BA820 – Project M4

Team B02

Steven Marathias

1. Refined Problem Statement & Focus (~0.5 page)

For Milestone 4, I am still focused on the same core question from earlier milestones: What patterns exist in contestants' tap out narratives, and do distinct exit themes show up that can be grouped into clear buckets? I stuck with this because the tap out text is one of the only places where we get the "why" behind leaving, and if those reasons fall into repeatable themes, we can turn them into a practical exit taxonomy that makes sense for producers, safety teams, and future contestants.

The question itself did not change, but my focus changed. In M2, the goal was proving that themes exist at all. In M4, the goal is making those buckets more stable and easier to defend, not just interesting. Instead of running one method and calling it a day, I treated my earlier approach as a baseline and ran controlled comparisons so I could explain what improves and why. I also leaned more into meaning based grouping because my real end goal is not keyword clusters, it is exit themes that still hold up even when people describe the same situation using different wording.

A few assumptions got tested in M4. First, I expected the narratives to be noisy, but I also expected some structure, and that assumption was validated. The themes that come out are not random, they keep pointing back to a small set of real exit categories. Second, I assumed a bag of words style approach would be enough to form clean buckets. That was partially challenged. It can work, but it depends a lot on exact phrasing, which is not ideal for theme discovery. Third, I assumed missing narratives would not matter much. That turned out to be wrong. Leaving empty reasons in the pipeline created its own cluster, which forced me to handle missing text explicitly before trusting the final buckets.

2. EDA & Preprocessing: Updates (~0.5 page)

From the earlier milestones, the most important setup work was isolating the tap-out narrative field and doing basic checks on the text itself. The dataset is small and the narratives are short, so quick EDA like how many narratives are missing and how long they are matters more than it normally would. A small amount of noise can change the outputs a lot when you only have around a hundred narratives.

In M4, the main preprocessing update that actually changed the results was handling empty narratives during the embeddings experiment. When I clustered embeddings using all narratives, one cluster ended up being made entirely of empty text. That is not a real theme, it is just missing data showing up as structure. So for the final embedding based buckets, I filtered out empty narratives before embedding and clustering. I kept the rest of preprocessing consistent on

purpose, because the point of M4 was not to reinvent cleaning. It was to run fair comparisons and show whether representation and method changes actually improve the theme buckets.

3. Analysis & Experiments (~2 pages)

Overall approach

My work follows the basic text-mining workflow from class. I start with narrative text, convert it into a numeric representation, and then apply unsupervised methods to find structure. In M4, I ran two main experiments. Experiment 1 is a controlled TF-IDF refinement meant to improve interpretability. Experiment 2 is the method upgrade, using embeddings to group narratives by meaning rather than exact wording. Throughout, the goal stayed the same: build exit buckets that are simple enough to name and consistent enough to defend.

Experiment 1: TF-IDF 1-gram vs TF-IDF 1–2 grams with NMF

What this is helping answer
This experiment tests whether exit themes show up more clearly when I allow short phrases instead of only single words. Tap out narratives are often written in short chunks like “missed family,” “food poisoning,” or “low BMI.” If I only use 1 gram, those concepts can get split up and the topics become harder to label.

Why it makes sense for this data:
TF-IDF is a strong baseline for short text because it highlights terms that differentiate narratives. NMF is also helpful because it tends to produce topics that are easier to interpret than other approaches, which matters a lot when the end goal is usable buckets.

What I tried
I ran the same NMF setup twice with the same number of topics:

- TF IDF with 1 grams only as the baseline

- TF IDF with 1–2 grams as the refinement
  Then I compared top terms per topic, topic sizes, and representative examples. I also looked at days lasted by topic, including a boxplot for the 1–2 gram version, because one way to sanity check themes is seeing whether buckets line up with different outcome patterns.

What worked and what did not:

The 1–2 gram version was easier to interpret because it captured phrase level meaning that maps directly to real labels. The biggest limitation is that TF IDF still depends on shared wording, so some buckets can overlap when contestants describe the same idea using different words. Overall, the refinement helped, but it also made it clear that keyword based topics are not always the cleanest way to get “themes.”

What surprised me:

The improvement from adding bigrams was bigger than I expected. It did not magically fix everything, but it made the topics easier to name and made the examples fit better, which is exactly what matters for this project.

Experiment 2: Embeddings-based exit theme buckets with clustering

What this is helping answer
This experiment is aimed at the main weakness of TF IDF. TF IDF is good for keyword patterns, but themes are really about meaning. Embeddings are designed to represent semantic similarity, so this experiment tests whether I can group exits into clearer buckets by meaning instead of exact phrasing.

Why it makes sense for this data
For short narratives, it is common for the same theme to show up with different wording. Embeddings are a natural upgrade when the goal is theme discovery. If the output buckets are going to be useful, they need to stay coherent even when people phrase things differently.

What I tried
I generated sentence embeddings for the narratives and clustered them using KMeans. I printed representative examples per cluster by selecting narratives closest to each cluster centroid. I then plotted days lasted by cluster and created a final wrap up table with theme labels, counts, percent of exits, average days lasted, and example narratives.

The biggest adjustment happened after the first run. One cluster was made entirely of empty narratives. That was not a real exit theme, it was missing data. So I filtered out empty narratives and reran the embeddings and clustering. After that, the clusters became much more interpretable. I also ran a stability check by changing the clustering seed and verifying that the cluster meanings stayed roughly similar.

What worked and what did not
The embedding clusters were noticeably easier to label than the TF IDF topics because the

groupings were more semantically consistent. After filtering empty narratives, the clusters lined up with clear exit categories like injury or exhaustion, starvation, illness, missing family, and low BMI or weight loss. The main “did not work” moment was the blank narrative cluster, but that actually helped the refinement story because it showed why missing text needs to be handled explicitly.

What surprised me
The embedding buckets felt closer to a real exit taxonomy with less manual translation than TF IDF topics. The representative examples also matched the cluster labels more naturally, which makes the results easier to explain in non-technical language.

4. Findings & Interpretations (~1 page)

The main result is that tap-out narratives consistently fall into a small set of repeatable exit themes. Even with short text, the buckets that come out are not random, and they match real-world categories that make sense.

Experiment 1 showed that adding 1–2 grams improves interpretability because a lot of exit reasons are naturally expressed as short phrases. That makes it easier to label topics and makes the “theme bucket” idea more realistic. At the same time, TF-IDF still leans on exact wording, which creates overlap when different contestants describe the same situation in different ways.

The clearest and most usable buckets came from Experiment 2. After filtering out empty narratives, embedding clustering produced clusters that were easy to name and supported by representative examples. These buckets matter because they separate different exit modes that have different real-world implications. Injury or exhaustion exits point toward safety monitoring and physical risk. Starvation and low BMI exits point toward longer-term physical decline and resource issues. Illness or food poisoning exits point toward health events that can escalate quickly. Missing family exits point toward non-survival constraints that still impact outcomes and might matter for preparation and mental readiness.

The days lasted plots help support the interpretation. Themes like low BMI or weight loss tend to look more like later-stage decline patterns, while some acute categories like injury can happen earlier. This is not a causal claim, but it is a useful validation that the buckets capture different patterns of experience, not just random text groupings.

Overall, M4 strengthened the original framing. The themes are real, and the embedding approach makes the buckets more coherent and easier to defend, which is the whole point of this milestone.

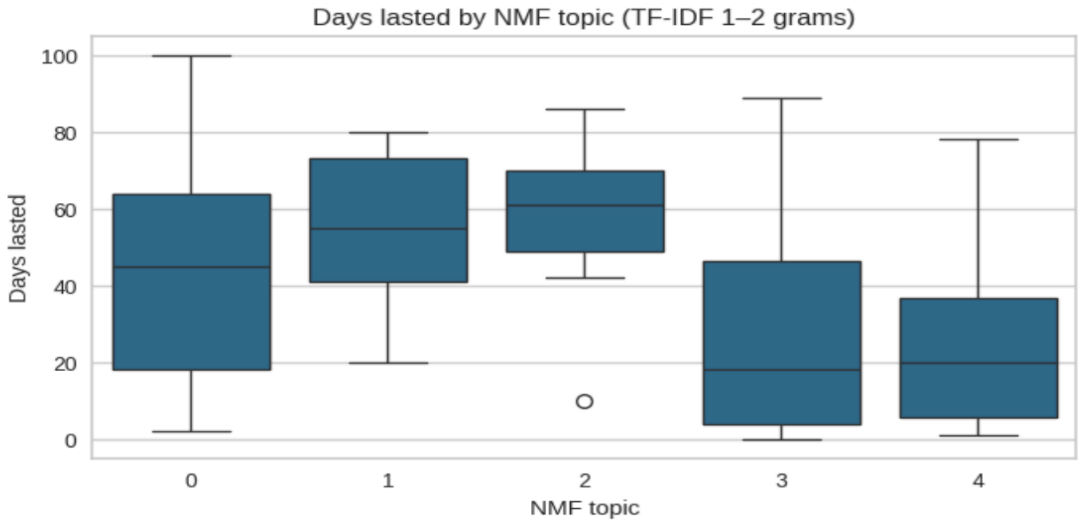
Shared GitHub Repository

Team repo: https://github.com/MikaIsmayilov/B02_BA820_Project

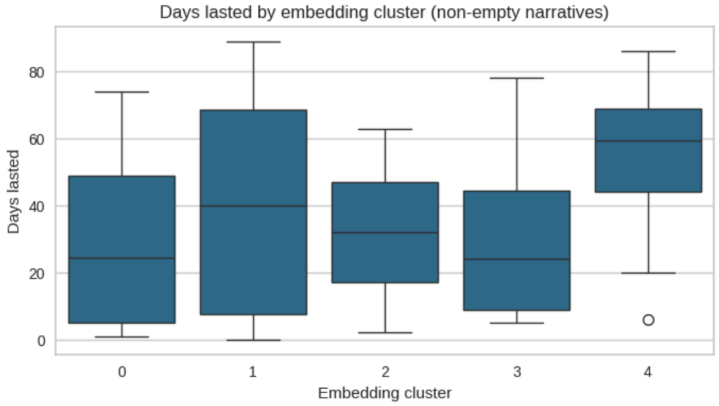
Project Proposal Team B02 BA820

Supplemental Material:

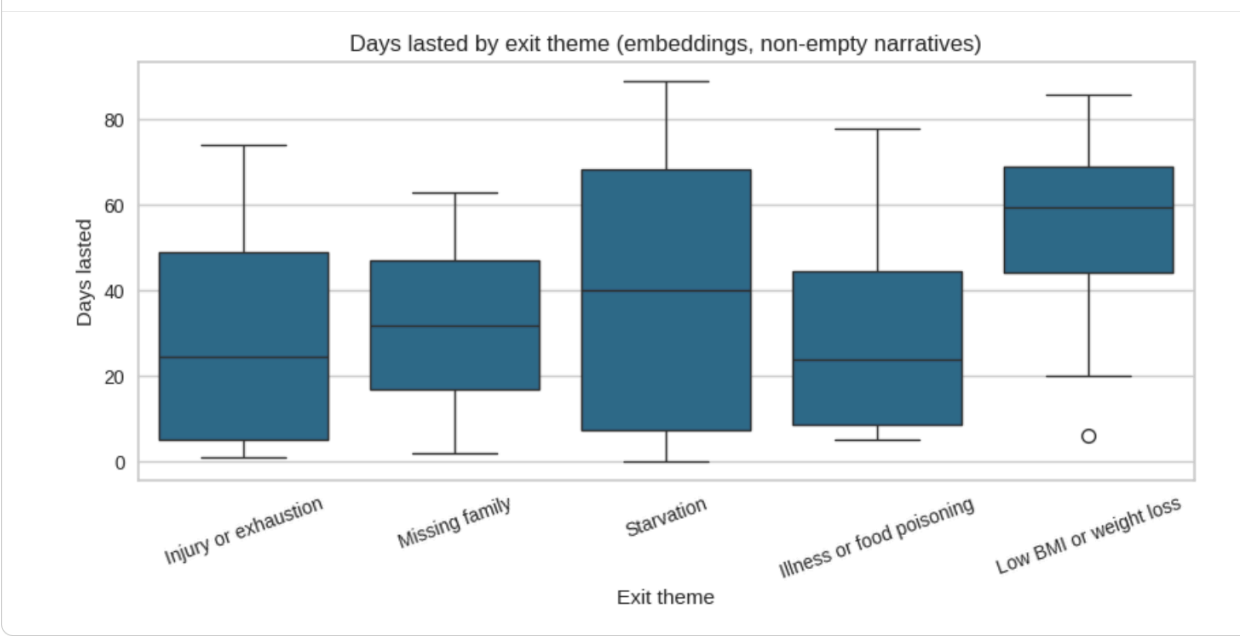
```
len(docs): 94 | len(topic_assign_12g): 94
Topic 0 examples:
-
- Realized he should actually be around for his mother's cancer
- Consuming salt water
Topic 1 examples:
- Starvation
- Starvation
- Starvation
Topic 2 examples:
- Lost the mind game
- BMI too low
- Systolic pressure too low
Topic 3 examples:
- Fear of storm
- Fear of bears
- Loss of ferro rod
Topic 4 examples:
- Felt content with what he had done
- Hunger and mental breakdown
- Fell off kayak into river
```



```
Cluster 0 representative examples:
- Fell, injured back and knee
- Broken ankle
- Felt too exhausted and drained to go on
Cluster 1 representative examples:
- Starvation
- Starvation
- Starvation
Cluster 2 representative examples:
- Missed his family
- Missed his family
- Missed his family
Cluster 3 representative examples:
- Food poisoning
- Food poisoning
- Food poisoning, infection
Cluster 4 representative examples:
- Low BMI, lost too much weight
- Low BMI, lost too much weight
- Low BMI, lost too much weight
```



	theme	n	avg_days_lasted	pct	example_reasons
1	Injury or exhaustion	28	29.32	33.3	[Lost the mind game, Felt content with what he...
4	Starvation	19	38.95	22.6	[Fear of storm, Fear of bears]
3	Missing family	15	32.33	17.9	[Realized he should actually be around for his...
0	Illness or food poisoning	12	28.42	14.3	[Consuming salt water, 2 broken teeth (molars)...
2	Low BMI or weight loss	10	53.50	11.9	[BMI too low, Systolic pressure too low]



Process Overview

1. Load survivalists data → select tap-out narrative field
2. Clean and standardize narratives → drop empty reasons for embedding run
3. Experiment 1: TF-IDF 1-gram vs TF-IDF 1–2 grams → NMF topics → compare top terms, topic sizes, and example narratives → days lasted by topic plot
4. Experiment 2: Sentence embeddings → KMeans clustering → representative examples per cluster → filter empty narrative cluster and rerun → stability check with different seed → days lasted by cluster plot
5. Finalize exit theme buckets → export summary table with theme name, percent, average days lasted, and example reasons for reporting

Use of Generative AI Tools:

I used Gen AI to help debug some code that I was stuck on, it also helped me create a road map for how i wanted M4 to turn out, and i used it to help sift through some of the slides and notebooks we have gone through the last week to help me best identify what methods I should use to dive deeper into my question.

Chat GPT code debugging: https://chatgpt.com/share/699d14d2-d198-800c-b751-541e09a0b066

RoadMap: https://chatgpt.com/share/699d176e-79dc-800c-837f-2bc432104ca9

More ML concepts that could help me:
https://chatgpt.com/share/699d1a27-c1f0-800c-ab8a-e333a99087fa