BA820 – Project M2

Cover Page

- **Project Title:** Interpretable Palette Archetypes in Bob Ross Paintings via Clustering and Association Rules
- **Section and Team Number:** B1 Team 3
- **Student Name:** Irene (Yihui) Tang

1. Refined Problem Statement & Focus

Research Question: **Do Bob Ross paintings form a small number of interpretable palette archetypes based on binary paint usage?**

Each painting can be represented as a binary vector over 18 paint indicators, which naturally treats the palette as a set. The goal of this milestone is to determine whether these binary palette sets fall into a small number of recurring "archetypes," and equally important whether those archetypes can be described in plain language through stable co-usage patterns.

The question is unchanged, but the success criterion changed: instead of expecting balanced clusters, I now evaluate whether the data supports one baseline archetype plus rare but coherent variants. M1 showed extremely strong repetition in paint usage: Titanium White appears in 99% of paintings, many exact paint "recipes" repeat across episodes, and pairwise Jaccard similarity is consistently high. These findings suggested that the dataset is unlikely to contain several balanced stylistic groups. Instead, the evidence pointed toward one dominant baseline palette and a small number of rare but structured variants.

Based on this, M2 shifts from the broad idea of "discovering structure" to the more specific task of identifying interpretable palette archetypes. The emphasis is no longer on finding balanced clusters but on understanding how a dominant palette coexists with a few specialized deviations.

Several M1 assumptions were tested in M2:

- **Binary palette representation is appropriate.** The 18 paint indicators capture palette choice well and preserve the repeated structure observed in M1. No additional feature engineering was needed.
- **The dataset contains real structure, not noise.** Clustering at k=5 produced one dominant cluster (n=385, 95.5%) and four small clusters (n=10, 5, 2, 1). This imbalance is consistent with M1's evidence of a shared baseline palette rather than a coding artifact.
- **Not all clusters can be interpreted the same way.** Very small clusters (n < 10) cannot support stable association-rule mining. Their interpretation must rely on simpler paint-frequency profiles rather than rule-based evidence.
- **Archetypes are likely "baseline vs. rare signatures," not multiple equally sized groups.** This was the most important update from M1: the structure of the dataset is defined by repetition and consistency, with only occasional departures.

The main challenge is determining whether the very small clusters represent meaningful stylistic variants or simply outliers. Because the imbalance is so extreme, additional validation, such as metadata linkage or alternative clustering formulations, may be needed in later milestones.

2. EDA & Preprocessing: Updates

M1 findings most relevant for M2. M1's EDA highlighted an extremely strong "common palette" baseline. For instance, Titanium_White appears in about 99% of paintings, which suggests the data may naturally form one dominant group plus a small number of rare groups, rather than several balanced archetypes. This directly shaped the M2 framing. Instead of aiming for "balanced segmentation," I treated "dominant baseline vs. rare signatures" as the main structure to interpret.

M1 also provided evidence that palette choices repeat in template-like ways. Jaccard similarity was generally high, M1 found many highly similar pairs, and exact paint "recipes" repeat across multiple paintings. Together, these patterns support the idea that palette composition is not random across episodes and motivate reframing the question in M2 as identifying interpretable palette archetypes.

I did not add a new round of EDA in M2 because the dataset and feature set are unchanged from M1, and the M1 EDA already contains the key justification for clustering and rule mining. The main M2 refinement was for interpretability in association-rule mining: I treated any paint with global usage ≥ 0.95 as "trivial" (which only includes Titanium_White) from rule mining so the rules would capture informative co-usage patterns instead of being dominated by a near-always-present paint.

The only preprocessing step added in M2 was for interpretability in association-rule mining. Paints with extremely high global usage (≥ 0.95) do not contribute useful information to rule discovery because they appear in almost every painting. To avoid rules dominated by trivial antecedents or consequents, I removed Titanium White before mining both global and cluster-level rules. This ensures that the resulting rules highlight informative co-usage patterns rather than restating the near-universal baseline.

3. Analysis & Experiments

Method 1: Agglomerative Clustering on Binary Paint Vectors (Jaccard)

The goal of Method 1 is to identify potential palette archetypes by grouping paintings with similar sets of paints. Because each painting is represented as a binary vector indicating whether each of the 18 paints is used, the palette is naturally treated as a set. I compared Jaccard and Hamming distances and found that Jaccard provides a wider and more meaningful range for set-based similarity, whereas Hamming compresses distances due to the large number of shared zeros. And, in the Bob Ross Painting dataset, archetypes are defined by which paints are used, not shared absences. Therefore, Jaccard better matches the palette-as-set interpretation and better reflects overlap in actual paint usage.

- **Jaccard distance range:** [0.000, 1.000]
- **Hamming distance range:** [0.000, 0.778]

I applied agglomerative clustering with average linkage, which avoids the chaining behavior of single linkage and does not force overly compact clusters as complete linkage sometimes does. I examined solutions at k = 3, 4, and 5. At k = 3 and k = 4, nearly all paintings collapsed into a single large cluster, which was too coarse to support interpretable archetypes. At k = 5, the first meaningful structure emerged: one dominant cluster of 385 paintings (~95.5%) and several much smaller clusters (sizes 10, 5, 2, and 1). The dendrogram confirmed that this imbalance reflects genuine structure rather than a coding issue, consistent with the near-universal use of core paints observed in M1.

To interpret the clusters, I summarized paint usage using lift, defined as the ratio of a paint's usage rate within a cluster to its global usage rate. Lift highlights paints that are unusually common within a cluster relative to the dataset overall. For very small clusters, lift becomes unstable, so I relied on raw paint-frequency profiles instead. This approach revealed that the dominant cluster closely mirrors global usage patterns, while the smaller clusters show distinct combinations of less common paints such as Liquid Clear, Black Gesso, and Prussian Blue. These patterns suggest that the palette structure consists of a standard baseline and a few rare but coherent variants.

Method 2: Association Rules to Explain Archetypes (Cluster-level & Global Baseline)

This method extracts stable paint co-usage patterns within each cluster to explain "what defines" each archetype, and compares them to global patterns.

It is very helpful for the research question because association rules are designed for binary co-occurrence. Running rules within clusters gives cluster-level signatures. Running a global baseline helps separate "rules that are common everywhere" from "rules that really characterize a specific archetype."

I used association rules in two ways. First, I mined rules from the full dataset to establish a global baseline independent of clustering. Second, I mined rules within each cluster to summarize cluster-specific "recipes," using the clustering labels to slice the data for interpretation rather than as a second clustering method.

Before mining rules, I removed Titanium_White from rule mining because its global usage is extremely high (≥ 0.95), and keeping it would result in many rules being dominated by a paint that appears in nearly every painting. For the global baseline, I used min_support = 0.10, min_confidence = 0.60, and min_lift = 1.30.

For within-cluster rules, I set MIN_CLUSTER_N = 10 to ensure stable rule mining. Clusters with n< 10 were skipped due to insufficient sample size. Within-cluster rules were computed for Cluster 3 (n=10, yielding 43 cluster-only rules) and Cluster 4 (n=385, yielding 2 rules that also appear globally). I used a cluster-specific support threshold min_support = max (3/n_cluster, 0.10) so that rules in smaller clusters still have a minimum of roughly three supporting paintings, yielding interpretable signatures.

The dominant cluster behaves like a baseline: under these thresholds, it produces only a small number of strong rules, and those mostly overlap with the global baseline. In contrast, the specialized cluster with n = 10 is much more distinctive. It produces many cluster-only rules concentrated around Liquid_Clear and Black_Gesso combinations, which gives a concrete, interpretable description of that palette mode. A limitation I accepted is that clusters with n < 10 are too small for stable rule mining. I did not treat their rule outputs as evidence; instead, I described those clusters using paint frequency profiles.

4. Findings & Interpretations

Bob Ross paintings form interpretable palette archetypes, but the structure is extremely imbalanced. Under the k=5 setting, one archetype dominates: 385 paintings (~95.5%) share a common baseline palette. The remaining paintings are split into a few small, specialized clusters (5, 2, 10, and 1 paintings), which represent rare palette modes rather than the mainstream approach.

The dominant group behaves like the show's default palette. Its paint usage looks almost the same as the dataset overall, and the few strong co-usage patterns that appear are the same ones you would expect across the entire series. In other words, this cluster mostly reflects the common and repeatable elements of Bob Ross's teaching style.

The small groups are rare but not random. The clearest example is the 10-painting cluster, which shows a consistent "signature" centered on Liquid_Clear and Black_Gesso. The fact that these signatures appear repeatedly within the same small group suggests that at least some rare clusters capture a distinct palette mode rather than noise.

This structure is useful if we think of the episode library as something people browse, search, or learn from. Most episodes fall into a clear "baseline palette" mode, while a small number stand out as genuinely different. Organizing the library around that split makes it easier to navigate. Viewers who feel the show is repetitive can skip to the unusual palette cases. For learners, the baseline group is a sensible place to start, and the rare groups function like "next step" variations once the basics feel familiar. The cluster signatures also read like simple palette recipes, which gives a plain-language way to describe what makes a rare archetype distinct.

If this were used in a streaming setting, the same split could be turned into a practical browsing and recommendation layer: a main track for the standard look, and a small "something different" section that helps people discover outliers without having to sift through hundreds of similar episodes.

5. Next Steps

The clustering is highly imbalanced. The next step is to connect clusters to episode metadata. For example, season, subject, theme, or background type, to see whether the small clusters line up with specific techniques or recurring themes. It is also worth testing imbalance-aware clustering ideas, such as running a second clustering pass after separating the dominant cluster, or using density-based methods that treat the dominant mode as "background" and retain structured outliers. For association rules, the next improvement would be to provide more visual summaries and to check which rules repeat across clusters.

At this stage, it is unclear whether the small clusters reflect genuine palette styles or are just rare combinations due to low frequency. The main decision is whether to treat the dominant cluster as a default archetype and view the remaining clusters as meaningful but rare modes. Linking clusters to metadata and testing an imbalance-aware approach would help answer this.

Appendix

Shared GitHub Repository (Required)

Link to shared team GitHub repository:
https://github.com/yujiunzou/BA820-Unsupervised-ML-Project

My individual M2 work (where to find my files):

- Branch: “*Irene-(Yihui)-Tang*”
- Folder: “M2_Irene/”
- Primary notebook (graded analysis): “*M2_Irene/Irene_Tang_M2_Individual_Assignment_Notebook.ipynb*”
- Supporting file (report PDF): “*M2_Irene/Irene_Tang_M2_Individual_Assignment_Report.pdf*”
- Folder documentation: “*M2_Irene/README.md*”

Link to my M2 Notebook:
https://github.com/yujiunzou/BA820-Unsupervised-ML-Project/blob/Irene-(Yihui)-Tang/M2_Irene/Irene_Tang_M2_Individual_Assignment_Notebook.ipynb

Supplemental Material (Highly Recommended)

- **Method 1: Clustering**

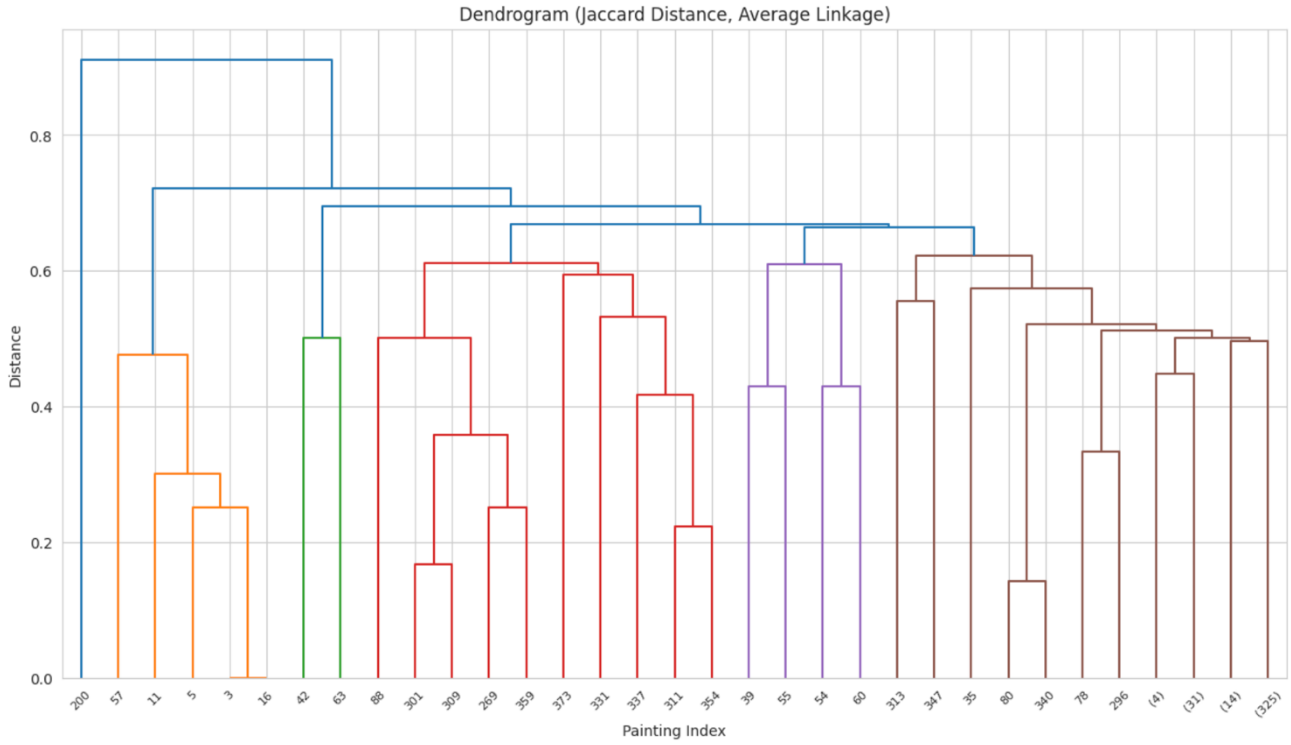


Figure 1: Dendrogram

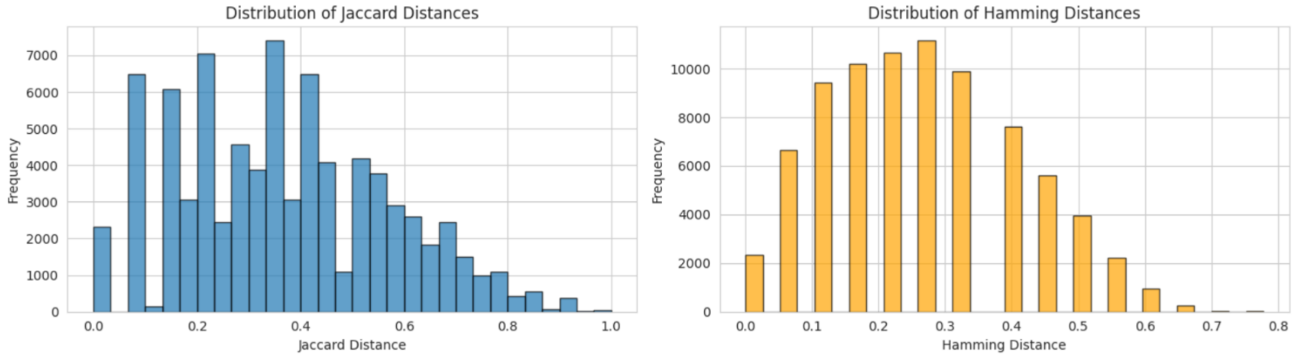


Figure 2: Jaccard vs Hamming distance distribution

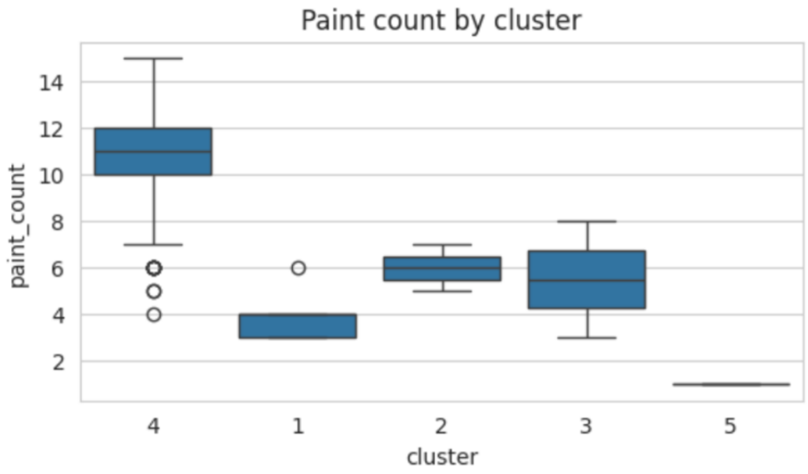
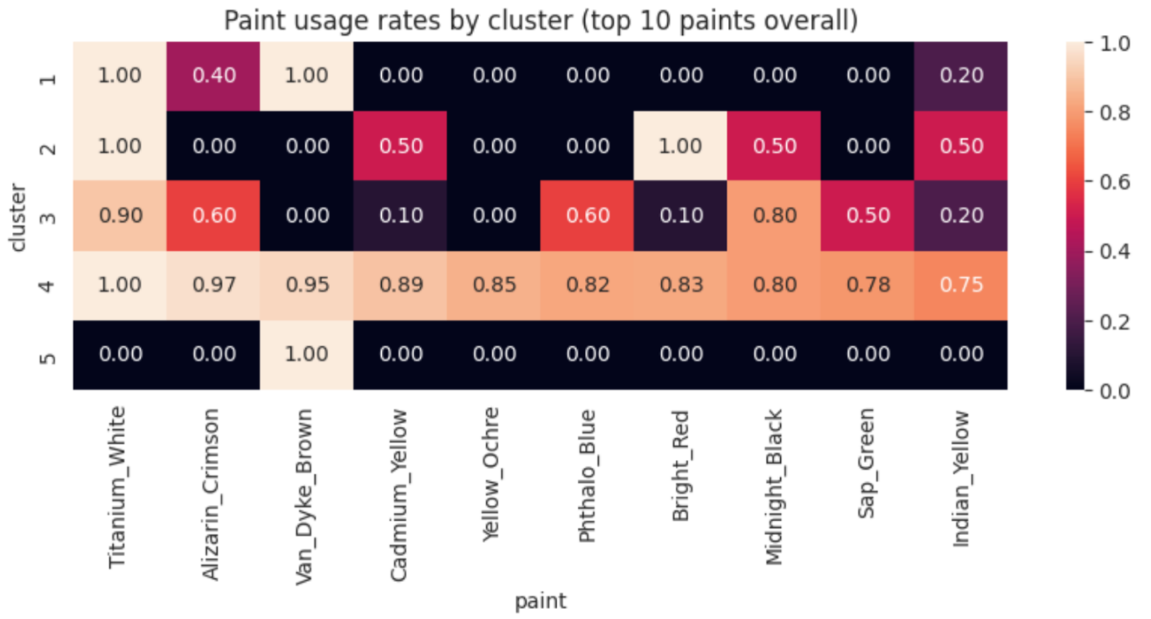


Figure 3: Palette complexity differs by archetype (paint count per painting)



cluster	Titanium_White	Alizarin_Crimson	Van_Dyke_Brown	Cadmium_Yellow	Yellow_Ochre	Phthalo_Blue	Bright_Red	Midnight_Black	Sap_Green	Indian_Yellow
1	1.00	0.40	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20
2	1.00	0.00	0.00	0.50	0.00	0.00	1.00	0.50	0.00	0.50
3	0.90	0.60	0.00	0.10	0.00	0.60	0.10	0.80	0.50	0.20
4	1.00	0.97	0.95	0.89	0.85	0.82	0.83	0.80	0.78	0.75
5	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 4: Cluster-level paint usage heatmap

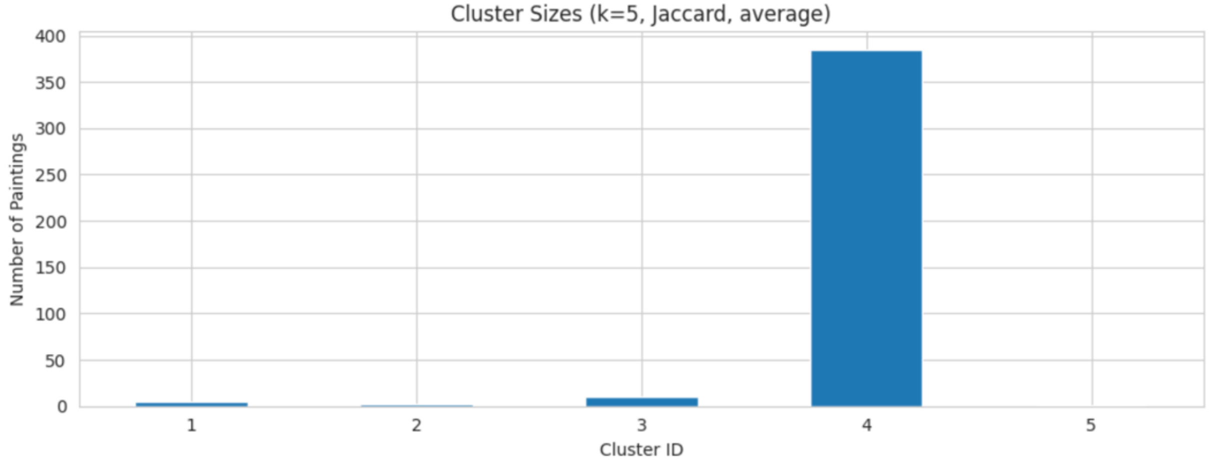
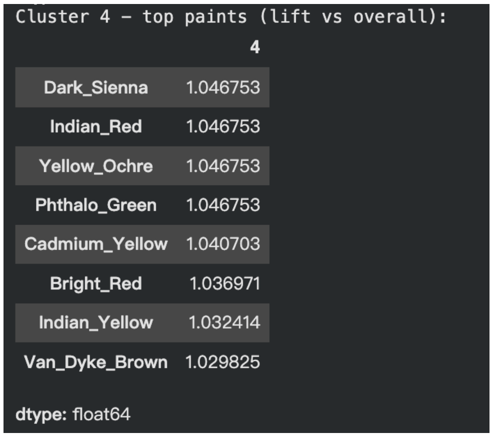


Figure 5: Cluster Size

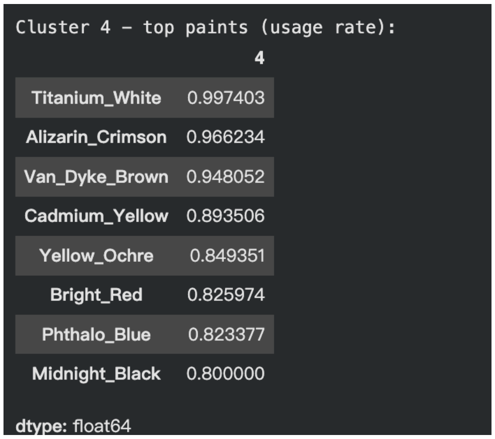


Cluster 4 – top paints (lift vs overall):

	4
Dark_Sienna	1.046753
Indian_Red	1.046753
Yellow_Ochre	1.046753
Phthalo_Green	1.046753
Cadmium_Yellow	1.040703
Bright_Red	1.036971
Indian_Yellow	1.032414
Van_Dyke_Brown	1.029825

dtype: float64

Table 1: Cluster 4 – Top lifted Paints vs. Overall Usage



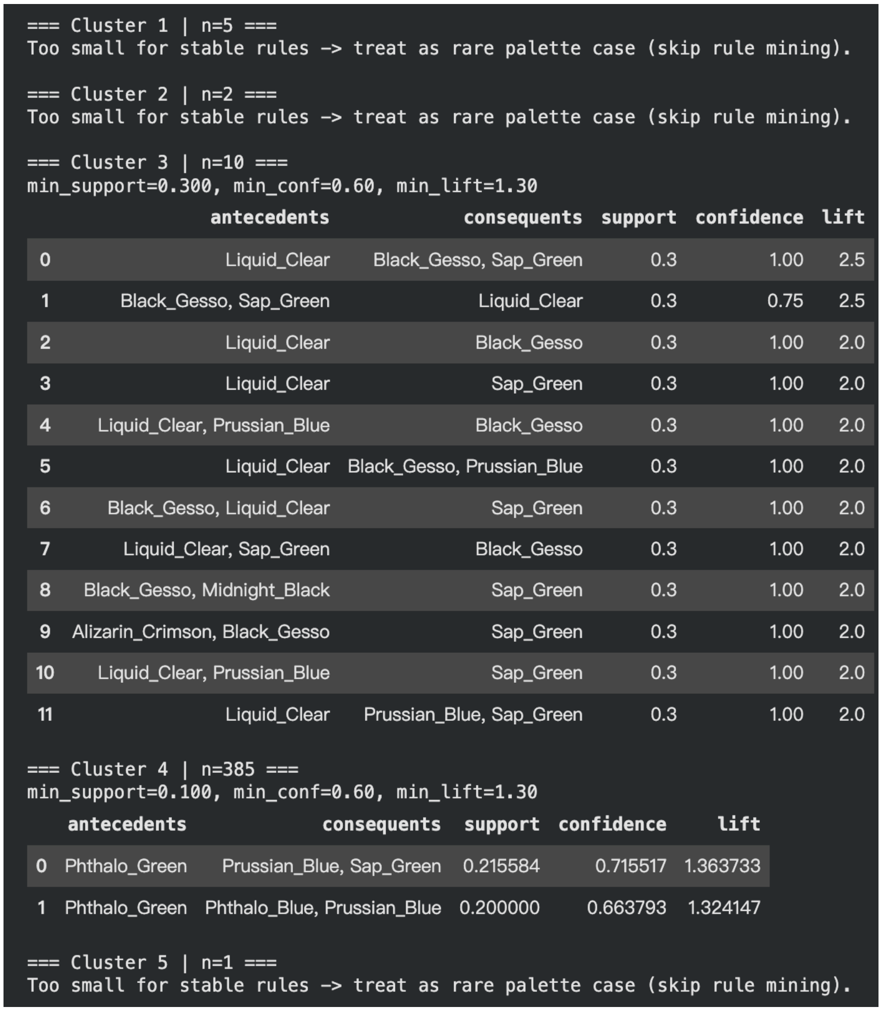
Cluster 4 – top paints (usage rate):

	4
Titanium_White	0.997403
Alizarin_Crimson	0.966234
Van_Dyke_Brown	0.948052
Cadmium_Yellow	0.893506
Yellow_Ochre	0.849351
Bright_Red	0.825974
Phthalo_Blue	0.823377
Midnight_Black	0.800000

dtype: float64

Table 1: Cluster 4 – Top Paints: Usage Rate

- **Method 2: Association Rules**



```
=== Cluster 1 | n=5 ===
Too small for stable rules -> treat as rare palette case (skip rule mining).

=== Cluster 2 | n=2 ===
Too small for stable rules -> treat as rare palette case (skip rule mining).

=== Cluster 3 | n=10 ===
min_support=0.300, min_conf=0.60, min_lift=1.30
```

	antecedents	consequents	support	confidence	lift
0	Liquid_Clear	Black_Gesso, Sap_Green	0.3	1.00	2.5
1	Black_Gesso, Sap_Green	Liquid_Clear	0.3	0.75	2.5
2	Liquid_Clear	Black_Gesso	0.3	1.00	2.0
3	Liquid_Clear	Sap_Green	0.3	1.00	2.0
4	Liquid_Clear, Prussian_Blue	Black_Gesso	0.3	1.00	2.0
5	Liquid_Clear	Black_Gesso, Prussian_Blue	0.3	1.00	2.0
6	Black_Gesso, Liquid_Clear	Sap_Green	0.3	1.00	2.0
7	Liquid_Clear, Sap_Green	Black_Gesso	0.3	1.00	2.0
8	Black_Gesso, Midnight_Black	Sap_Green	0.3	1.00	2.0
9	Alizarin_Crimson, Black_Gesso	Sap_Green	0.3	1.00	2.0
10	Liquid_Clear, Prussian_Blue	Sap_Green	0.3	1.00	2.0
11	Liquid_Clear	Prussian_Blue, Sap_Green	0.3	1.00	2.0

```
=== Cluster 4 | n=385 ===
min_support=0.100, min_conf=0.60, min_lift=1.30
```

	antecedents	consequents	support	confidence	lift
0	Phthalo_Green	Prussian_Blue, Sap_Green	0.215584	0.715517	1.363733
1	Phthalo_Green	Phthalo_Blue, Prussian_Blue	0.200000	0.663793	1.324147

```
=== Cluster 5 | n=1 ===
Too small for stable rules -> treat as rare palette case (skip rule mining).
```

Table 3: Association Rules Table

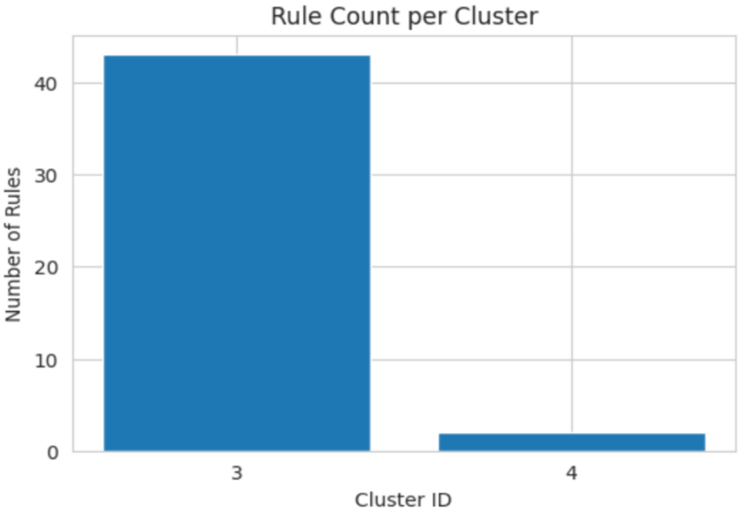


Figure 6: Rule Count per Cluster

Process Overview

1. **Data loading:** Load the Bob Ross dataset (403 paintings) and select the 18 paint indicator columns.
2. **Data validation:** Confirm paint indicators are binary, with no missing values/duplicates in the feature matrix.
3. **Core EDA (from M1):** Summarize global paint usage rates and assess palette similarity using pairwise Jaccard similarity to confirm recurring palette templates.
4. **Feature setup for M2:** Use the 18 paint indicators as binary vectors; treat each painting's palette as a set of paints.
5. **Method 1: Clustering:**
   a) Compute Jaccard distance between paintings
   b) Run agglomerative clustering (average linkage)
   c) Compare k = 3, 4, 5 and select k = 5 based on interpretability
6. **Cluster interpretation:**
   a) Report cluster sizes
   b) Summarize cluster paint profiles
7. **Method 2: Association rules (global & within-cluster):**
   a) Remove trivial paints
   b) Mine global rules as a baseline
   c) Mine within-cluster rules for clusters with n ≥ 10 and compare against the global baseline
8. **Insight generation:** Translate outputs into a non-technical interpretation: a dominant "standard palette" archetype plus a small number of rare, structured archetypes with clear signature paint combinations.

Use of Generative AI Tools

Link: https://chatgpt.com/share/698a78fb-edd4-8010-997f-9dc6413a8a78

I used ChatGPT to organize the assignment requirements and to build a checklist for a final submission review. I also discussed my research question with ChatGPT and asked for possible method directions to consider. To confirm that my distance metric matched the data and research goal, I asked about the differences between Jaccard and Hamming distance and why Jaccard is appropriate for binary set-based paint usage.

During the coding stage, I asked about the structure of Apriori-based association rule mining in Python. ChatGPT provided example code patterns that I adapted in my notebook. When I encountered repeated warning messages in the output, I asked for ways to manage them; I implemented one of the suggested approaches at the beginning of the notebook to keep the output readable.

Because my clustering results included several small clusters, I asked for guidance on setting reasonable thresholds for rule mining in small samples. And I asked how to simplify the comparison between within-cluster rules and a global baseline in Python, and ChatGPT suggested a compact way to construct comparable rule keys and check overlap. I also used ChatGPT to sanity-check an apparent result mismatch (few rules in the dominant cluster vs.

many rules in a small cluster) and to confirm whether this pattern is expected given cluster genericity and threshold choices rather than a coding issue.

In addition, I used ChatGPT to learn the basic steps for creating folders and working with branches in GitHub. And how to write the direction to my individual M2 notebook from the link in plain, clear words. At the end of the project, I used ChatGPT for a final grammar and spelling check of the report. And ask for an example of a high-level process flow.