BA820 - Project M3

Team B02

Members: Akbar Wibowo, Burak Ataseven, Mika Ismayilli, and Steven Marathias

1. Integrated Problem Framing and Updated Questions

After Milestone 2, the project is centered on one main idea: audience reception of Alone may look steady on the surface, but it might actually break into recognizable patterns across episodes and seasons. The Milestone 2 work approached that idea from different angles that fit together. One analysis focused on IMDb ratings and the number of votes to see whether higher and lower rated content naturally forms groups. Another brought in viewership to help separate "highly rated" from "widely seen," since those are not always the same thing. A third looked at tap out narratives to see whether common exit themes appear across seasons, and a fourth examined survival item choices to capture recurring strategy patterns. Put together, these approaches move the project from basic summaries toward a clearer question: where do meaningful differences show up, and what signals help explain them.

With that framing in place, the team is now working on four questions, each updated through integration in Milestone 3. The core reception question, "Are IMDb ratings mostly consistent, or are there clear high and low rated episodes or seasons," has been refined, since the focus is now on grouping episodes and seasons into segments rather than only comparing averages. The viewership based segmentation question has also been refined, because adding viewership helps interpret whether a rating pattern reflects audience opinion, audience reach, or both. The tap out narrative question has been reframed as a supporting layer, used to help interpret differences seen in the reception segments instead of standing alone as its own endpoint. The item co-occurrence question has been reframed in the same way, since it provides structured context on contestant strategies that may align with broader season patterns. Finally, earlier demographic summaries have been deprioritized and kept only for context, because integration clarified that they are helpful for background understanding but do not directly support the main goal of identifying and interpreting meaningful groupings.

2. Recap of Individual M2 Contributions

Akbar: In Milestone 2, I focused on segmenting episodes using IMDb rating and vote count. I standardized the inputs and applied both KMeans and hierarchical clustering with Ward linkage, using elbow and silhouette diagnostics to choose a reasonable number of groups. This helped translate rating variation into clearer reception buckets, and I used cluster summaries and season by cluster tables to check whether certain seasons consistently fell into the same buckets. One limitation is that IMDb signals can mix perceived quality with participation, so episodes with very low vote counts may introduce noise and should be interpreted cautiously.

Burak: I extended the reception view by adding viewership and clustering in a three feature space using viewers, rating, and votes. I tested multiple cluster counts and used visualizations to evaluate separation. A consistent pattern was a cleaner top popular cluster with high viewers and strong IMDb signals, while the remaining groups overlapped more. This added an exposure versus reception framing, though viewership can dominate the clustering and compress differences among mid range episodes

Steve: I focused on tap out reason narratives using NLP. I built a text cleaning pipeline, ran text EDA, and tested topic modeling with NMF and TF IDF, with LDA explored as an alternative. I also ran embedding based clustering with sentence embeddings and KMeans, supported by silhouette style diagnostics. This work added the why layer by producing interpretable exit themes that we can later connect to performance or reception patterns. The main limitation is that short and noisy text can make themes sensitive to preprocessing choices and labeling

Mika: For my contribution for M2, I looked at whether survival item selection patterns among different contestants differ in strategies. For that, I firstly converted the loadout data into a binary basket format(1/0) and then I used the Apriori algorithm and association rule mining on the data. After tuning the thresholds, I settled on a medium support of 20% and a confidence of 70% for the most balanced signals and noise. This uncovered strong item pairings like axe, pot, and sleeping bag (76% reliability and a lift of 2.16). From that, I deduced that there is a clear divide between the necessities and strategic differentiators. The main limitation was the attempts to link item combinations directly to survival outcomes because of a small sample size.

3. Integration Strategy and Synergy Effort

Our Milestone 3 integration strategy was to turn four separate Milestone 2 notebooks into one coherent workflow that still preserves each person's angle, so we can compare approaches side-by-side. Instead of forcing everything into one model, we treated the work as connected "views" of the same show: audience reception (IMDb ratings), contestant outcomes (days lasted / demographics), tap-out narratives (text), and loadout strategy (items). The main integration work was making sure everyone's code runs off consistent, shared datasets and identifiers (season, episode, participant), and then choosing a small, representative set of outputs from each section that directly support the team's updated questions.

In terms of reuse, we kept the core pipelines from each M2 contribution. The ratings work was reused directly (standardize features to hierarchical and KMeans to elbow/silhouette checks to cluster summaries and season by cluster breakdowns). The survivalist EDA pieces were reused as supporting context (age/days distribution ns, gender breakdowns, correlations, and tap-out reasons). The narrative section was reused mainly for feasibility checks (how much text exists, duplicates, word-count distribution) so we know whether unstructured text is usable later. The loadout section reused the association-rule work (Apriori) as a separate "strategy-side" lens. What we modified was mostly integration hygiene: aligning variable names and dataframe handoffs, adding quick sanity checks (missingness, duplicates, row counts), and standardizing output formatting so the appendix doesn't look like four disconnected notebooks.

We also discarded or deprioritized pieces that didn't help comparison or were misleading in an integrated notebook. That included redundant EDA visuals, plot types that weren't interpretable given the data shape, and any methods that didn't connect clearly to the integrated questions. The biggest incompatibility we had to resolve was that different sections operate at different units of analysis (episode-level ratings vs contestant-level outcomes vs item-level choices vs text narratives), so not everything can be merged into one table without forcing bad assumptions. We briefly considered building a single unified model across all signals, but we deprioritized it for M3 because it would require heavier feature engineering and careful matching across levels; instead, we focused on a clean, comparable integration now, and saved deeper combination or "one-model" experiments for the next milestone.

4. Integrated Analysis and Results

After integrating everyone's M2 work into one notebook and rerunning everything off the same cleaned tables, the results became easier to compare and easier to explain. In M2, each analysis was useful, but it sometimes felt like we were telling separate stories with slightly different assumptions. In M3, the integration step forced consistency, so differences we see are more likely to come from the data instead of preprocessing choices. Three integrated outputs stood out most: episode segmentation using ratings and viewership (Appendix C: Episode segmentation figures/tables), exit theme buckets from tap out narratives (Appendix A: Survivalists/outcomes + narrative text figures/tables), and strategy bundles from loadout coselection (Appendix B: Loadouts/association rule figures/tables).

4.1 Integrated episode segmentation

Using IMDb rating and vote count, we still see clear reception buckets like we did in M2 (Appendix C: K-means cluster summary and representative episodes; Appendix C: elbow and silhouette diagnostics). The main improvement is consistency. We applied the same missing value rules and the same scaling across the full dataset, which makes the segmentation more stable and the cluster summaries easier to defend (Appendix C: preprocessing/missingness checks; Appendix C: cluster summary table).

When we add viewership, the structure becomes more interpretable (Appendix C: scatter plots using viewers vs rating with cluster overlays; Appendix C: sensitivity/variant clustering outputs if included). One group consistently separates as the high exposure bucket, meaning episodes with high viewership and strong engagement signals (Appendix C: viewer–rating scatter; Appendix C: cluster profile table including viewers). After that, the remaining episodes overlap more, which suggests that beyond the top tier, the boundary between mid range episode groups is naturally fuzzy (Appendix C: elbow/silhouette plots; Appendix C: viewer–rating scatter). The bigger insight is that popularity and reception are related but not identical (Appendix C: votes vs rating scatter; Appendix C: viewers vs rating scatter). Some episodes are widely watched without being uniquely high rated, and some episodes are highly rated without being the most watched (Appendix C: cluster representative episodes table; Appendix C: cross-tab by season if referenced). That matters because it supports two different business questions: what drives reach and what drives satisfaction once someone watches.

Compared to M2, the integrated version is better because ratings, votes, and viewers are treated as one shared segmentation framework instead of separate experiments (Appendix C: integrated clustering outputs). It also makes it easier to compare segment behavior across seasons using the same definitions of each bucket (Appendix C: Season × cluster distribution table and/or row-normalized percentages).

4.2 Integrated exit themes from tap out narratives

On the text side, integration made theme results more stable and easier to interpret (Appendix A: tap-out narrative preprocessing checks; Appendix A: theme outputs/visuals). Even with short narratives, the same types of exit language show up repeatedly, which supports grouping exits into a small set of buckets that can be described in plain English (Appendix A: theme summaries and example narratives; Appendix A: any topic/cluster visualization used).

A meaningful pattern is that exit themes line up with survival behavior (Appendix A: days lasted by tap-out reason/theme table/plot). Slow decline themes like sustained hunger, gradual health breakdown, and long run fatigue tend to appear later and align with longer survival (Appendix A: distribution of days lasted by theme; Appendix A: supporting summary table). Sudden disruption themes like injury language, fear driven exits, and fast escalating risk tend to show up earlier and align with shorter survival (Appendix A: days lasted by theme; Appendix A: supporting summary table). We are not claiming causality, but it still matters because it turns narrative text into a practical signal. It suggests there are different exit modes that could be used for training and for how safety teams think about monitoring risk over time (Appendix A: examples of narratives per bucket/theme).

Compared to M2, the integrated version is better because themes are presented alongside outcomes, not in isolation (Appendix A: combined theme + survival duration outputs). We also reduced noise by standardizing the same cleaning decisions before topic modeling and embedding based clustering (Appendix A: narrative cleaning/word-count diagnostics; Appendix A: model outputs). The main limitation stays the same: short text and theme labeling can be sensitive to preprocessing choices, so deeper diagnostics belong in the appendix (Appendix A: narrative word count distribution; Appendix A: duplicate/missing narrative checks).

4.3 Integrated loadout strategy bundles

Association rules show contestants do not choose items independently (Appendix B: association rule outputs and visualizations). There is a common backbone of frequently selected items, plus repeating combinations that function like strategy packages, such as trapping oriented, fishing focused, or hunting oriented setups (Appendix B: most frequent items table/bar chart; Appendix B: association-rule metric plots and/or top rules table). This is useful because it gives us a way to talk about preparation styles instead of treating the loadout as ten separate decisions.

Compared to M2, integration makes this more useful because loadouts are aligned cleanly at the contestant level and can be connected to outcomes in the same workflow (Appendix B: merged loadout + outcomes sanity checks; Appendix A/B: any combined table). That means bundles can be used to generate hypotheses, like whether certain strategy packages tend to correlate with longer survival, and whether those relationships change depending on the season environment (Appendix B: candidate bundle examples; Appendix A/B: any exploratory linkage to survival duration if included). We still have to be careful, because association rules reflect co occurrence rather than causality and the "best" loadout depends heavily on location and conditions, but integration gives us a stronger base for those comparisons (Appendix B: rule interpretation visuals/metrics; Appendix B: limitations/coverage notes if shown).

4.4 Why this matters for our question

Our question is whether distinct patterns exist that can be grouped into clear buckets people can actually use. After integration, we can point to buckets at three levels (Appendix A–C overview). At the episode level, buckets separate popularity from reception, which supports content and performance decisions (Appendix C: cluster summaries, season-by-cluster tables, and key scatter plots). At the narrative level, buckets separate slow decline exits from sudden disruption exits, which supports training and safety planning (Appendix A: theme outputs and survival-duration comparisons). At the preparation level, bundles summarize loadout strategy into repeatable packages, which can support onboarding and guidance for future contestants (Appendix B: association rules and frequent-item patterns).

5. Insights Gained Through Integration

The biggest thing we learned is that consistency matters just as much as the model choice. In M2, each analysis looked reasonable on its own, but once we integrated everything into one workflow, we saw how easy it is for small preprocessing differences to create different stories. Standardizing how we handled missing values, scaling, and joins made the outputs more stable and made our interpretations easier to defend (see Appendix: integration sanity checks and shared preprocessing diagnostics).

Integration also helped us see where our work was overlapping. The IMDb based clustering and the viewership plus IMDb clustering were not really separate projects, they were two angles on the same segmentation question. Seeing them side by side helped us treat them as complementary, one lens focused more on reception and engagement, the other focused more on exposure and reach. That made the final narrative cleaner and helped us avoid repeating ourselves (see Appendix C: episode segmentation outputs and viewers versus rating sensitivity figures).

We also learned more about tradeoffs that were not obvious in M2. The structured clustering is relatively stable and easy to summarize, but it does not explain why contestants leave. The text based exit themes give us that why layer, but they are more sensitive to cleaning choices and labeling. Integration made it clear that interpretability is not just about picking an interpretable model, it is also about choosing representations that hold up when we rerun things consistently (see Appendix C: clustering diagnostics and cluster summaries; Appendix A: narrative availability and word count diagnostics, plus exit theme outputs).

Finally, integration forced us to be disciplined about the unit of analysis. Episodes live at the episode level, while loadouts and narratives live at the contestant level, and mixing those without strict joins can lead to misleading comparisons. If we were starting over, we would lock the integrated dataset structure and definitions first, then run the models, and we would plan earlier for stability checks on the text themes since preprocessing sensitivity turned out to be one of the biggest risks.

6. Limitations, Open Questions, and Next Steps

Before the final milestone, our main job is to tighten the integrated work into one clean, defensible story. That starts with locking the integrated dataset structure and being strict about units of analysis, episodes stay episode level, while loadouts and tap out narratives stay contestant level. From there, we need to finalize the "bucket" definitions we plan to report, episode segments, exit theme buckets, and loadout strategy bundles, and make sure we summarize each one in plain language so the final deliverable reads like one connected framework instead of separate sections. We also need to add brief stability checks using what we already ran, mainly to show that our clusters and themes do not change dramatically under small, reasonable preprocessing choices.

Integration also made several limitations clearer. The biggest is alignment risk, mixing episode level outcomes with contestant level features can create misleading comparisons if joins are not handled carefully. We also rely on proxies. IMDb votes reflect engagement and who chooses to rate, not just quality, and viewership can dominate clustering even when it is not the main thing we want to interpret. On the unstructured side, tap out narratives are short and inconsistent, so exit themes are inherently sensitive to cleaning and labeling. Finally, association rules on loadouts are descriptive, they capture co occurrence, not causality, and item effectiveness likely depends on season conditions and contestant skill, which we cannot fully control for.

A few questions remain unresolved or newly emerged after integration. We still need to understand how stable our buckets are across seasons, since environments and casting change over time. It is also unclear whether exit modes are actually tied to reception and engagement, or whether those links are driven more by editing and storytelling than what happened in the field. Another open question is whether loadouts reflect real strategic differences or mostly shared constraints, meaning the signal may come from smaller deviations rather than the core items most people bring. These open questions matter because they define where we should be cautious in the final milestone and where we should frame conclusions as hypotheses instead of hard claims.

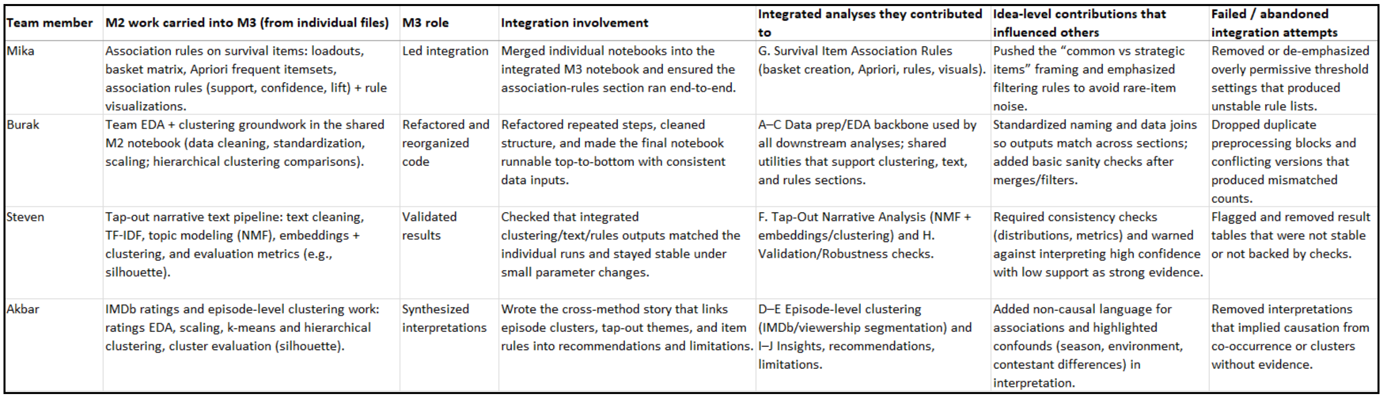
Appendix

Shared GitHub Repository (Required)

https://github.com/MikaIsmayilov/B02_Alone_TVShow_Unsupervised_Project/tree/main

All the M3 files are in the following folder: M3_Integrated&Synergized_Analysis

Contribution Table

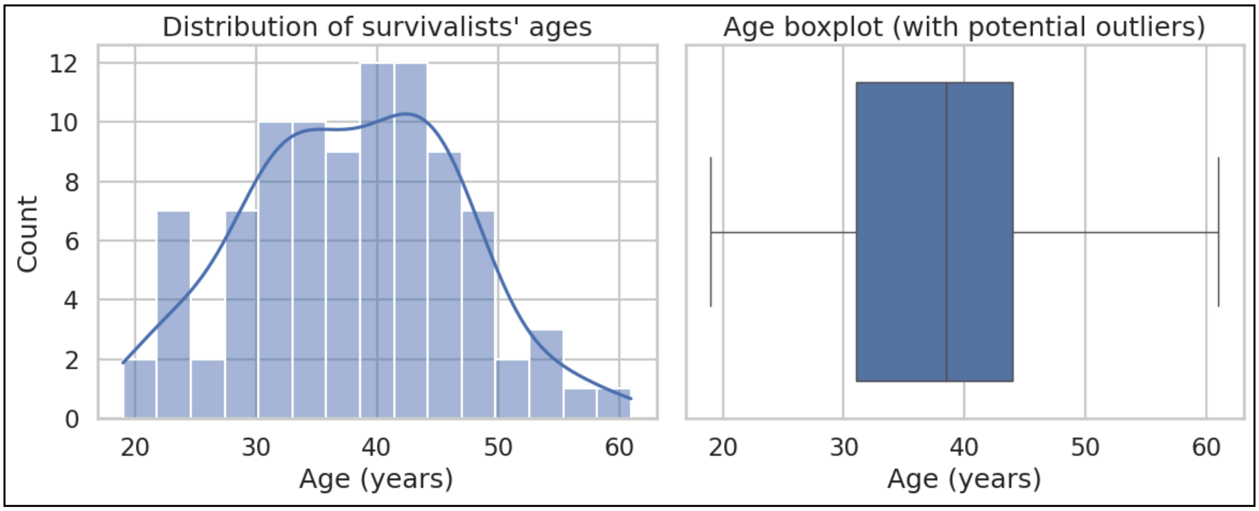


Team member	M2 work carried into M3 (from individual files)	M3 role	Integration involvement	Integrated analyses they contributed to	Idea-level contributions that influenced others	Failed / abandoned integration attempts
Mika	Association rules on survival items: loadouts, basket matrix, Apriori frequent itemsets, association rules (support, confidence, lift) + rule visualizations.	Led integration	Merged individual notebooks into the integrated M3 notebook and ensured the association-rules section ran end-to-end.	G. Survival Item Association Rules (basket creation, Apriori, rules, visuals).	Pushed the “common vs strategic items” framing and emphasized filtering rules to avoid rare-item noise.	Removed or de-emphasized overly permissive threshold settings that produced unstable rule lists.
Burak	Team EDA + clustering groundwork in the shared M2 notebook (data cleaning, standardization, scaling; hierarchical clustering comparisons).	Refactored and reorganized code	Refactored repeated steps, cleaned structure, and made the final notebook runnable top-to-bottom with consistent data inputs.	A–C Data prep/EDA backbone used by all downstream analyses; shared utilities that support clustering, text, and rules sections.	Standardized naming and data joins so outputs match across sections; added basic sanity checks after merges/filters.	Dropped duplicate preprocessing blocks and conflicting versions that produced mismatched counts.
Steven	Tap-out narrative text pipeline: text cleaning, TF-IDF, topic modeling (NMF), embeddings + clustering, and evaluation metrics (e.g., silhouette).	Validated results	Checked that integrated clustering/text/rules outputs matched the individual runs and stayed stable under small parameter changes.	F. Tap-Out Narrative Analysis (NMF + embeddings/clustering) and H. Validation/Robustness checks.	Required consistency checks (distributions, metrics) and warned against interpreting high confidence with low support as strong evidence.	Flagged and removed result tables that were not stable or not backed by checks.
Akbar	IMDb ratings and episode-level clustering work: ratings EDA, scaling, k-means and hierarchical clustering, cluster evaluation (silhouette).	Synthesized interpretations	Wrote the cross-method story that links episode clusters, tap-out themes, and item rules into recommendations and limitations.	D–E Episode-level clustering (IMDb/viewership segmentation) and I–J Insights, recommendations, limitations.	Added non-causal language for associations and highlighted confounds (season, environment, contestant differences) in interpretation.	Removed interpretations that implied causation from co-occurrence or clusters without evidence.

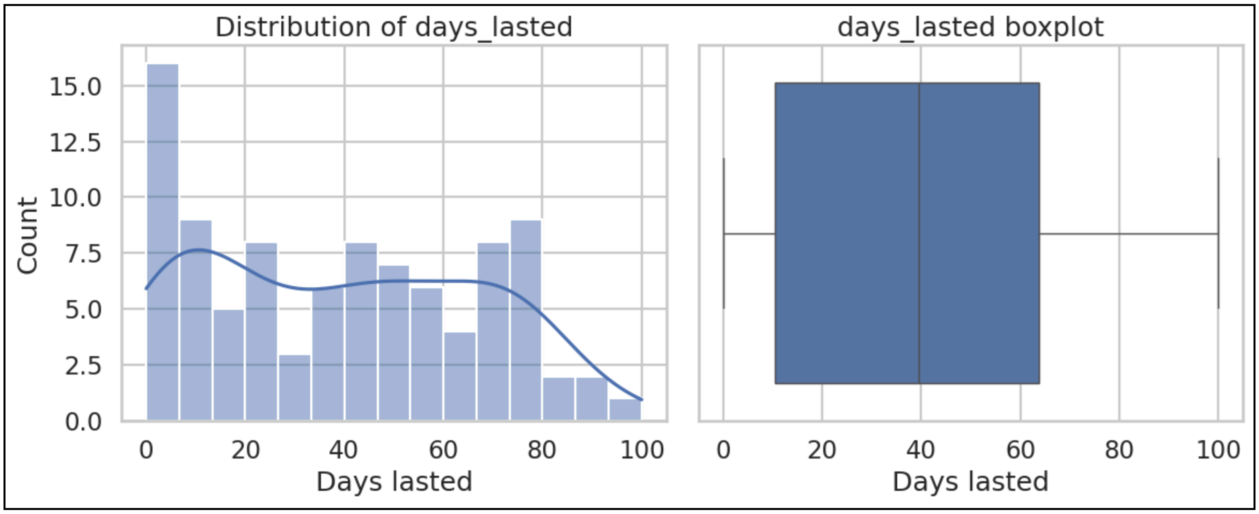
Supplemental Material

A. Survivalists / outcomes EDA (contestant side)

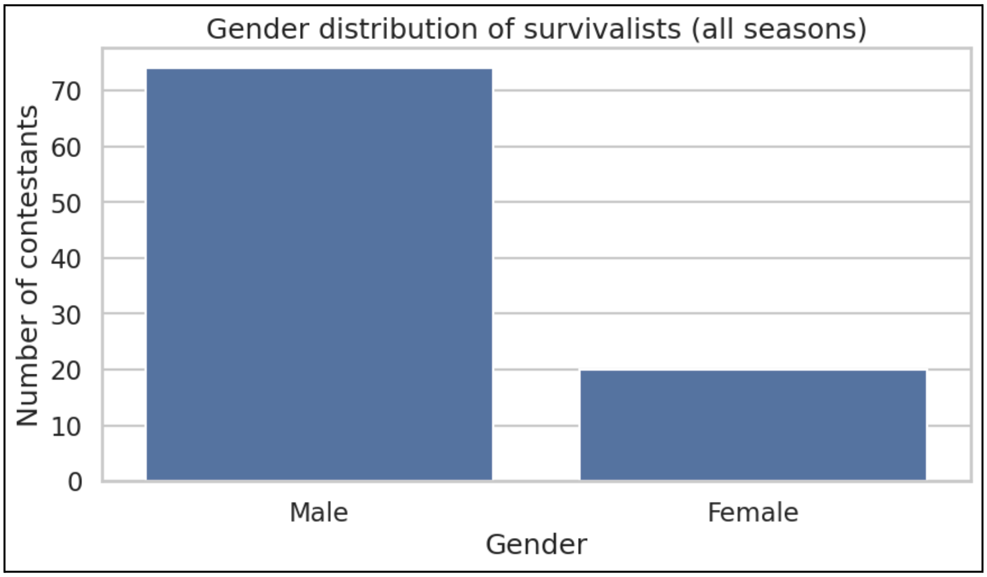
A.1 Participants’ ages, distribution and outliers

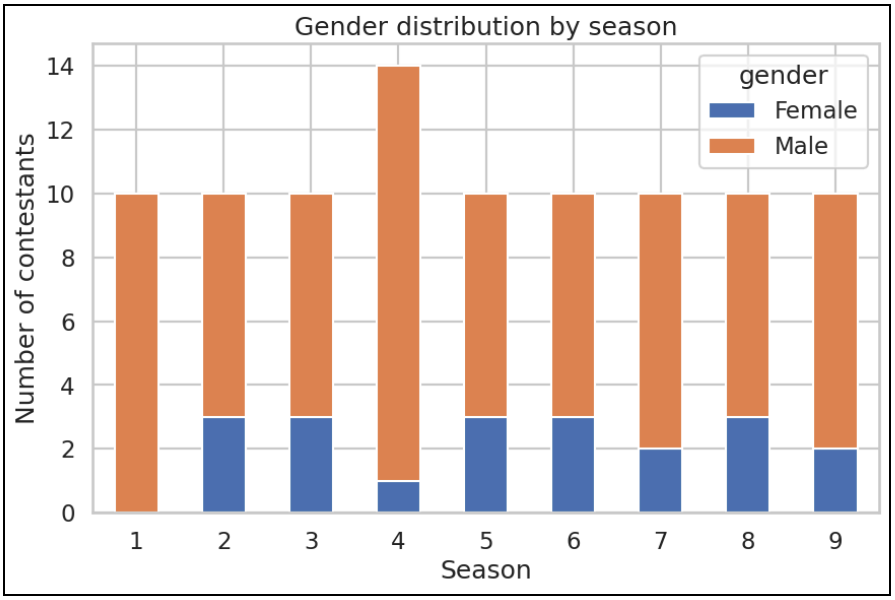


A.2 Survival duration, days lasted distribution and skewness

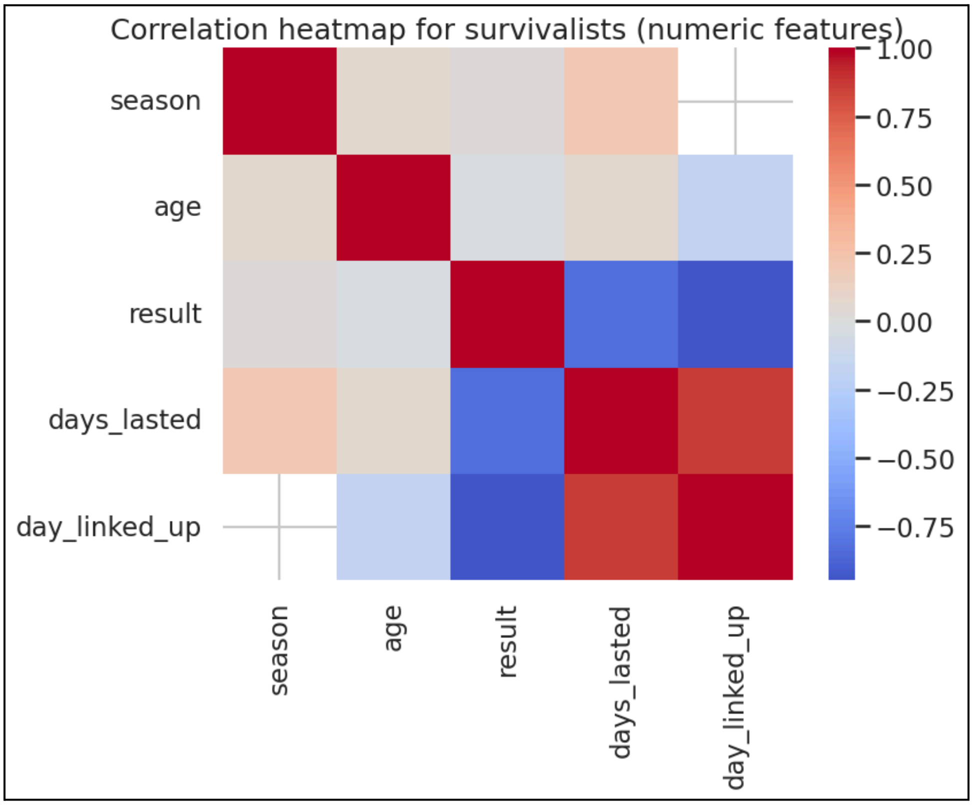


A.3 Gender distribution across all seasons

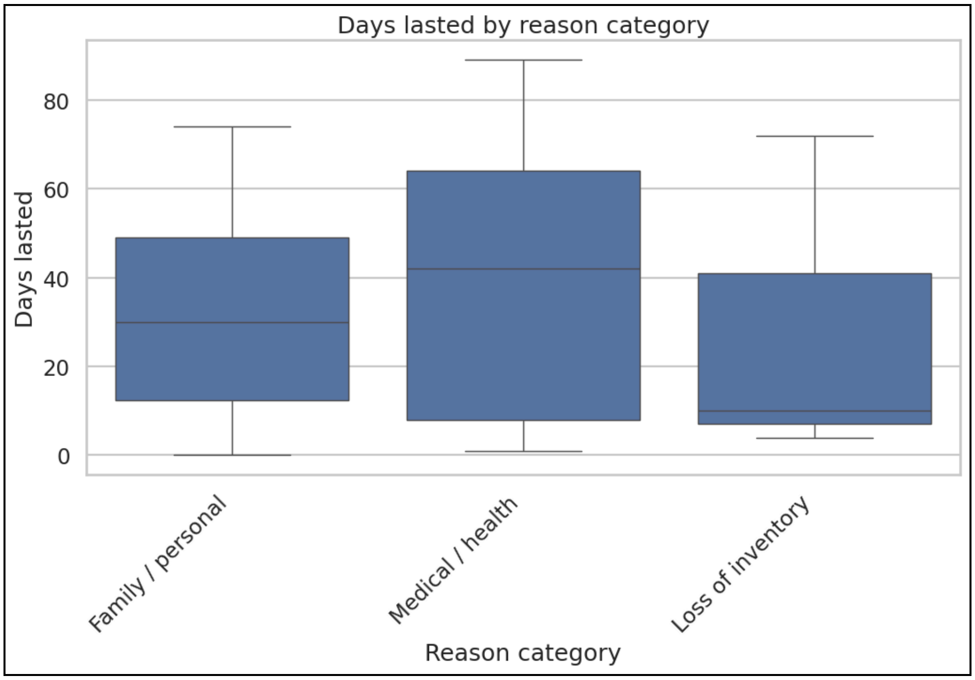




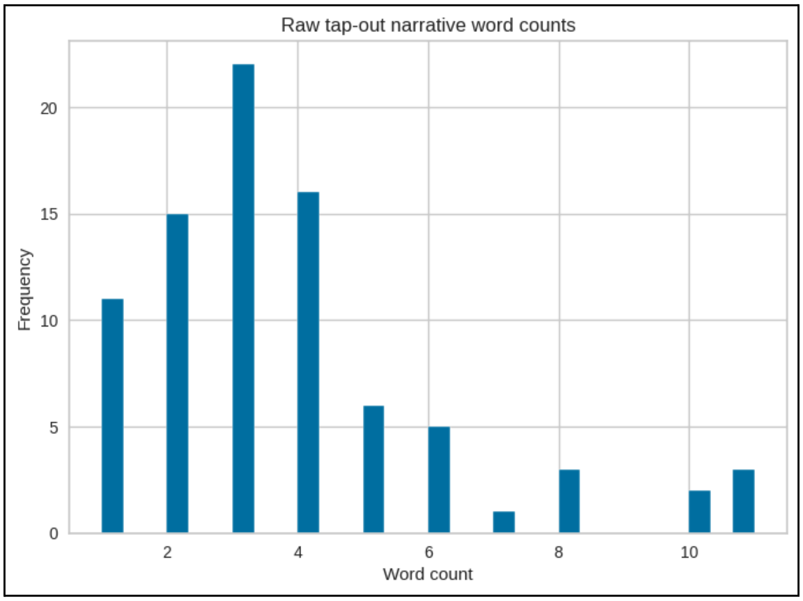
A.4 Correlation heatmap, survivalists numeric variables



A.5 Days lasted by tap out reason category

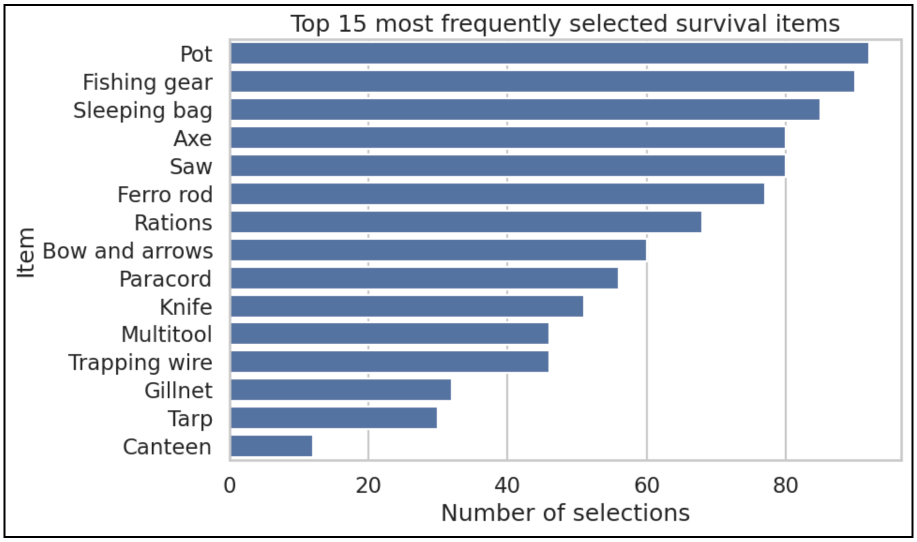


A.6 Tap out narrative word count distribution

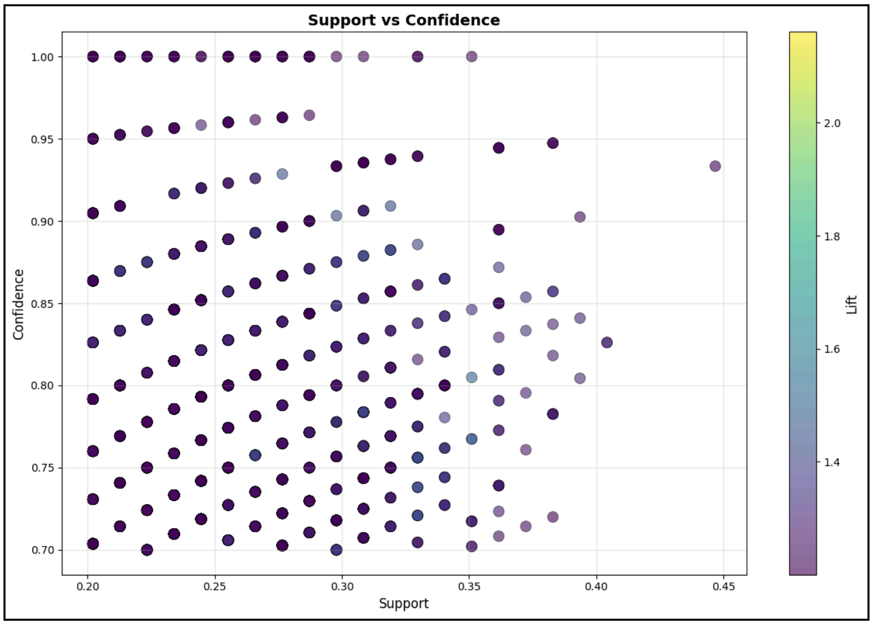


B. Loadouts / items EDA (item strategy side)

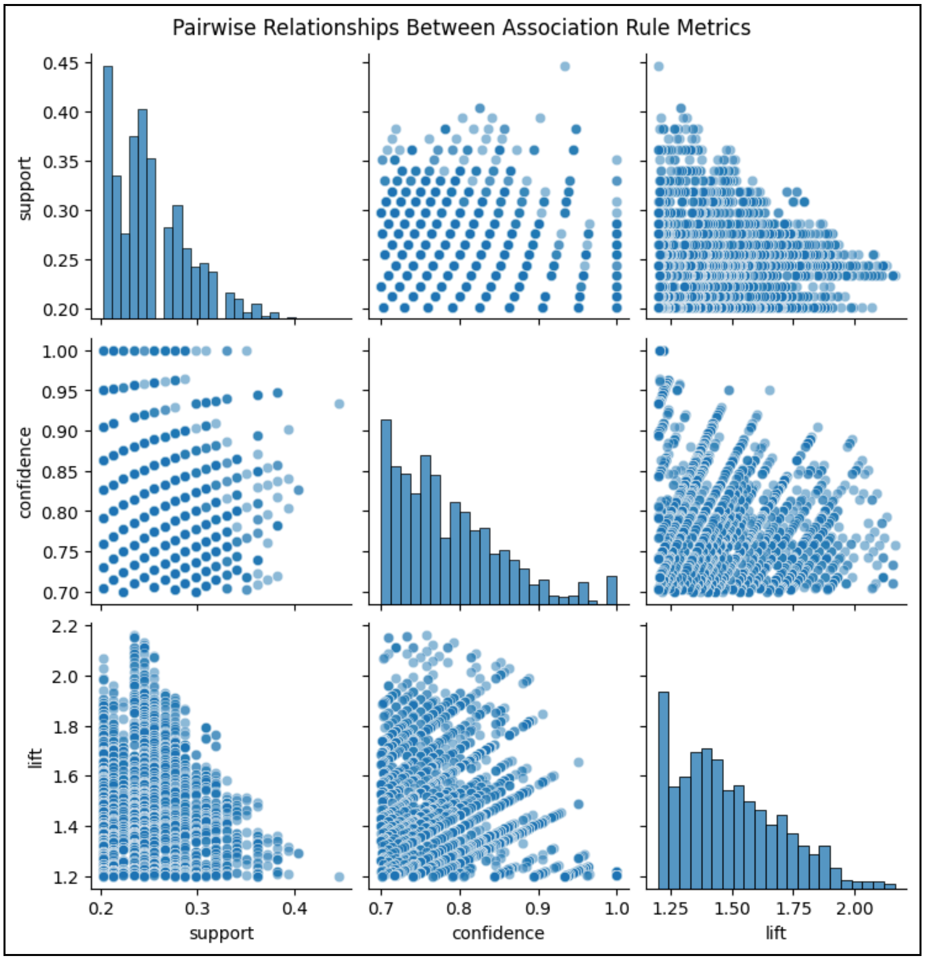
B.1 Most frequently selected survival items



B.2 Support vs Confidence (colored by Lift)

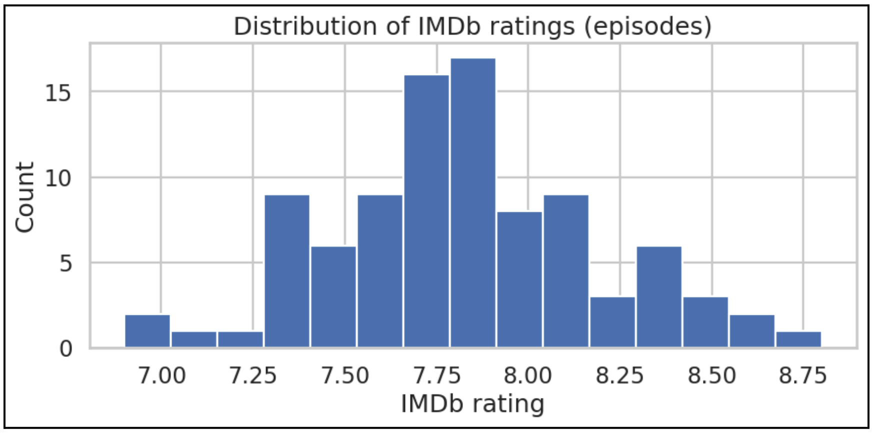


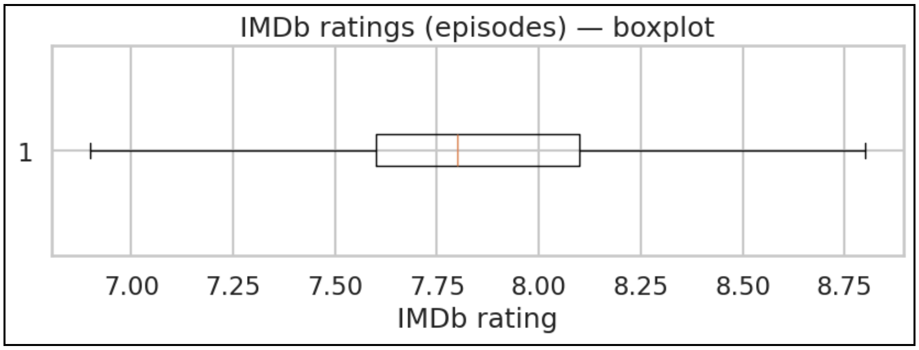
B.3 Item frequency by outcome group



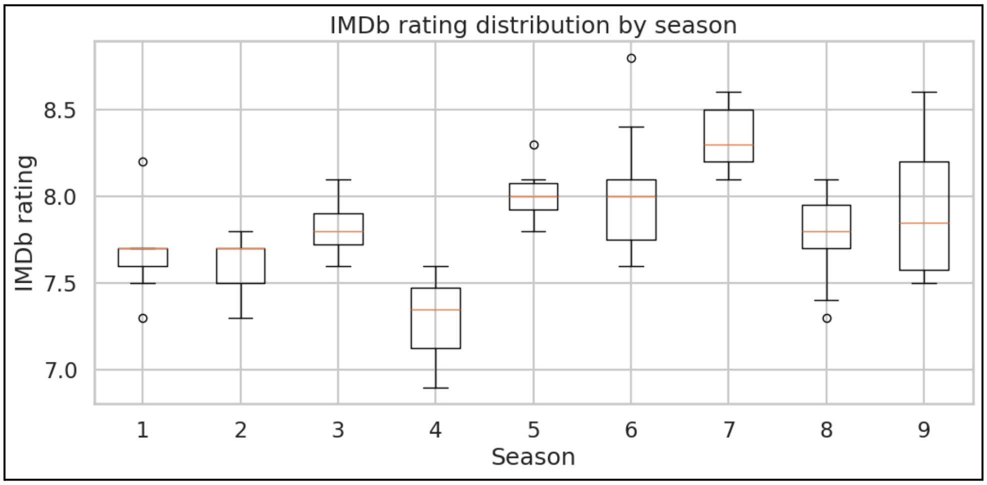
C. Episodes / Ratings (audience reception side)

C.1 Episode rating distribution (hist + boxplot)

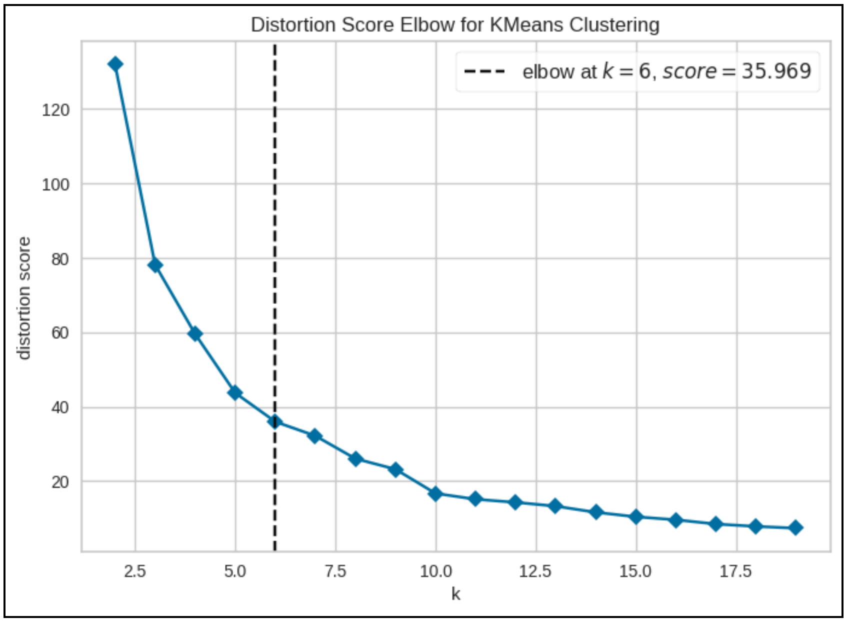




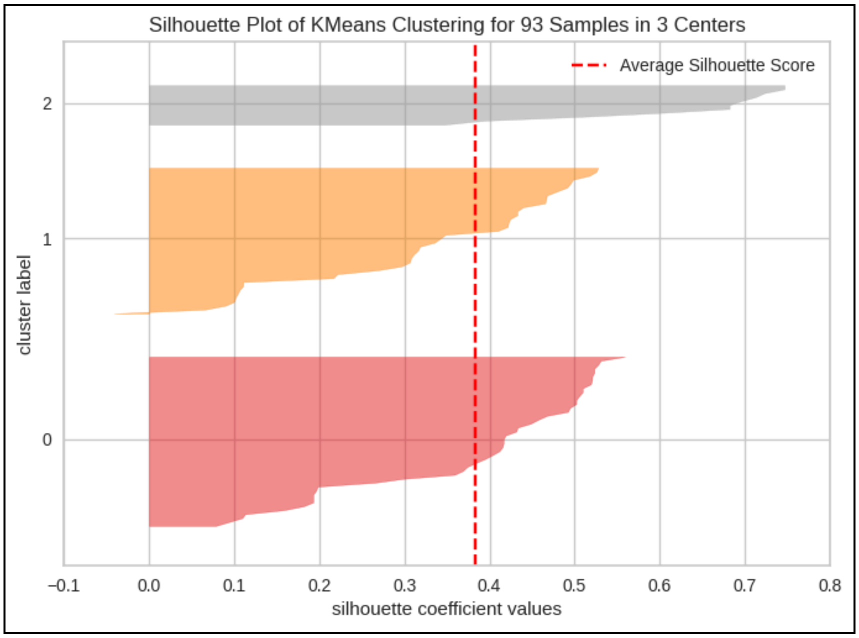
C.2 Ratings by season (season boxplot)



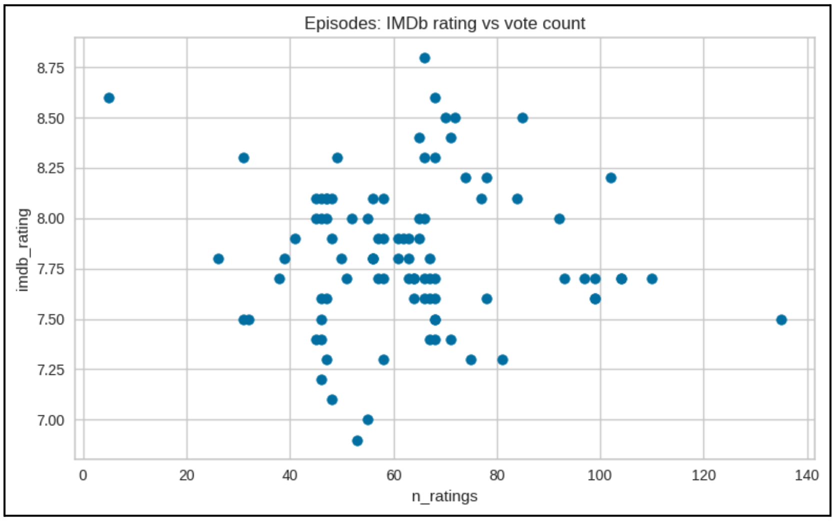
C.3 K-means elbow plot



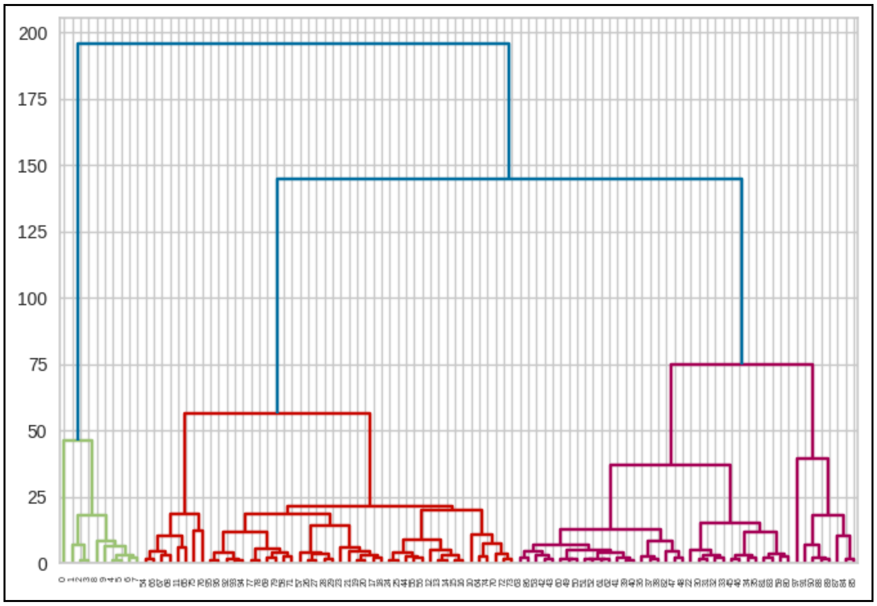
C.4 K-means silhouette plot (chosen k)

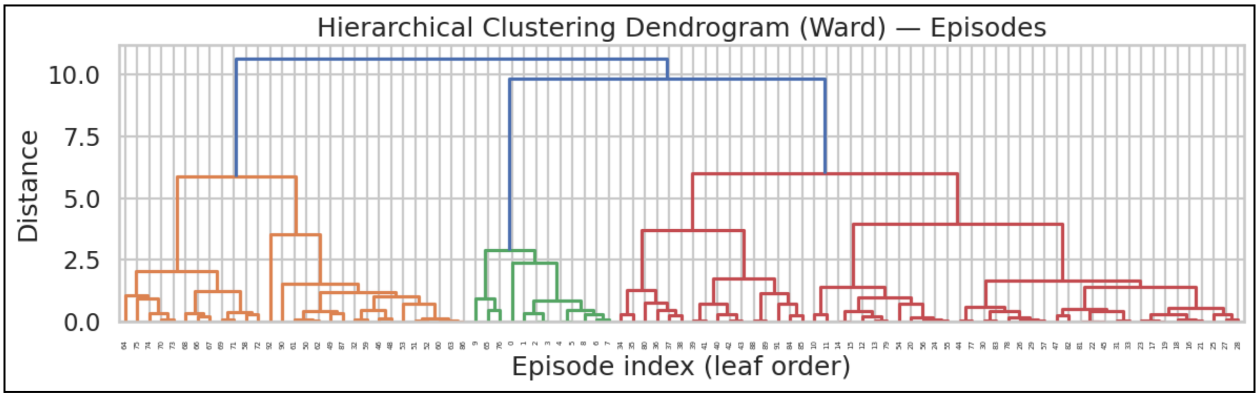


C.5 Scatter: votes vs rating (context for reliability)

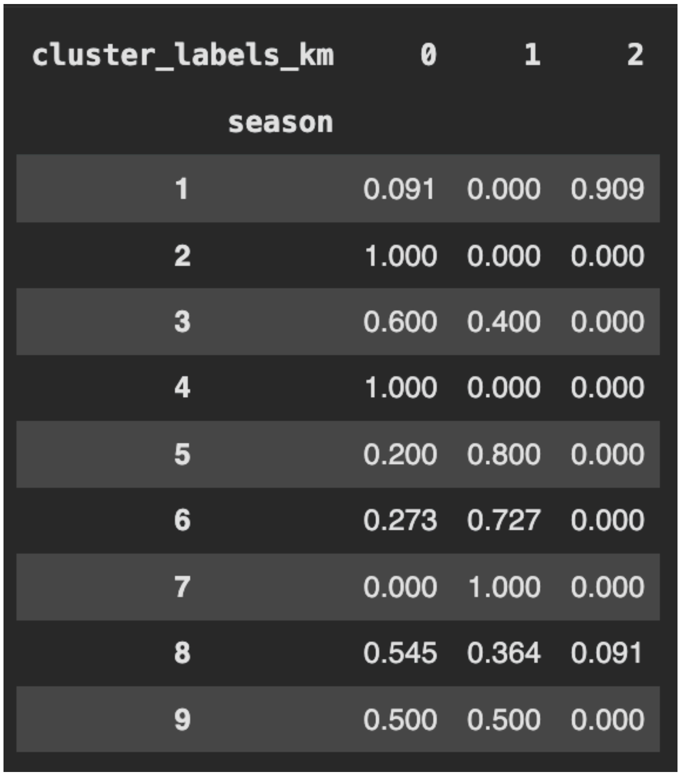


C.6 Dendrogram (episodes and seasons)



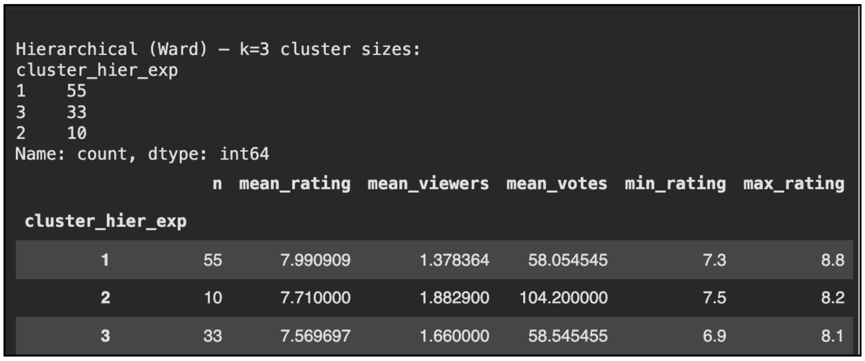


C.7 Season × cluster distribution (row-normalized table)



cluster_labels_km	0	1	2
season			
1	0.091	0.000	0.909
2	1.000	0.000	0.000
3	0.600	0.400	0.000
4	1.000	0.000	0.000
5	0.200	0.800	0.000
6	0.273	0.727	0.000
7	0.000	1.000	0.000
8	0.545	0.364	0.091
9	0.500	0.500	0.000

C.8 Sensitivity check (ONLY ONE): either k=3 scatter on expanded features or a small table summarizing k=3/4/5 cluster profiles



```
Hierarchical (Ward) — k=3 cluster sizes:
cluster_hier_exp
1    55
3    33
2    10
Name: count, dtype: int64
```

	n	mean_rating	mean_viewers	mean_votes	min_rating	max_rating
cluster_hier_exp						
1	55	7.990909	1.378364	58.054545	7.3	8.8
2	10	7.710000	1.882900	104.200000	7.5	8.2
3	33	7.569697	1.660000	58.545455	6.9	8.1

Process Overview

1. Data Review

We reviewed the four core tables (seasons, episodes, survivalists, and loadouts) and confirmed what each table represents and how they connect through shared keys.

2. Cleaning and Standardization

We fixed column names and data types, handled missing values, removed duplicates where needed, and standardized item names so the same gear did not appear under multiple labels.

3. Data Integration

We merged tables to create two analysis views: an episode and season view for ratings and episode features, and a survivalist and loadout view for item choice patterns.

4. Exploratory Data Analysis

We summarized key distributions and trends, checked relationships with simple tables and plots, and used EDA to spot outliers and confirm that joins worked as expected.

5. Clustering Analysis

We created numeric features for episode-level analysis, scaled features when needed, ran clustering to find groups with similar patterns, and compared methods to see if clusters stayed consistent.

6. Text Narrative Analysis

We cleaned the tap-out narratives, extracted themes with topic modeling, and used embeddings with clustering to test whether the narratives form clear groups.

7. Association Rule Mining

We converted loadouts into a 0/1 basket matrix, ran Apriori to find frequent itemsets, generated association rules, and interpreted results using support, confidence, and lift.

8. Validation and Comparison

We checked whether findings stayed similar under small parameter changes, reviewed outputs for internal consistency across sections, and kept interpretations tied to what each method can support.

9. Synthesis and Takeaways

We combined results into one story, highlighted the main takeaways for contestant strategy and show design, and ended with limits and clear next steps.

Use of Generative AI Tool

For this milestone of the project, we had limited usage of GenAI due to the nature of this step. All the code seen has been directly copied from our previous stages(individual M2 files). Thus, in this step, GenAI was used to brainstorm a strategy in the beginning as well as fixing part of the code that would envelop the output in errors.

M3 Roadmap Brainstorming:
https://chatgpt.com/share/699518c3-c5b4-8002-92cd-0a69b723104f

Fix for Error Flood For Mika’s Part of Code:
https://chatgpt.com/share/69953332-299c-8002-b36c-9735a07c215f