Let me first look at the CSV to understand its structure.Now let me look at the full structure more carefully — particularly the `cleaned_ingredients` column:Good, I understand the structure. Now let me build the complete Python script:Here's the script. The only dependency is `pip install pandas scikit-learn numpy` — all standard, no API keys or heavy models needed.

How it works:

1. **Parses ingredients** from the `cleaned_ingredients` column (handles the numpy-style string format)

2. **Builds substitution groups** using Union-Find over ~170 curated substitute pairs (butter↔ghee↔olive oil, sugar↔honey, basmati rice↔longgrain rice, etc.) plus **auto-discovered** pairs from your corpus (handles fresh/dried prefixes, plurals, "zest of" suffixes, etc.)

3. **Canonicalizes** each recipe's ingredients — so "ghee" and "butter" map to the same token, making recipes with either one appear more similar

4. **TF-IDF** vectorizes the canonical ingredient lists (with `sublinear_tf` to dampen ubiquitous ingredients like "salt")

5. **Truncated SVD** reduces to 64-dimensional dense embeddings, then L2-normalizes so cosine similarity = dot product

Output: A CSV with columns `Name, emb_0, emb_1, ..., emb_63` — plug directly into your recommender model.

To run:

```
pip install pandas scikit-learn numpy
python recipe_embeddings.py
```

You can tune `EMBEDDING_DIM` (line 18) and add/remove pairs in `SUBSTITUTION_PAIRS` for your specific corpus. The script also prints all discovered substitution groups and a pairwise similarity sanity check.