Task 2: Cuisine-Map Mining

Dheer Gupta

May 2025

1 Overview

My objective in Task 2 was to build an exploratory *cuisine map* that reveals how restaurant cuisines relate to one another in the Yelp dataset. I (1) aggregated review text by cuisine, (2) engineered several text representations, (3) computed pairwise cosine similarities, (4) visualized those similarities as heat maps, and finally (5) clustered cuisines with two different algorithms and two values of k.

2 Data Preprocessing

I loaded `yelp_academic_dataset_business.json` and `yelp_academic_dataset_review.json`, filtered to businesses tagged Restaurants, and exploded the category field so each review held a single cuisine label. After removing non-cuisine categories and requiring at least 500,000 characters of text per cuisine, I ended up with **60 cuisines** for analysis.

3 Text-Representation Experiments

The LDA model revealed interpretable topics including Mexican cuisine (enchiladas, carnitas, tequila), Asian cuisine (ramen, sashimi, tempura), Mediterranean/Indian cuisine (hummus, naan, masala), and Italian cuisine (pepperoni, bruschetta, lasagna).

For every matrix I normalized rows to unit length and computed the cosine similarity matrix $S = XX^\top$.

3.1 Heat-Map Snapshots

Figure 1 compares the TF-IDF and LDA similarity maps; Raw-TF looked noticeably noisier, so I omitted it to save space.

4 Clustering the Cuisine Space

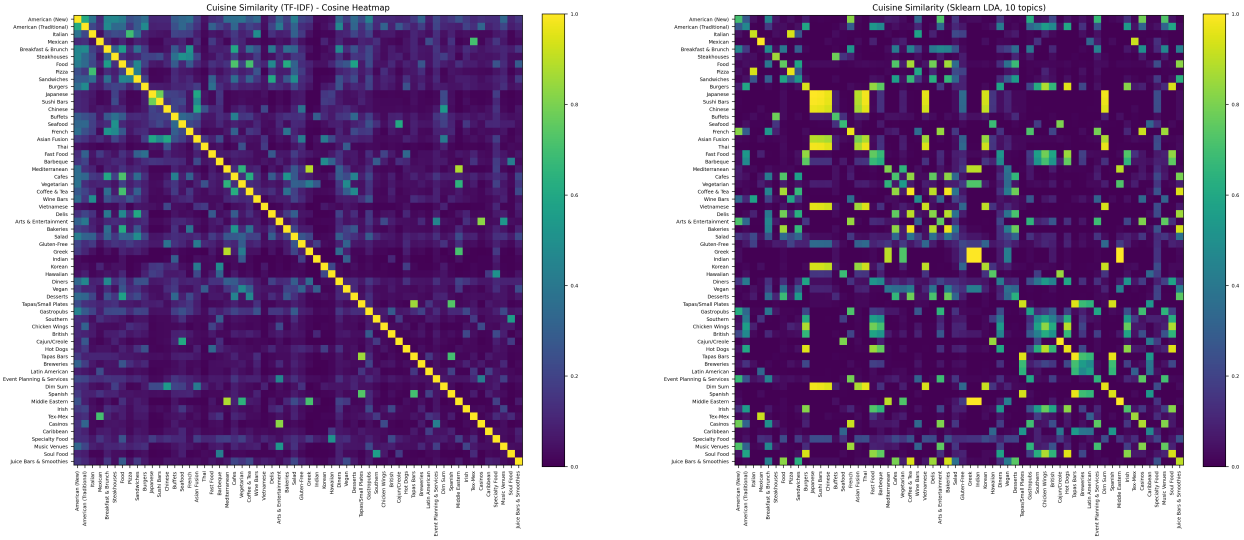
I clustered the LDA vectors with **K-Means** and **Agglomerative** (average-link, cosine) at $k = 4$ and $k = 8$. Each algorithm assigns a color to axis labels in the following heat maps.

Quick Observations

- $k = 4$ reveals four macro groups: (i) American/bar food, (ii) Asian (Sushi, Japanese, Thai, Korean), (iii) Latin/Mexican, and (iv) Mediterranean/veg-leaning.

Table 1: Feature spaces I built and why I tried them

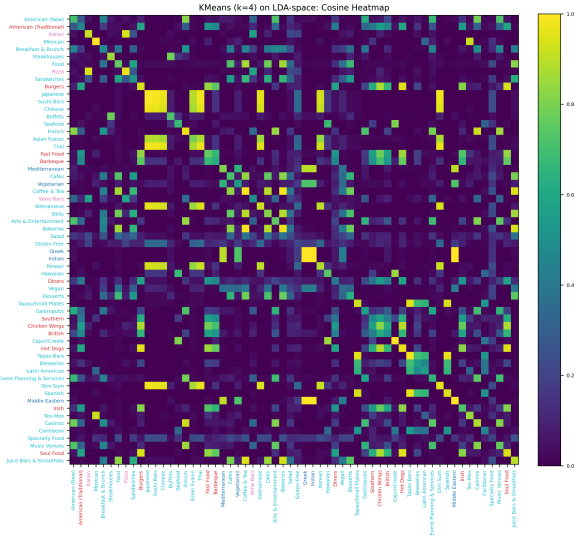
Representation	How I built it	Rationale
Raw TF	2,000-feature `CountVectorizer` with max_df=0.90, min_df=2; counts → L2-normalized cuisine vectors	Baseline overlap; highlights sheer word sharing.
TF-IDF	Same vocab + `TfidfTransformer`	Down-weights ubiquitous tokens like food and place; sharpens signal.
LDA (10 topics)	`LatentDirichletAllocation` with 50 iterations on the TF matrix; each cuisine → 10-D topic mixture	Compresses documents into salient themes—useful for clustering.



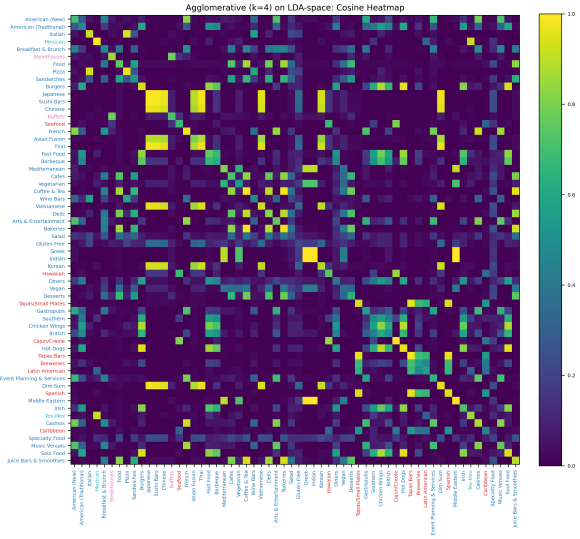
(a) TF-IDF cosine heat map

(b) LDA (10 topics) cosine heat map

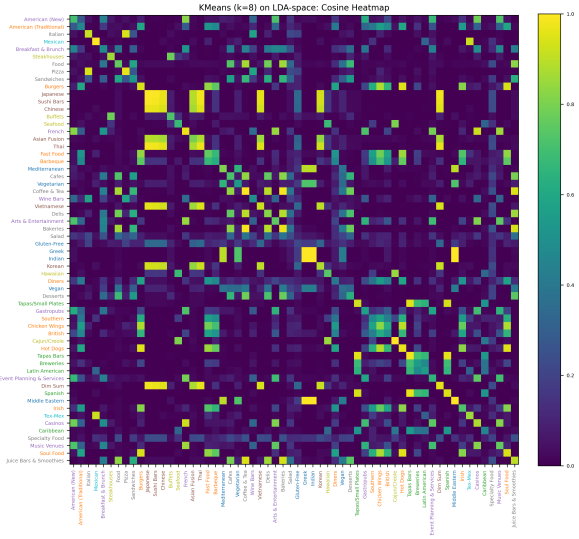
Figure 1: TF-IDF versus LDA similarity structure. LDA shows sharper block-diagonal patterns.



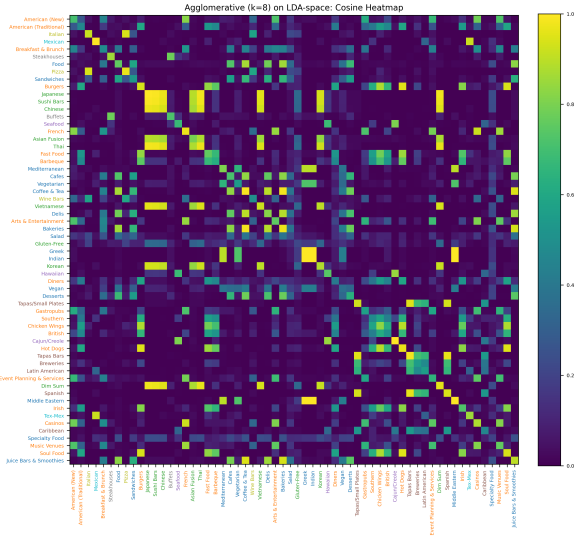
(a) K-Means $k = 4$



(b) Agglomerative $k = 4$



(c) K-Means $k = 8$



(d) Agglomerative $k = 8$

Figure 2: Four clustering variants of the LDA space.

- $k = 8$ splits those groups finer; e.g., Sushi peels away from general Japanese, and a Caribbean+Hawaiian tropical cluster appears in the Agglomerative run.

5 Evaluation Analysis

I performed quantitative evaluation using silhouette scores and a validation test:

1. **Silhouette Analysis.** K-Means with $k = 8$ achieved the highest silhouette score (0.614), followed by Agglomerative $k = 8$ (0.527), K-Means $k = 4$ (0.293), and Agglomerative $k = 4$ (0.230).

2. **Validation Test.** I checked that Indian cuisine was properly represented in the topic space. The model assigned Indian cuisine 99.99% weight to Topic 4, which contained appropriate terms like naan and masala alongside Mediterranean cuisine terms.

The results showed that higher k values and K-Means clustering produced more coherent cuisine groupings according to the silhouette metric.

6 Practical Implementation

- **Discovery.** Vegetarians see Mediterranean, Middle-Eastern, and Salad cafés clustering near one another—an easy pathway for new dining options.

- **Recommendation engines.** A Hawaiian-food lover could branch into Caribbean, which the map marks as closest in flavor profile.

- **Culinary research.** Schools might notice a gap between Korean and Latin flavors—an untapped fusion niche.

7 Future Scope

1. Build an interactive UMAP scatter so users can hover to see exemplar restaurants.

2. Use automatic cluster-count selection (gap statistic, Bayesian GMM) instead of hand-picking k.

3. Blend TF-IDF with word-embedding averages to capture semantic similarity beyond bag-of-words.