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 Introduction and Overview

Association Rule Mining (ARM), often referred to as **Market Basket Analysis** in business intelligence, 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. 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].

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.

1.2 Objectives

The SimplyCast behavioral dataset presents a high-cardinality environment with over 39,000 distinct milestones. Traditional descriptive statistics are insufficient for uncovering the latent logic of user interaction within such a sparse and complex feature space. This project utilizes Association Rule Mining (ARM) to transition from simple frequency counts to a structured understanding of user intent and platform utility.

1.3 Analytical Bipartition: Action vs. Habit

To provide a holistic view of the SimplyCast ecosystem, the research is divided into two distinct analytical granularities. This dual-lens approach allows for the differentiation between transient software mechanics and long-term professional objectives.

1.3.1 Objective I: Tactical Session-Level Discovery

The primary goal of the session-level analysis is to map the **Micro-Mechanics** of the user interface. By focusing on a single interaction window, this objective seeks to:

- Identify immediate, sequential transitions between milestones.
- Uncover functional coupling (e.g., feature sets that are consistently used in tandem).
- Provide data-driven insights for **UI/UX Optimization**, such as streamlining menu hierarchies and feature bundling.

1.3.2 Objective II: Strategic User-Level Synthesis

The user-level analysis shifts focus toward identifying **Macro-Habits** and long-term behavioral intent. By aggregating a user's entire historical footprint, this objective aims to:

- Decode the "Strategic Fingerprint" of different user classes.
- Synthesize established rules into **Professional Personas** (e.g., Content Creators vs. Data Investigators).
- Enable **Predictive Customer Success** strategies, allowing SimplyCast to personalize onboarding and identify churn risks when a user's behavioral patterns deviate from their established archetype.

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 6

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 algorithm was executed on the machine with specification mentioned in Table 1, ensuring that the "Exponential Explosion" of candidate itemsets remained computationally manageable.

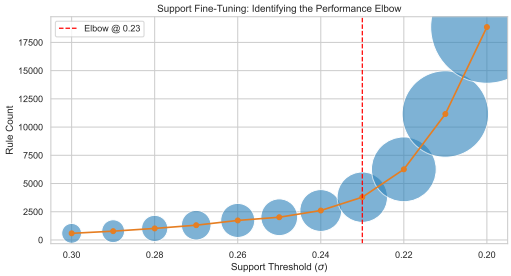

Figure 1: User Support Tuning

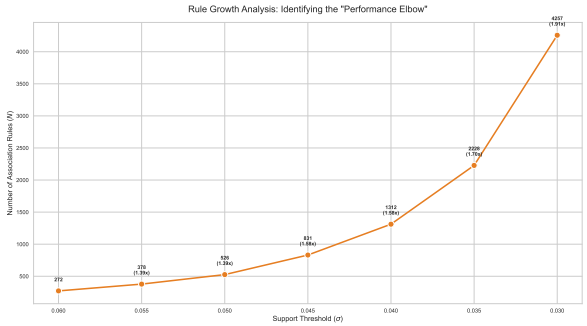

Figure 2: Session Support Tuning

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.

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.

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

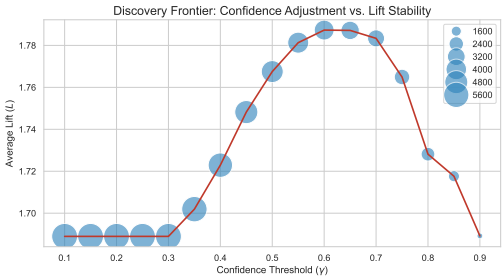

Figure 3: User Confidence Tuning

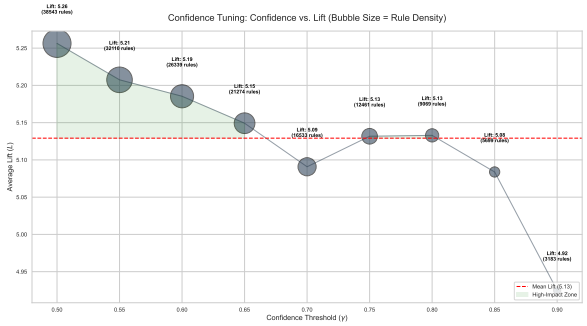

Figure 4: Session Confidence Tuning

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 & in the confidence range $0.75 - 0.80$ for Session transitions Figure 4.

- **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.

In addition, interactive plot for session hyper-parameter turning was generated using **plotly** library. A static version of which is provided in Figure 10.

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 1 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).

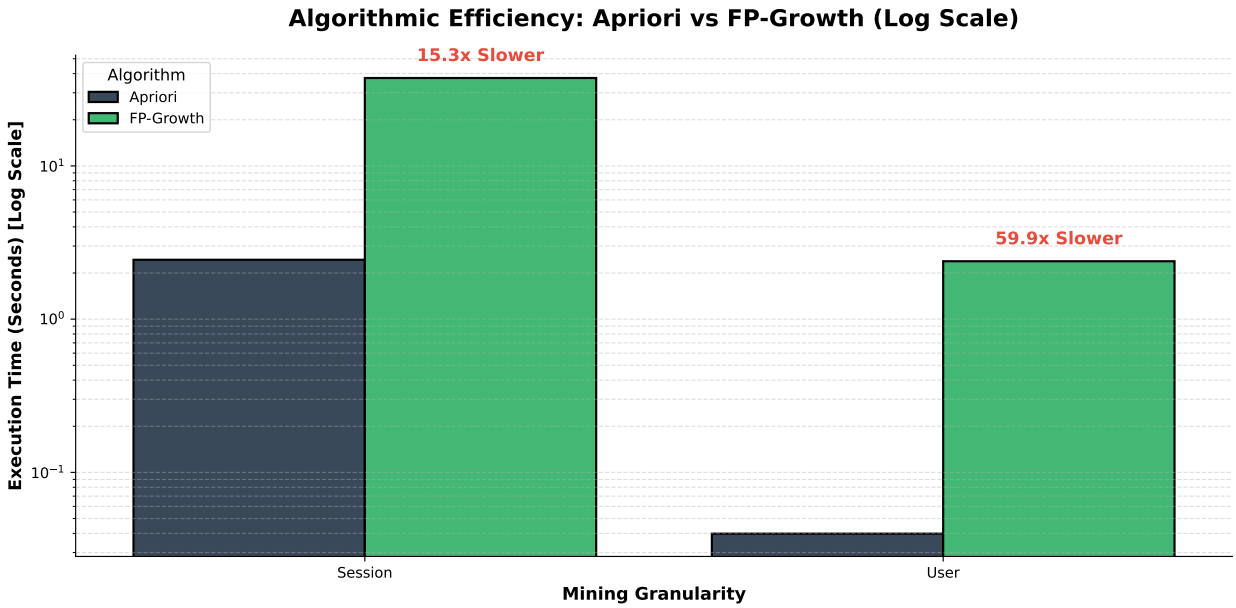

Figure 5: Apriori v FPG Performance Comparison

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 9), 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 9 and Figure 8, 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 Results and Summary

This chapter synthesizes the "Elite 20" rulesets into actionable personas and strategic recommendations, transitioning from mathematical generation to platform auditing. Please note that all the rulesets are provided in section 7 in Table 2 for users' basket and Table 3 for sessions' basket.

5.1 The Elite Metric Strategy: User & Session

While initial tuning identified the "Support Elbow" at ($\sigma = 0.23, \gamma = 0.60$) for user & ($\sigma = 0.045, \gamma = 0.80$)), we implemented a secondary **Elite Filter** to isolate the most deterministic signatures:

- **Reliability Gate:** Ensures conditional probability is high enough to justify UI modifications. For session-rulesets $\gamma = 0.9$ was filter was applied.
- **Interest Gate:** For user rulesets Lift $L \geq 3.0$ confirms behaviors are **3x more likely** than random chance, likewise for session it Lift was set to $L \geq 7.0$, filtering trivial patterns like *Dashboard* $\rightarrow$ *Login* in favor of specialized workflows.

5.2 Persona Profiling

Analysis of the top associations reveals three distinct archetypes:

1. **The Content Creator:** Focused on design milestones (`AddImage`, `UploadImage`). Prioritizes aesthetic output over data inquiry.
2. **The Data Investigator:** Relies on analytics (`OpensAndBounces`, `OpenData`). Treats the platform as a data intelligence engine to refine audience segments.
3. **The Campaign Manager:** An operational orchestrator defined by `Dashboard` navigation and the final `SendNow` execution.

5.3 Actionable Insights

- **Feature Bundling:** SimplyCast should prioritize bundling features with $L \geq 3.0$ (e.g., linking `OpenData` directly to `OpensAndBounces`) to minimize functional distance.
- **Churn Prevention:** Strategic fingerprints serve as health diagnostics. Deviation from established habits (e.g., a "Data Investigator" stopping report audits) acts as an indicator of warning.
- **Customization:** Marketing and on-boarding should be persona-specific, offering creative tips to "Creators" and advanced analytics to "Investigators."

5.4 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.

5.5 Algorithmic Insights: The Apriori vs FP-Growth

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.

5.6 Business Implications for SimplyCast

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

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

In conclusion, **Market Basket Analysis** via 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.

6 Appendix A

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.

Video Link

Youtube

Chat Sessions

- Gemini
- DeepSeek 1
- DeepSeek 2

Benchmarking Machine Specification

Table 1: Machine Architecture Specifications

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

Visualizations

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

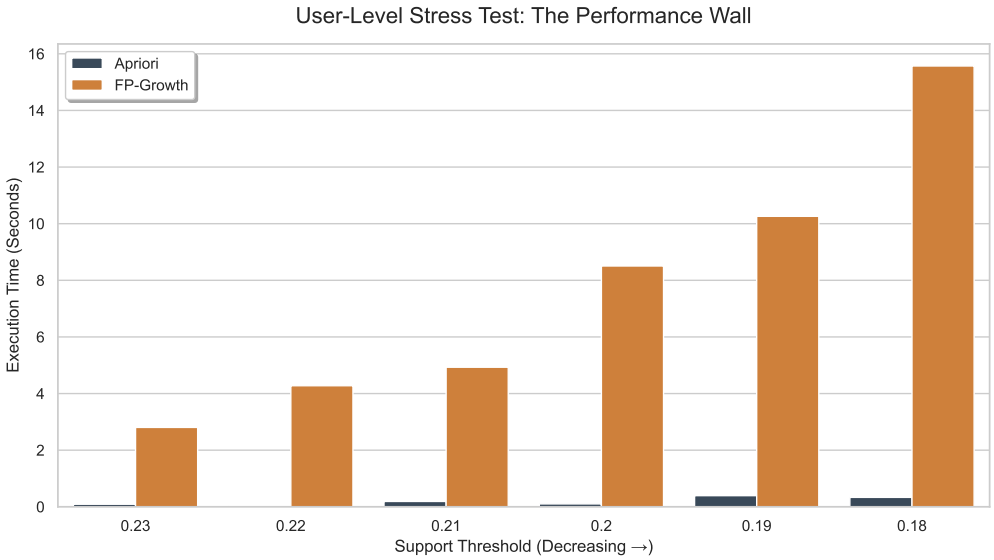

Figure 6: User-Level Performance Wall: Comparison of Apriori and FP-Growth efficiency.

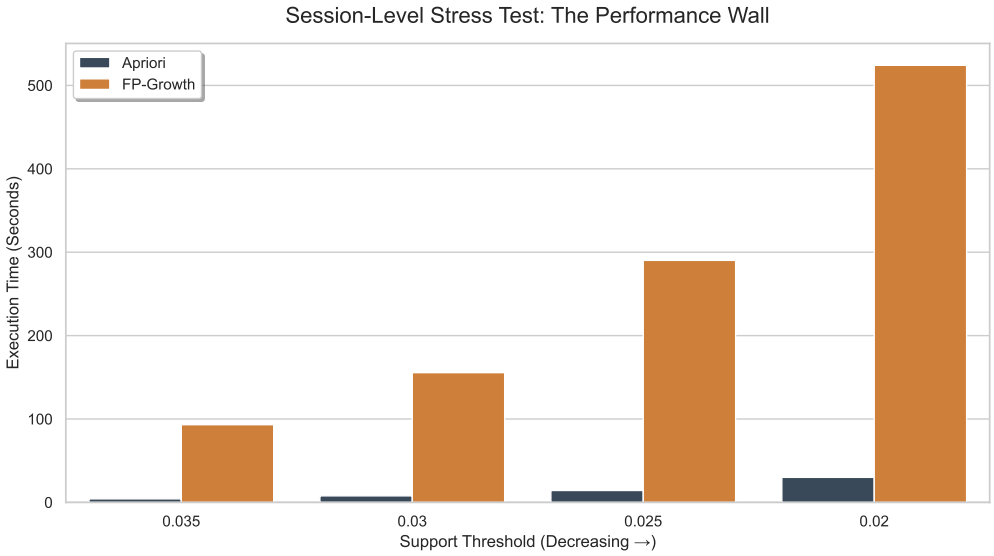

Figure 7: Session-Level Performance Wall: Comparison of Apriori and FP-Growth efficiency.

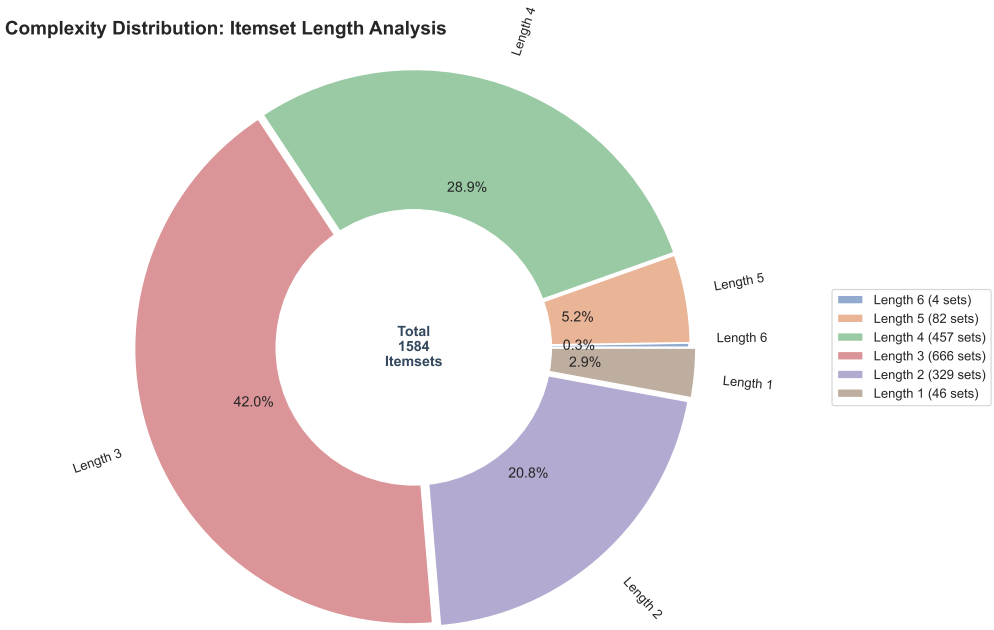

Figure 8: Session Itemset Distribution

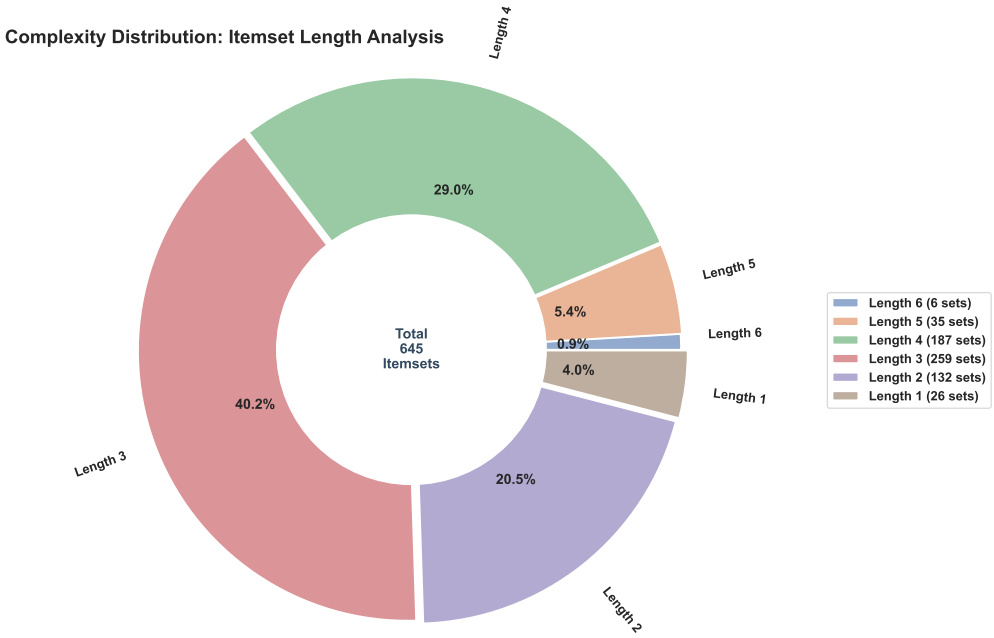

Figure 9: User Itemset Distribution

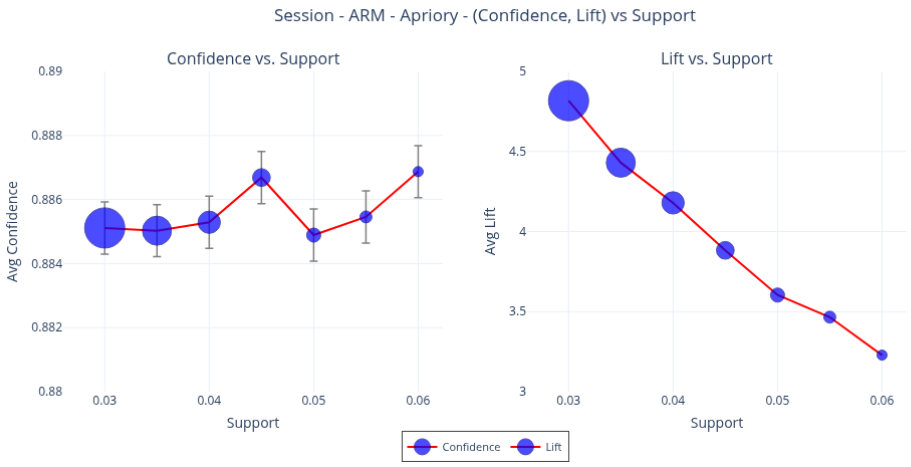

Figure 10: Session hyper-parameter Tuning

7 Appendix B: Extended Ruleset Analysis

This section provides the full technical breakdown of the elite association rules mined at both the user and session levels. These rules represent the most reliable and significant behavioral patterns identified during the analysis.

User-Level Strategic Workflows

The following table details the top-tier rules mined from user-aggregated baskets. These rules define the *Content Creator*, *Data Investigator*, and *Campaign Manager* personas discussed in section 5 by capturing the relationships between high-level platform interactions.

Table 2: User-Level Elite Ruleset: Behavioral Signatures and Metrics

ID	Antecedents	Consequents	Supp.	Conf.	Lift
1	OpenData	OpensAndBounces	0.233	0.934	3.51
2	OpensAndBounces	OpenData	0.233	0.878	3.51
3	OpenReport, OpenData	OpensAndBounces	0.233	0.934	3.51
4	OpenReport, OpensAndBounces	OpenData	0.233	0.878	3.51
5	OpenData	OpenReport, OpensAndBounces	0.233	0.934	3.51
6	OpensAndBounces	OpenReport, OpenData	0.233	0.878	3.51
7	OpenReport, ReportsTab, SendNow	OpensAndBounces, ManageTab	0.237	0.791	3.14
8	OpensAndBounces, ManageTab	OpenReport, ReportsTab, SendNow	0.237	0.943	3.14
9	OpensAndBounces, SendNow	OpenReport, ManageTab, ReportsTab	0.237	0.996	3.14
10	OpenReport, ManageTab, ReportsTab	OpensAndBounces, SendNow	0.237	0.748	3.14
11	AddImage, SendNow	ManageTab, UploadImage	0.243	0.835	3.14
12	ManageTab, UploadImage	AddImage, SendNow	0.243	0.914	3.14
13	OpensAndBounces, ReportsTab, SendNow	OpenReport, ManageTab	0.237	1.000	3.14
14	OpensAndBounces, SendNow	OpenReport, ManageTab	0.238	1.000	3.14
15	OpenReport, ManageTab	OpensAndBounces, SendNow	0.238	0.748	3.14
16	OpenReport, ManageTab	OpensAndBounces, ReportsTab, SendNow	0.237	0.745	3.14
17	OpensAndBounces, ManageTab	OpenReport, SendNow	0.238	0.946	3.14
18	OpenReport, SendNow	OpensAndBounces, ManageTab	0.238	0.790	3.14
19	OpensAndBounces, ManageTab, ReportsTab	OpenReport, SendNow	0.237	0.946	3.14
20	OpenReport, SendNow	OpensAndBounces, ManageTab, ReportsTab	0.237	0.787	3.14

Session-Level Tactical Rules

The session-level rules capture granular, short-term interactions. These are particularly useful for real-time feature recommendations and identifying immediate intent during a single platform visit.

Table 3: Session-Level Elite Ruleset: Interaction Sequences and Metrics

ID	Antecedents	Consequents	Supp.	Conf.	Lift
1	UploadImage, TxtBIUS	TxtFontSizeColor, AddImage	0.048	0.930	10.00
2	OpenReportList, OpenReport, OpensAndBounces	OpenData	0.048	0.904	8.93
3	OpenReportList, OpensAndBounces	OpenReport, OpenData	0.048	0.902	8.93
4	OpenReportList, OpensAndBounces	OpenData	0.048	0.903	8.92
5	OpenReportList, SendNow	OpenReport, ReportsTab, ManageTab	0.048	0.955	8.89
6	OpenReportList, ReportsTab, SendNow	OpenReport, ManageTab	0.048	0.962	8.84
7	OpenReportList, SendNow	OpenReport, ManageTab	0.048	0.960	8.81
8	OpenData, ManageTab	OpenReport, ReportsTab	0.058	0.996	5.98
9	OpenReportList, SendNow, ManageTab	OpenReport, ReportsTab	0.048	0.995	5.98
10	OpenReportList, ProjPreview, ManageTab	OpenReport, ReportsTab	0.046	0.994	5.97
11	OpensAndBounces, ManageTab	OpenReport, ReportsTab	0.049	0.993	5.97
12	OpenReportList, SendNow	OpenReport, ReportsTab	0.050	0.992	5.96
13	SEDragResize, SEDeleteElement, SEDragMove	SEDragIn	0.048	0.912	5.96
14	OpenReportList, ManageTab	OpenReport, ReportsTab	0.071	0.992	5.96
15	TxtFontSizeColor, UploadImage, CaptionedImages	AddImage	0.045	1.000	5.95
16	UploadImage, CaptionedImages	AddImage	0.064	1.000	5.95
17	TxtFontSizeColor, UploadImage, ImageProperties	AddImage	0.050	1.000	5.95
18	UploadImage, MultiImageElemen	AddImage	0.047	1.000	5.95
19	ReEditProj, CaptionedImages, UploadImage	AddImage	0.045	1.000	5.95
20	TxtFontSizeColor, UploadImage, SEDragResize	AddImage	0.060	0.999	5.94

Observation on Lift and Granularity

As observed in the tables, session-level rules yield significantly higher Lift values (up to $L = 10.00$) compared to user-level rules ($L \approx 3.14$). This highlights the difference between *tactical consistency* and *strategic habit*. While users have broad strategic patterns that repeat across months, their session-to-session behaviors are highly deterministic, particularly during specialized tasks like content editing and data auditing.

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. 2).

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