M3_ML

IT 10 JA

Yesterday 3:29 PM

Irene Tang created the group chat.

Your personal ChatGPT memory is never used in group chats.

Invite with link

KZ joined the group chat.

joined the group chat.

Irene Tang renamed the group chat.

janice joined the group chat.

Today 4:31 PM

Irene Tang 4:31 PM

IT

@ChatGPT I ran hierarchical clustering with Jaccard on the dataset. k=5, result: 385/10/5/2/1. can i treated the dominant cluster as a baseline and interpret the small clusters using lift vs baseline and within-cluster association rules?

Jennifer Zhu joined the group chat.

ChatGPT

Yeah, that's a totally reasonable way to read it — just be careful about a few traps.

- **Dominant cluster as "baseline"**: works *if* it's genuinely a "general population" cluster and not just "missingness / low-activity / common tags." Quick check: compare basic stats (avg set size, sparsity, missing rate) between the big cluster and others.
- **Lift vs baseline for small clusters**: good idea. Compute for each item (or pair)

  $$\text{lift}_{c\ vs\ base}(x) = \frac{P(x \mid c)}{P(x \mid base)}$$

  and consider also **log lift** + a **minimum support** so you don't over-interpret tiny counts.
- **Significance / stability** (super important with sizes 10/5/2/1): use Fisher's exact / binomial CI or at least a "must appear ≥2 (or ≥3)" rule; otherwise lift will explode for rare stuff.

- **Within-cluster association rules**: fine, but for tiny clusters I'd cap it to *descriptive* rules (high confidence + decent absolute count). If you can, use rules with a **shared-minimum-support across clusters** to compare fairly.
- **Sanity check**: also compute lift for the big cluster *vs the rest* to confirm it's actually baseline-ish.

Show less

If you tell me what "items" are (tags? symptoms? products?) and the typical set size per row, I can suggest concrete thresholds for support/confidence and a good way to report it without overclaiming.

Jennifer Zhu was removed from the group chat.

Irene Tang

IT

@ChatGPT What other approaches could I try to better surface meaningful structure beyond the dominant baseline?

ChatGPT

A few good "next moves" to surface structure when one giant cluster eats everything:

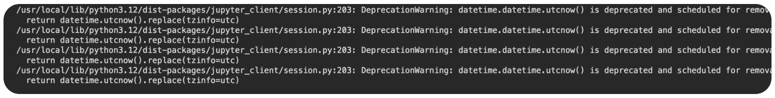
- **Try a different distance / linkage**
  Jaccard + hierarchical can lump "mostly-common" rows together. Test **cosine on TF-IDF (or BM25)** vectors, or **Hamming** if it's truly binary. Also swap linkage (average/complete/Ward*—Ward needs Euclidean).
- **Reweight common items**
  If some items appear everywhere, they dominate similarity. Do **IDF weighting** (treat each row as a sparse vector) so rare-but-informative items matter more.
- **Filter / cap ultra-common + ultra-rare features**
  Drop items with, say, **>80–90% prevalence** (uninformative) and items with **<2–3 occurrences** (too noisy), then recluster.
- **Model-based clustering for sparse binaries**
  Use **Latent Class Analysis / Bernoulli mixture** (soft clusters) or **topic-model style** approaches. These often reveal "archetypes" even when hard clustering collapses.
- **Community detection on a graph**
  Build a graph of rows (kNN by Jaccard/cosine) or an **item co-occurrence graph**, then run **Leiden/Louvain**. This can split the "baseline" into subcommunities.
- **Dimensionality reduction + clustering**
  Use **UMAP** (with Jaccard/cosine) then **HDBSCAN** (density-based) to find small dense pockets without forcing everything into k clusters.

- **Stability-driven selection**
  Bootstrap rows/features and see which subclusters are **stable**. Often the "real" structure is what survives resampling, not what maximizes a single metric.

If you tell me roughly: (1) #rows, (2) avg items-per-row, and (3) whether items are super-skewed (a few very common ones), I can point you to the 2–3 most promising options and concrete parameter ranges.

Show less

Irene Tang



```
/usr/local/lib/python3.12/dist-packages/jupyter_client/session.py:203: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for remov
  return datetime.utcnow().replace(tzinfo=utc)
/usr/local/lib/python3.12/dist-packages/jupyter_client/session.py:203: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for remova
  return datetime.utcnow().replace(tzinfo=utc)
/usr/local/lib/python3.12/dist-packages/jupyter_client/session.py:203: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for remova
  return datetime.utcnow().replace(tzinfo=utc)
/usr/local/lib/python3.12/dist-packages/jupyter_client/session.py:203: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for remova
  return datetime.utcnow().replace(tzinfo=utc)
```

IT

@ChatGPT how to remove this?

ChatGPT

That's just a **DeprecationWarning** from `jupyter_client` using `datetime.utcnow()` in Python 3.12.

You have 3 practical options:

✅ 1. Just suppress the warning (easiest)

Add this at the top of your notebook:

python Copy code

```python
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

If you only want to silence that specific module:

python Copy code

```python
warnings.filterwarnings(
    "ignore",
    category=DeprecationWarning,
    module="jupyter_client"
)
```

✅ 2. Upgrade Jupyter packages (recommended)

It's coming from:

bash Copy code

```bash
jupyter_client/session.py
```

So upgrade:

bash Copy code

```bash
pip install --upgrade jupyter_client jupyter_core ipykernel
```

Then restart the kernel.

(Newer versions remove the deprecated call.)

✅ 3. Temporary environment fix (if it persists)

If you're on Python 3.12 and the package hasn't caught up yet, sometimes downgrading helps:

bash Copy code

```bash
pip install "jupyter_client<8"
```

But upgrading is usually better.

💡 This warning is harmless — it won't break your clustering code. It's just noisy.

If you tell me your Python + Jupyter versions, I can suggest the cleanest fix.

Show less

@ChatGPT For this Bob Ross painting project clustering part, last time we used fcluster with the "maxclust", which resulted in one dominant cluster containing most paintings, but I want to see how other ways of cutting the clusters work, instead of forcing a fixed number of clusters, so this way we are not just accepting the fact that our data is mainly falling into one big cluster. can i set up thresholds? and i want to try different thresholds

ChatGPT

Yes — exactly 👍 you don't have to force `maxclust`. You can cut the dendrogram by **distance threshold** instead.

Instead of:

python Copy code

```python
from scipy.cluster.hierarchy import fcluster
labels = fcluster(Z, t=5, criterion="maxclust")
```

Try cutting by **distance**:

python Copy code

```python
labels = fcluster(Z, t=0.6, criterion="distance")
```

Here, `t` = the height where you cut the dendrogram.

🔎 How to choose thresholds?

1. Plot the dendrogram

python Copy code

```python
from scipy.cluster.hierarchy import dendrogram
import matplotlib.pyplot as plt

plt.figure(figsize=(10,5))
dendrogram(Z)
plt.show()
```

Look for large vertical gaps — those are natural cut points.

---

2. Try multiple thresholds

python Copy code

```python
for t in [0.3, 0.5, 0.7, 1.0]:
    labels = fcluster(Z, t=t, criterion="distance")
    print(t, len(set(labels)))
```

This lets you see how cluster count changes smoothly instead of forcing k.

🔥 Other good alternatives

- `criterion="inconsistent"` → cuts based on inconsistency statistics
- Try **complete linkage** instead of average (often breaks big loose clusters)
- Use **HDBSCAN** if you want density-based clusters that naturally leave some points unclustered

---

If your dominant cluster is happening because distances are small overall, distance-threshold cutting is much better than `maxclust`.

If you tell me:

- linkage method?
- typical Jaccard distance range?
- number of paintings?

I can suggest smart threshold ranges to try.

Show less

i also want to explore each cluster size with each threshold, like adding some calculations. I wrote these calculations that i thought would be helpful to see. n_clusters = len(np.unique(labels))

largest_pct = cluster_sizes.max() / len(labels)
smallest_pct = cluster_sizes.min() / len(labels)
imbalance_ratio = cluster_sizes.max() / cluster_sizes.min() but how can i make this section more consolidated? i tried adding in the for loop but didn't do it correctly so it is not iterating properly.

ChatGPT

Yes — best way is to compute everything **inside the threshold loop** and store results in a list (or DataFrame). That keeps it clean and actually iterative.

Here's a compact version:

python Copy code

```python
import numpy as np
import pandas as pd
from scipy.cluster.hierarchy import fcluster

thresholds = [0.3, 0.5, 0.7, 1.0]
results = []

for t in thresholds:
    labels = fcluster(Z, t=t, criterion="distance")

    cluster_sizes = np.bincount(labels)[1:]  # ignore zero index
    n_clusters = len(cluster_sizes)

    largest_pct = cluster_sizes.max() / len(labels)
    smallest_pct = cluster_sizes.min() / len(labels)
    imbalance_ratio = cluster_sizes.max() / cluster_sizes.min()

    results.append({
        "threshold": t,
        "n_clusters": n_clusters,
        "largest_pct": largest_pct,
        "smallest_pct": smallest_pct,
        "imbalance_ratio": imbalance_ratio
    })
```

```
summary_df = pd.DataFrame(results)
print(summary_df)
```

This gives you a clean table like:

| threshold | n_clusters | largest_pct | smallest_pct | imbalance_ratio |

If you want it even more compact:

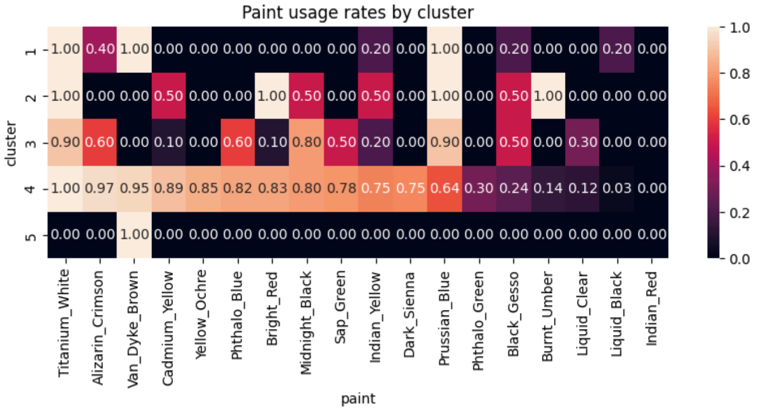
python Copy code

```
for t in thresholds:
    labels = fcluster(Z, t=t, criterion="distance")
    sizes = np.bincount(labels)[1:]
    print(
        t,
        len(sizes),
        sizes.max()/len(labels),
        sizes.min()/len(labels),
        sizes.max()/sizes.min()
    )
```

That'll let you quickly see how imbalance changes as you move the cut height 👀

Show less

Today 5:49 PM



cluster	Titanium_White	Alizarin_Crimson	Van_Dyke_Brown	Cadmium_Yellow	Yellow_Ochre	Phthalo_Blue	Bright_Red	Midnight_Black	Sap_Green	Indian_Yellow	Dark_Sienna	Prussian_Blue	Phthalo_Green	Black_Gesso	Burnt_Umber	Liquid_Clear	Liquid_Black	Indian_Red
1	1.00	0.40	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	1.00	0.00	0.20	0.00	0.00	0.20	0.00
2	1.00	0.00	0.00	0.50	0.00	0.00	1.00	0.50	0.00	0.50	0.00	1.00	0.00	0.50	1.00	0.00	0.00	0.00
3	0.90	0.60	0.00	0.10	0.00	0.60	0.10	0.80	0.50	0.20	0.00	0.90	0.00	0.50	0.00	0.30	0.00	0.00
4	1.00	0.97	0.95	0.89	0.85	0.82	0.83	0.80	0.78	0.75	0.75	0.64	0.30	0.24	0.14	0.12	0.03	0.00
5	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Can you help me name the five clusters and organize the traits of them?

10

@ChatGPT

ChatGPT

Yep 👍 — based on the paint-usage patterns in your heatmap, here's a **clean interpretable naming + traits** pass:

🎨 Cluster 4 — "Classic Full-Palette Landscapes"

Traits

- High usage across many paints (white, browns, blues, greens, yellows)
- Looks like the *baseline Bob Ross recipe*
- Balanced warm + cool colors
  👉 Good candidate for your "dominant baseline" cluster.

🌈 Cluster 3 — "Color-Rich / Expressive Scenes"

Traits

- Strong Alizarin Crimson + Phthalo Blue + Sap Green
- More saturated colors than cluster 4
- Likely sunsets / vibrant compositions.

🖤 Cluster 2 — "Dark-Base High-Contrast"

Traits

- Black Gesso + Midnight Black + Burnt Umber spikes
- Fewer total paints but strong contrasts
- Suggests dramatic or night scenes.

❄️ Cluster 1 — "Cool Minimalist Mix"

Traits

- Titanium White + Prussian Blue + Van Dyke Brown dominate
- Very sparse elsewhere
- Probably snow, water, or subdued palettes.

🟫 Cluster 5 — "Earth-Tone Outliers"

Traits

- Almost only Van Dyke Brown
- Extremely sparse usage → likely niche or edge cases
- Treat more like an **outlier micro-cluster** than a core style.

If you want something more "ML-paper ready," I can also give you:

- a **rule-based naming formula** (auto-generate cluster labels from top paints)
- or a 1-line description template so all five sound consistent 👍 .

Show less

Today 7:22 PM

janice

Report.docx -.pdf
PDF

@ChatGPT help me check this file for grammar and spelling errors, if any correction needed list the original text and suggestions

JA

@ChatGPT

ChatGPT

Got it 👍 I'll list **only sentences/phrases that need correction**, with:

- **Original**
- **Suggested revision**

🔎 Grammar & Spelling Corrections

1️⃣ Extra punctuation

Original:

> Although we used different unsupervised methods,, our results were very consistent
> —

Show more

ChatGPT

Got it 👍 I'll list **grammar/spelling issues and suggested corrections** in a clear format:

1️⃣ Punctuation / Typo Errors

❌ Original:

> Although we used different unsupervised methods,, our results were very consistent
> —

✅ Suggestion:

> Although we used different unsupervised methods, our results were very consistent
> —

(double comma removed)

❌ Original:

> 1groups of colors working together

✅ Suggestion:

groups of colors working together

(remove the stray “1”)

❌ Original:

We also kept association rule mining as a way to complementarily interpret the clusters.While hierarchical clustering showed...

✅ Suggestion:

We also kept association rule mining as a way to complementarily interpret the clusters. While hierarchical clustering showed...

(add space after period)

❌ Original:

Jaccard- based hierarchical clustering.

✅ Suggestion:

Jaccard-based hierarchical clustering.

(remove space after hyphen)

❌ Original:

2reflects Bob Ross’s consistent canonical color usage...

✅ Suggestion:

reflects Bob Ross’s consistent canonical color usage...

(remove stray “2”)

❌ Original:

35. **“Insights Gained Through Integration”**

✅ Suggestion:

5. Insights Gained Through Integration

(remove stray “3”)

❌ Original:

4So far, text information has only been explored at a basic level, we observed that title similarity is much weaker than palette similarity.

✅ Suggestion:

So far, text information has only been explored at a basic level. We observed that title similarity is much weaker than palette similarity.

(remove “4” + fix comma splice)

2️⃣ Grammar / Sentence Structure Improvements

❌ Original:

even though we work independently.

✅ Suggestion:

even though we worked independently.

(maintain past tense consistency)

❌ Original:

Because the data is binary, we focused on similarity-based clustering and association rules, and looked at how the methods support or challenge each other instead of evaluating them separately.

✅ Suggestion (cleaner parallel structure):

Because the data is binary, we focused on similarity-based clustering and association rules, examining how the methods support or challenge each other rather than evaluating them separately.

❌ Original:

Instead of fixing the number of clusters using an arbitrary k value, we shifted to exploring distance thresholds in hierarchical clustering. We also then analyzed cluster size distributions...

✅ Suggestion:

Instead of fixing the number of clusters using an arbitrary k value, we shifted to exploring distance thresholds in hierarchical clustering. We then analyzed cluster size distributions...

(remove redundant “also”)

❌ Original:

First, our analysis is limited to binary paint usage.

✅ Suggestion:

First, our analysis is limited to binary paint usage data.

(more precise wording)

❌ Original:

Second, by introducing dendrogram threshold exploration justified the presence of a dominant mega cluster and showed that it is not driven by inappropriate parameter selection, the extreme imbalance in cluster sizes remains a limitation.

This sentence is grammatically broken.

✅ Suggested rewrite:

Second, although dendrogram threshold exploration justified the presence of a dominant mega-cluster and showed that it is not driven by inappropriate parameter selection, the extreme imbalance in cluster sizes remains a limitation.

❌ Original:

Their limited sample size makes it difficult to draw strong statistical conclusions about rare palette types.

✅ Suggestion:

The limited sample sizes of these clusters make it difficult to draw strong statistical conclusions about rare palette types.

(clear reference)

❌ Original:

There are very small clusters that lack strong statistical support, it is difficult to determine whether they represent meaningful stylistic variants or as low frequency combinations.

Comma splice + grammar issue.

✅ Suggested rewrite:

There are very small clusters that lack strong statistical support, making it difficult to determine whether they represent meaningful stylistic variants or simply low-frequency combinations.

❌ Original:

Our next step will focus on deeper text analysis and integration between title, and palette structure.

✅ Suggestion:

> Our next step will focus on deeper text analysis and integration between title and palette structure.

(remove extra comma)

❌ Original:

> patterns or simply tied to specific production periods.

✅ Suggestion:

> patterns or are simply tied to specific production periods.

(missing verb)

3️⃣ Minor Style Suggestions (Optional but improves polish)

- Replace **"mega cluster"** → **"mega-cluster"**
- Keep cluster names stylistically consistent (capitalize consistently)
- Consider replacing "it is more like" → "they are more like" (agreement with clusters)

Overall Assessment

Your writing is actually very strong structurally. Most issues are:

- minor punctuation
- a few comma splices
- 3–4 broken sentences
- stray numbering artifacts (probably from formatting copy/paste)

Nothing conceptually wrong — just polish-level fixes.

Show less

If you'd like, I can also:

- Rewrite the full document into a polished final submission version
- Or mark edits inline like a tracked-edit version
- Or tighten it to sound more academic/formal

Just tell me the target tone 👌