A Novel Way to Estimate High Utility Mining Using Matrix Based Approach

Bujja Abigna[1], Dharini R[1], Harini N[1], Dr. S J Vivekanandan[2*]
[1]*UG Scholar, Department of Computer Science and Engineering, Dhanalakshmi College of Engineering, Tambaram, Chennai-601301*
[2]*Assistant Professor, Department of Computer Science and Engineering, Dhanalakshmi College of Engineering, Tambaram, Chennai-*

Abstract—**High Utility Itemset Mining (HUIM) serves as a vital tool for extracting important data patterns from extensive transactional databases through its analysis of both item occurrence and product profitability. Traditional frequent pattern mining techniques use support values as their main measurement method while they fail to consider item utility which results in frequent pattern discovery of unimportant patterns. The current HUIM methods use unchanging threshold parameters together with multiple database scanning processes which results in increased system resource use and extended operation time. This paper presents a matrix-based method that enables effective high utility pattern extraction through the implementation of dynamic percentile-based threshold selection. The proposed method transforms transactional data into a structured utility matrix which eliminates unnecessary database scanning while enhancing system performance. The calculation of utility values utilizes quantity and profit information, whereas the calculation of support values depends on the frequency of occurrence. Percentile analysis generates dynamic thresholds which enable the system to select itemsets that hold significant meaning at any time. The proposed system classifies discovered itemsets into High Frequency High Utility (HFHU), High Frequency Low Utility (HFLU), and Low Frequency High Utility (LFHU) categories, with special emphasis on identifying low-frequency yet highly profitable patterns. The experimental results show that the proposed method enhances pattern discovery performance while it delivers valuable insights about sales patterns and profit margins which assist decision makers in their work. The system proves its enhanced capabilities through performance testing which involves large-scale transactional datasets. The interactive visualization dashboard developed through Streamlit enables users to perform intuitive analysis while FastAPI backend API layer implementation provides support for real-world applications. The matrix-based representation delivers operational efficiency advantages by decreasing the need for multiple database accesses while it enhances the system's processing performance. The high-utility patterns which have been discovered assist organizations in making strategic decisions for their inventory control and sales planning and profit maximization activities. The proposed model improves usability through its visualization tools and API-based connectivity while enabling organizations to implement the system in their actual business operations.**

Keywords: **Utility Mining, Apriori Algorithm, High Utility Itemset Mining (HUIM), Transactional Data Analysis, Percentile-Based Thresholding, Matrix-Based Mining, Support and Utility, Low Frequency High Profit Items.**

I. INTRODUCTION

Frequent Pattern Mining (FPM) is a vital data mining method for finding useful connections between items in big transactional databases. It is used a lot in different fields, like analysing the retail market, making recommendation systems, predicting sales, and man-aging inventory. The main goal of frequent pattern mining is to find itemsets that often show up together in transactions. This helps businesses understand how customers buy things and make their business plans better. The Apriori algorithm is one of the first and most popular algorithms for finding frequent patterns. Apriori finds frequent itemsets by looking at support values, which show how often an itemset shows up in the database.Apriori is good at finding common pat-terns, but it needs to scan the database many times and makes a lot of candidate itemsets, which makes the process more complicated and takes longer. Also, Apriori only looks at how often items are sold and not how

much money they make or how important they are.

High Utility Itemset Mining (HUIM) techniques were created to get around these problems. HUIM looks at both the number of items and the profit they make to find more valuable patterns, which is different from traditional frequent pattern mining methods. Utility mining helps you find itemsets that make a lot of money but don't show up very often. These itemsets are very important for overall revenue. Several HUIM algorithms have been created to make performance better and cut down on the search space by using efficient pruning methods and better data structures.

In the past few years, matrix-based methods have been suggested to make frequent and utility mining processes work better. In matrix-based methods, trans-actional data is changed into a structured matrix for-mat, with rows for transactions and columns for items. This change cuts down on repeated database scans and speeds up calculations, making it good for pro-cessing large amounts of data.

But a lot of current systems use fixed minimum support and utility threshold values that users have to choose themselves. If you choose these thresholds incorrectly, you might end up with too many unimpor-tant patterns or miss important ones. Also, traditional methods often don't work well to find low-frequency but high-profit items, which are important for making better decisions and boosting sales.

This research presents a matrix-based high utility frequent pattern mining model combined with dy-namic percentile-based threshold selection to tackle these challenges. The suggested method changes trans-actional data into a utility matrix, finds support and utility values, and uses percentile interpolation to find threshold values in real time. The system also puts itemsets into useful groups, such as High Frequency High Utility (HFHU), High Frequency Low Utility (HFLU), and Low Frequency High Utility (LFHU). The model also compares sales from one month to the next to find patterns that are profitable over time. The suggested method aims to make computations faster, improve the accuracy of finding patterns, and give useful information about sales and profitability trends. This will help businesses make better decisions and plan their strategies.

The contemporary data-driven environment re-quires organizations to extract patterns from their data while demonstrating those patterns through effective presentation methods and their capacity to manage extensive data sets. The existing systems from traditional times do not provide the ability to expand their operations nor do they support visual representation functions. The real-world applications process requires the implementation of efficient min-ing techniques together with interactive dashboard systems and system integration mechanisms.

1.1 Associate Rule Mining

Association rule mining, a data mining methodology, serves to uncover latent relationships and valuable pat-terns within extensive transactional databases. The primary objective is to pinpoint item sets frequently co-occurring and to formulate rules elucidating their associations. Organizations leverage insights derived from these relationships to analyze consumer buy-ing behaviors, informing decisions regarding product placement and the development of data-driven busi-ness strategies.

Association rules are generally structured as follows: $X \rightarrow Y$, where X represents the antecedent and Y rep-resents the consequent. Evaluating the effectiveness of these rules necessitates three key metrics: support, confidence, and lift. Support indicates the frequency of a specific item set within the dataset. Confidence quantifies the probability of the consequent occurring given the presence of the antecedent. Lift evaluates the interdependence between two items by contrasting their observed co-occurrence with the co-occurrence expected under the assumption of independence. As-sociation rule mining is widely applied in diverse real-world contexts, including retail analytics, market bas-ket analysis, recommended systems, and predictive sales models.

II. LITERATURE SURVEY

High Utility Itemset Mining (HUIM) has garnered considerable attention in recent years owing to its capability to discern lucrative patterns from extensive transactional datasets. Utility mining is better for real-world uses like retail analysis and making business decisions because it looks at both the frequency and profit of items, unlike traditional frequent itemset mining methods that only look at support values.

Recent research has concentrated on enhancing the efficiency and scalability of utility mining algorithms. Fournier-Viger et al. (2020) provided a thorough ex-amination of high-utility pattern mining methodolo-gies and underscored the significance of effective data structures and pruning techniques in mitigating com-putational complexity. Their research highlighted that utility-based mining yields more significant outcomes than conventional frequency-based methods.

Lin et al. (2020) put forward effective ways to mine high-utility itemsets from big datasets by making the most of memory and execution time. Their study showed that better algorithms can better handle large amounts of transactional data. Ahmed et al. (2021) also came up with a dynamic threshold-based mining method that changes the threshold values based on the features of the dataset. This method cut down on the generation of patterns that weren't needed and made mining more accurate.

Kiran and Fournier-Viger (2022) reported addi-tional progress, examining recent advancements in utility mining and emphasising the necessity of scal-able solutions for contemporary data environments. Li et al. (2023) suggested a matrix-based method that changes transactional data into matrix form. This cuts down on the number of times the database needs to be scanned and speeds up calculations.Zhang et al. (2024) recently came up with dynamic utility thresh-old selection techniques that make it easier to find important patterns, especially for low-frequency, high-profit items.

III.PROBLEM STATEMENT

The retail and e-commerce and sales management do-mains now produce massive amounts of transactional data which have developed into a widespread trend during recent years. The process of deriving valuable insights from extensive datasets serves as a fundamen-tal requirement to discover profitable products and to enhance sales methods and to aid business decision processes. The traditional methods of frequent pat-tern mining which use the Apriori algorithm as their primary approach, focus their research on discovering itemsets which appear frequently through the appli-cation of support value measurements. The methods fail to identify true valuable patterns because they re-strict their examination to item relationships without considering item utility and profit value.

High Utility Itemset Mining (HUIM) techniques were developed to solve frequency-based method re-strictions through the inclusion of utility elements which include profit and quantity measurements. The existing systems use minimum support and utility thresholds, which require value item set identifica-tion. The process of selecting threshold values presents challenges which frequently result in the production of excessive unimportant patterns or the omission of crucial high-profit item sets. The algorithms need to scan the database multiple times which results in both extended processing time and increased memory re-quirements for operations with extensive transactional databases. The current methods fail to recognize low-frequency items which generate substantial profits. The items exist in transactions at low frequencies but produce substantial revenue for the business.

The decision-making process becomes less effective because businesses fail to recognize crucial patterns, which activates a chain reaction that results in lower profitability. The existing systems need more precise methods which enable users to adjust thresholds dynamically, together with structured data representa-tion that enhances mining performance. The project requires a system which achieves both efficient opera-tions and large-scale capabilities through its combi-nation of matrix-based data representation and dy-namic threshold selection and utility-based classification methods.

The system transforms transactional data into a utility matrix, which creates support and utility values through efficient computational methods, while its dynamic thresholding system uses percentile-based methods to detect critical patterns. The system detects low-frequency high-profit items through its monthly sales comparison function, which generates insights to support business decisions and enhance profitability evaluation.

The current systems cannot process large datasets because they were built to handle smaller data vol-umes and their visualization systems are not effective. The extracted patterns present problems for users to understand because users lack the necessary tools to interpret them.

IV.EXISTING SYSTEM

The primary method that traditional systems use for data mining operations relies on Frequent Pattern Mining (FPM) methods to detect connections between different items in their transactional databases. The existing systems use Apriori algorithm as their main al-gorithm which finds frequent itemsets according to the established minimum support requirements.Existing systems use traditional frequent pattern mining meth-ods which include FP-Growth and Eclat algorithms in addition to the Apriori algorithm. The FP-Growth algorithm reduces the need for repeated candidate gen-eration by constructing a compact Frequent Pattern Tree (FP-Tree) structure which improves performance when processing large datasets. The Eclat algorithm employs a vertical data representation method which allows transaction identifiers to be used for efficient support value computation through set intersection operations.

The algorithms provide better performance than Apriori in specific situations but their main function relies on frequency-based measures which do not assess the utility or profit value of items. Traditional meth-ods prevent businesses from discovering low-frequency items which generate high profits, thus making it diffi-cult to make successful business decisions.The Apriori algorithm creates candidate itemsets through an iter-ative process which requires multiple database scans to identify frequent pattern. The method identifies commonly occurring itemsets but it uses frequency as its only measurement while excluding the profit value of items. The result leads to the situation where items with high profitability get hidden because they occur less often.

Traditional systems developed High Utility Itemset Mining (HUIM) techniques to address the problems of frequency-based mining. HUIM algorithms use prod-uct quantities together with profit values to find high utility itemsets. Multiple classical HUIM algorithms have been created including utility-list based and tree-based approaches to achieve better mining results. The methods identify profitable patterns through suc-cessful execution; however, they need users to estab-lish minimum utility thresholds which the system will use to determine pattern identification. The process of choosing threshold values inappropriately leads to two outcomes, which include generating too many pat-terns that lack relevance and losing essential patterns.

The system faces an additional problem because it needs to repeatedly scan extensive databases of trans-actions, which results in more complicated computa-tions and longer times to complete tasks. The tradi-tional mining algorithms use the original transactional data format to process data, which results in their inability to operate efficiently with extensive datasets. The existing systems lack structured data represen-tations through the use of matrices, which leads to processing delays and redundant data handling.

The existing methods lack the capability to adjust their dynamic threshold settings for itemset classifica-tion according to item frequency and utility charac-teristics. The system also fails to compare sales data over different months, which shows how sales numbers develop over time. The traditional systems can not detect low-frequency high-profit items, which busi-nesses use to make better decisions and increase their profitability.

Existing systems need to develop new systems be-cause they currently identify frequent or high-utility itemsets through fixed threshold dependency. The database handling of existing systems requires mul-tiple database scans,

which results in high computa-tional costs and limits their ability to find profitable patterns. The existing challenges require the devel-opment of a new system that combines matrix-based computation with dynamic threshold selection and improved classification methods.

V.PROPOSED SYSTEM

The system which we are proposing creates an ad-vanced matrix-based high utility frequent pattern min-ing model which solves the problems that both tra-ditional frequent pattern mining and utility mining methods encounter. The proposed system uses a util-ity matrix which essential transforms transactional data into structured form to enhance computational efficiency because it eliminates the need for multiple database scans and uses unchanging threshold data.

The proposed system processes database transac-tional data by first executing data cleaning opera-tions which eliminate all missing data and inconsistent values. The cleaned data is then transformed into a matrix format which shows transactions as rows and items as columns. The matrix cell value for each item in a transaction shows its utility value which is deter-mined through the multiplication of quantity and unit profit. The matrix-based representation method de-creases database scanning requirements while increas-ing processing speed when compared to conventional methods.

The proposed system introduces an upgraded function which implements dynamic percentile-based threshold selection as its main method for identifying minimum support and utility values. The system uses percentile interpolation techniques to calculate support and utility values for itemsets while it deter-mines threshold levels through automatic calculations. The adaptive system identifies important patterns through its automatic process which does not require any parameter value adjustments.

The proposed system establishes a classification system which divides itemsets into three essential categories.
High Frequency High Utility (HFHU): Item-sets that occur frequently and generate high profit
High Frequency Low Utility (HFLU): Itemsets that occur frequently but generate low profit
Low Frequency High Utility (LFHU): Itemsets that occur rarely but generate high profit
The system focuses on recognizing Low Frequency High Profit (LFHP) products which traditional systems overlook because these products have substantial business value. The system conducts monthly sales comparisons to track utility value changes which oc-cur during different time periods for improved trend analysis and decision-making processes.

The system uses Oracle Database for data storage and Python-based analytical modules to perform ma-trix computation and utility calculation and classifi-cation and visualization operations. The system gen-erates outputs which include bar charts and category-wise comparisons to display results in an understand-able and simple way.

The proposed system achieves four goals which in-clude better mining efficiency and improved pattern discovery accuracy and decreased computational com-plexity and identification of valuable itemsets which provide business insights.

5.1 Itemset Lattice Generation

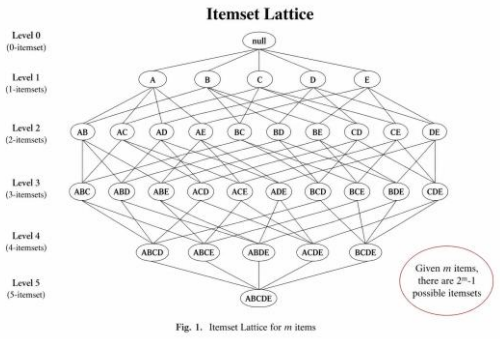

Figure 1. Itemset Lattice Representation for Generating All Possible Itemsets

The itemset lattice structure enables systematic generation of all possible item combinations. The lat-tice system displays all item subsets through a hier-archical structure which begins with the null set and proceeds to show complete itemsets.
The total number of possible itemsets for a set of m items equals 2m 1. Each level of the lattice represents itemsets which increase in size. The mining process

5.2 Scalability Enhancement

The developed system undergoes testing with exten-sive datasets that include over 10000 transactional records. The matrix-based method demonstrates its ability to process actual data through testing which shows its effectiveness and capacity to handle substan-tial workloads.

5.3 Visualization Module

Streamlit was used to create an interactive dashboard which displays utility trends and category distribu-tions and monthly comparison data through graphical representations to help users understand the results better.

5.4 API Integration

The FastAPI framework handles backend API devel-opment to create endpoints which display mining re-sults through top utility items and LFHU items and monthly trends endpoints. This enables integration with external systems.

VI. MATHEMATICAL MODEL

1) Basic Definitions

$$I = i_1, i_2, \ldots, i_m \quad (1)$$

$$T = T_1, T_2, \ldots, T_n \quad (2)$$

$$T_i \subseteq I \quad (3)$$

2) Utility Calculation

$$U(i,j) = Q(i,j) \times P(j) \quad (4)$$

$$TU(T_i) = U(i,1) + U(i,2) + U(i,3) + \cdots + U(i,m) \quad (5)$$

$$U(j) = U(1,j) + U(2,j) + U(3,j) + \cdots + U(n,j) \quad (6)$$

3) Support

$$Support(j) = \frac{\text{Number of transactions containing } j}{\text{Total number of transactions}} \quad (7)$$

4) Mean and Max

$$S_{mean} = \frac{Support(1) + Support(2) + \cdots + Support(m)}{m} \quad (8)$$

$$S_{max} = \max(Support(1), Support(2), \ldots, Support(m)) \quad (9)$$

$$U_{mean} = \frac{Utility(1) + Utility(2) + \cdots + Utility(m)}{m} \quad (10)$$

$$U_{max} = \max(Utility(1), Utility(2), \ldots, Utility(m)) \quad (11)$$

5) Percentile Calculation

$$P_s = \frac{S_{mean}}{S_{max}} \times 100 \quad (12)$$

$$P_u = \frac{U_{mean}}{U_{max}} \times 100 \quad (13)$$

6) Position Formula

$$Pos_s = (P_s/100) \times m \quad (14)$$

$$Pos_u = (P_u/100) \times m \quad (15)$$

7) Sorting

$$S_{sorted} = \text{sort}(S) \quad (16)$$

$$U_{sorted} = \text{sort}(U) \quad (17)$$

8) Threshold Selection

$$S_{th} = \text{value at position } Pos_s \text{ in } S_{sorted} \quad (18)$$

$$U_{th} = \text{value at position } Pos_u \text{ in } U_{sorted} \quad (19)$$

9) Classification

$$HFHU : Support(X) \geq S_{th} \wedge Utility(X) \geq U_{th} \quad (20)$$

$$HFLU : Support(X) \geq S_{th} \wedge Utility(X) < U_{th} \quad (21)$$

$$LFHU : Support(X) < S_{th} \wedge Utility(X) \geq U_{th} \quad (22)$$

Table 1. Transaction Dataset

TID	Rice	Oil	Sugar	Tea	Biscuits
T1	2	1	1	0	1
T2	1	2	0	1	0
T3	1	1	1	0	1
T4	0	2	1	1	0
T5	2	0	1	1	1

VII. WORKED EXAMPLE

7.1 Transaction Dataset

7.2 Profit Values

$$P = 5, 4, 3, 6, 2 \quad (23)$$

7.3 Item Utility

$$Utility(Rice) = 30 \quad (24)$$

$$Utility(Oil) = 24 \quad (25)$$

$$Utility(Sugar) = 15 \quad (26)$$

$$Utility(Tea) = 18 \quad (27)$$

$$Utility(Biscuits) = 6 \quad (28)$$

7.4 Support Values

$$Support(Rice) = 0.8 \quad (29)$$

$$Support(Oil) = 0.8 \quad (30)$$

$$Support(Sugar) = 1.0 \quad (31)$$

$$Support(Tea) = 0.6 \quad (32)$$

$$Support(Biscuits) = 0.6 \quad (33)$$

7.5 Threshold Calculation

$$S_{mean} = \frac{0.8 + 0.8 + 1.0 + 0.6 + 0.6}{5} = 0.76 \quad (34)$$

$$S_{max} = 1.0 \quad (35)$$

$$P_s = 76 \quad (36)$$

$$Pos_s = 3.8 \quad (37)$$

$$S_{th} = 0.8 \quad (38)$$

$$U_{mean} = \frac{30 + 24 + 15 + 18 + 6}{5} = 18.6 \quad (39)$$

$$U_{max} = 30 \quad (40)$$

$$P_u = 62 \quad (41)$$

$$Pos_u = 3.1 \quad (42)$$

$$U_{th} = 24 \quad (43)$$

7.6 Final Classification

Table 2. Final Classification

Item	Support	Utility	Category
Rice	0.8	30	HFHU
Oil	0.8	24	HFHU
Sugar	1.0	15	HFLU
Tea	0.6	18	LFHU
Biscuits	0.6	6	Low Value

VIII. CONCLUSION

The model effectively identifies both frequent high-profit items and rare high-profit items using dynamic thresholding and matrix-based computation.

IX.METHODOLOGY

Step 1: Data Collection The Data Collection Layer is responsible for obtaining transactional data from the MySQL database. The database stores structured sales transaction records that include details such as transaction ID, item name, quantity, unit profit, category, and transaction date. MySQL is used as the backend database management system due to its reliability, efficient data storage capabilities, and support for structured query operations. This layer ensures secure and efficient retrieval of transactional data required for further processing. The collected data is passed to the preprocessing layer, where it is cleaned and prepared for utility computation and pattern mining operations.

Step 2: Data Preprocessing Delete Missing values, correct inconsistent entries and transform data to the appropriate form such that analysis can be performed. Step 3: Creation of utility matrix.After preprocessing the transactions database we get utility matrix which has transactions as row entries and items as column entries.The values of each cell in the matrix are calculated using Quantity and profit attribute.

Step 3: Utility Matrix Construction The process of preprocessing transactional data leads to the creation of a utility matrix which displays transactions as row entries while item entries appear as column en-tries. The matrix values are determined through the application of quantity and profit data.

Step 4: Support and Utility Calculation The system determines support values based on the occur-rence count of items in database and utility based on contribution to the business profit by an item.
Step 5: Percentile-Based Threshold Selection
The approach dynamically selects thresholds which calculate support and utility threshold using percentiles rather than having a static threshold to improve performance of pattern selection.
Step 6: Itemset Classification Based on support and utility values system classifies itemsets into three categories i.e HFHU, HFLU and LFHU.
Step 7: Monthly Comparison and Visualiza-tion Visualizations Our System will compare sales from current month and that of previous month to determine decrease in utility and highlight profitable LOW frequency items. Graphs and tables are used to represent our results giving a better understanding of data.

X.SYSTEM ARCHITECTURE

The system architecture uses multiple operational layers which work together for handling transactions and finding important itemsets. The system operates through dedicated functions which each layer uses to exchange information with its neighboring layers for maintaining operational efficiency. This organization system enables better system expansion and makes it easier to handle system components while delivering accurate analytical results.

1. Data Source Layer

The system uses MySQL database system to store transactional data because MySQL provides reliable performance for handling structured database opera-tions. The database system maintains essential trans-action records which include details about Item Name, Quantity, Unit Profit, Category. Data retrieval opera-tions achieve quick results through the application of indexing techniques which also reduce the duration needed for query execution.

2. Data Preprocessing Layer

The collected dataset needs preparation work before analysis begins to ensure its trustworthy status. The record inspection process identifies missing data, du-plicate entries, and Abnormalities during this inspec-tion procedure. The system removes unwanted records to achieve better data quality results. The system re-quires standardized formatting after data cleansing to ensure consistent product name and quantity and profit information distribution. The processed dataset transforms into a structured format which enables further computational processing.

3. Utility Computation Layer

The prepared data from this stage of the process transforms into a matrix form which enables quick calculation operations. The utility values for items are calculated by multiplying the quantity with unit profit value. The representation of transactions takes place through rows which display individual items as column entries. The matrix design enables better performance because it reduces the need for database retrieval operations when managing extensive transaction data.

4. Frequent Pattern Mining Layer

The system uses support values to discover repeated item combinations through this analysis. The sys-tem detects purchasing patterns through frequency analysis which identifies products that customers fre-quently buy together. The purchasing patterns create customer behavior patterns which lead to subsequent classification work.

5. Classification Layer

The stage creates itemset groups according to their calculated support and utility values. The system uses percentile-based methods to automatically gen-erate minimum support and utility limits instead of using manual threshold definitions. The system classi-fies itemsets into three categories based on computed values which include High Frequency High Utility (HFHU) and High Frequency Low Utility (HFLU) and Low Frequency High Utility (LFHU). The classi-fication process identifies items which have a major impact on profit creation.

6. Visualization Layer

The system displays processed results through graphical representations which enhance user com-prehension after they complete classification. The system creates tables and bar

charts and monthly comparison diagrams to display performance trends and revenue changes. The visual outputs enable users to identify significant patterns which exist in the data without needing to manually check the entire transaction dataset.

7. API Integration Layer

The final layer enables external programs to con-nect with the mining system. The FastAPI framework enables this feature through its structured API end-points which give access to processed output informa-tion. External systems can access critical information through the system which includes high-utility items and classified itemsets and trend summaries. The sys-tem provides integration capabilities which enhance usability in real-world applications.

XI.IMPLEMENTATION

The implementation process started with setting up a MySQL database environment. This allowed for man-agement of transactional data. The database tables were created to hold product data and transaction records. The database stores records that contain data fields such as Transaction ID and Item Name and Quantity and Unit Profit and Category.

The database also created indexes on columns that users frequently access. This enhances query perfor-mance. Decreases data response time during retrieval operations.

Step 1: Setting Up the MySQL Environment

We began by setting up the MySQL database en-vironment. This was the step in the implementation process. The MySQL database environment is used for transactional data management. We created database tables to hold product data and transaction records.

The database stores one record that contains the fol-lowing data fields: Transaction ID, Item Name, Quan-tity, Unit Profit and Category. We also established indexes on columns that users frequently access to enhance query performance.

This helps to decrease data response time during re-trieval operations. The MySQL database environment is used for management of transactional data.

Step 2: Connecting to the Database Using Python

We used Python to connect to the MySQL database. This connection allows Python programs to access stored transaction records directly from the database. We used MySQL Connector and PyMySQL database connector libraries to establish the connection.

The database configuration was completed to estab-lish MySQL database connections through Python. This connection enables Python programs to access stored transaction records from the MySQL database.

We used MySQL Connector and PyMySQL database connector libraries to connect to the database.

Step 3: Getting and Loading the Data

We retrieved the required transaction data from the database. Loaded it into Python data structures. These data structures include DataFrames. The data structures provide methods to process data.

This enables users to perform sorting and filtering and aggregation operations. We loaded the dataset into memory. This enables preprocessing and mining tasks to execute

The data structures provide methods to process the data. This includes sorting and filtering and aggrega-tion operations.

Step 4: Cleaning and Preparing the Data

The dataset required examination to discover all missing values and incorrect data points. These needed correction. We improved the dataset through data cleaning procedures.

These procedures removed entries and duplicate rows and inconsistent records. We arranged the cleaned dataset into a format.

This enables computations and pattern mining pro-cedures to function properly. The dataset required examination to discover all missing values.

Step 5: Creating the Utility Matrix

We transformed the prepared dataset into a utility matrix. We used processing libraries such as Pandas and NumPy. The matrix contains rows that represent transactions.

Its columns display all product options. The matrix holds values that show the utility value. This is derived from multiplying item quantity with unit profit.

The matrix-based structure enables computation. It minimizes the need to frequently access databases. The utility matrix is used to calculate the utility value.

Step 6: Calculating Support and Utility Values

We determined the support and utility values for

every item. We used established equations. The support values indicate how often items appear across all transactions.

The utility values indicate profit provided by each item. The computed values create a foundation. This enables researchers to discover patterns in their dataset.

The support and utility values are used to identify itemsets. We calculated the support and utility values for every item.

Step 7: Determining Threshold Limits

We applied analysis to automatically set threshold limits. These limits determined support and utility points.

Of allowing researchers to set these limits manually we used percentile analysis. The determined percentile values functioned as thresholds.

These thresholds identified itemsets. The system achieves adaptability through its dynamic threshold selection method.

This provides results on datasets with varying sizes and data distributions. We used analysis to determine threshold limits.

Step 8: Classifying Itemsets

The threshold values determined through computation enabled the creation of three itemset categories. These are High Frequency High Utility (HFHU) High Frequency Low Utility (HFLU) and Low Frequency High Utility (LFHU).

This classification enables the identification of items that generate profit. Even if they occur frequently in transactions these items are identified.

The classification enables the identification of items that generate profit. We classified itemsets into three categories.

Step 9: Creating the Visualization Dashboard

The mining results required visualization capabilities. This led to the creation of an interface through Streamlit.

The dashboard presents outputs. These include bar charts and utility comparisons and monthly performance trends.

The visual elements help users to comprehend patterns that exist in the data. Without needing to check the data users can understand the patterns.

We created a visualization dashboard to present the mining results.

Step 10: Integrating the API, for Result Ac-cess

A backend service using FastAPI was established. This enables users to access mining results through API endpoints.

The endpoints enable applications to request processed data. This includes utility items and classified itemsets and statistical summaries.

The system integration allows users to deploy real-world applications. It improves system accessibility. We integrated the API for result access.

The API integration enables users to access mining results through API endpoints.

XII.RESULT AND DISCUSSION

Researchers conducted tests on the suggested matrix-based utility mining system by using sales data from transactions that recorded multiple product categories. The system successfully calculated support and util-ity values for each item and applied percentile-based dynamic thresholds to identify meaningful patterns. The results show that the proposed method effectively identified Low Frequency High Utility (LFHU) items which include high-value products that appear less fre-quently but contribute significantly to overall profit. The dynamic percentile-based approach provided bet-ter results than traditional fixed-threshold methods be-cause it reduced unnecessary pattern generation while improving classification accuracy. Graphical visualiza-tion techniques used bar charts and monthly comparison graphs as tools to display utility trends across different categories. The analysis also enabled compar-ison between current month and previous month sales, which helped to identify profit variations and high-performing products. The proposed method showed increased efficiency while using lower computational power to find profitable patterns in transactional datasets. The system demonstrated efficient perfor-mance on large-scale datasets, which confirmed its ability to handle increasing workload. The visualiza-tion dashboard integration improved usability, which established clear graphical insights for users. The API layer enables developers to implement the system in real-world environments while creating links to other systems.

XIII.CONCLUSION

A matrix-oriented framework is introduced for high utility itemset mining, employing percentile-driven dynamic thresholding to improve pattern discovery within transactional databases. This framework con-verts transactional data into a utility matrix,

facilitating the computation of support and utility values, and categorizing itemsets into HFHU, HFLU, and LFHU groups.

This novel approach overcomes the drawbacks of conventional techniques by avoiding redundant database scans and eliminating reliance on static thresholds. By analyzing low-frequency, high-profit items, the system aids retail and sales strategies with actionable insights.

The findings indicate that matrix-based frameworks enhance processing efficiency, while dynamic thresh-olding techniques improve the identification of significant itemsets. The generated visualizations offer users a clearer understanding of the relationships between sales performance and profitability.

The system becomes more appropriate for actual use because it now includes scalability testing and in-teractive visualization together with API integration.

XIV.FUTURE WORK

The system functions well for transactional data anal-ysis but requires enhancements in three specific areas. The first area requires real-time data processing to monitor sales patterns throughout the day. The second area requires advanced machine learning techniques to forecast which items will generate future profits. The third area requires distributed computing frame-works to enable the model to process extremely large datasets. Web-based dashboards will allow users to create interactive visualizations and generate reports. The system now uses extra parameters to analyze seasonal trends and customer behavior patterns. The system will gain from these upgrades which will boost its ability to scale and use in real-world situations. The upcoming improvements will introduce three fea-tures which include real-time data streaming, machine learning-based prediction, and cloud deployment for large-scale distributed environments.

REFERENCES

[1] In 2020, Fournier-Viger, Lin, and Kiran's review article on Pattern Mining of High Utility Itemsets was featured in *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. https://doi.org/10.1002/widm.1359

[2] Lin, Li, and Fournier-Viger's work, "Efficient Algorithms for Mining High Utility Itemsets from Large Databases," appeared in *IEEE Access* in 2020. https://doi.org/10.1109/ACCESS.2020.296615

[3] Ahmed, Abdullah, and Hossain's "An Efficient Utility Mining Approach Using Dynamic Threshold-ing" was published in the *Journal of Big Data* in 2021. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00474-5

[4] Kiran and Fournier-Viger's "Current Research Insights in High Utility Itemset Mining" was published in *Knowledge and Information Systems* in 2022. https://doi.org/10.1007/s10115-021-01636-4

[5] Li, Lin, and Fournier-Viger introduced "Matrix-based High Utility Pattern Mining" in 2023, a method designed for enhanced efficiency, as documented in their IEEE publication. https://doi.org/10.1109/TKDE.2022.3149278