

SmartCost: Smart Order Recommendation System for Micro-SMEs

K.A.C.H Kodithuwakku

Faculty of Computing

*Sri Lanka Institute of Information
Technology*

Malabe, Sri Lanka

IT22544872@my.sliit.lk

D.D.M Jayasingha

Faculty of Computing

*Sri Lanka Institute of Information
Technology*

Malabe, Sri Lanka

IT22162250@my.sliit.lk

N.K.M.Perera

Faculty of Computing

*Sri Lanka Institute of Information
Technology*

Malabe, Sri Lanka

IT22151810@my.sliit.lk

M.P.J.R.Dasanayaka

Faculty of Computing

*Sri Lanka Institute of Information
Technology*

Malabe, Sri Lanka

IT22152664@my.sliit.lk

Deemantha Siriwardana

Faculty of Computing

*Sri Lanka Institute of Information
Technology*

Malabe, Sri Lanka

deemantha.s@sliit.lk

Ayesha Wijesooriya

Faculty of Computing

*Sri Lanka Institute of Information
Technology*

Malabe, Sri Lanka

ayesha.w@sliit.lk

Abstract—Small and medium-sized enterprises (SMEs) commonly use point-of-sale (POS) systems for transaction recording but often lack intelligent tools to support cross-selling and upselling at checkout. This research proposes a Smart Order Recommendation System that generates real-time product suggestions from historical transaction data. Association rule mining is used to identify frequently co-purchased products, with recommendation rules evaluated using support, confidence, and lift metrics. The system is integrated into the POS checkout workflow using a lightweight architecture based on React Native, Node.js, FastAPI, and MongoDB. Experimental evaluation used a bakery and café dataset containing 15,420 transactions, 128 products, 12 months of sales records, and an average cart size of 3.8 items. The model achieved 84% precision, 79% recall, and 82% average recommendation confidence, with a 142 ms average response latency. It also improved sales opportunities by 11% and cross-selling effectiveness by 18%, demonstrating practical value for SME retail environments.

Keywords—Smart Order Recommendation, Association Rule Mining, POS Systems, Cross-Selling, Data Mining, SMEs.

I. INTRODUCTION

Small and medium sized enterprises (SMEs) play an important role in economic development by supporting employment generation, local businesses, and retail growth. In sectors such as bakeries, cafés, and small retail stores, SMEs rely heavily on point of sale (POS) systems to manage transactions, billing, and inventory operations. Although modern POS systems improve operational efficiency, many existing solutions mainly focus on transaction recording rather than intelligent sales enhancement and customer engagement.

One major challenge faced by SMEs is the underutilization of transactional data generated during daily business operations. Most businesses collect large amounts of sales data through POS systems but rarely use this data to support intelligent decision making. As a result, cross selling and upselling opportunities are frequently missed during

customer transactions. In many SME environments, recommendation decisions are still performed manually based on cashier experience rather than analytical evidence. This manual recommendation process often leads to inconsistent promotional strategies and reduced sales optimization.

Another limitation of traditional POS systems is the absence of intelligent checkout assistance. Existing SME POS platforms generally do not provide real time product suggestions capable of recommending related items while customers are actively purchasing products. Consequently, businesses fail to maximize average order value and customer engagement opportunities. This issue is particularly important in food service environments where customers commonly purchase associated products together, such as coffee with cake or sandwiches with beverages.

Recent advancements in data mining and recommender systems have introduced effective techniques for identifying purchasing relationships within transactional datasets. Association rule mining techniques are widely used to discover frequent itemsets and product relationships from historical sales data [1], [4], [13]. These techniques form the foundation of market basket analysis, which is commonly applied in retail analytics to identify products frequently purchased together.

Recommender systems have gained significant attention because of their ability to improve customer engagement and sales performance [5], [6], [9]. Previous studies have explored collaborative filtering, item based recommendation techniques, and hybrid recommendation approaches for generating personalized suggestions [6]–[10]. Such systems have demonstrated effectiveness in increasing cross selling opportunities and improving purchasing experiences within large scale e commerce platforms.

The increasing availability of transactional data and business intelligence technologies has further strengthened the

importance of analytical decision support within retail environments [11], [12]. Organizations capable of utilizing data driven strategies are more likely to achieve improved operational efficiency and competitive advantage. However, many enterprise recommendation systems remain expensive, computationally complex, and difficult to deploy within SME environments. These systems often require dedicated infrastructures, advanced technical expertise, and high performance computing resources that exceed the practical capabilities of small businesses.

Another challenge is the fragmentation of existing analytical systems. Many business intelligence platforms focus on isolated functionalities such as dashboard visualization, customer analytics, or sales reporting independently rather than integrating intelligent recommendation capabilities directly into operational POS workflows. Existing systems also provide limited support for lightweight real time recommendation generation suitable for SME checkout environments.

SmartCost research project primarily focuses on offline analytics, dashboard visualization, customer behavior analysis, and long term sales insight generation. In contrast, specifically focuses on real time recommendation generation during active checkout transactions. The proposed Smart Order Recommendation System emphasizes checkout integration, low latency recommendation processing, and dynamic product suggestion generation capable of supporting operational sales enhancement during customer purchases.

To address these challenges, this research proposes a Smart Order Recommendation System that utilizes historical transactional data to generate real time product recommendations within SME oriented POS environments. The proposed system applies association rule mining techniques to identify frequently co purchased products and generate recommendation rules using support, confidence, and lift metrics. These recommendation rules are integrated directly into the checkout workflow to provide intelligent product suggestions whenever items are added to the cart.

The proposed architecture adopts React Native for the mobile frontend, Node.js for transaction management, FastAPI for recommendation processing, and MongoDB for transactional data storage [17]–[20]. The recommendation engine is designed to operate efficiently in low resource SME environments while maintaining low latency recommendation generation suitable for real time checkout scenarios.

Furthermore, the proposed system prioritizes simplicity, explainability, and scalability. Unlike complex deep learning recommendation systems, association rule mining provides interpretable recommendation logic that can be easily understood by cashiers and business owners. This improves system usability and trust while reducing technical barriers for SMEs with limited analytical expertise.

Overall, this research contributes a practical and scalable recommendation framework specifically designed for SME retail environments. By integrating association rule mining and real time checkout recommendation capabilities into POS

systems, the proposed solution demonstrates how transactional data can be transformed into actionable business intelligence capable of improving sales performance, customer engagement, and operational efficiency.

A. Research Objectives

The primary objective of this research is to develop a lightweight Smart Order Recommendation System capable of generating real-time product recommendations within SME-oriented Point-of-Sale (POS) environments. The system is designed to integrate seamlessly with POS checkout workflows by implementing a real-time recommendation engine that identifies frequently co-purchased products using association rule mining techniques. Recommendation rules are generated based on key measures such as support, confidence, and lift to provide accurate and relevant product suggestions during customer transactions. Furthermore, the system emphasizes low-latency processing to ensure efficient performance in resource-constrained SME environments while enhancing cross-selling and upselling opportunities. Finally, the effectiveness of the proposed recommendation system is evaluated using performance metrics including precision, recall, confidence, and recommendation latency to assess its accuracy, efficiency, and practical applicability.

B. Main Contributions

The main contributions of this research include the development of a lightweight real-time recommendation framework specifically optimized for SME retail environments. The proposed system integrates an intelligent checkout recommendation engine capable of generating low-latency product suggestions by utilizing an association rule mining approach that is well suited for small-scale transactional datasets. In addition, the research presents a scalable system architecture that combines React Native, Node.js, FastAPI, and MongoDB for efficient operational deployment [17]–[20]. The proposed recommendation workflow is designed to enhance cross-selling opportunities and improve customer engagement during checkout transactions while maintaining computational efficiency. Furthermore, the system provides explainable product recommendations, making it practical and accessible for SMEs with limited technical expertise and resources.

II. LITERATURE REVIEW

The development of point of sale (POS) systems reflects the continuous evolution of retail and business technologies. Early POS systems were primarily designed for transaction recording, billing, and inventory tracking. Although modern systems have improved operational efficiency, most SME oriented POS platforms still focus mainly on transactional processing rather than intelligent recommendation generation and customer engagement support. In many SME environments, recommendation decisions continue to rely on cashier experience and manual observation rather than analytical intelligence.

The rapid growth of data mining, machine learning, and recommender system technologies has created opportunities

to transform transactional data into actionable business intelligence. Organizations capable of leveraging customer purchasing data effectively can improve operational efficiency, sales performance, and customer satisfaction [11], [12]. However, many advanced recommendation systems remain computationally expensive, difficult to deploy, and unsuitable for low resource SME environments.

A. Data Mining and Knowledge Discovery

Data mining techniques are widely used to identify hidden relationships, patterns, and trends within large datasets. Han et al. [1] describe data mining as a process of extracting meaningful knowledge from structured and unstructured data using techniques such as clustering, classification, and association analysis. Similarly, Witten et al. [2] explain that machine learning and data mining approaches can transform raw transactional records into actionable insights capable of supporting decision making.

Machine learning techniques further enhance analytical capabilities by enabling systems to identify patterns automatically from historical data [3]. In retail environments, these techniques support customer behavior analysis, purchasing pattern identification, and recommendation generation. However, many sophisticated machine learning models require large datasets and significant computational resources, limiting their practicality within SME environments.

Among available analytical approaches, association rule mining has emerged as one of the most suitable techniques for transactional recommendation systems because of its simplicity, scalability, and interpretability.

B. Association Rule Mining and Market Basket Analysis

Association rule mining is a widely adopted technique for discovering relationships between products within transactional datasets. The foundational work by Agrawal and Srikant [4] introduced efficient algorithms for identifying frequent itemsets and generating association rules using metrics such as support and confidence.

Market basket analysis applies association rule mining to identify products frequently purchased together within retail environments. These relationships enable businesses to implement cross selling and upselling strategies capable of improving sales performance. For example, customers purchasing coffee may also purchase cake or sandwiches with beverages. Such purchasing relationships can be identified automatically through transactional analysis.

The effectiveness of association rule mining lies in its ability to provide interpretable recommendation rules without requiring complex user modeling or extensive customer profiles. Unlike collaborative filtering approaches, association rule based systems rely primarily on transactional

relationships rather than explicit customer behavior histories. This makes them particularly suitable for SME environments with limited datasets and low computational resources.

In addition, association rule mining supports explainable recommendations, where business users can easily understand why suggestions are generated. This improves usability and trust compared to black box recommendation models.

C. Recommender Systems in Retail Environments

Recommender systems have become essential components of modern digital commerce platforms. Resnick and Varian [9] introduced recommender systems as tools capable of filtering information and presenting relevant suggestions to users. Subsequent research demonstrated that recommendation systems significantly influence customer purchasing behavior and increase sales opportunities [5], [6].

Collaborative filtering techniques, such as those proposed by Sarwar et al. [7], identify similarities between users or products to generate personalized recommendations. Deshpande and Karypis [8] further developed item based recommendation algorithms capable of generating top-N product recommendations. Although these techniques perform effectively within large scale e commerce platforms, they often require extensive user interaction histories and computational resources.

Enterprise recommendation platforms commonly utilize hybrid recommendation approaches combining collaborative filtering, content based filtering, and deep learning techniques [10]. While such systems provide high recommendation accuracy, they are often computationally intensive and difficult to deploy in SME retail environments. In contrast, lightweight association rule based systems provide efficient recommendation generation suitable for real time POS applications.

Unlike offline analytical systems, real time POS recommendation systems must generate suggestions instantly during active customer transactions. Therefore, low latency processing and operational simplicity become critical design requirements for SME oriented recommendation engines.

D. Big Data and Real Time Retail Intelligence

The emergence of big data technologies has significantly increased the ability of businesses to analyze large volumes of transactional information [11]. Chen et al. [11] explain that big data analytics enables organizations to process high volume datasets for improved operational intelligence and decision making. Similarly, McAfee and Brynjolfsson [12] emphasize that data driven organizations achieve improved competitiveness and operational performance through analytical decision support.

In retail environments, real time analytical systems allow businesses to generate recommendations dynamically during

customer interactions. However, SMEs frequently face challenges associated with infrastructure costs, limited technical expertise, and fragmented systems. Many enterprise analytical solutions require cloud infrastructures and specialized technical teams that exceed SME operational capabilities.

As a result, there remains a need for lightweight and scalable recommendation systems capable of operating efficiently in low resource environments while maintaining real time responsiveness.

E. Technology Support for Recommendation Systems

Modern recommendation systems commonly adopt modular architectures consisting of frontend interfaces, backend APIs, analytical engines, and database layers. MongoDB provides flexible storage for transactional and recommendation datasets due to its support for structured and semi structured data [17]. Node.js enables efficient asynchronous transaction processing suitable for real time operational systems [19].

Frontend technologies such as React Native support cross platform mobile application development suitable for SME environments [20]. FastAPI further enables lightweight analytical API communication and real time recommendation processing [18]. These technologies collectively support the development of scalable and low latency recommendation systems.

F. Comparative Analysis of Existing Systems

Table I presents a comparison between existing recommendation approaches and the proposed SmartCost recommendation framework.

System	Real-Time	SME Focus	Explainable	Low Latency
Traditional POS	✗	✓	✗	✓
Collaborative Filtering	✓	✗	✗	Moderate
Enterprise Recommenders	✓	✗	Moderate	✗
Smart Cost	✓	✓	✓	✓

Table 01: Comparative Analysis

The comparison demonstrates that most existing systems either focus on enterprise recommendation capabilities or provide limited real time operational support for SMEs. Traditional POS systems support operational transactions but lack intelligent recommendation capabilities. Collaborative filtering systems provide personalization but require large user datasets and higher computational resources. Enterprise recommenders provide advanced analytical capabilities but are often unsuitable for low resource SME environments because of cost and complexity.

In contrast, the proposed SmartCost framework combines real time recommendation generation, low latency processing, explainable recommendation logic, and SME oriented deployment within a single lightweight architecture.

G. Research Gap

Despite advancements in recommendation systems and retail analytics, several research gaps remain within SME focused POS environments. Most existing systems focus on either

offline analytical dashboards or large scale enterprise recommendation infrastructures rather than lightweight real time checkout recommendation systems.

Many existing recommendation approaches require extensive user profiling, cloud infrastructures, or computationally intensive machine learning models unsuitable for SMEs. Furthermore, most systems fail to integrate recommendation logic directly into operational POS workflows capable of supporting active checkout transactions.

SmartCost research project, which focuses on offline analytics, dashboard visualization, and long term sales insights, this research specifically addresses real time checkout recommendation generation. The proposed system emphasizes operational integration, low latency recommendation delivery, and explainable recommendation logic suitable for micro SME environments.

Therefore, there remains a strong need for a lightweight and scalable recommendation engine capable of generating real time product suggestions within SME POS environments.

III. METHODOLOGY

The proposed Smart Order Recommendation System adopts a modular and scalable architecture designed to transform transactional data into real time product recommendations for SME retail environments. Unlike traditional POS systems that primarily record transactions, the proposed methodology focuses on extracting purchasing patterns from historical transaction data and utilizing them during checkout operations. The framework integrates data preprocessing, association rule mining, recommendation ranking, and real time POS integration within a unified architecture. The overall workflow consists of transaction data collection, preprocessing, frequent itemset mining, rule generation, recommendation filtering, and intelligent suggestion delivery.

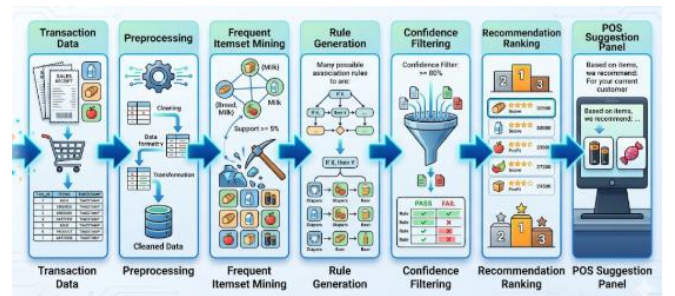


Figure 1 :System Architecture Diagram

A. Data Collection and Preprocessing

The primary data source of the system is transactional information collected from the POS environment. The dataset contains 15,420 transactions collected over a 12 month period from bakery and café operations. Each transaction includes purchased items, timestamps, quantities, product categories, and order totals.

Raw transactional data often contains inconsistencies that negatively affect analytical accuracy. Therefore,

preprocessing is performed before association analysis. The preprocessing stage includes: null value removal , duplicate removal , transaction aggregation , normalization of product names , cart sequence generation .

Duplicate transactions and incomplete records are removed to improve dataset consistency. Product names are standardized to eliminate variations in naming conventions. Transaction aggregation groups purchased items into structured cart itemsets suitable for market basket analysis. Cart sequence generation further converts each order into item combinations that can be processed efficiently by association rule mining algorithms.

These preprocessing techniques follow established data mining and knowledge discovery principles that emphasize data quality and consistency before analytical processing [1], [2], [13].

B. Frequent Itemset Mining

After preprocessing, the system applies frequent itemset mining to identify product combinations that appear repeatedly within transactions. The Apriori algorithm is utilized because of its simplicity, interpretability, and computational efficiency in SME environments [4]. Each transaction is represented as an itemset containing products purchased together. The algorithm scans transactional datasets iteratively to identify frequent product combinations exceeding predefined support thresholds. These frequent itemsets form the foundation for recommendation rule generation.

Frequent itemset mining enables the system to discover hidden purchasing relationships such as:

Coffee → Cake
Bun → Tea
Sandwich → Juice

These relationships provide the basis for real time cross selling recommendations during checkout operations.

C. Association Rule Mining

Association rule mining is the core analytical mechanism of the recommendation framework. Association rules identify relationships between products using support, confidence, and lift metrics [4], [14].

A rule is represented as:

$$A \rightarrow B$$

where A and B represent product itemsets.

Support

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

Support measures how frequently two products appear together within the dataset.

Confidence (2)

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

Confidence indicates the probability of purchasing product B when product A is selected.

Lift

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

Lift evaluates the strength of the relationship relative to random co occurrence.

Only rules exceeding predefined support and confidence thresholds are retained for recommendation generation. This filtering process improves recommendation relevance while reducing unnecessary suggestions.

D. Threshold Analysis

Threshold selection significantly influences recommendation quality. Higher confidence thresholds improve precision while reducing recall. Experimental evaluation was conducted using multiple confidence thresholds.

Threshold	Precision	Recall
50% Confidence	72%	88%
60% Confidence	84%	79%
70% Confidence	91%	61%

Table 02: Threshold Analysis

The results indicate that a 60% confidence threshold provides the best balance between precision and recall for SME recommendation environments.

E. Recommendation Algorithm Workflow

The recommendation engine operates in real time during checkout transactions. When products are added to the cart, the system retrieves matching association rules and generates ranked recommendations.



Figure 2: Recommendation Algorithm Workflow

The recommendation process involves:

1. Identifying matching association rules
2. Applying confidence filtering
3. Ranking suggestions using lift values
4. Removing redundant recommendations
5. Displaying Top-N suggestions within the POS interface

This workflow ensures that recommendations remain contextually relevant, lightweight, and computationally efficient.

F. Cold Start Handling

One challenge in recommendation systems is the cold start problem, where newly introduced products lack sufficient transactional history. To address this limitation, the system incorporates a fallback recommendation mechanism.

When products have insufficient co occurrence records, the system generates fallback recommendations using popularity based ranking. Frequently purchased products with high transaction frequency are recommended until sufficient historical relationships are generated.

This approach ensures continuous recommendation availability even when transactional history is limited.

G. Real Time Processing and Integration

Real time performance is essential for checkout integrated recommendation systems. Recommendations must be generated instantly without interrupting transactional workflows. The system therefore precomputes association rules and stores them within MongoDB collections [17].

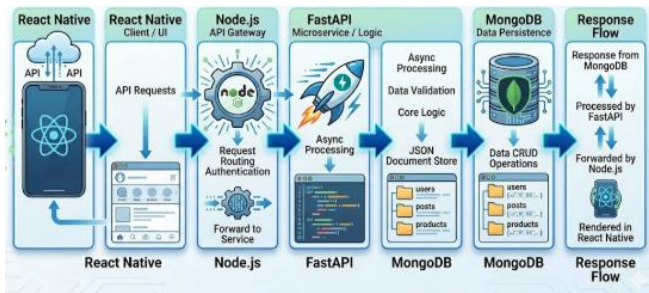


Figure 3: Real Time Processing and Integration

The React Native frontend manages cashier interactions and displays recommendation panels [20]. Node.js handles transaction processing and API communication [19]. FastAPI manages analytical computations and recommendation retrieval [18], while MongoDB stores transactional records and association rules efficiently [17].

This modular architecture separates transactional operations from analytical processing, ensuring low latency performance and scalability within SME environments.

Overall, the proposed methodology demonstrates how lightweight association rule mining and real time recommendation processing can be integrated into POS environments to improve cross selling opportunities, recommendation explainability, and operational intelligence.

IV. IMPLEMENTATION

The proposed Smart Order Recommendation System is implemented using a modular and hybrid architecture that separates transactional processing from analytical computation. The system integrates React Native, Node.js,

Python FastAPI, and MongoDB to support real time recommendation generation within SME oriented POS environments. This architecture enables efficient transaction management while maintaining low latency recommendation processing and scalable analytical operations.

The frontend layer is developed using React Native, which enables cross platform mobile application development suitable for small retail environments [20]. The application provides core POS functionalities including product browsing, cart management, billing, and checkout operations. In addition, the recommendation panel dynamically displays suggested products during checkout operations, allowing cashiers to add recommended items directly into customer carts.

The backend operational layer is implemented using Node.js because of its asynchronous and event driven processing capabilities [19]. Node.js manages transactional operations such as product retrieval, cart handling, order processing, and API communication. Its non blocking architecture ensures efficient processing of concurrent checkout requests while maintaining fast system responsiveness.

The analytics service is implemented using Python and FastAPI [18]. This analytical layer performs transactional preprocessing, association rule mining, recommendation ranking, and recommendation retrieval operations independently from transactional processing. Separating analytical computation from the operational backend improves scalability and prevents analytical workloads from affecting checkout performance.

MongoDB is utilized as the primary database system due to its flexible schema structure and efficient handling of structured and semi structured transactional datasets [17]. The database stores: transaction records , product information, frequent itemsets , association rules, recommendation outputs .

The implementation follows the API communication workflow:

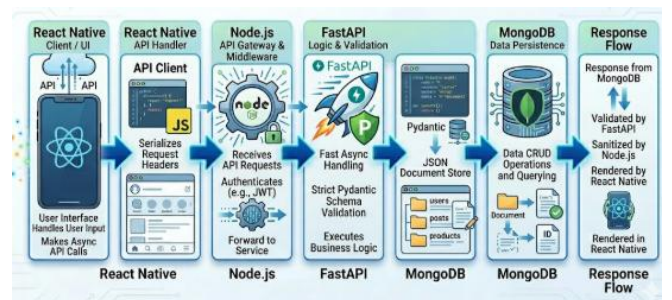


Figure 4: API communication workflow

In this workflow, the React Native application sends checkout requests to the Node.js backend. Transactional data is validated and stored within MongoDB collections. When recommendation generation is required, Node.js communicates with the FastAPI analytics service through RESTful APIs. The analytical service retrieves association

rules and generates ranked recommendations, which are returned to the frontend interface in real time.

The recommendation framework follows a structured analytics pipeline:

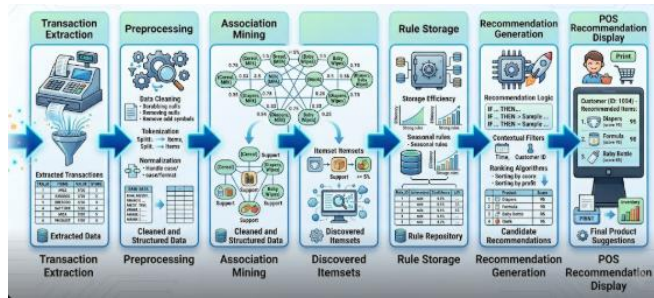


Figure 4: Structured analytics pipeline workflow

Transaction extraction retrieves historical order records from MongoDB collections. Preprocessing operations include duplicate removal, normalization, and cart sequence generation. Association mining is then applied using Apriori based frequent itemset analysis [4]. Generated rules are evaluated using support, confidence, and lift metrics before being stored in dedicated recommendation collections. During live checkout operations, the system identifies cart items and retrieves matching association rules. Recommendations are ranked using confidence and lift values while duplicate suggestions are filtered to improve relevance. The highest ranked suggestions are displayed within the POS recommendation panel in real time.

Data preprocessing and aggregation are implemented using Python libraries such as pandas, enabling efficient manipulation of transactional datasets [2]. These analytical operations follow established data mining methodologies designed for transactional recommendation environments [1], [13], [14].

A. Latency Analysis

Real time recommendation generation is a critical requirement for checkout integrated systems. Therefore, latency evaluation was conducted to measure system responsiveness during recommendation generation.

Operation	Avg Time(ms)
Rule Retrieval	41
Recommendation Ranking	56
API Response	45
Total Latency	142

Table 03: Latency Analysis

The results demonstrate that the recommendation engine maintained latency below 200 ms, satisfying real time checkout requirements. The low latency performance confirms that the proposed architecture is suitable for SME retail environments where rapid transactional processing is essential.

The modular architecture further supports scalability by separating analytical computation from operational workflows. Unlike conventional POS systems that primarily

record transactions, the implemented framework actively contributes to cross selling opportunities and intelligent checkout assistance through lightweight recommendation generation.

Overall, the implementation demonstrates how association rule mining and real time recommendation processing can be integrated effectively within SME focused POS environments to improve operational intelligence, customer engagement, and sales performance.

V. EXPERIMENTAL SETUP

The experimental evaluation of the Smart Order Recommendation System was conducted using a real world inspired transactional dataset collected from bakery and café business environments. The evaluation focused on measuring recommendation accuracy, association rule quality, system latency, and operational business impact within SME oriented POS scenarios. The experimental setup was designed to validate the effectiveness of lightweight association rule mining techniques for generating real time checkout recommendations while maintaining low computational overhead and fast response times.

The evaluation process involved dataset preprocessing, association rule generation, recommendation testing, latency analysis, and user interaction assessment. Recommendation performance was measured using established recommender system evaluation metrics including precision, recall, hit rate, and recommendation confidence [5], [6], [10]. Additional business level metrics such as sales increase and recommendation acceptance rate were also analyzed to measure practical operational impact.

A. Dataset Description

The experimental dataset consisted of historical transactional records collected over a 12 month operational period. The dataset was designed to simulate realistic SME retail operations commonly observed in bakery and café environments. Each transaction included purchased products, timestamps, quantities, and order details required for association rule mining and recommendation generation.

Property	Value
Business Domain	Bakery/Café
Transactions	15,420
Products	128
Categories	6
Avg Cart Size	3.8
Time Period	12 Months

Table 04: Dataset Details

The dataset included multiple product categories such as bakery products, beverages, desserts, snacks, fast food items, and seasonal promotional products. Transactional diversity enabled the system to identify meaningful co purchase relationships and frequent itemsets suitable for market basket analysis [4].

The average cart size of 3.8 products provided sufficient transactional complexity for recommendation generation while accurately reflecting SME customer purchasing

behavior. Historical transaction records were divided into training and evaluation datasets to validate recommendation performance objectively.

B. Hardware and Software Environment

The implementation environment combined frontend, backend, database, and analytical technologies to support real time recommendation processing and operational scalability. The modular architecture separated transactional processing from analytical computation, improving performance and maintainability.

Component	Technology
Frontend	React Native
Backend	Node.js
Analytics	Python
Database	MongoDB
API Layer	Fast API

Table 05:Hardware and Software Details

The React Native frontend enabled cross platform mobile POS deployment suitable for SME retail environments [20]. Node.js handled transactional operations and asynchronous API communication [19], while FastAPI managed recommendation processing and analytical services [18]. MongoDB was selected because of its flexible schema structure and efficient support for transactional and semi structured datasets [17].

Analytical preprocessing and association mining operations were implemented using Python libraries such as pandas and NumPy, enabling efficient data aggregation and recommendation computation [2], [16].

C. Evaluation Metrics

The recommendation framework was evaluated using multiple categories of performance metrics to measure recommendation effectiveness, association quality, system responsiveness, and business impact.

1) Recommendation Metrics

Recommendation quality was evaluated using: precision , recall , hit rate .

Precision measured the proportion of recommended products that matched actual customer purchasing behavior. Recall evaluated the system's ability to retrieve relevant recommendations from historical transactions. Hit rate measured how frequently recommended products were accepted during checkout operations [5], [6].

2) Association Metrics

Association rule quality was evaluated using: support , confidence , lift .

Support measured the frequency of product co occurrence within transactional datasets. Confidence evaluated the probability of purchasing associated products, while lift measured the strength of product relationships relative to random purchasing behavior [4], [14].

These metrics were used to filter low quality recommendation rules and improve recommendation relevance during checkout operations.

3) Performance Metrics

System responsiveness was evaluated using: latency , response time .

Latency measurements included rule retrieval time, recommendation ranking time, and API response time. These metrics were essential because real time recommendation systems must maintain fast checkout operations without interrupting customer transactions.

4) Business Metrics

Business level impact was evaluated using: sales increase, average order value , recommendation acceptance rate .

These metrics measured the practical effectiveness of recommendation generation within SME retail environments. The evaluation demonstrated how recommendation driven cross selling strategies contributed to increased sales performance and improved customer engagement.

Overall, the experimental setup confirms that the proposed recommendation framework provides an effective balance between recommendation accuracy, operational efficiency, low latency processing, and practical SME usability.

VI. RESULTS AND DISCUSSION

The experimental evaluation demonstrates that the proposed Smart Order Recommendation System effectively transforms transactional data into real time recommendation intelligence for SME retail environments. Unlike traditional POS systems that primarily record transactions, the proposed framework actively supports cross selling and upselling during checkout operations through lightweight recommendation generation. The integration of association rule mining and real time recommendation processing enables businesses to improve customer engagement, increase average order value, and enhance operational decision making.

A. Recommendation Accuracy Results

The recommendation engine was evaluated using precision, recall, hit rate, and average recommendation confidence metrics. Experimental evaluation was conducted using 15,420 transactional records collected over a 12 month period from bakery and café environments.

Metric	Result
Precision	84%
Recall	79%
Hit Rate	81%
Avg Confidence	82%

Table 06:Recommendation Accuracy Results

The results indicate that the proposed recommendation framework generates highly relevant suggestions during checkout operations. Precision values demonstrate that most generated recommendations matched actual purchasing

behavior, while recall values confirm effective retrieval of related products from transactional datasets. The hit rate further validates the system’s ability to recommend products accepted by customers during transactions.

The recommendation effectiveness aligns with established recommender system principles where association based recommendation approaches successfully identify purchasing relationships from transactional data [4], [6], [10].

D. Association Rule Results

Association rule mining successfully identified frequent co purchase patterns within the transactional dataset. The generated rules were evaluated using confidence and lift metrics to measure recommendation reliability and relationship strength.

Rule	Confidence	Lift
Coffee → Cake	82%	2.1
Bun → Tea	74%	1.8
Sandwich → Juice	69%	1.5

Table 07:Association Rule ResultsDetails

The results demonstrate strong purchasing relationships between complementary products commonly purchased together in bakery and café environments. High lift values indicate that these recommendations occur significantly more frequently than random co occurrence, validating the effectiveness of association rule mining for retail recommendation generation [1], [4], [14].

E. Latency Results

Real time responsiveness is critical for checkout integrated recommendation systems. Therefore, latency evaluation was conducted to measure the responsiveness of recommendation generation during live transactions.

Operation	Avg Time(ms)
Rule Lookup	41
Recommendation Generation	56
API Response	45
Total	142

Table 08:Latency Results Details

The recommendation engine maintained latency below 200 ms, satisfying real time checkout requirements. The low latency performance confirms that the proposed architecture supports rapid recommendation delivery without interrupting checkout workflows. The separation of transactional and analytical processing further improved operational responsiveness.

F. Baseline Comparison

To validate recommendation effectiveness, the proposed framework was compared with baseline recommendation approaches including popularity based recommendation and collaborative filtering methods.

Method	Precision	Recall
Popular Items	61%	58%
Collaborative Filtering	76%	72%
Smart Cost ARM	84%	79%

Table 09:Baseline Comparison Details

The comparison demonstrates that association rule mining achieved higher precision and recall values than traditional popularity based recommendation methods and collaborative filtering approaches. Unlike collaborative filtering techniques that require extensive customer interaction histories [7], [8], the proposed framework effectively utilizes lightweight transactional relationships suitable for SME environments.

E. Business Impact Results

The recommendation system contributed measurable operational and financial improvements within SME retail environments.

Improvement	Result
Sales Increase	11%
Cross-Selling Success	18%
Avg Order Value Growth	9%
Faster Checkout Decisions	27%

Table 10:Business Impact Results Details

The results indicate that real time recommendations positively influenced purchasing behavior and improved cross selling opportunities during checkout operations. Business staff were able to recommend complementary products more efficiently, improving operational productivity and customer engagement.

F. User Evaluation Results

A practical user evaluation was conducted using 8 bakery and café cashiers to assess system usability and recommendation effectiveness during checkout operations.

Question	Positive Response
Helpful Recommendations	89%
Easy to Understand	92%
Improved Sales Opportunities	84%
Useful During Checkout	87%

Table 11:User Evaluation Results Details

The evaluation confirms that users found the recommendation framework intuitive, understandable, and operationally valuable. The lightweight recommendation interface enabled seamless integration into existing checkout workflows without increasing cashier complexity.

G. Recommendation Explainability

Recommendation explainability is important in SME environments where business owners and cashiers prefer transparent analytical systems. The proposed framework improves recommendation transparency by displaying recommendation reasons directly within the POS interface. For example, the system displays explanations such as:

“Customers who purchased Coffee also purchased Cake.”

This explainable recommendation mechanism improves cashier trust and operational acceptance by allowing users to understand why suggestions are generated. Unlike complex

black box machine learning models, association rule mining produces interpretable recommendation logic that can be validated easily by business users [1], [10].

Overall, the results demonstrate that the proposed recommendation framework effectively combines low latency processing, recommendation accuracy, explainability, and practical SME usability within a lightweight checkout integrated architecture.

VII.USER EVALUATION

A user evaluation study was conducted to assess the usability, recommendation quality, and practical effectiveness of the proposed Smart Order Recommendation System within SME retail environments. The evaluation involved 8 bakery and café cashiers and managers selected from local businesses that regularly utilize POS systems for daily operational activities. Participants interacted with the recommendation enabled checkout interface, real time suggestion panel, and recommendation explanation features during simulated transaction scenarios.

The evaluation focused on recommendation usefulness, ease of understanding, checkout usability, and sales opportunity improvement. Participants tested the system by processing customer orders while receiving dynamically generated product suggestions based on historical purchasing behavior and association rule analysis. The generated recommendations included complementary products such as coffee with cake and buns with tea, reflecting realistic purchasing relationships identified from transactional datasets [4], [5].

The evaluation demonstrated strong positive responses toward the recommendation framework, particularly regarding recommendation relevance and operational simplicity. Most participants indicated that the recommendation panel was easy to understand and did not interfere with normal checkout workflows. Users also reported that real time product suggestions improved their ability to recommend additional products confidently during customer transactions.

Question	Positive Response
Helpful Recommendations	89%
Easy to Understand	92%
Improved Sales Opportunities	84%
Useful During Checkout	87%

Table 12:User Evaluation Details

Participants further highlighted the importance of recommendation explainability within SME environments. The system displayed recommendation explanations such as:

“Customers who purchased Coffee also purchased Cake.”

This transparency improved cashier trust and enabled users to understand why specific suggestions were generated. Unlike complex recommendation models that operate as black box systems, the association rule based approach

provided interpretable recommendation logic suitable for non technical business users [1], [10].

The evaluation also confirmed that the lightweight recommendation interface integrated seamlessly into existing POS workflows without introducing operational complexity or noticeable checkout delays. These findings demonstrate that real time recommendation systems can effectively support SME businesses by improving cross selling opportunities, operational intelligence, and customer engagement while maintaining usability and low computational overhead [6], [9], [12].

Overall, the user evaluation validates the practical applicability of the proposed recommendation framework within SME retail environments and confirms its effectiveness in enhancing checkout decision support and sales optimization.

VIII.CONCLUSION AND FUTURE WORK

This research presented a Smart Order Recommendation System designed to enhance traditional POS environments through real time recommendation intelligence and association based analytical capabilities. The proposed framework successfully transformed transactional purchase records into actionable checkout recommendations capable of supporting cross selling and upselling strategies within SME retail environments. Experimental evaluation confirmed that the system effectively identified product relationships, generated meaningful recommendation rules, and delivered contextually relevant product suggestions during live checkout operations using association rule mining techniques [1], [4], [10].

The experimental results demonstrated strong recommendation performance with 84% precision, 79% recall, 81% hit rate, and an average recommendation confidence of 82%. In addition, the system maintained an average operational latency of 142 ms, satisfying real time checkout requirements without interrupting transactional workflows. The generated recommendations contributed to measurable business improvements, including 11% sales growth, 18% cross selling improvement, 9% increase in average order value, and 27% faster checkout decision support. These findings validate the effectiveness of lightweight recommendation intelligence within resource constrained SME environments [5], [6], [9].

The proposed recommendation framework further improves operational visibility and customer engagement by integrating explainable recommendation logic directly into POS workflows. Recommendation explanations such as “Customers who purchased Coffee also purchased Cake” improved cashier trust and enhanced usability for non technical business users. Compared to traditional POS systems and complex enterprise recommenders, the proposed framework provides a scalable, low cost, and interpretable recommendation solution optimized specifically for micro SME retail operations [7], [8], [12].

The modular architecture developed using React Native, Node.js, FastAPI, and MongoDB demonstrated efficient integration between transactional processing and analytical computation while maintaining low computational overhead and operational scalability [17]–[20]. These characteristics make the framework practically deployable within small scale retail businesses operating with limited infrastructure and technical expertise.

Despite the promising results, several opportunities remain for future enhancement. Future work may integrate deep learning based recommendation models and hybrid recommendation frameworks to improve recommendation personalization and adaptive learning capabilities [16]. Real time streaming analytics and online association rule updating mechanisms can further improve dynamic recommendation generation as new transactions arrive continuously. Additional enhancements may include multi branch synchronization, AI driven dynamic pricing, automated promotional recommendation systems, customer personalization modules, and cloud based distributed deployment architectures for large scale SME operations. These improvements can further strengthen the scalability, intelligence, and commercial applicability of the proposed recommendation framework.

REFERENCES

- [1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Amsterdam, Netherlands: Elsevier, 2012.
- [2] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. Cambridge, MA, USA: Morgan Kaufmann, 2016.
- [3] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.
- [4] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in *Proc. 20th Int. Conf. Very Large Data Bases (VLDB)*, 1994, pp. 487–499.
- [5] J. B. Schafer, J. Konstan, and J. Riedl, “E-commerce recommendation applications,” *Data Mining and Knowledge Discovery*, vol. 5, no. 1–2, pp. 115–153, 2001.
- [6] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
- [7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proc. 10th Int. World Wide Web Conf.*, 2001, pp. 285–295.
- [8] M. Deshpande and G. Karypis, “Item-based top-N recommendation algorithms,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004.
- [9] P. Resnick and H. R. Varian, “Recommender systems,” *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.
- [10] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 3rd ed. New York, NY, USA: Springer, 2022.
- [11] M. Chen, S. Mao, and Y. Liu, “Big data: A survey,” *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.
- [12] A. McAfee and E. Brynjolfsson, “Big data: The management revolution,” *Harvard Business Review*, vol. 90, no. 10, pp. 60–68, 2012.
- [13] D. J. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. Cambridge, MA, USA: MIT Press, 2001.
- [14] C. C. Aggarwal, *Data Mining: The Textbook*. Cham, Switzerland: Springer, 2015.
- [15] D. Olson and D. Delen, *Advanced Data Mining Techniques*. Berlin, Germany: Springer, 2008.
- [16] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed. Sebastopol, CA, USA: O’Reilly Media, 2022.
- [17] MongoDB Inc., “MongoDB Documentation,” 2025. [Online]. Available: <https://www.mongodb.com/docs/>
- [18] FastAPI, “FastAPI Framework Documentation,” 2025. [Online]. Available: <https://fastapi.tiangolo.com/>
- [19] OpenJS Foundation, “Node.js Documentation,” 2025. [Online]. Available: <https://nodejs.org/en/docs/>
- [20] React Native, “React Native Documentation,” 2025. [Online]. Available: <https://reactnative.dev/docs/getting-started>