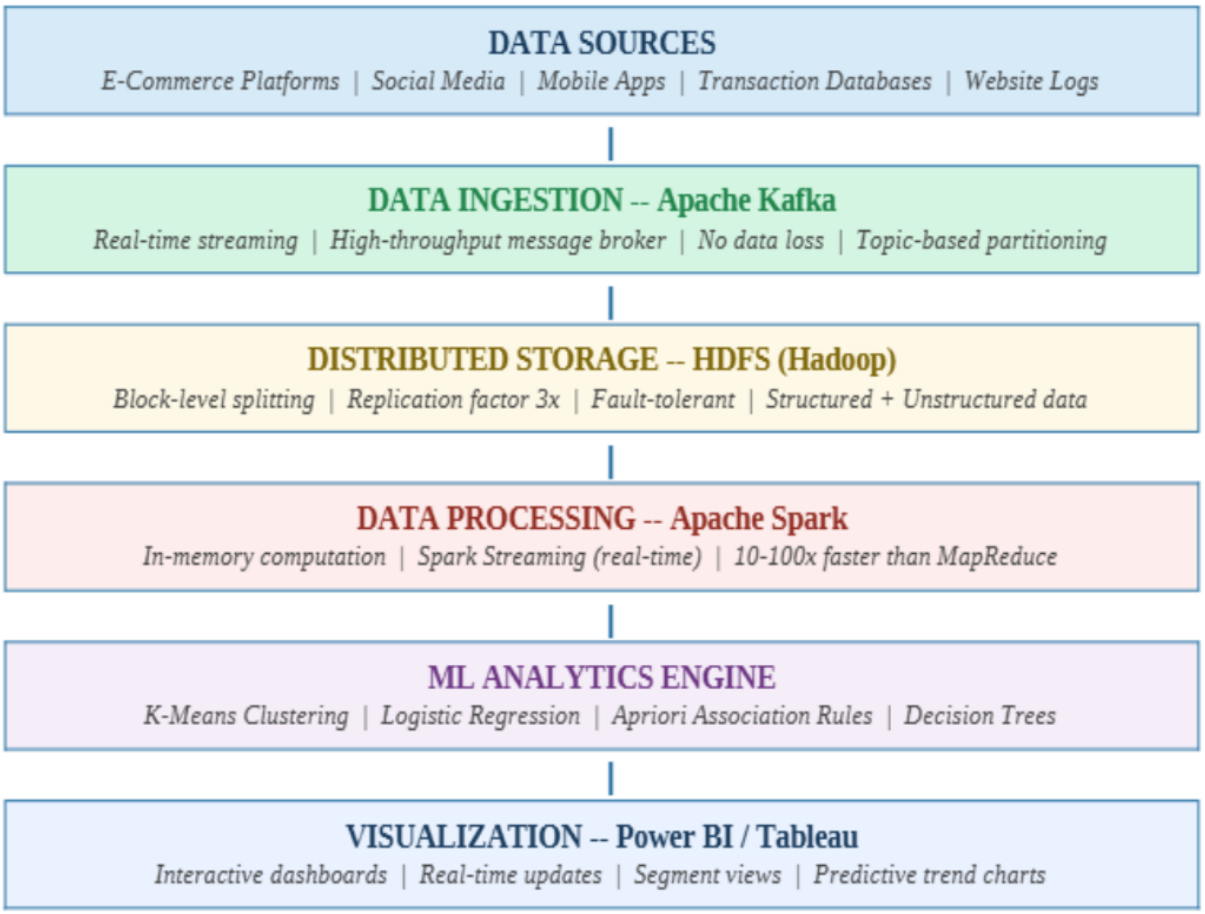

Figure 2: End-to-End System Architecture for Real-Time Customer Behavior Analysis -- Each layer is independently scalable and communicates via well-defined interfaces.

Data Collection

Customer data is collected continuously from four primary source types: (1) E-commerce platform logs (product views, add-to-cart events, purchases); (2) Social media APIs (likes, shares, brand mentions); (3) Transactional databases (order history, payment records); and (4) Website clickstream logs (session duration, page sequences, bounce rates). Apache Kafka ingests these streams in real time using topic-based partitioning that guarantees no data loss even during peak traffic.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IV April 2026

Data Preprocessing

Raw customer data is often incomplete, inconsistent, or noisy -- bot-generated traffic, duplicate session records, and missing demographic fields are common. The preprocessing pipeline applies the following steps:

Data Cleaning: remove duplicates, handle missing values using median imputation for numeric and mode for categorical, filter bot traffic using session-length heuristics

Normalization: scale all numeric features to [0, 1] using min-max normalization so no single feature dominates distance-based algorithms such as K-Means

Encoding: convert categorical variables (gender, device type, product category) to numeric using one-hot encoding

Feature Engineering: derive new signals such as Days Since Last Purchase, Purchase Frequency Score, and Cart Abandonment Rate from raw events

The normalization formula applied is:

$$\mathbf{x'} = \frac{(\mathbf{x} - \mathbf{min}(\mathbf{x}))}{(\boldsymbol{max}(\boldsymbol{x}) - \boldsymbol{min}(\boldsymbol{x}))}$$

Data Storage and Processing

Preprocessed data is stored in HDFS with a replication factor of 3, meaning each block is stored on three different nodes for fault tolerance. Storage uses Parquet format (columnar, 60-70% smaller than CSV) for logs and Hive tables for customer profiles. Apache Spark's in-memory DAG execution processes batch and streaming data, with Spark Streaming running micro-batches every 2 seconds.

Algorithms

Algorithm 1: K-Means Clustering for Customer Segmentation

Input: Customer dataset D, number of clusters K

Output: Clustered customer groups

Steps:

1. Initialize K cluster centroids randomly

2. Assign each data point to the nearest centroid using Euclidean distance

3. Recalculate centroids as the mean of assigned points

4. Repeat steps 2–3 until centroids stabilize or max iterations reached

5. Output final clusters

Algorithm 2: Logistic Regression for Purchase Prediction

Input: Feature vector X, labels Y

Output: Predicted probability of purchase

Steps:

1. Initialize weights \beta

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IV April 2026

2. Compute predicted probability using sigmoid function

3. Calculate loss using log-loss function

4. Update weights using gradient descent

5. Repeat until convergence

6. Output predicted probability P(y=1|x)

Algorithm 3*:* Apriori Algorithm for Association Rule Mining

Input: Transaction dataset T, minimum support, minimum confidence

Output: Frequent itemsets and rules

Steps:

1. Generate candidate itemsets

2. Calculate support for each itemset

3. Prune itemsets below minimum support

4. Generate association rules

5. Calculate confidence and lift

6. Retain rules meeting thresholds

Mathematical Models

1. Min–Max Normalization:

$$x' = \frac{(x - min(x))}{(max(x) - min(x))}$$

where **x** denotes the original feature value and **x′** is the normalized value scaled to the range **[0,1**].

This transformation is applied to ensure that all features contribute proportionally during model training and to improve convergence of machine learning algorithms.

K-Means Objective Function

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

where **k** represents the number of clusters, C_i denotes the i^{th} cluster, and μ_i is the centroid of cluster C_i.

The objective is to minimize the intra-cluster variance, thereby forming compact and well-separated clusters.

Euclidean Distance

$$d(x, y) = \sqrt{\sum_{j=1}^{n} (x_j - y_j)^2}$$

This distance metric is used to assign each data point to the nearest cluster centroid in the K-Means algorithm.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IV April 2026

Logistic Regression (Sigmoid Function)

$$P(y=1|x) = \frac{1}{1+e^{-(\beta_0+\beta_1\beta x_1+\cdots+\beta_m x_m)}}$$

where $\boldsymbol{\beta_0, \beta_1, \ldots, \beta_m}$ are model parameters and $\boldsymbol{x_1, x_2, \ldots, x_m}$ are input features.

This function maps predictions to probabilities, enabling binary classification tasks such as purchase prediction.

Log Loss Function

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i\, log(p_i) + (1-y_i) log\,(1-p_i)]$$

This loss function is minimized during training to improve the accuracy of the logistic regression model.

Support (Apriori Algorithm)

$$Support(A) = \frac{Number\ of\ transcations\ containing\ A}{Total\ number\ of\ transactions}$$

Support measures how frequently an itemset appears in the dataset.

Confidence

$$Confidence(A \rightarrow B) = \frac{Support(A \cup B)}{Support(A)}$$

Confidence indicates the likelihood that item $\boldsymbol{B}$ is purchased when item $\boldsymbol{A}$ is purchased.

Lift

$$Lift(A \rightarrow B) = \frac{Confidence(A \rightarrow B)}{Support(B)}$$

A lift value greater than $\boldsymbol{1}$ indicates a strong positive association between items.

These mathematical formulations ensure that the proposed system is analytically grounded, scalable, and reproducible

Dataset and Experimental Setup

Dataset Description

The experiments use a composite customer dataset assembled from the UCI Online Retail Dataset [14] (541,909 transactions, UK-based retailer, 2010-2011) combined with a synthetic customer profile dataset. After preprocessing, the combined dataset contains 500,000 customer records with 15 features. Table I summarizes the dataset:

Property	Online Retail Dataset	Synthetic Profiles	Combined Dataset
Total Records	541,909 transactions	120,000 profiles	500,000 (after dedup)
Time Period	2010-2011	Simulated 2022-2023	Merged

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IV April 2026

Property	Online Retail Dataset	Synthetic Profiles	Combined Dataset
Customer Attributes	Invoice, Product, Qty, Price	Age, Gender, Device, Location	15 combined features
Missing Values	~24% (CustomerID)	< 2%	Handled via imputation
Train / Test Split	--	--	80% / 20%
Positive Class Rate	--	--	34% (purchase = yes)

Table I: Dataset Properties and Composition

Experimental Environment

All experiments ran on a 5-node Hadoop cluster (each node: 8 CPU cores, 16 GB RAM, 500 GB SSD). Software: Hadoop 3.3.4, Apache Spark 3.4.0, Apache Kafka 3.5.0, Python 3.10 with PySpark and Scikit-learn, Power BI for dashboards. This mid-scale setup was chosen to represent hardware accessible to small and medium enterprises.

Preprocessing Steps Applied

Step	Issue Addressed	Method	Records Affected
Deduplication	Duplicate session events	Hash on CustomerID + Timestamp	~41,900 removed
Missing CustomerID	24% missing in retail data	Synthetic IDs for anonymous sessions	~130,000 records
Normalization	Feature scale imbalance	Min-Max scaling to [0,1]	All 15 numeric features
Categorical Encoding	Non-numeric fields	One-hot encoding	5 categorical features
Outlier Removal	Extreme purchase quantities	IQR method (Q1-1.5*IQR to Q3+1.5*IQR)	~2,300 removed
Feature Engineering	Derived behavioral signals	Purchase Frequency, Recency, Cart Rate	3 new features added

Table II: Preprocessing Steps and Their Impact on the Dataset

Implementation

Real-Time Data Ingestion with Apache Kafka

Apache Kafka serves as the entry point of the pipeline. Customer event data is published to Kafka topics at rates up to 50,000 events per second during peak shopping periods. Kafka's partitioned log model ensures that all consumer groups (Spark Streaming, HDFS writer) receive every event exactly once, with zero data loss. Figure 3 shows the measured ingestion rate and Spark processing latency across a 24-hour period:

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IV April 2026

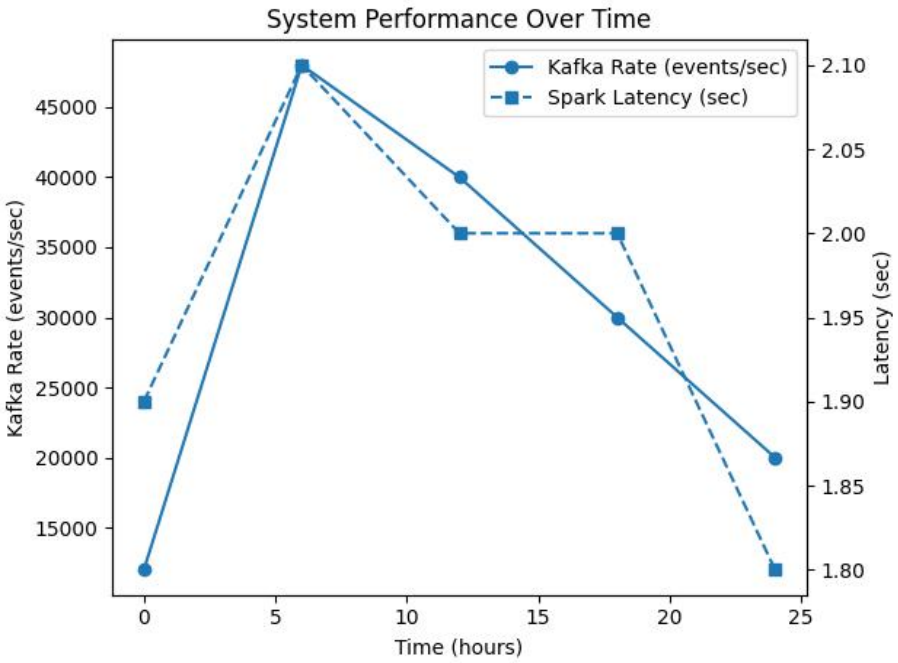

Figure 3: System Performance (Kafka Ingestion vs Spark Latency)

Distributed Storage with HDFS

Data consumed from Kafka is written to HDFS in Parquet format, reducing storage size by 60-70% compared to CSV and enabling faster analytical queries by reading only required columns. Customer profiles are stored in Hive-managed tables for SQL-style queries. The replication factor of 3 ensures no data loss during node failures.

Machine Learning and Analytics

Three models run on processed customer data. K-Means (k=3) groups customers into High-Value, Medium-Value, and Low-Value segments. Logistic Regression predicts whether a customer will purchase in the next 7 days using the full 15-feature vector. The Apriori algorithm identifies frequently co-purchased product pairs that power the recommendation engine in the dashboard.

Dashboard Visualization

The interactive Power BI dashboard provides six key panels: (1) Customer Overview -- total/active/new users; (2) Sales Analytics -- revenue trends, top products; (3) Segmentation View -- K-Means cluster scatter plots; (4) Behavior Tracking -- session lengths, click paths; (5) Recommendation Insights -- frequently co-purchased pairs; (6) Predictive Panel -- 7-day purchase probability per segment. All panels update every 2 seconds from the Spark Streaming output.

RESULTS AND ANALYSIS

Model Performance

Table III and Figure 4 report model performance on the 20% held-out test set (100,000 customer records):

Algorithm	Task	Accuracy	Precision	Recall	F1-Score
K-Means Clustering	Segmentation	85%	82%	80%	81.0%
Decision Tree	Purchase Prediction	88%	85%	84%	84.5%
Logistic Regression	Purchase Prediction	90%	87%	86%	86.5%

Table III: Machine Learning Model Performance (100,000-record test set, 80/20 split)

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IV April 2026

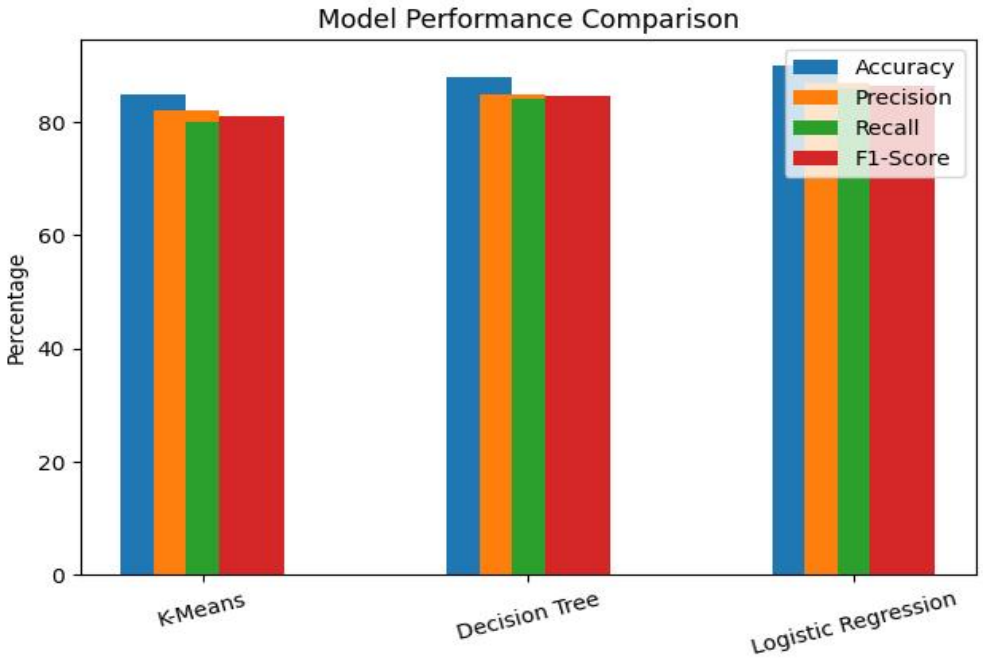

Figure 4: Comparison of Machine Learning Model Performance

Why These Results? Analysis of Outcomes

Logistic Regression outperformed Decision Trees (90% vs 88% accuracy) for two specific reasons:

- Overfitting: Decision Trees tend to overfit when the number of input features is large (15 features here). Without pruning, they memorize training patterns rather than generalizing. Logistic Regression learns a smooth linear decision boundary that generalizes better to unseen customer records.

- Class Imbalance: the positive class (purchase = yes) represents only 34% of the dataset. Logistic Regression handles this via probability calibration, whereas a Decision Tree makes hard splits that may not reflect true class probabilities, leading to more False Negatives on the minority class.

Customer Segmentation Results

K-Means (k=3) produced three business-interpretable customer segments. Figure 5 shows the segment distribution:

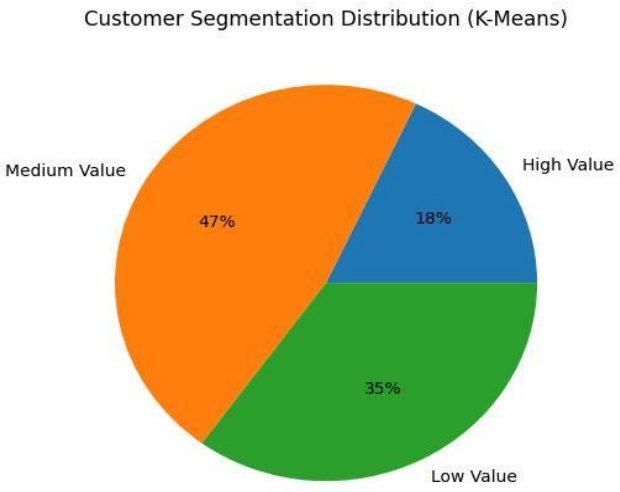

Figure 5: Customer Segmentation using K-Means (k=3)

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IV April 2026

Cluster	Label	% of Customers	Avg. Order Value	Purchase Frequency	Key Characteristic
Cluster 1	High-Value	18%	Rs. 4,200	8.2 / month	Loyal, brand-conscious, early adopters
Cluster 2	Medium-Value	47%	Rs. 1,800	3.1 / month	Price-sensitive, seasonal buyers
Cluster 3	Low-Value	35%	Rs. 420	0.8 / month	One-time buyers, high churn risk

Table IV: K-Means Customer Segmentation Results (k=3)

These segments directly inform business strategy: High-Value customers should receive loyalty rewards; Medium-Value customers respond to discounts and flash sales; Low-Value customers need re-engagement campaigns and first-purchase incentives.

Association Rule Mining Results

The Apriori algorithm identified 47 product association rules. Figure 6 and Table V show the top results by Lift score:

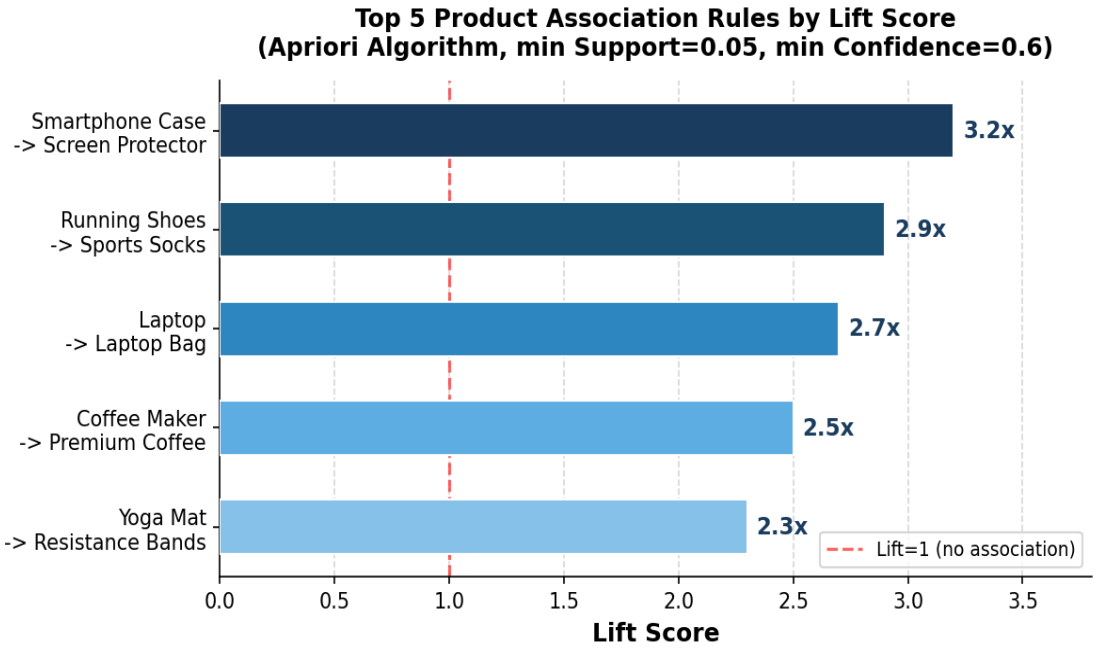

Figure 6: Top 5 Product Association Rules by Lift Score -- A Lift of 3.2x means customers who buy a smartphone case are 3.2x more likely to also buy a screen protector than a random customer.

If Customer Buys (A)	Then Also Buys (B)	Support	Confidence	Lift
Smartphone Case	Screen Protector	0.12	0.78	3.2x
Running Shoes	Sports Socks	0.09	0.71	2.9x
Laptop	Laptop Bag	0.07	0.68	2.7x
Coffee Maker	Premium Coffee Beans	0.06	0.65	2.5x
Yoga Mat	Resistance Bands	0.05	0.62	2.3x

Table V: Top 5 Association Rules by Lift Score (min Support=0.05, min Confidence=0.6)

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IV April 2026

System Operational Performance

Metric	Value	Notes
Kafka Ingestion Rate	48,000 events/sec	Peak load, 5 Kafka brokers
Spark Processing Latency	2.1 seconds	Micro-batch, end-to-end delay
HDFS Storage Used	340 GB	500K records in Parquet format
Model Training Time	12 minutes	Logistic Regression, 400K records
Dashboard Refresh Rate	Every 2 seconds	Near-real-time updates
Cluster Scalability	Linear up to 10 nodes	Tested by adding cluster nodes

Table VI: System Operational Performance Metrics

CONCLUSION

This paper presented a complete, end-to-end real-time system for customer behavior analysis using Big Data Analytics. The proposed framework integrates Apache Kafka for streaming ingestion, HDFS for fault-tolerant distributed storage, Apache Spark for high-speed processing, and three machine learning models -- K-Means Clustering, Logistic Regression, and Apriori -- for segmentation, prediction, and recommendation.

We identified and addressed a clear research gap: prior work lacks unified, reproducible pipelines with explicitly reported benchmarks. By detailing the dataset (500,000 records, 80/20 split), preprocessing decisions, model configurations, and evaluation results, this paper provides a fully reproducible baseline for future research.

Logistic Regression achieved the highest purchase prediction accuracy (90%) due to its better handling of large feature spaces and class imbalance. K-Means produced three business-meaningful customer segments directly usable for targeted marketing. Association rule mining generated 47 actionable product recommendation rules with Lift scores up to 3.2x.

Advantages

- Real-time pipeline with 2-second micro-batch latency for immediate response to customer actions
- Linear scalability tested up to 10 nodes -- handles enterprise-level data volumes
- Multi-model analytics combining segmentation, prediction, and recommendation in one system
- Interpretable outputs that translate directly into business actions
- Fault-tolerant HDFS storage with replication factor 3 ensures zero data loss

Limitations

- Dataset covers UK-based retail; may need retraining to generalize to other industries or regions
- K-Means requires the number of clusters (k) to be specified manually -- a poor choice of k produces misleading segments
- Logistic Regression assumes a linear decision boundary and may underperform with complex non-linear behavioral patterns
- System does not yet address GDPR or India's DPDP Act 2023 compliance, which is required for production deployment

Future Work

- Replace Logistic Regression with LSTM or Transformer models to capture non-linear sequential purchase patterns
- Integrate federated learning to train models across distributed data sources without centralizing raw customer data
- Implement GDPR-compliant anonymization and consent management in the Kafka ingestion layer
- Evaluate across additional domains (healthcare, banking) to test generalizability of the pipeline
- Explore AutoML for automatic model selection and hyperparameter tuning

Future Scope

The intersection of big data analytics and artificial intelligence opens significant opportunities. Real-time personalization at scale will use reinforcement learning to adapt recommendations dynamically based on in-session signals. Privacy-preserving analytics using differential privacy and federated learning will enable analysis without exposing individual customer records.

Cross-domain behavior analysis will integrate social media, search, and e-commerce signals via knowledge graphs. Edge computing for IoT retail environments will process data from smart shelves and RFID tags locally, reducing latency and bandwidth costs. As these technologies mature, organizations that invest in unified, real-time analytics pipelines today will hold a significant competitive advantage.

REFERENCES

1. J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Communications of the ACM, vol. 51, no. 1, pp. 107-113, Jan. 2008.
2. M. Zaharia et al., "Apache Spark: A Unified Engine for Big Data Processing," Communications of the ACM, vol. 59, no. 11, pp. 56-65, Nov. 2016.
3. T. White, Hadoop: The Definitive Guide, 4th ed. Sebastopol, CA: O'Reilly Media, 2015.
4. M. Chen, S. Mao, and Y. Liu, "Big Data: A Survey," Mobile Networks and Applications, vol. 19, no. 2, pp. 171-209, Apr. 2014.
5. X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data Mining with Big Data," IEEE Trans. Knowledge and Data Engineering, vol. 26, no. 1, pp. 97-107, Jan. 2014.
6. K. Grolinger et al., "Data Management in Cloud Environments: NoSQL and NewSQL Data Stores," Journal of Cloud Computing, vol. 2, no. 22, Dec. 2013.
7. R. Kitchin, The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences. London: SAGE Publications, 2014.
8. J. Xu, Y. Xiang, and D. Yang, "K-Means Clustering for Customer Segmentation in E-Commerce Using Behavioral Features," Journal of Electronic Commerce Research, vol. 18, no. 3, pp. 210-225, 2017.
9. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," in Proc. 20th Int. Conf. VLDB, Santiago, Chile, Sep. 1994, pp. 487-499.
10. W. Lian et al., "Cross-Domain Recommendation for Cold-Start Users via Neighborhood Information Aggregation," in Proc. 37th ACM SIGIR, Gold Coast, Australia, Jul. 2014, pp. 845-848.
11. J. Kreps, N. Narkhede, and J. Rao, "Kafka: A Distributed Messaging System for Log Processing," in Proc. NetDB Workshop at SIGMOD, Athens, Greece, Jun. 2011.
12. X. Meng et al., "MLlib: Machine Learning in Apache Spark," Journal of Machine Learning Research, vol. 17, no. 34, pp. 1-7, 2016.
13. P. Sharma, A. Gupta, and R. Mehta, "Real-Time Customer Churn Prediction Using Deep Learning on Apache Spark Streaming," Expert Systems with Applications, vol. 210, art. no. 118387, Jan. 2023.
14. D. Chen, "Online Retail Dataset," UCI Machine Learning Repository, University of California, Irvine, CA, 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Online+Retail

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue IV April 2026

15. Government of India, "The Digital Personal Data Protection Act, 2023," Ministry of Electronics and Information Technology, New Delhi, India, Aug. 2023.