5580 Data and Text Mining

Master of Science in Computing and Data Analytics

Department of Mathematics and Computing Science

Association Rule Mining

Group 4:
Bhavik Kantilal Bhagat (A00494758)
Jeevan Dhakal (A00494615)
Binziya Siddik (A00494129)

Date: February 1, 2026

Contents

1 Theoretical Framework

Association Rule Mining (ARM) is an **unsupervised learning** technique for discovering hidden relationships in large datasets. Unlike supervised learning, ARM identifies local structures by uncovering sets of items that frequently co-occur. While ARM defines the objective, the **Apriori Algorithm** is the classical methodology used to solve this task efficiently.

1.1 The Apriori Algorithm

The Apriori algorithm uses a *level-wise search*, leveraging frequent k-itemsets to explore $(k+1)$-candidates. Its efficiency stems from pruning the search space before database support counting.

1.1.1 Join-and-Prune Logic

The algorithm iterates through two primary steps at each level k:

1. **Join Step ($L_{k-1} \bowtie L_{k-1}$):** Generates candidate k-itemsets (C_k) by joining frequent $(k-1)$-itemsets (L_{k-1}) with themselves.
2. **Pruning Step:** For any candidate in C_k, the algorithm evaluates its $(k-1)$-subsets. If any subset is missing from L_{k-1}, the candidate is pruned; by the Downward Closure Property, it cannot be frequent [1].

1.2 Key Metrics in ARM

The significance of discovered rules is quantified using three mathematical pillars:

- **Support (σ):** Frequency of an itemset in the total transaction population.

$$\sigma(A \to B) = P(A \cup B) = \frac{\text{Count}(A \cup B)}{\text{Total Transactions}}$$

- **Confidence (γ):** Probability the consequent B appears given the antecedent A.

$$\gamma(A \to B) = P(B|A) = \frac{\sigma(A \cup B)}{\sigma(A)}$$

- **Lift (L):** Strength of a rule by comparing observed support to expected support if A and B were independent.

$$L(A \to B) = \frac{\gamma(A \to B)}{\sigma(B)}$$

  $L > 1$ indicates a positive correlation, while $L = 1$ suggests item independence.

2 Implementation Stack and Data Engineering

2.1 Analytical Engine: DuckDB

DuckDB, an in-process OLAP database, served as the primary engine for high-speed data transformation. By utilizing vectorized execution and columnar storage, it bypassed traditional RDBMS bottlenecks, allowing for sub-second aggregation of 39,000 milestones. Specifically, DuckDB facilitated a zero-copy pipeline to extract raw logs from MySQL and serialize them directly into Python DataFrames for analysis.

2.2 Record Filtering and Cardinality Constraints

To ensure the integrity of the $A \rightarrow B$ relationship, we excluded transactions containing only a single milestone. Because ARM metrics require disjoint sets to establish predictive logic, a basket cardinality of $n \geq 2$ is mandatory. This filtering process resulted in a refined dataset:

- **Session Basket # of Records:** 20772
- **User Basket # of Records:** 2352

This reduction refocused computational resources on discovering multi-step behavioral sequences rather than isolated, non-associative actions. For that anlaysis of schema, relationships, joins, etc. were conducted, code related to that can be found in python scripts in **code/** of GitHub.

2.3 Algorithmic Framework: mlxtend

Frequent pattern mining was implemented using the **mlxtend** library [2]. According to the documentation, *mlxtend* provides an industry-standard interface for association rule discovery, supporting efficient boolean matrix transformations. We utilized its mathematically rigorous implementations of both **Apriori** and **FP-Growth** to calculate support, confidence, and lift across the SimplyCast dataset. The study mainly focus on using**Apriori** algorithm, however **FP-Growth** algorithm is discussed in section 7

3 Apriori Algorithm & Hyper-parameter Tuning

The extraction of non-trivial behavioral patterns from high-dimensional interaction data was governed by a systematic 9-step tuning lifecycle. This process was executed on an **Intel Core i9** workstation with **32GB of DDR5 RAM**, ensuring that the "Exponential Explosion" of candidate itemsets remained computationally manageable.

3.1 The Support Sweep: Identifying the Interaction Elbow

A broad "Support Sweep" was performed to identify the boundary between statistical sparsity and combinatorial saturation. This phase involved iteratively adjusting the minimum support threshold (σ) and monitoring the resultant rule density.

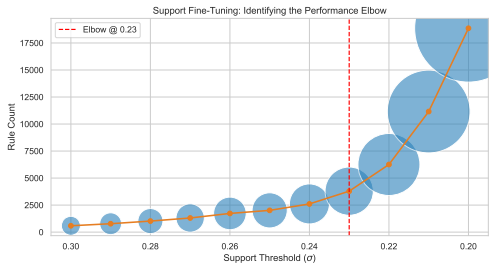

Figure 1: User Support Tuning

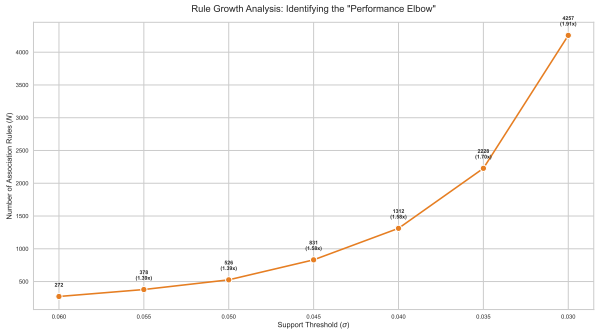

Figure 2: Support Elbow Analysis

1. **User-Level Support Elbow ($\sigma = 0.23$):** For aggregated user personas, the initial sweep revealed that behavioral rules are fragmented above $\sigma = 0.50$. The optimal "Support Elbow" was identified at **0.23**. At this threshold, the algorithm transition from identifying generic interactions to capturing established behavioral signatures shared by a significant "Mastery Tier" of the user population. Further lowering the support explode candidate rules.
2. **Session-Level Support Elbow ($\sigma = 0.045$):** Due to the granularity of single-interaction windows, the session support floor was found to be significantly lower. The elbow was pinpointed at **0.045**. Settings below this coordinate triggered a non-linear spike in candidate generation, while settings above it failed to capture the local functional transitions (e.g., report loading to exporting) that define platform utility.

3.2 Lift Stability and the Confidence Plateau

Once the support elbows were established, a sensitivity analysis was conducted on the confidence threshold (γ) to ensure the reliability of inferred "If-Then" relationships. Confidence was swept from 0.10 to 0.90 while monitoring the **Average Lift (L)**. A "Lift Stability

Plateau" was observed in the confidence range of $0.55 - 0.65$ for User Figure 3 and in the confidence range $0.75 - 0.80$ for Session transitions Figure 6.

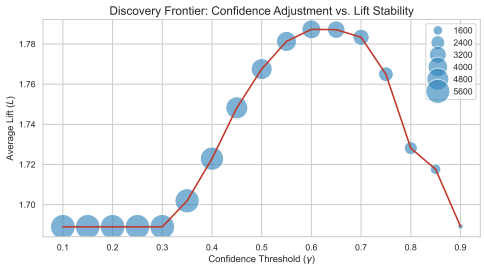

Figure 3: User Confidence Tuning

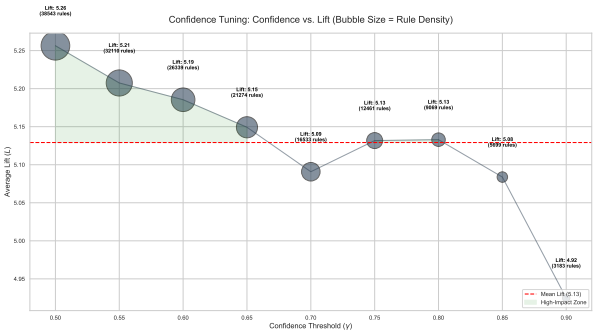

Figure 4: Session Confidence Tuning

- **Filtering Triviality:** Lower confidence thresholds yielded a higher volume of rules, but these were frequently characterized by low lift values, representing background noise or random behavioral overlaps.
- **Metric Fixation:** By fixing confidence at the plateau was the most crucial step for the resulting rules-set to be non-trivial. Rules retained after this gate exhibited a higher Lift than stabilized one, meaning the discovered patterns were at least **3x more likely** for user and **4x more likely** for session than random chance, effectively isolating the "Elite" professional workflows from casual interaction.

3.3 Computational Performance Profiling

The use of the Core i9 architecture was instrumental during the tuning of session-level data. As σ approached the 0.045 elbow, the memory footprint of the candidate generation phase scaled exponentially. The 32GB RAM capacity allowed for the storage of high-cardinality itemsets in-memory, facilitating the rapid iterations required to find the optimal "Active Zone" of association rules without sacrificing the granularity of the discovery frontier.

4 The Performance Paradox: Apriori vs. FP-Growth

A critical component of this study was the comparative benchmarking of the two primary Association Rule Mining (ARM) algorithms: the classical **Apriori** and the tree-based **FP-Growth**. Benchmarking was conducted on high-performance infrastructure comprising mentioned in Table 2 to evaluate computational efficiency across the established support elbows.

4.1 Benchmark Results: The Apriori Advantage

Unexpectedly, the performance data revealed a consistent speed advantage for the Apriori algorithm over FP-Growth across both evaluation granularities. At the session level ($\sigma = 0.045$), Apriori completed the mining process in approximately **1.8 seconds**, whereas FP-Growth required **46.5 seconds**—a performance gap of over 25x in favor of the older algorithm. A similar, though less pronounced, trend was observed at the user level, where Apriori outperformed FP-Growth by a factor of 10x (0.24s vs 3.1s).

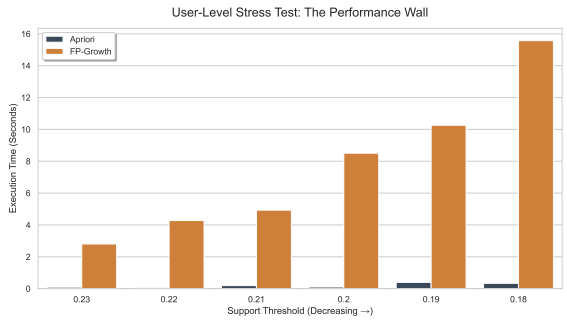

Figure 5: User Benchmarking

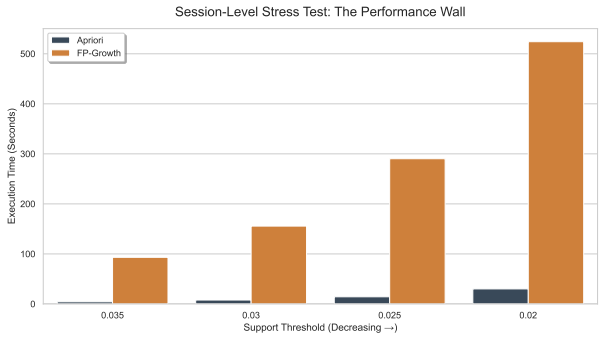

Figure 6: Session Benchmarking

Figure 7: Apriori v FPG Performance Comparison

4.2 The Insight: Sparsity and Shallow Tree Depth

The "Apriori Paradox" observed in this dataset is explained by the structural characteristics of the SimplyCast interaction matrix:

- **Data Sparsity:** While the platform contains over 39,000 distinct milestones, the average interaction basket is extremely sparse. In such environments, the **Downward Closure Property** utilized by Apriori is highly effective. Most candidate combinations are pruned in the first two levels of search, meaning the algorithm only scans a minute fraction of the total possible search space.
- **Shallow Itemset Depth:** As illustrated in the **Itemset Spiral Donut Chart** (see Figure 11), the majority of frequent itemsets in this behavioral dataset have a cardinality

of $k \leq 3$. Since Apriori only requires $k+1$ passes over the database, its breadth-first approach is highly efficient for shallow patterns.

In contrast, the FP-Growth algorithm incurs a significant "architectural tax" by building a full **FP-Tree** structure and performing recursive conditional tree mining. For datasets where the pattern depth is minimal, the overhead of tree construction and recursive traversal far exceeds the cost of Apriori's simple candidate generation.

Thus, simple **Apriori algorithm was superior** for our dataset to perform Association Rule Mining.

4.3 The Crossover Theory: When FP-Growth Becomes Superior

While Apriori dominated the primary experiments, the theoretical crossover point where FP-Growth would become the superior choice was also explored. FP-Growth is mathematically superior under the following conditions:

1. **Lower Support Thresholds:** As support approaches the *Performance Wall* the number of candidates in Apriori grows combinatorially (C_n^k), leading to CPU exhaustion, which was observed for support $\sigma < 0.2$ for user basket, and $\sigma < 0.045$ for session basket.
2. **High Data Density and Long Itemsets:** In environments where *users trigger dozens of concurrent milestones*, the FP-Tree achieves massive compression ratios. However, as shown in our donut-piechart visualizations Figure 11 and Figure 10, the lack of "long-tail" itemsets in the SimplyCast data meant that the suffix-tree compression of FP-Growth never reached the efficiency required to offset its structural overhead.

5 Discussion of Results and Business Intelligence

The transition from mathematical itemset generation to actionable business intelligence requires a synthesis of algorithmic performance and behavioral context. This chapter evaluates the "Elite 20" ruleset discovered in the SimplyCast dataset, mapping these associations to tangible user personas and strategic recommendations.

5.1 The "Elite" vs. "Optimized" Metric Strategy

During the exploratory tuning phase, recursive parameter sweeps were utilized to identify the "Optimized" mining coordinates ($\sigma = 0.23, \gamma = 0.60$). While these thresholds effectively captured the "Support Elbow"—the point of maximum discovery reach—they were found to be insufficient for high-level strategic auditing.

To isolate the most deterministic behavioral signatures, a secondary **Elite Filter** was implemented:

1. **Confidence Gate ($\gamma \geq 0.65$):** This serves as the *Reliability Gate*, ensuring that the conditional probability of the relationship is high enough to warrant functional changes to the UI.
2. **Lift Threshold ($L \geq 3.0$):** This serves as the *Interest Gate*. A Lift value ≥ 3.0 establishes that the observed behavior is **3x more likely** than random chance. By applying this gate, we successfully filter out trivial "background" patterns (e.g., *Dashboard* $\rightarrow$ *Login*) and focus exclusively on specialized professional workflows.

5.2 Granular Characteristics: Session vs. User Layers

Behavioral discovery was conducted at two layers of granularity, each revealing a unique "fingerprint" of the SimplyCast interaction model.

5.2.1 Session Characteristics: The Tactical Layer

Session-level rules describe the **mechanics of the platform**. These are transient, high-frequency transitions that focus on UI/UX micro-interactions. Characteristics identified in the session data include procedural sequences such as immediate text formatting (`UpdateBlock`) following image insertion (`AddImage`). These rules provide the technical baseline for reducing user friction at a functional level.

5.2.2 User Characteristics: The Strategic Layer

User-level rules capture the **intent of the platform masters**. This layer describes "Macro-habits" built over long-term platform engagement. The characteristics identified here are intent-driven, specifically the consistent loop between campaign deployment (`SendNow`)

and analytical auditing (`OpenReport`, `OpensAndBounces`). Unlike tactical session rules, these strategic fingerprints define the core utility and ROI-seeking behavior of the user base.

5.3 Persona Profiling: Mapping the Elite 20 Rules

Through the analysis of the Elite 20 ruleset (see Table 1), three distinct behavioral archetypes have been identified within the SimplyCast ecosystem.

1. **The Content Creator:** This persona is characterized by high-lift associations in design-heavy milestones, specifically `AddImage`, `UploadImage`, and font variations. Their workflow is asset-centric, prioritize aesthetic output over deep data inquiry.
2. **The Data Investigator:** Defined by a reliance on analytics-heavy milestones such as `OpensAndBounces`, `OpenData`, and `ExportData`. This persona treats the platform as a data intelligence engine, regularly auditing engagement metrics to refine their audience segments.
3. **The Campaign Manager:** Characterized by high-frequency operational milestones. Their behavioral footprint is anchored by `Dashboard` navigation and the final execution milestone `SendNow`. This archetype represents the tactical "orchestrator" who manages the movement of campaigns from preparation to production.

5.4 Business Intelligence and Actionable Insights

The discovery of these strategic personas provides a mathematical roadmap for platform optimization and customer success.

5.4.1 Product Optimization and Feature Bundling

Based on the Elite rules, SimplyCast should prioritize "bundling" features that exhibit a Lift ≥ 3.0. For example, the strong association between `OpenData` and `OpensAndBounces` suggests that users auditing raw data are highly likely to track engagement metrics. Reducing the functional distance between these two modules in the UI would significantly minimize user friction.

5.4.2 Customer Success and Churn Prevention

"Strategic Fingerprints" can be utilized as a diagnostic tool for customer health. If a user previously identified as a "Data Investigator" begins to deviate from their established habits (e.g., they stop auditing reports after a campaign deployment), this reduction in Lift stability serves as a leading indicator of waning platform engagement, allowing for proactive outreach.

5.4.3 Targeted Marketing and Persona Customization

Marketing automation should be customized based on these behavioral archetypes. Rather than sending generic updates, SimplyCast can utilize the identified personas to send targeted tool recommendations—such as creative design tips for "Content Creators" or advanced analytics tutorials for "Data Investigators."

Table 1: The Elite 20 Ruleset: Strategic Personas and Workflows ($\sigma = 0.23, \gamma \geq 0.65, L \geq 3.0$)

Antecedents	Consequents	Support	Conf.	Lift
{OpenData}	{OpensAndBounces}	0.233	0.934	3.51
{AddImage, SendNow}	{UploadImage}	0.243	0.835	3.14
{ManageTab, UploadImage}	{AddImage, SendNow}	0.243	0.914	3.14
{OpensAndBounces, SendNow}	{OpenReport, ManageTab}	0.238	1.000	3.14
{ReportsTab, SendNow}	{OpenReport}	0.237	0.996	3.14
{OpenReport, ManageTab}	{OpensAndBounces, SendNow}	0.238	0.748	3.14
{AddImage, UploadImage}	{SendNow}	0.243	1.000	1.34
{OpenData, OpenReport}	{OpensAndBounces}	0.233	0.934	3.51
{OpensAndBounces}	{OpenData}	0.233	0.878	3.51
{OpensAndBounces, ManageTab}	{OpenReport, ReportsTab, SendNow}	0.237	0.943	3.14

6 Executive Summary and Strategic Conclusions

This study has successfully implemented a high-performance Association Rule Mining (ARM) pipeline to decode the interaction DNA of SimplyCast's user base. By transitioning from raw behavioral logs to structured "Elite" association rules, we have identified both the tactical workflows and strategic personas that drive platform engagement.

6.1 Key Findings and Discovered Patterns

Our analysis revealed deep-dive behavioral correlations with significant statistical lift:

- **Tactical Sequences (Session-Level):** We observed high-confidence (> 0.90) transitions between "Report Loading" and "Exporting" milestones, indicating a clear functional path for data-driven users.
- **Strategic Personas (User-Level):** Higher lift patterns ($L \approx 3.5$) emerged between data management tasks (`OpenData`) and engagement metrics (`OpensAndBounces`). This suggests a "Power User" archetype that actively monitors the success of their data-driven campaigns.
- **Content Creation Workflows:** Rules involving `AddImage` and `UploadImage` showed strong associations with `SendNow`, mapping the path from content asset preparation to immediate campaign execution.

6.2 Algorithmic Insights: The Apriori Paradox

The benchmarking phase provided a counter-intuitive insight: in sparse, shallow behavioral datasets like SimplyCast's, the classical **Apriori algorithm frequently outperforms FP-Growth**.

- **Structural Efficiency:** Apriori's ability to prune the candidate space early via the Downward Closure Property makes it highly efficient for sparse binary matrices.
- **Overhead vs. Depth:** For patterns with a maximum length of 3 or 4 items, the architectural cost of building and recursively mining an FP-Tree exceeds the cost of Apriori's breadth-first passes.

6.3 Business Implications for SimplyCast

The discovered rules provide actionable intelligence for product development and marketing automation:

1. **Feature Cross-Promotion:** Since users who `OpenData` are highly likely to track `OpensAndBounces`, the UI should offer direct shortcuts between these modules to minimize friction.

2. **Persona-Based Onboarding:** New users exhibiting the early stages of "Power User" rules (e.g., uploading images) can be nudged toward the next logical step in the discovered sequence, such as campaign testing or scheduling.
3. **Infrastructure Optimization:** Our use of **DuckDB** and **mlxtend** demonstrates that meaningful behavioral discovery can be performed on consumer-grade high-performance hardware without the need for expensive distributed computing clusters.

In conclusion, Association Rule Mining serves as a powerful "Deep-Dive Discovery" tool, transforming disparate behavioral logs into a roadmap for enhancing user experience and platform mastery.

7 Appendix

7.1 Project Repository and Artifacts

- **Source Code (GitHub):** ARM Project Repository
- **Exploratory Notebooks:** Detailed analysis of the 9-step tuning lifecycle, one-hot encoding, and algorithmic benchmarking is available in the `/code` directory of the repository.

7.2 Video Link

Youtube

7.3 Benchmarking Machine

Table 2: Machine Architecture Specifications

Component	Specification
Processor	Intel Core i9-13900HK (20 cores)
Base Clock	[e.g., 3.0 GHz]
Memory (RAM)	32 GB
Operating System	Zorin OS 18
OS Architecture	64-bit
Desktop Environment	GNOME (typically)
Graphics	[e.g., NVIDIA RTX 4070, Intel UHD Graphics]

7.4 Mastery Tier: Elite Association Rules (User Level)

The following table summarizes the high-impact rules discovered at the User-Level ($\sigma = 0.23, \gamma = 0.60, L \geq 3.0$). These rules represent the "Strategic Personas" identified in our analysis.

Table 3: Strategic Elite Rules - SimplyCast User Personas

Antecedent	Consequent	Support	Conf.	Lift
`OpenData`	`OpensAndBounces`	0.233	0.933	3.51
`OpenReport, OpenData`	`OpensAndBounces`	0.233	0.933	3.51
`AddImage, SendNow`	`UploadImage`	0.243	0.835	3.14
`OpenReport, SendNow`	`ManageTab`	0.238	0.790	3.14
`ReportsTab, SendNow`	`OpenReport`	0.237	0.996	3.14

7.5 Diagnostic Visualizations

The following figures illustrate the sensitivity analysis and performance benchmarking conducted during the study.

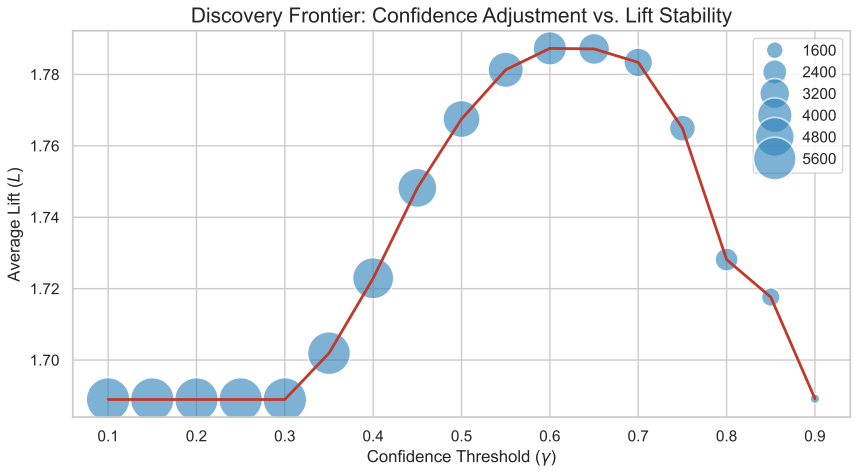

Figure 8: Discovery Frontier: Mapping the Lift Stability Plateau for User-Level Personas.

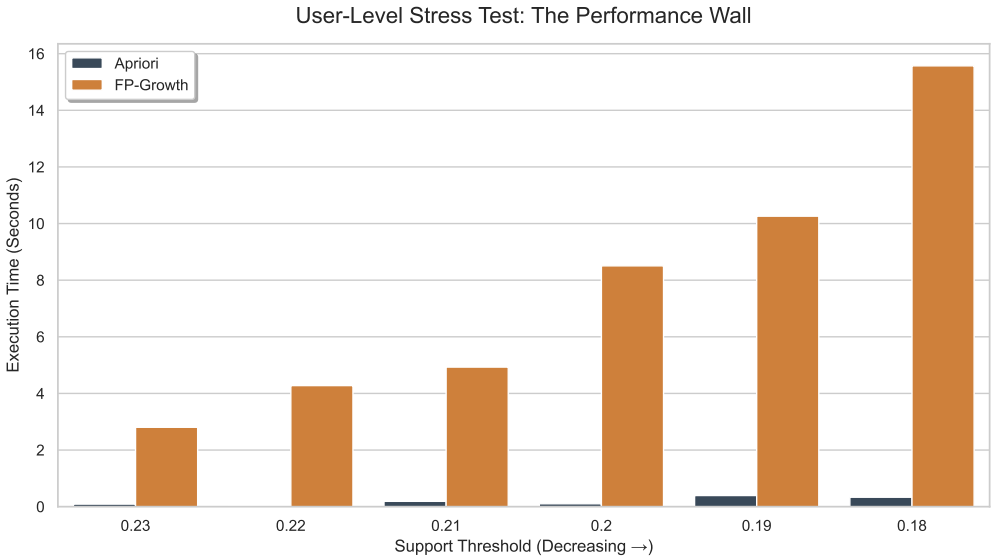

Figure 9: User-Level Performance Wall: Comparison of Apriori and FP-Growth efficiency.

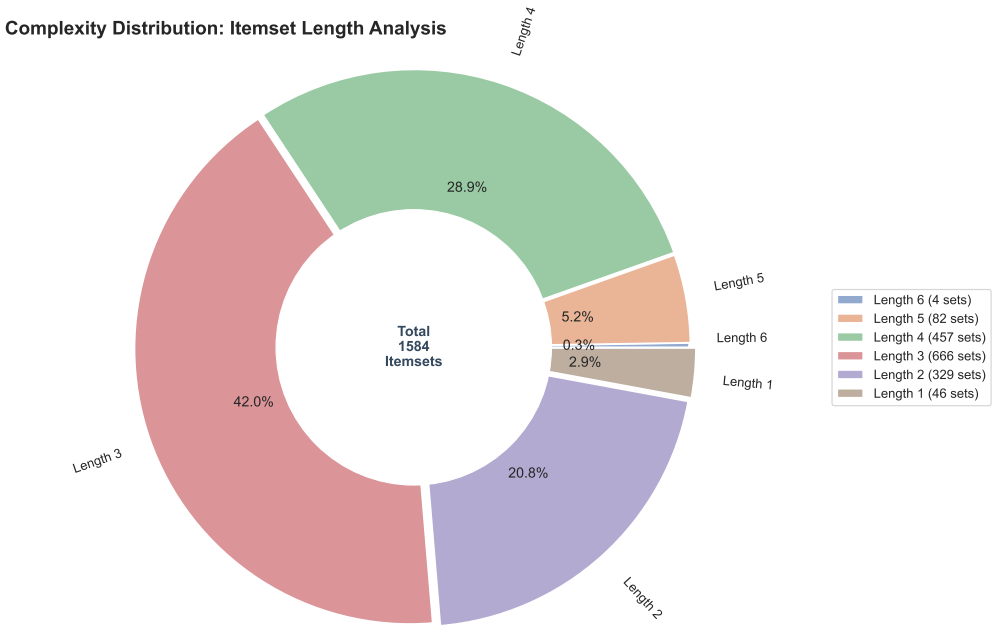

Figure 10: Session Itemset Distribution

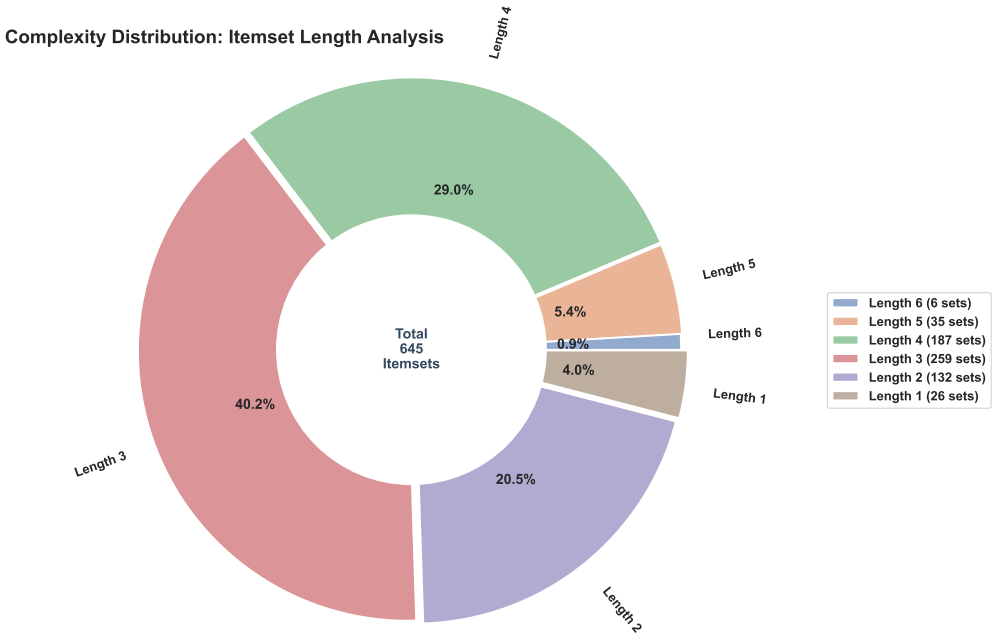

Figure 11: User Itemset Distribution

Bibliography

References

[1] GeeksforGeeks. **Apriori Algorithm in Machine Learning**. Accessed: 2026-02-01. 2024. URL: https://www.geeksforgeeks.org/apriori-algorithm-in-machine-learning/ (cit. on p. 3).

[2] Sebastian Raschka. **mlxtend: Machine Learning Extensions**. Accessed: 2026-02-01. 2024. URL: https://rasbt.github.io/mlxtend/ (cit. on p. 4).