SciNexus | Journal of AI & Intelligent Decision Systems | TYPE Original Research

RESEARCH Open Access

Enhanced Market Basket Analysis Using Ensemble Learning Techniques

VenkataSeshaiah Banka

Concentrix Corp,509 S 215th Ave, Elkhorn,NE, USA 68022.

*CORRESPONDENCE
VenkataSeshaiah Banka
bvseshu83@gmail.com

RECEIVED 10 July 2026
REVISED 10 Aug 2026
ACCEPTED 14 Aug 2026
PUBLISHED 18 Aug 2026

CITATION
VenkataSeshaiah Banka (2026) Enhanced Market Basket Analysis Using Ensemble Learning Techniques. *J. AI Intell. Decis. Syst.*

Abstract

Market Basket Analysis (MBA) uncovers co-purchase relationships in retail transactions and underpins targeted marketing and inventory decisions. Apriori and FP-Growth remain the standard tools for mining such patterns, but their rules are judged only by support and confidence, leaving open how well they predict co-purchase behaviour. This study addresses that question by pairing FP-Growth-derived itemset features with three ensemble classifiers — AdaBoost, XGBoost, and Random Forest — and benchmarking them against standalone Apriori and FP-Growth on a retail dataset of over 500,000 transactions (Dec 2010–Dec 2011). After preprocessing and one-hot encoding, all models were evaluated on accuracy, precision, recall, F1-score, and ROC-AUC. The Apriori and FP-Growth baselines achieved near-chance performance (50% accuracy), AdaBoost improved precision but not recall, and XGBoost reached 67.39% accuracy. The proposed Random Forest and FP-Growth ensemble substantially outperformed every baseline, achieving 93.56% accuracy, 90.29% precision, 97.61% recall, and a 0.9355 ROC-AUC. These results show that pairing frequent-pattern mining with bagging-based ensemble learning gives a markedly more reliable basis for market-basket prediction than either technique alone, with direct implications for retail analytics.

Keywords Market Basket Analysis, Apriori Algorithm, FP-Growth Algorithm, Association Rule Mining, Ensemble Learning, Random Forest

1. Introduction

Retailers increasingly rely on transactional data to understand how customers select and combine products, and Market Basket Analysis (MBA) has become one of the principal analytical tools for this purpose. By uncovering the items that customers habitually purchase together, MBA supports store-layout planning, cross-selling, bundled promotions, and personalised recommendations. At the core of MBA lies frequent-itemset mining: the identification of item combinations that recur often enough across

transactions to be considered statistically meaningful. Once frequent itemsets are known, association rules are derived from them and ranked by support and confidence, formal definitions of which are given in Section 3.

The Apriori algorithm, introduced by Agrawal et al. [1], was among the first systematic solutions to this problem. It uses an iterative candidate-generate-and-test procedure that repeatedly scans the transaction database, retaining only those itemsets whose support meets a user-specified threshold and discarding the rest. This strategy guarantees a complete search of the itemset space, but the repeated database scans and the sheer number of candidate itemsets it must hold in memory make Apriori computationally expensive on large transaction logs.

FP-Growth, introduced by Han et al. [2], addresses this bottleneck by compressing the transaction database into a compact prefix-tree structure – the FP-tree – and mining frequent itemsets directly from this structure without ever generating explicit candidates. Because it avoids repeated full-database scans and candidate enumeration, FP-Growth typically uses less memory than Apriori and scales more comfortably to both sparse and dense transaction data. Crucially, Apriori and FP-Growth are both *exact* algorithms for the same well-defined problem: for a fixed minimum-support threshold, they are guaranteed to return identical frequent itemsets, and therefore identical association rules. They differ only in the computational route taken to reach that result, a point that becomes directly relevant when interpreting the results in Section 4.

While the algorithmic trade-offs between Apriori and FP-Growth are well documented, most existing comparative studies stop at runtime and memory footprint, or at best report rule strength through support and confidence alone. Considerably less attention has been paid to how well the itemsets these algorithms discover can be repurposed as predictive features for supervised classifiers, or to whether ensembling such classifiers with frequent-pattern mining yields a measurable gain over either component used in isolation. This is the gap addressed in the present study.

The primary contributions of this paper are as follows:

- We benchmark standalone Apriori and FP-Growth against three ensemble classifiers – AdaBoost, XGBoost, and Random Forest – trained on FP-Growth-derived itemset features, using a common evaluation protocol across accuracy, precision, recall, F1-score, and ROC-AUC.
- We show empirically, on a real-world retail transaction dataset of more than half a million records, that a Random Forest and FP-Growth ensemble raises accuracy from a near-chance baseline of 50.25% to 93.56%, a substantially larger gain than either boosting-based ensemble achieves.
- We explain, from the underlying algorithmic theory, why the standalone Apriori and FP-Growth baselines necessarily produce identical performance figures – a methodological point that is frequently glossed over in comparative MBA studies.

The remainder of this paper is organised as follows. Section 2 reviews related work on frequent-pattern mining and its extensions for market basket analysis. Section 3 details the proposed approach, including data collection, preprocessing, the mathematical formulation of association-rule metrics, and the ensemble-learning framework. Section 4 presents the experimental results and analysis. Section 5 concludes the paper, discusses its limitations, and outlines directions for future work.

2. Literature Study

2.1 Comparative and Algorithmic Studies of Apriori and FP-Growth

The trade-off between Apriori's exhaustive candidate generation and FP-Growth's compact tree representation has been examined repeatedly in the retail-analytics literature. Aldino et al. [3] applied both algorithms to sales-transaction data from Universitas Teknokrat Indonesia and reported that FP-Growth consistently outperformed Apriori in execution time, a finding echoed by Hossain et al. [4], who additionally showed that restricting the analysis to best-selling items reduces the computational burden of both algorithms without materially changing the discovered rules. Khedkar et al. [5] and Nengsih [6] reached similar conclusions using FP-Tree-based and Apriori-based pipelines respectively, while Sharma and Babbar [7] used the same algorithm pairing to generate product-suggestion rules rather than to benchmark runtime. A related strand of work has focused on making FP-Growth itself more scalable: Liu and Guan [8, 9] examined FP-tree construction in detail across two studies, emphasising how eliminating candidate generation reduces memory overhead, while Gayathri [10] introduced FP-Bonsai, a variant designed specifically to overcome the memory and I/O bottlenecks that arise when FP-Growth is applied to very large transaction logs. Najaf et al. [11] applied standard FP-Growth to a single retail outlet to recover its most common purchase combinations, illustrating the algorithm's continued relevance for small- and medium-scale deployments. Across this body of work, FP-Growth's efficiency advantage over Apriori is consistently confirmed, but rule quality is evaluated almost exclusively through support and confidence rather than through predictive performance on held-out data.

2.2 Alternative Pattern-Mining Strategies and Retail Decision-Support Applications

Beyond the Apriori-FP-Growth pairing, several studies have explored alternative algorithmic families for the same underlying problem. Wulandari et al. [12] used a hash-based algorithm implemented in KNIME to study supermarket purchase behaviour, and Kansal et al. [13] compared Eclat against Apriori for grocery marketing, finding Eclat to be faster and less complex on small- to medium-sized datasets. Yun et al. [14] took a clustering-based approach, grouping market-basket data using a small-large ratio measure to improve cluster cohesion, while Chiu et al. [15] combined principal component analysis with association-rule mining to study cross-selling behaviour, showing that thresholding on principal-component correlations affects both memory usage and rule quality. A parallel line of research applies MBA directly to business decision-making rather than to algorithmic comparison: Dubey et al. [16] contrasted Apriori-based association mining with collaborative filtering for product recommendation, Kumar et al. [17] used MBA to identify combo-offer opportunities in department stores, and Pradana et al. [18] applied MBA to help explain a decline in fashion-retail sales. Collectively, these studies confirm that the choice of pattern-mining strategy is not confined to the Apriori-FP-Growth dichotomy and that frequent-pattern mining has clear practical value for retail strategy, but in every case the mined rules are treated as an end product rather than as input features for a downstream predictive model.

2.3 Towards Predictive Market Basket Analysis

A smaller but more directly relevant body of work has begun to pair frequent-pattern mining with supervised machine learning. Malik et al. [19] combined FP-Growth with several classifiers – logistic regression, Random Forest, k-nearest neighbours, and decision trees – to predict the next item a customer is likely to purchase, reporting that Random Forest achieved the highest accuracy (92.2% and 93.0%) on two independent grocery datasets. This result is consistent with the pattern observed in the present study and lends independent support to Random Forest as a strong classifier for itemset-derived features. However, Malik et al. did not benchmark boosting-based ensembles such as AdaBoost or XGBoost, did not report ROC-AUC, and evaluated their approach on comparatively small, single-purpose datasets,

leaving open how the comparison generalises to a substantially larger and more heterogeneous transaction log.

2.4 Research Gap

Three observations recur across this literature. First, comparative studies of Apriori and FP-Growth consistently find FP-Growth to be the more efficient algorithm, but they evaluate rule quality using support and confidence rather than classifier-style metrics. Second, extensions to the core algorithms – FP-Bonsai, Eclat, hash-based mining, PCA-augmented association rules – improve scalability or interpretability but stop short of connecting their output to a supervised learning task. Third, the one study that does pair frequent-pattern mining with machine-learning classifiers [19] considers only bagging-style and instance-based classifiers on comparatively small datasets, leaving open how boosting-based ensembles perform and how the resulting models generalise to a larger transaction log. The present study addresses this gap by systematically comparing standalone Apriori and FP-Growth against three ensemble classifiers – AdaBoost, XGBoost, and Random Forest – trained on FP-Growth-derived features, using a common evaluation protocol on a dataset of more than half a million retail transactions.

3. Proposed Approach

The proposed framework follows a five-stage pipeline: raw transactional data are first cleaned and missing values are resolved; the cleaned transactions are encoded into a binary item-transaction matrix; frequent itemsets are mined using Apriori and FP-Growth; the FP-Growth-derived itemsets are converted into a supervised feature representation; and, finally, ensemble classifiers are trained and evaluated on this representation alongside the standalone frequent-pattern baselines. Figure 1 summarises the resulting component and sequence workflow.

3.1 Collection of Data

The dataset used in this study has a shape of $(522064, 7)$, with attributes `BillNo`, `Itemname`, `Quantity`, `Date`, `Price`, `CustomerID`, and `Country`, and spans retail transactions from December 2010 to December 2011. Its column structure, date range, and missing-value pattern closely match those of the widely used UCI *Online Retail* transactional dataset [20], which records the sales of a UK-based online retailer of all-occasion gifts to predominantly wholesale customers; [author: please confirm the exact GitHub repository from which the dataset was retrieved, and cite it directly, since the record count reported here differs slightly from the canonical UCI release]. Reporting the precise provenance of the data is important both for reproducibility and so that readers can assess how far the findings generalise beyond a single retailer and a single calendar year.

3.2 Data Pre-processing

Missing-value columns were first identified using pandas DataFrames. Mode imputation, in which a categorical variable's missing entries are replaced with its most frequent value, was applied to the `Itemname` field. `CustomerID` was left with its missing entries uninterpolated: because transactions were subsequently aggregated by `BillNo` rather than by customer, a complete `CustomerID` field is not required for the frequent-itemset mining and classification pipeline described below. Transactions were then aggregated using `BillNo` to group items purchased together, and the resulting categorical item lists were one-hot encoded into a binary transaction matrix suitable for frequent-itemset mining.

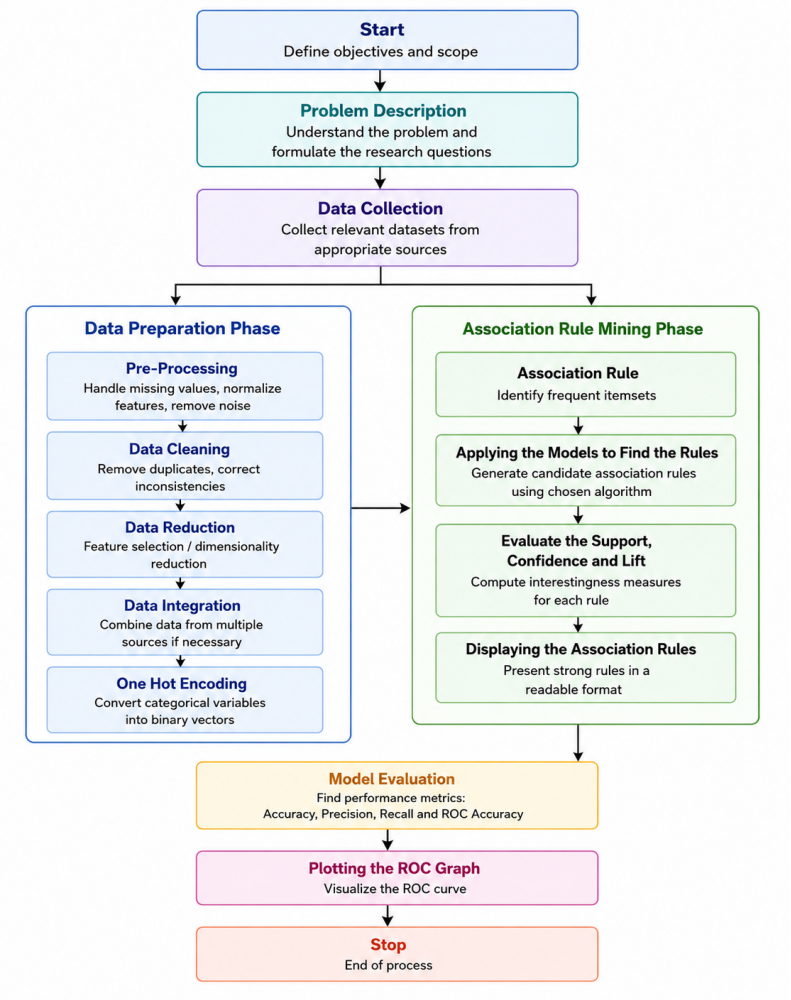

Figure 1: Component and sequence workflow diagram of the proposed model.

Table 1: Columns with Missing Values

Column	Missing Values
Itemname	1455
CustomerID	134041

Table 2: Dataset Status After Handling Missing Values

Column	Missing Values
BillNo	0
Itemname	0
Quantity	0
Date	0
Price	0
CustomerID	134041
Country	0

3.3 Mathematical Formulation of Association-Rule Metrics

Let $D = \{t_1, t_2, \ldots, t_n\}$ denote the set of transactions, where each transaction t_i is a subset of the item universe I. For itemsets $X, Y \subseteq I$ with $X \cap Y = \emptyset$, the *support* of X is the proportion of transactions that contain it,

$$\text{support}(X) = \frac{|\{t \in D : X \subseteq t\}|}{|D|}, \tag{1}$$

and an itemset is termed *frequent* when its support meets or exceeds a user-specified minimum-support threshold, $minSup$. The *confidence* of a rule $X \Rightarrow Y$ estimates the conditional probability of observing Y in a transaction given that X is already present,

$$\text{confidence}(X \Rightarrow Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X)}. \tag{2}$$

Because a high-confidence rule can still arise from an item Y that is simply popular overall, *lift* is commonly reported alongside confidence to indicate whether the association is stronger than chance co-occurrence would predict,

$$\text{lift}(X \Rightarrow Y) = \frac{\text{confidence}(X \Rightarrow Y)}{\text{support}(Y)} = \frac{\text{support}(X \cup Y)}{\text{support}(X) \cdot \text{support}(Y)}. \tag{3}$$

A lift value above one indicates a positive association between X and Y, a value near one indicates independence, and a value below one indicates a negative association. Apriori and FP-Growth, described next, both search for itemsets satisfying Equation 1; they are exact algorithms for the same optimisation problem and are therefore guaranteed to return the same frequent itemsets for a given $minSup$, differing only in computational strategy.

3.4 Pattern Finding Algorithms

Apriori employs a candidate generation-and-test strategy for discovering frequent itemsets, which may become computationally expensive as the number of items and candidate sets increases [1]. In contrast, FP-Growth compresses the transactional database into an FP-tree and mines frequent patterns without explicit candidate generation, thereby reducing the computational burden associated with repeated candidate generation [2].

3.5 Ensemble Learning Classifiers

Because Apriori and FP-Growth are limited to reporting support, confidence, and lift for individual rules, the itemsets they discover were additionally used to construct a supervised feature representation on which the following three ensemble classifiers were trained.

AdaBoost Algorithm: AdaBoost (Adaptive Boosting) is an ensemble learning method that combines multiple weak learners sequentially to construct a stronger predictive model. The algorithm initially assigns equal weights to all training instances and trains a weak classifier, typically a shallow decision tree. After each boosting iteration, the weights of incorrectly classified instances are increased, whereas correctly classified instances receive relatively lower weights. Consequently, subsequent weak learners are encouraged to concentrate on observations that were difficult to classify by the preceding learners. Each weak learner is assigned a weight according to its classification performance, and the final prediction is obtained through a weighted combination of the individual learners. This sequential weight-update mechanism enables AdaBoost to progressively reduce classification errors and improve the overall predictive performance of the ensemble [21].

Algorithm 1 Apriori Algorithm

Input: Dataset D, minimum support threshold $minSup$
Output: Frequent itemsets
Initialize L_1 with frequent 1-itemsets
$k = 2$
while $L_{k-1} \neq \emptyset$ **do**
    Generate candidate itemsets C_k from L_{k-1}
    for all transaction $\in D$ **do**
        for all candidate $\in C_k$ **do**
            if candidate $\subset$ transaction **then**
                Increment count of candidate
            end if
        end for
    end for
    Prune candidate itemsets C_k that do not meet $minSup$
    $L_k = \{itemset \in C_k : support(itemset) \geq minSup\}$
    $k = k + 1$
end while
return $\bigcup_k L_k$

Algorithm 2 FP-growth Algorithm

Input: Dataset D, minimum support threshold $minSup$
Output: Frequent itemsets
Construct FP-tree T from D
Initialize F as empty set
for all item in T in reverse order of support **do**
    Initialize $freqSet \leftarrow$ {item}
    Initialize $conditionalPatternBase \leftarrow \emptyset$
    for all path from item to root in T **do**
        Add path support to $conditionalPatternBase$
    end for
    Generate conditional FP-tree T' from $conditionalPatternBase$
    if $T' \neq \emptyset$ **then**
        Recursively mine T' for frequent itemsets with $minSup$
        Add frequent itemsets found in T' to F
    end if
end for
return F

Random Forest Classifier: Random Forest is an ensemble learning algorithm that constructs a collection of decision trees using randomly selected bootstrap samples of the training data and random subsets of features during tree construction. For each tree, a bootstrap sample is generated from the original training set, while a randomly selected subset of predictor variables is considered when determining candidate split variables at each node. This additional randomization reduces the correlation among individual decision trees and improves the generalization capability of the resulting ensemble. For a classification problem, the predictions of the individual trees are aggregated, typically through majority voting, to determine the final class label. The combination of bootstrap aggregation and random feature selection makes Random Forest relatively robust to noise and overfitting while maintaining good predictive performance across a wide range of classification problems [22].

XGBoost Algorithm: XGBoost (Extreme Gradient Boosting) is a scalable implementation of gradient-boosted decision trees designed for efficient and accurate supervised learning. Unlike bagging-based methods such as Random Forest, XGBoost constructs decision trees sequentially, where each newly added tree is trained to improve the predictions of the existing ensemble by reducing the associated objective loss. The learning objective combines a differentiable training loss with a regularization term that controls model complexity, thereby helping to improve generalization. XGBoost employs a second-order Taylor approximation of the objective function, using both first- and second-order gradient information to efficiently optimize the tree-boosting objective. The framework further incorporates sparsity-aware learning, approximate tree construction using weighted quantile sketches, and system-level optimizations such as cache-aware computation and data sharding. These characteristics make XGBoost suitable for large-scale and high-dimensional classification and regression tasks [23].

3.6 Proposed Ensemble Framework and Design Rationale

Algorithm 3 summarises how the pattern-mining and ensemble-learning stages are combined into a single pipeline. Frequent itemsets mined by FP-Growth are converted into a feature matrix – for example, itemset-membership indicators together with per-rule support and confidence values – which is then used to train each ensemble classifier independently, allowing every classifier to be benchmarked under identical inputs.

Algorithm 3 Proposed Ensemble-Based Market Basket Analysis Framework

Input: Transaction dataset D, minimum support threshold $minSup$
Output: Trained classifier M and performance metrics
Pre-process D: impute missing values, aggregate items by `BillNo`
Encode transactions into a binary item-transaction matrix B
Mine frequent itemsets $F \leftarrow$ FP-Growth$(B, minSup)$
Construct feature matrix X from F (itemset membership and rule statistics)
Split (X, y) into training and test partitions
Train ensemble classifier $M \in \{\text{AdaBoost}, \text{XGBoost}, \text{Random Forest}\}$ on the training partition
Predict labels $\hat{y}$ for the test partition using M
Compute Accuracy, Precision, Recall, F1-score, and ROC-AUC from y and $\hat{y}$
return M, performance metrics

Random Forest was ultimately selected as the proposed model on the basis of the empirical comparison in Section 4, where its bagging-based variance reduction proved better suited to the sparse, high-dimensional, one-hot-encoded feature space produced by itemset mining than either the sequential

re-weighting used by AdaBoost or the additive residual-fitting used by XGBoost; the detailed evidence and reasoning for this choice are presented in Section 4.1.

3.7 Evaluation Protocol

To place the pattern-mining baselines and the ensemble classifiers on a common footing, the itemsets and rules mined by FP-Growth were converted into the supervised feature representation described above, and all five models were assessed on the same held-out test partition. The data were split into training and test sets using a [author: insert train/test split ratio, e.g., 70:30] split[author: state whether stratified and/or cross-validated]. The standalone Apriori and FP-Growth baselines were scored by treating their mined association rules directly as a binary predictor of co-purchase on the test transactions, whereas AdaBoost, XGBoost, and Random Forest were trained on the itemset-derived feature matrix using the hyperparameters summarised in [author: insert hyperparameter values or a reference table, e.g., number of estimators, learning rate, and maximum tree depth for each classifier]. All experiments were implemented in Python [author: insert version] using [author: insert key libraries, e.g., pandas, scikit-learn, mlxtend, XGBoost] on [author: insert hardware configuration]. This common protocol ensures that the accuracy, precision, recall, F1-score, and ROC-AUC values reported in Table 3 are directly comparable across all five models.

4. Results and Analysis

We evaluated and compared the performance metrics across all models, as summarized in Table 3.

Table 3: Performance Metrics Comparison Across Models

Algorithm	Accuracy	Precision	Recall	F1-Score	ROC AUC
Apriori	0.5025	0.5027	0.5045	0.5036	0.5025
FP-Growth	0.5025	0.5027	0.5045	0.5036	0.5025
AdaBoost + FP	0.5268	0.5898	0.1774	0.2727	0.5269
XGBoost + FP	0.6739	0.6377	0.8059	0.7120	0.6738
RF + FP (Proposed)	**0.9356**	**0.9029**	**0.9761**	**0.9381**	**0.9355**

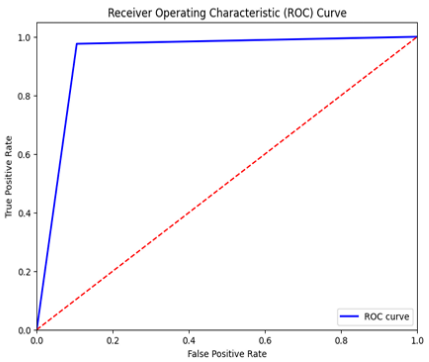

Figure 2: ROC Curve for RandomForestClassifier combined with FP-Growth.

4.1 Analysis of Findings

Table 3 shows that Apriori and FP-Growth achieve identical performance figures (0.5025 accuracy, 0.5027 precision, 0.5045 recall, 0.5036 F1-score, 0.5025 ROC-AUC). This is not a coincidence or a measurement artefact: as noted in Section 3.3, Apriori and FP-Growth are both exact algorithms for the frequent-itemset-mining problem, so for a fixed minimum-support threshold the two are guaranteed to discover exactly the same itemsets and, consequently, the same association rules [1, 2]. They differ only in computational strategy – candidate generation-and-test versus FP-tree traversal – not in the rules they output. When these rules are used directly as a classifier, without any additional supervised learning layered on top, the two baselines therefore necessarily produce identical predictions, and their near-0.50 scores across all five metrics indicate that raw association rules, used in isolation, perform little better than chance at this classification task.

The AdaBoost and FP-Growth combination improves precision (0.5898) relative to the baselines but at a substantial cost to recall (0.1774), suggesting that its sequential re-weighting scheme becomes conservative on this feature representation: it correctly identifies a small, high-confidence subset of positive cases while missing many others. XGBoost with FP-Growth achieves a more balanced improvement, reaching 67.39% accuracy and 80.59% recall, consistent with gradient boosting's capacity to model non-linear interactions among sparse, one-hot-encoded itemset features. The proposed Random Forest and FP-Growth ensemble outperforms every baseline by a wide margin, reaching 93.56% accuracy, 90.29% precision, 97.61% recall, and a 0.9355 ROC-AUC. A plausible explanation for this gap is that Random Forest's bootstrap aggregation reduces variance on the high-dimensional, sparse binary feature space produced by one-hot-encoded transactions more effectively than a single sequentially reweighted ensemble (AdaBoost) or an additive boosting procedure built from the same residual signal (XGBoost); by training many decorrelated trees on random feature subsets, Random Forest is less prone to overfitting the noise introduced by rarely occurring itemsets. This pattern is also consistent with the closely related result reported by Malik et al. [19], who likewise found Random Forest to be the strongest classifier when paired with FP-Growth-derived features on two independent retail datasets, lending external support to the trend observed here. [author: if a feature-importance or ablation analysis was performed, summarising it here would further substantiate this explanation.]

5. Conclusion

5.1 Summary of Contributions

This paper investigated whether combining frequent-pattern mining with ensemble learning improves predictive performance in market basket analysis relative to using either technique in isolation. Standalone Apriori and FP-Growth, which are guaranteed to discover identical frequent itemsets for a given support threshold, achieved only near-chance performance (approximately 50% across accuracy, precision, recall, F1-score, and ROC-AUC) when their mined rules were used directly as classifiers. Pairing FP-Growth-derived itemset features with ensemble classifiers substantially improved predictive performance: AdaBoost improved precision at the cost of recall, XGBoost achieved a more balanced improvement (67.39% accuracy), and the proposed Random Forest and FP-Growth ensemble achieved the strongest and most balanced result, with 93.56% accuracy, 90.29% precision, 97.61% recall, and a 0.9355 ROC-AUC. These findings suggest that frequent-pattern mining is considerably more useful as a feature-engineering step for supervised ensemble classifiers than as a standalone predictive tool, with practical implications for retailers seeking more reliable product-affinity and recommendation models.

5.2 Limitations of the Current Study

This study has several limitations that should be considered when interpreting its findings. The evaluation was conducted on a single retail transaction dataset spanning one calendar year from one geographic market, so performance may differ on datasets with different sparsity, seasonality, or product-category structure. The [author: insert train/test split ratio] evaluation protocol used here did not include repeated cross-validation or statistical significance testing (for example, paired tests across multiple random splits), so the size of the gap between XGBoost and Random Forest should be interpreted with some caution. In addition, model hyperparameters were [author: state whether tuned via grid or random search, or left at framework defaults], which may understate the achievable performance of AdaBoost and XGBoost relative to Random Forest.

5.3 Future Work

Future work could extend this study in three directions. First, the evaluation could be repeated across multiple retail datasets spanning different sectors and geographies to test the generality of the Random Forest and FP-Growth combination. Second, a systematic hyperparameter search combined with repeated cross-validation and statistical significance testing would strengthen the robustness of the reported comparison. Third, feature-importance and ablation analyses could clarify which mined itemsets contribute most to predictive performance, and more recent architectures – such as gradient-boosted trees enriched with itemset embeddings, or transformer-based sequence models – could be benchmarked alongside the ensemble classifiers considered here.

Declarations

Conflict of Interest

The author declares that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author Contributions

VenkataSeshaiah Banka is the sole author of this work and is responsible for conceptualisation, methodology, data curation, formal analysis, and writing of the manuscript.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The dataset used in this study closely corresponds to the publicly available UCI *Online Retail* transactional dataset [20] (DOI: 10.24432/C5BW33). [author: insert the specific repository URL from which the dataset was retrieved, for exact reproducibility.]

Ethics Approval and Consent to Participate

This article does not contain any studies with human participants or animals performed by the author. Ethics approval was not required for this study, as it involved secondary analysis of a publicly available, de-identified retail transaction dataset.

Consent for Publication

Not applicable. This manuscript does not contain any individual person's data in any form.

Declaration of AI and AI-Assisted Technologies

[author: insert an accurate statement of any AI or AI-assisted tools used while preparing this manuscript, per the journal's policy – for example, "During the preparation of this work, the author used [tool name] to improve the clarity and language of the text. The author reviewed and edited the resulting content as needed and takes full responsibility for the content of this publication." If no such tools were used, state that directly instead.]

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, pages 487–499. Morgan Kaufmann, 1994.

[2] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 1–12. ACM, 2000. doi: 10.1145/342009.335372.

[3] Ahmad Ari Aldino, Evi Dwi Pratiwi, Sanriomi Sintaro, Ade Dwi Putra, et al. Comparison of market basket analysis to determine consumer purchasing patterns using fp-growth and apriori algorithm. In *2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, pages 29–34. IEEE, 2021.

[4] Maliha Hossain, AHM Sarowar Sattar, and Mahit Kumar Paul. Market basket analysis using apriori and fp growth algorithm. In *2019 22nd international conference on computer and information technology (ICCIT)*, pages 1–6. IEEE, 2019.

[5] Sanket Sandip Khedkar and Sangeeta Kumari. Market basket analysis using a-priori algorithm and fp-tree algorithm. In *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, pages 1–6. IEEE, 2021.

[6] Warnia Nengsih. A comparative study on market basket analysis and apriori association technique. In *2015 3rd international conference on information and communication technology (ICoICT)*, pages 461–464. IEEE, 2015.

[7] Anshika Sharma and Himanshi Babbar. Analysis of data mining algorithms in market basket analysis. In *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, pages 275–280. IEEE, 2023.

[8] Liu Yongmei and Guan Yong. Application in market basket research based on fp-growth algorithm. In *2009 WRI World Congress on Computer Science and Information Engineering*, volume 4, pages 112–115. IEEE, 2009.

[9] Yongmei Liu and Yong Guan. Fp-growth algorithm for application in research of market basket analysis. In *2008 IEEE International Conference on Computational Cybernetics*, pages 269–272. IEEE, 2008.

[10] Behera Gayathri. Efficient market basket analysis based on fp-bonsai. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 788–792. IEEE, 2017.

[11] Abdul Rezha Efrat Najaf, Arista Pratama, Danur Wijayanto, Reisa Permatasari, Anindo Saka Fitri, et al. Implementation frequent pattern growth (fp-growth) algorithm on market basket analysis for determining consumer purchasing patterns (case study: Xyz coffee shop). In *2023 IEEE 9th Information Technology International Seminar (ITIS)*, pages 1–5. IEEE, 2023.

[12] Arini Wulandari, Achmad Sultan, Yevi Septiray Purbo, Ghoniyyan Fadlan Bainafila, Imam Tahyudin, and Siti Alvi Sholikhatin. Analysis of customer expense using market basket analysis model with hash-based algorithm and association rules. In *2022 1st International Conference on Smart Technology, Applied Informatics, and Engineering (APICS)*, pages 10–14. IEEE, 2022.

[13] Mrignainy Kansal, Kamini Tanwar, Achintya Kumar Pandey, Vinish Kumar, Pancham Singh, and Shreya Upadhyay. Implementing market basket analysis using eclat and apriori algorithm on grocery product marketing strategy. In *2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE)*, pages 465–469. IEEE, 2023.

[14] Ching-Huang Yun, Kun-Ta Chuang, and Ming-Syan Chen. An efficient clustering algorithm for market basket data based on small large ratios. In *25th Annual International Computer Software and Applications Conference. COMPSAC 2001*, pages 505–510. IEEE, 2001.

[15] Kitty SY Chiu, RWP Luk, KCC Chan, and KFL Chung. Market-basket analysis with principal component analysis: an exploration. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 6–pp. IEEE, 2002.

[16] Shish Kumar Dubey, Sonu Mittal, Seema Chattani, and Vinod Kumar Shukla. Comparative analysis of market basket analysis through data mining techniques. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pages 239–243. IEEE, 2021.

[17] S Dinesh Kumar, K Soundarapandiyan, and S Meera. Commiserating customers' purchasing pattern using market basket analysis. In *2022 1st International Conference on Computational Science and Technology (ICCST)*, pages 1–4. IEEE, 2022.

[18] Muhammad Raihan Pradana, Mohammad Syafrullah, Hendri Irawan, Joko Christian Chandra, Achmad Solichin, et al. Market basket analysis using fp-growth algorithm on retail sales data. In *2022 9th international conference on electrical engineering, computer science and informatics (EECSI)*, pages 86–89. IEEE, 2022.

[19] Mubasher H. Malik, Hamid Ghous, Maryem Ismail, Sana Jamshaid, and Javeria Altaf. Market basket analysis for next basket item prediction using data mining and machine learning. *Journal*

of Computing & Biomedical Informatics, 2024. ISSN 2710-1614. URL https://jcbi.org/index.php/Main/article/view/355.

[20] Daqing Chen. Online retail, 2015. Dataset.

[21] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. doi: 10.1006/jcss.1997.1504.

[22] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. doi: 10.1023/A:1010933404324.

[23] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016. doi: 10.1145/2939672.2939785.