Home-Cooked Food Classification: Identifying Latent Ingredient Groupings through Network Science Clustering, and Developing Machine Learning Models for Food Processing Assessment

Students:

Harsh Vishwakarma (2022205)

Dhairya (2022157)

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science and Engineering

Date of Submission: 28 November 2025

BTP Track: Research

BTP Advisor(s)
Ganesh Bagler

Indraprastha Institute of Information Technology, Delhi
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled "Home-Cooked Food Classification: Identifying Latent Ingredient, Groupings through Network Science Clustering, and Developing, Machine Learning Models for Food Processing Assessment", submitted by me for the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under the guidance of Ganesh Bagler. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the award of any other degree.

Student Name
Harsh Vishwakarma
Dhairya

Place: New Delhi
Date: 28 Nov 2025

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Advisor Name
Prof. Ganesh Bagler

Place & Date:
New Delhi, 28 Nov 2025

Abstract

This report presents three integrated modules developed for the B.Tech project.(1) The first module constructs an ingredient network from a dataset of 39,774 recipes and applies relevance weighting, k-nearest neighbor sparsifica- tion, and Louvain community detection to uncover distinct ingredient-based flavor commu- nities and classify recipes into eight coherent clusters. (2) The second module builds super- vised machine learning models to predict Nova food processing levels using nutrient profiles, achieving strong performance—especially with a 102-feature nutrient dataset—and deploying the best model through a Streamlit interface. (3) The third module performs semantic cluster- ing of 270 culinary process terms, using word embeddings and clustering algorithms to form meaningful groups of cooking actions that enhance recipe understanding and representation.

Figure 1: Ultra Processed Food V/s Home made Food

Project GitHub Repositories

- Home-Cooked Food Protocol Classification
- Numerical Model: Nutrient & NOVA Classification
- Food Pairing Analysis

Acknowledgments

I want to extend my heartfelt thanks to my guide, Prof. Ganesh Bagler. His guidance, knowledge, and constant encouragement made this project possible. I am grateful to the IIIT-Delhi faculty, research scholars, and the Complex System Lab members for their datasets.

Contents

Chapter 1

Introduction

In the contemporary food environment, ultra- processed foods (UPFs) have become deeply embedded in daily consumption patterns. Supermar- ket shelves are increasingly dominated by packaged products formulated not from whole ingredients, but from industrially derived substances—refined starches, seed oils, protein isolates, flavor en- hancers, and emulsifiers—designed to maximize shelf life, hyper-palatability, and profit. As a re- sult, many modern "foods" are constructed more like engineered products than meals prepared from traditional cooking practices. Chris van Tulleken, a leading voice in UPF re- search, captures this transformation succinctly:

"UPF has a long, formal scientific def- inition, but it can be boiled down to this: if it's wrapped in plastic and has at least one ingredient that you wouldn't usually find in a standard home kitchen, it's UPF."

This framing highlights a key concern—many everyday convenience foods rely on components the average person would neither purchase nor in- tentionally cook with, such as maltodextrin, modi- fied starches, invert sugar syrup, hydrogenated oils, or synthetic stabilizers. A striking example is the modern "breakfast bar," which is frequently marketed as wholesome or energy-boosting. Despite imagery of nuts, fruit, and grains, the actual composition often centers around low-cost commodity crops like corn, soy, or wheat—chemically restructured into glucose syrup, soy protein isolate, artificial flavors, and binding agents. These ingredients reduce production costs and enhance shelf stability, but they also move the product far away from the identity of minimally processed food. This raises an increasingly relevant question for both researchers and consumers: what are the chances that foods prepared at home could also fall into patterns of ultra-processing? While home cooking is traditionally associated with fresher and less-processed ingredients, the growing availability of pre-mixed sauces, instant bases, synthetic flavor packets, and ready-made batters means that UPF elements can enter homemade dishes indirectly. A "homemade" pasta dish, for instance, becomes ultra- processed if the sauce relies heavily on industrial stabilizers or flavor additives not typically found in a household pantry. Thus, even domestic cooking can be influenced by the broader industrial food ecosystem.

Chapter 2

Problem Statement

Although ultra-processed foods (UPFs) are com- monly associated with industrially manufactured products, the growing infiltration of UPF ingre- dients—such as stabilizers, refined syrups, flavor bases, and premixed packets—has increasingly blurred the line between "homemade" and "pro- cessed." Many dishes prepared at home now rely on components that contain industrial additives or derivatives of cheap commodity crops, raising an important question: to what extent can homemade food also become ultra-processed, and how can this be identified objectively? Addressing this challenge requires more than conventional food labeling or anecdotal observation. It demands computational tools capable of analyzing the underlying structure of recipes, the nature of their ingredients, and the nutrient com- position that reflects varying levels of processing. By examining homemade foods through the lenses of ingredient networks, nutrient-based prediction, and process semantics, this project aims to de- velop computational methods that clarify how pro- cessing characteristics emerge—even within home kitchens—and contribute to a more transparent un- derstanding of modern food preparation.

Chapter 3

Related Work

3.1 Machine Learning and NLP Models to Predict Food Processing

A closely relevant study is the work by Nalin Arora et al [1] . "Machine Learning and Natural Lan- guage Processing Models to Predict the Extent of Food Processing." The authors highlight that the dramatic rise in ultra-processed food (UPF) con- sumption has been linked to numerous adverse health outcomes, making it essential to develop computational frameworks that can accurately as- sess the level of food processing. Their study inte- grates the FNDDS dataset of food items with nutri- ent profiles and NOVA processing levels to build a range of machine learning, deep learning, and NLP-based models. Beginning with a comprehen- sive 102-nutrient panel, they progressively coarse- grained the feature set to 65 and then 13 nutrients by removing flavonoids and ultimately aligning with the FDA-mandated nutrient list. Among tradi- tional models, the LightGBM Classifier performed best on the 102-nutrient dataset (F1-score: 0.9411, MCC: 0.8691), while Random Forest achieved the strongest results for the 65-nutrient version (F1- score: 0.9345, MCC: 0.8543). For the FDA 13- nutrient panel, Gradient Boost recorded the highest performance (F1-score: 0.9284, MCC: 0.8425). Notably, the authors also developed NLP-based models that demonstrated state-of-the-art predic- tive capabilities. In addition to identifying the most informative nutrients, the study introduced a user- friendly web server for processing-level prediction. This work provides a strong methodological foun- dation for research focused on food processing clas- sification and is directly relevant to the modeling approach used in the present project.

Chapter 4

Dataset Description

his study makes use of three primary datasets, each supporting a different component of the overall analysis. Together, they enable ingredient-level modeling, nutrient-based food processing predic- tion, and integration with NOVA classification la- bels.

4.1 Recipe Ingredients Dataset

The Recipe Ingredients Dataset from Kaggle [2] contains the ingredient lists for over 39,000 recipes across multiple cuisines. This dataset is used to:

- construct the ingredient co-occurrence graph,
- compute ingredient relevance scores,
- perform community detection and derive ingredient-driven recipe clusters.

4.2 NOVA Food Processing / FNDDS Dataset (CoSyLab)

The NOVA Food Processing dataset provided by CoSyLab, IIIT-Delhi [4] integrates nutrient profiles from the Food and Nutrient Database for Dietary Studies (FNDDS) with NOVA processing labels. This dataset is used to:

- train machine learning models for predicting food processing level,
- validate classification performance across multiple nutrient feature sets,
- map nutrient compositions to processing-level categories.

4.3 RecipeDB: Home-Cooked Recipe Text and Culinary Processes

4.3.1 RecipeDB: Home-Cooked Recipe Text and Culinary Processes

RecipeDB [3] is a comprehensive repository of structured, home-cooked and traditional recipes collected from diverse global cuisines. Each recipe entry includes standardized ingredient lists, preparation descriptions, cooking instructions, and culinary metadata.

In this project, RecipeDB is specifically used to:

- obtain recipe text from which culinary process terms (verbs and actions) are extracted,
- build a curated lexicon of cooking processes relevant to home-cooked food,
- generate semantic embeddings of processes for clustering via Word2Vec and PCA,
- study how different cooking actions group into different culinary categories.

4.4 Exploratory Data Analysis (EDA)

The following analysis summarizes the Exploratory Data Analysis (EDA) performed on the primary datasets utilized in this project: the NOVA Food Processing / FNDDS nutrient datasets and the RecipeDB general recipe dataset. The goal of the EDA is to understand the structure, completeness, distributions, and key characteristics of each dataset before model development.

EDA of NOVA–FNDDS Datasets (13, 65, 102 Nutrients)

The FNDDS datasets combine nutrient composition data with NOVA food processing labels. Three feature variants (13, 65, and 102 nutrients) were analyzed to understand how increasing feature dimensionality affects data quality and discriminative potential.

Key Observations

- **Dataset Size:** All three feature variants (13, 65, and 102 nutrients) contain the same number of food items; only the number of nutrient attributes differs.
- **Missing Values:**
  - The 13-feature set has very low missingness (typically $< 1\%$).
  - The 65- and 102-feature sets show moderate missingness in micronutrient features but remain manageable through median imputation.

- **NOVA Class Distribution:** The dataset is mildly imbalanced, with NOVA 3 or NOVA 4 categories slightly dominating.

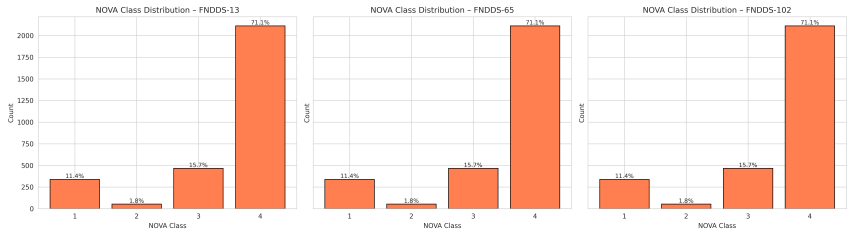

Figure 4.1: Nova class distribution of FNDDS dataset

- **Nutrient Variability:** Increasing the number of nutrient features captures more fine-grained micronutrient patterns. Nutrients such as sodium, added sugars, and saturated fat exhibit strong differences across NOVA classes, contributing to better class separation.

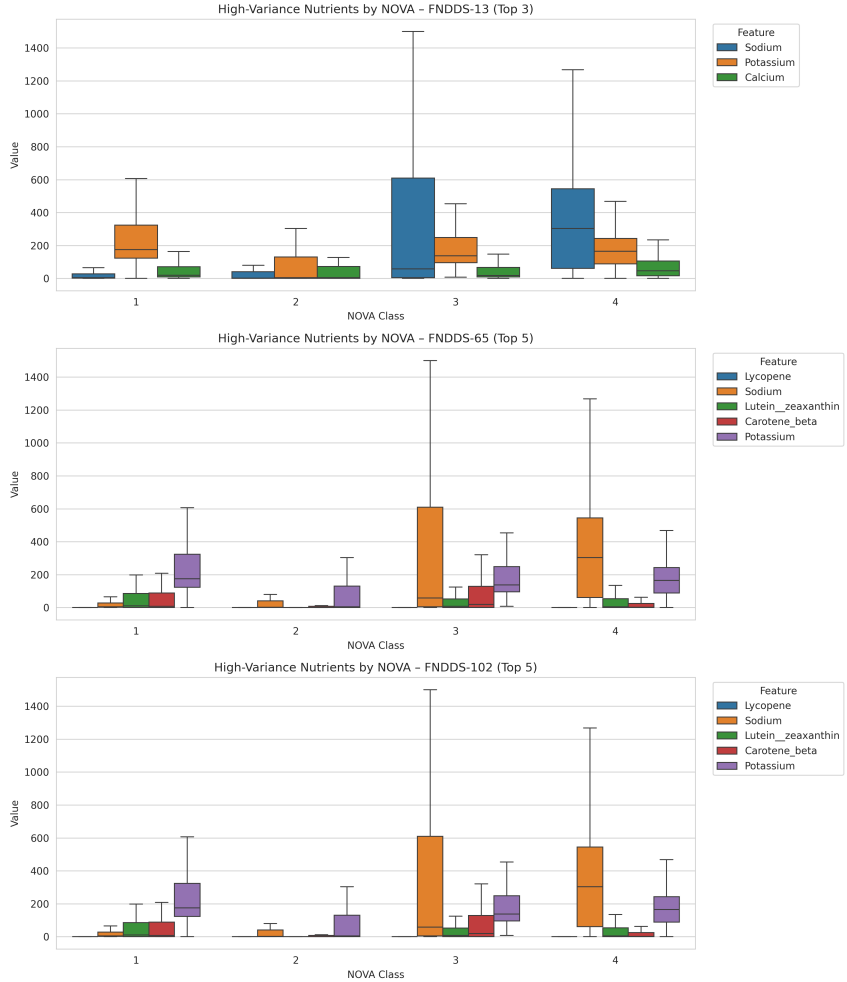

Figure 4.2: FNDDS boxplots high variance

- **Correlation Structure:**
  - In the 13-feature set, correlations primarily reflect coarse macronutrient relationships.

– In the 65- and 102-feature sets, richer correlation clusters emerge, indicating improved feature separability among processing categories.

EDA of RecipeDB – Home-Cooked Recipe Dataset

RecipeDB provides structured recipe texts including ingredient lists, cooking instructions, and associated metadata. This dataset is used for extracting culinary process verbs and understanding common cooking patterns.

Key Observations

- **Dataset Size:** The dataset contains a large and diverse collection of global recipes.
- **Ingredient Structure:**
  - Recipes typically contain between 8–10 ingredients on average.
  - The most frequent ingredients include common staples such as *salt*, *water*, *oil*, *onion*, and *garlic*.
- **Instruction Characteristics:** Cooking instructions vary widely in length, but most fall within a moderate word-count range.
- **Culinary Process Verbs:**
  - Frequent cooking actions include *mix*, *heat*, *fry*, *boil*, *bake*, and *sauté*.
  - These verbs form well-defined semantic clusters when embedded using Word2Vec and visualized through PCA projections.

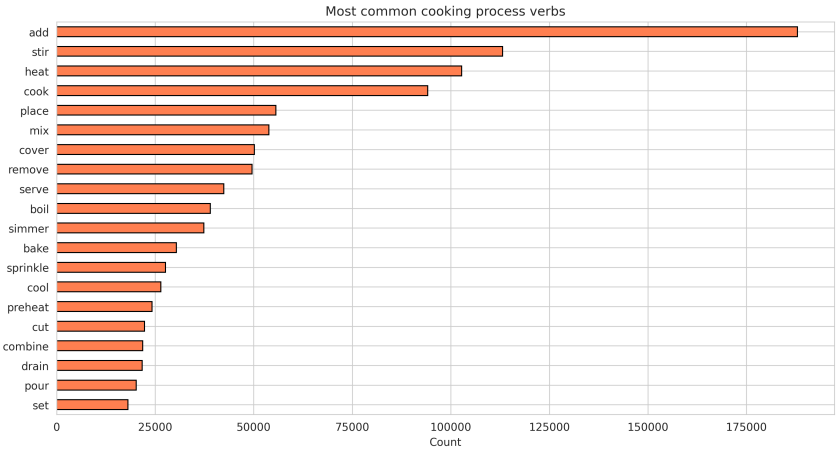

Figure 4.3: Top 20 verbs

Chapter 5

Recipe Ingredient Network Analysis

5.1 Dataset Overview

The original dataset contains:

- **Dataset shape:** 39,774 recipes, 3 columns
- **Total cuisines:** 20
- **Total unique ingredients:** 6,714

5.2 Ingredient Filtering

To reduce sparsity, ingredients appearing too rarely or too frequently were removed:

- **Filtering thresholds:** min_freq = 5, max_freq = 500
- **Ingredients after filtering:** 3,183 (from 6,703)
- **Recipes retained:** 39,023

5.3 Network Construction

A co-occurrence network was constructed where:

- Nodes represent recipes
- Edges represent ingredient-overlap between recipes
- **Overlap matrix size:** (39023×39023)

- **Non-zero entries:** 7,805,664
- **Final network:** 39,023 nodes, 72,231 edges

5.4 Network Statistics

- **Nodes:** 39,023
- **Edges:** 72,231
- **Average degree:** 3.70
- **Degree range:** 0–182
- **Average clustering coefficient:** 0.2099
- **Network density:** 9.5×10^{-5}

5.5 Degree Distribution

- Mean: 3.70
- Median: 0.00
- Std: 8.14
- 25th percentile: 0.00
- 75th percentile: 4.00
- 90th percentile: 11.00
- 95th percentile: 19.00
- 99th percentile: 40.00

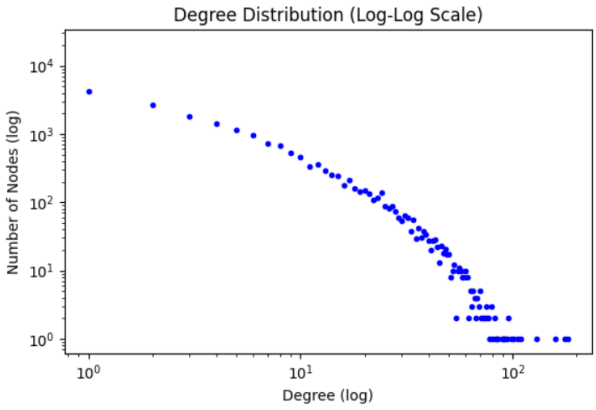

Figure 5.1: Degree Distribution log scale

5.6 Top 10 Most Connected Recipes

1. Recipe 29182 — 182 connections
2. Recipe 1667 — 175 connections
3. Recipe 26165 — 158 connections
4. Recipe 25996 — 130 connections
5. Recipe 41524 — 109 connections
6. Recipe 29544 — 105 connections
7. Recipe 2069 — 101 connections
8. Recipe 40636 — 99 connections
9. Recipe 20439 — 95 connections
10. Recipe 10437 — 95 connections

These represent the **network hubs**.

5.7 Clustering Coefficient Analysis

- Mean: 0.2099
- Median: 0.0000
- Std: 0.3468
- Max: 1.0000
- Min: 0.0000

The wide spread indicates strong local clustering for certain cuisine families.

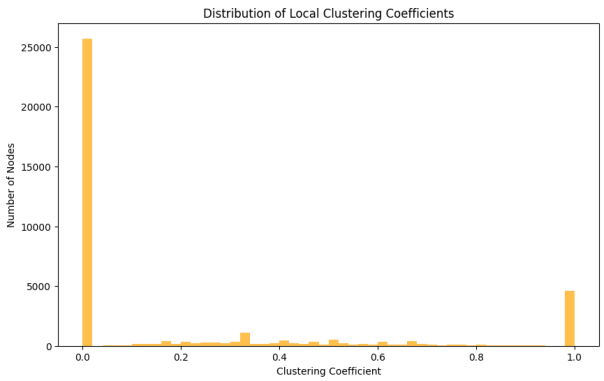

Figure 5.2: Distribution of Local Clustering Coefficients

5.8 Path-Length and Connectivity Analysis

The full graph is disconnected; therefore, analysis was performed on the largest connected component.

- **Largest component size:** 16,744 nodes
- **Average path length:** 6.35
- **Path length range:** 0 to 18
- **Most common path length:** 6

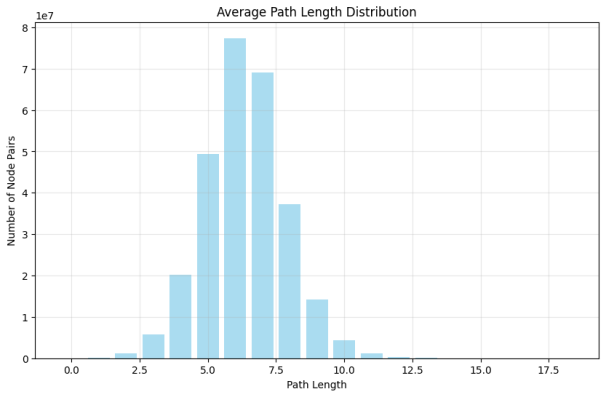

Figure 5.3: Average Path Length Distribution

5.9 Power-Law Analysis

Empirical degree distribution fit:

- Power-law slope (γ): -2.256
- $R^2 = 0.903$

Conclusion: The network strongly follows a **power-law distribution**, indicating scale-free properties.

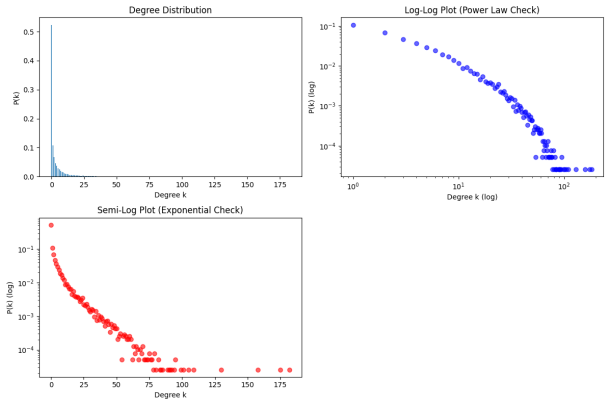

Figure 5.4

5.10 Small-World Property Analysis

- Average clustering: 0.4707
- Random graph clustering: 0.0005
- Observed path length: 6.35
- Random graph path length: 4.55
- **Small-world coefficient:** $\sigma = 665.78$

Result: The network displays **strong small-world characteristics**.

5.11 Top Hub Recipes

The top 20 hub recipes were identified along with cuisine type and example ingredients.
Top ranked HUB: Recipe 29182 (italian)

- **Connections:** 182
- **Total Ingredients:** 10
- **Sample Ingredients:** sliced ham, Pillsbury™ refrigerated crescent dinner rolls, roasted red peppers, provolone cheese, hot pepper

These hubs represent influential nodes that bridge diverse cuisines.

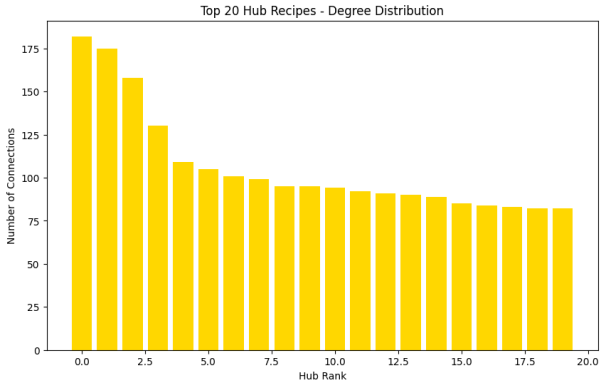

Figure 5.5: Number of connections vs Hub rank

5.12 Community Detection

- **Total communities detected:** 21,252
- Largest communities: sizes 1578, 1199, 1187, 955, 925, 910, 858, 836, 749, 691

Small communities of size 1 represent isolated or unique recipes.

5.13 Community Analysis (Top 10 Largest Communities)

For the largest communities:

- Purity ranges: 0.29 to 0.82
- Entropy ranges: 1.30 to 3.08
- Dominant cuisines: Mexican, Italian, Indian, Thai, Chinese, Moroccan

Example (Community 19283, 1578 recipes):

- Purity: 0.80
- Entropy: 1.33
- Dominant cuisine: Mexican
- Top cuisines: Mexican (1269), Southern US (92), Italian (77)

5.14 Cuisine Similarity (Jaccard Index)

Significant similarities:

- Italian and French: 0.583
- Southern US and Mexican: 0.556
- Italian and Mexican: 0.550
- Chinese and Thai: 0.488
- Thai and Vietnamese: 0.475

These capture shared culinary traditions and ingredient overlaps.

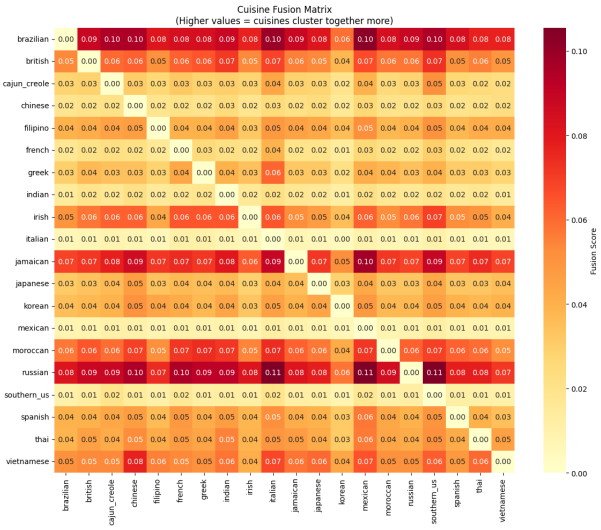

Figure 5.6: Cuisine Fusion Matrix

5.15 Cuisine Isolation Analysis

Isolation score ranges from 0 (highly mixed) to 1 (isolated).

- Most isolated cuisines: Mexican (0.769), Korean (0.640), Thai (0.634), Indian (0.624)
- Most mixed cuisines: Irish (0.128), Russian (0.165), Filipino (0.171)

Top Inter-Cuisine Link Pairs

- French and Italian: 2413 connections
- Thai and Vietnamese: 1176 connections
- Italian and Southern US: 944 connections
- Chinese and Thai: 904 connections

These pairs indicate strong cross-cultural culinary relationships.

Chapter 6

NOVA Class Prediction Using Nutrient Datasets

6.1 Overview of the Datasets

This study employs three structured nutrient datasets to predict the NOVA food processing classification of food items. Each dataset contains numerical nutrient profiles, along with a categorical target column, `nova_class`, representing the processing level according to the NOVA system (ranging from minimally processed foods to ultra-processed foods).

- **102-Nutrient Dataset**: A comprehensive feature set including macronutrients, micronutrients, amino acids, minerals, phytochemicals, and flavonoids (102 features total).
- **65-Nutrient Dataset**: A reduced representation obtained by removing phytochemicals/flavonoids while retaining essential macro- and micronutrient information.
- **FDA 13-Nutrient Dataset**: A minimal and highly interpretable nutrient panel following FDA labeling standards, consisting of only 13 nutrients.

All datasets share the same target variable, `nova_class`, enabling consistent modeling comparisons across different levels of feature granularity.

6.2 Columns and Structure

Each dataset includes:

- **Nutrient Features**: Numerical columns such as calories, proteins, fats, sugars, vitamins, minerals, and other nutritive compounds depending on the dataset's granularity.
- **Target Column**: `nova_class`, indicating the processing category.
- **Metadata Columns**: Standardized identifiers (e.g., food code, description).

6.3 UPF Identification Using Nutrient Profiles

Ultra-Processed Foods (UPFs) exhibit distinct nutrient patterns—typically higher levels of added sugars, unhealthy fats, sodium, preservatives, and altered nutrient ratios arising from industrial processing. By training supervised learning models on these nutrient patterns, the system can classify foods into their likely NOVA categories. As a result, even when the ingredient list is not available (as is common for homemade or crowdsourced foods), nutrient profiles alone can serve as a reliable predictor of whether an item is ultra-processed.

6.4 Model Training and Cross-Validation Strategy

Twelve supervised machine learning models were evaluated: Logistic Regression, SVM, KNN, Decision Tree, Random Forest, Extra Trees, Gradient Boosting, LightGBM, XGBoost, MLP, Naive Bayes, and AdaBoost.

Each model was trained using three cross-validation strategies:

1. **Stratified K-Fold**: Ensures class proportion consistency in each fold.
2. **SMOTE K-Fold**: Performs oversampling within each fold to reduce class imbalance.
3. **SMOTE + Stratified K-Fold**: Combines both benefits for balanced and structured evaluation.

For each model, the best-performing CV variant was selected using the **Matthews Correlation Coefficient (MCC)**, as it is particularly suitable for imbalanced multi-class problems.

Chapter 7

Model Architecture Summary

This section summarizes the core architectural configurations used for the twelve machine learning models trained on the nutrient datasets. Only the key structural hyperparameters used in this study are listed.

LightGBM

- Gradient Boosting Decision Trees (GBDT)
- 1000 estimators, learning rate = 0.02
- Max depth = 10, num leaves = 64
- Feature fraction = 0.8, bagging fraction = 0.8
- L1 and L2 regularization
- Multiclass objective

XGBoost

- Boosted tree classifier (multi:softmax)
- 800 estimators, learning rate = 0.03
- Max depth = 9, min child weight = 2
- Subsample = 0.8, colsample_bytree = 0.8
- Gamma = 0.2, L1/L2 regularization

Random Forest

- 500 trees
- Max depth = 18, min samples split = 3
- Max features = `sqrt`
- Class weight = balanced
- Bootstrap enabled

Gradient Boosting

- 500 boosting rounds, learning rate = 0.05
- Max depth = 6
- Min samples split = 4, leaf = 2
- Subsample = 0.8, max features = `sqrt`

Extra Trees

- 500 randomized trees
- Max depth = None
- Min samples split = 4, leaf = 2
- Max features = `sqrt`

K-Nearest Neighbors

- $k = 5$
- Distance-based weighting
- Metric: Minkowski ($p = 2$)

Multi-Layer Perceptron (MLP)

- Hidden layers: (128, 64)
- Activation: ReLU
- Optimizer: Adam
- L2 regularization ($\alpha = 0.0001$)

- Learning rate: adaptive
- Max iterations: 500, early stopping enabled

Decision Tree

- Criterion: gini
- Max depth: None
- Min samples split = 4, leaf = 2

Support Vector Machine (SVM)

- Kernel: RBF
- $C = 1.0$, gamma = scale
- Probability output enabled

AdaBoost

- 500 boosting rounds
- Learning rate = 0.05
- Base estimator: Decision Tree (depth = 2)

Logistic Regression

- Penalty: L2
- Solver: lbfgs
- Max iterations: 1000

Naïve Bayes

- Gaussian Naïve Bayes
- var_smoothing = 10^{-9}

7.1 Best Model Results Across Datasets

For each dataset, the following tables report the highest performance obtained by each model across all CV variants, based on the MCC metric. These results provide a comparative understanding of how feature granularity affects classification performance.

102-Nutrient Dataset

Table 7.1: Top performance metrics on 102-nutrient dataset.

Model	Best Accuracy	Best F1	Best MCC
AdaBoost	0.8354	0.8235	0.6178
Decision Tree	0.8593	0.8613	0.7004
Extra Trees	0.9094	0.9066	0.7976
Gradient Boosting	0.9263	0.9243	0.8350
KNN	0.8700	0.8682	0.7103
LightGBM	0.9320	0.9308	0.8488
Logistic Regression	0.8387	0.8273	0.6242
MLP	0.8485	0.8544	0.6946
Naive Bayes	0.5495	0.6173	0.3748
Random Forest	0.9182	0.9163	0.8182
SVM	0.7522	0.6892	0.4203
XGBoost	0.9256	0.9238	0.8338

65-Nutrient Dataset

Table 7.2: Top performance metrics on 65-nutrient dataset.

Model	Best Accuracy	Best F1	Best MCC
AdaBoost	0.8350	0.8240	0.6137
Decision Tree	0.8566	0.8582	0.6938
Extra Trees	0.9131	0.9099	0.8041
Gradient Boosting	0.9263	0.9242	0.8345
KNN	0.8747	0.8729	0.7209
LightGBM	0.9286	0.9272	0.8414
Logistic Regression	0.8246	0.8098	0.5885
MLP	0.8582	0.8632	0.7127
Naive Bayes	0.5253	0.5810	0.3559
Random Forest	0.9182	0.9182	0.8227
SVM	0.7522	0.6894	0.4212
XGBoost	0.9276	0.9259	0.8385

FDA 13-Nutrient Dataset

Table 7.3: Top performance metrics FDA 13-nutrient dataset.

Model	Best Accuracy	Best F1	Best MCC
AdaBoost	0.8364	0.8266	0.6174
Decision Tree	0.8704	0.8716	0.7215
Extra Trees	0.9118	0.9095	0.8053
Gradient Boosting	0.9175	0.9154	0.8150
KNN	0.8734	0.8711	0.7177
LightGBM	0.9242	0.9228	0.8312
Logistic Regression	0.7956	0.7629	0.4950
MLP	0.8010	0.8151	0.6273
Naive Bayes	0.5663	0.6075	0.3462
Random Forest	0.9165	0.9144	0.8137
SVM	0.7822	0.7293	0.4709
XGBoost	0.9189	0.9170	0.8187

7.2 Best Performing Model

To identify the most effective classifier for predicting NOVA processing levels, model performance was compared across all three nutrient datasets using **Accuracy** as the primary evaluation metric. Accuracy is a widely used measure for overall correctness in multi-class classification, making it suitable for assessing general model performance across the diverse feature sets.

Across the 102-nutrient, 65-nutrient, and FDA 13-nutrient datasets, **LightGBM consistently achieved the highest accuracy values**, making it the best-performing model overall. The best accuracies obtained by LightGBM on each dataset are:

- **102-Nutrient Dataset:** Accuracy = 0.9319
- **65-Nutrient Dataset:** Accuracy = 0.9286
- **FDA 13-Nutrient Dataset:** Accuracy = 0.9242

Chapter 8

Home-Cooked Food Processing Techniques Clustering

This section describes the methodology used to extract, represent, and cluster cooking processes from home-cooked recipe text. Two datasets were used in this pipeline: (1) the RecipeDB steps dataset containing free-text cooking instructions, and (2) the RecipeDB processes dataset containing extracted culinary process tokens. The objective is to build meaningful clusters of food-processing actions (e.g., boil, saute, grill, knead) using Word2Vec embeddings and multiple clustering algorithms.

8.1 Datasets Used

Two datasets from RecipeDB were used for process extraction, embedding, and clustering. Both datasets provide complementary information: one supplies structured culinary process labels, while the other provides free-text cooking instructions used to train the Word2Vec model.

8.1.1 RecipeDB Step Text Dataset

This dataset contains the complete free-text cooking instructions for each recipe. It is used to train the Word2Vec model by providing natural language context for cooking operations.

The dataset contains the following columns:

Identifiers: recipe_id, recipe_title

Geography: continent, region, subregion, subsubregion, subsubsubregion

Ingredients: ingredients, ingredient_list

Time: prep_time, cook_time, servings, source

Instruction text: steps, description

Metadata: author, recipe_url, image_url

Diet labels: vegan, pescetarian, ovo_vegetarian, lacto_vegetarian, ovo_lacto_vegetarian, non_vegetarian

Nutrients: Protein, Total lipid (fat), Energy, Carbohydrate (by difference)

This dataset contains free-text `steps` fields describing how home-cooked recipes are prepared. It provides rich linguistic context for modeling cooking verbs and actions. The dataset is used for:

- training the Word2Vec model using natural recipe instructions,
- capturing contextual co-occurrence patterns between culinary operations,
- providing semantic grounding for embedding process terms.

8.1.2 RecipeDB Processes Dataset

This dataset contains structured metadata for each recipe, including a curated `Processes` column from which all candidate food-processing terms were extracted.

The dataset contains the following columns:

Identifiers: Recipe_id, Recipe_title, url, img_url

Time Attributes: cook_time, prep_time, total_time, servings

Geographical Info: Region, Sub_region, Continent

Nutritional Values: Carbohydrate (g), Energy (kcal), Protein (g), Total lipid (fat) (g)

Metadata: Calories, Source, Utensils

Process Information: Processes (used to extract process tokens)

Diet Labels: vegan, pescetarian, ovo_vegetarian, lacto_vegetarian, ovo_lacto_vegetarian

This dataset contains a curated `Processes` column with multiple food-processing actions per recipe, encoded using the "——" separator. It is used to:

- extract candidate process tokens,
- clean, normalize, and deduplicate them,
- generate the final list of unique process terms for clustering.

Table 8.1: Top processes ranked by frequency of occurrence in the full recipe corpus.

Process	**Frequency**
add	187,921
heat	102,759
cook	94,106
stir	83,544
place	55,651
mix	53,863
cover	50,143
remove	49,561
serve	42,398
boil	39,022

These high-frequency processes reflect fundamental kitchen operations, which also explains their tendency to cluster into broader "basic cooking action" groups in our embedding-based clustering pipeline.

Table 8.2: Top 10 Recipes With the Highest Number of Processes

Recipe Title	**Process Count**
5-Cheese Crab Lasagna	121
Murgh Makhani (Butter Chicken)	90
Italian Rum Cake	84
Vegetable Lasanga	83
Cod Brandade	82
Shell's Potato Soup With Carrots	81
German Chocolate Cheesecake	75
Cuban Opera Cake	73
Boston Creme Doughnuts	73
Irish Cream Choco Mousse Cake	73

8.2 Process Extraction and Cleaning

A multi-stage cleaning and normalization pipeline was used:

1. **Token extraction**: process strings were split using the "——" delimiter.
2. **Lowercasing and punctuation removal**: each token was standardized to a clean form.
3. **Validation**: only alphabetic, hyphenated, or multi-word process expressions were retained.

4. **Stopword filtering**: generic words ("and", "with", etc.) were removed.
5. **Final list**: a sorted set of unique, validated process tokens was produced.

This list of unique processes is saved for inspection and later embedding.

8.3 Word2Vec Embedding Model

To represent cooking processes numerically, a Word2Vec [10] Skip-Gram model was trained.

8.3.1 Training Corpus

Two sources were combined:

- Recipe step instructions as natural language sentences.
- Each unique process token treated as a standalone "sentence" to ensure proper representation.

8.3.2 Word2Vec Architecture

The embedding model was trained using the following configuration:

- Architecture: Skip-Gram (SG = 1)
- Vector size: 100 dimensions
- Window size: 8
- Minimum token count: 1
- Epochs: 10
- Workers: 4 threads

For multi-word process expressions, embeddings were computed as the mean of individual token vectors.

8.4 Clustering Algorithms

Four unsupervised clustering methods were applied to the process embeddings:

8.4.1 K-Means Clustering

- Number of clusters: 8
- Multiple initializations (n_init = 20)

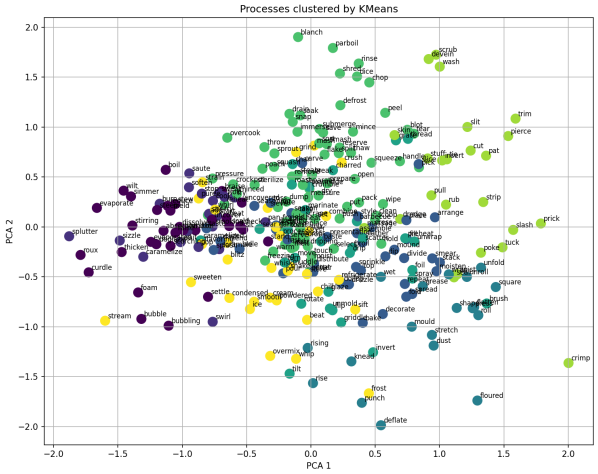

Figure 8.1: KMeans Clustering

8.4.2 Agglomerative / MiniBatch K-Means

- Ward linkage, used when the number of processes is below a threshold.
- If number of processes exceeds 2000, MiniBatch K-Means is used as a memory-efficient alternative.

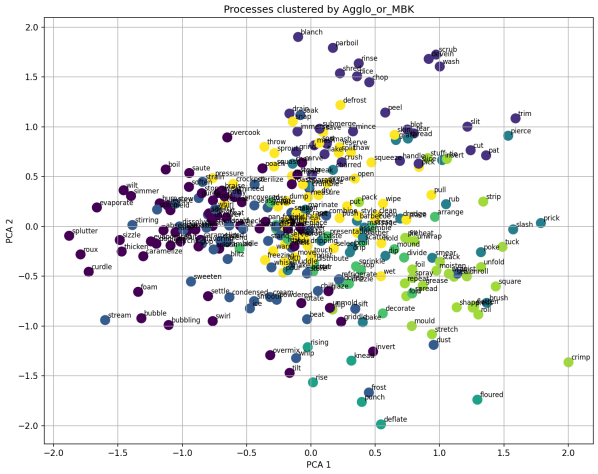

Figure 8.2: Agglo Clustering

8.4.3 DBSCAN

- eps = 0.7
- min_samples = 2
- Detects dense clusters and flags noise points.

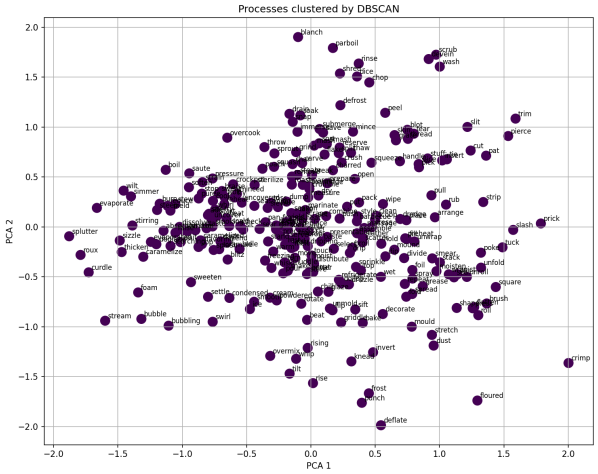

Figure 8.3: DBSCAN Clustering

8.4.4 Gaussian Mixture Models (GMM)

- Number of components: 8
- Provides soft probabilistic clustering.

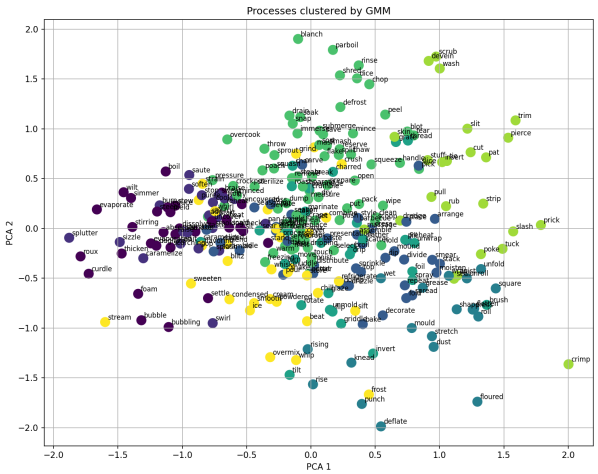

Figure 8.4: GMM Clustering

All embeddings and cluster assignments were stored in `unique_processes_clusters.csv`.

8.5 Visualization and Cluster Summaries

To visualize cluster structure:

- PCA was applied to reduce 100-dimensional embeddings to 2D.
- Scatter plots were generated for each clustering method.
- Optional annotation displays process labels directly on the plots.

Additionally, human-readable summaries were generated for each method, listing:

- cluster number,
- number of processes in each cluster,
- sample processes from each group.

The clustered protocols are highlighted at the end of the report.

8.6 Clustering Evaluation Metrics

To assess the quality of clustering across different algorithms, three standard internal validation metrics were computed: Silhouette Score, Calinski–Harabasz Index, and Davies–Bouldin Score. DBSCAN was excluded from metric computation due to producing an insufficient number of clusters.

Table 8.3: Vertical comparison of clustering performance metrics.

Metric	Value
KMeans	
Silhouette Score	0.357
Calinski–Harabasz Index	7.48
Davies–Bouldin Score	3.3103
Agglomerative / MBK	
Silhouette Score	0.216
Calinski–Harabasz Index	6.91
Davies–Bouldin Score	3.4477
GMM	
Silhouette Score	0.341
Calinski–Harabasz Index	7.45
Davies–Bouldin Score	3.3184
DBSCAN	
Silhouette Score	–
Calinski–Harabasz Index	–
Davies–Bouldin Score	–

8.7 Observations from Clustering Evaluation

The internal clustering metrics indicate that **KMeans** provides the best overall performance among the tested algorithms. It achieves the highest Silhouette score (0.0357) and the highest Calinski–Harabasz index (7.48), along with a comparatively lower Davies–Bouldin score, suggesting more compact and better-separated clusters.

GMM performs similarly to KMeans, with only marginally lower scores across all metrics, indicating that both methods capture comparable structure within the cooking process embeddings.

Agglomerative (or MiniBatchKMeans substitute) shows the weakest performance, with the lowest Silhouette and Calinski–Harabasz scores and the highest Davies–Bouldin value, reflecting overlapping and poorly separated clusters.

DBSCAN fails to identify meaningful cluster structure, placing most points into the noise class, likely due to the lack of strong density variations in the high-dimensional embedding space.

Overall, the results suggest that cooking process verbs form a continuous semantic space, making **KMeans** the most suitable clustering method for this dataset, with GMM serving as a secondary alternative.

Chapter 9

Processing Intensity Scoring Methodology

To quantify the extent of food processing represented by each culinary process term, we compute a unified **Processing Intensity Score (PIS)** for all 270 unique cooking processes. This score integrates linguistic complexity, embedding–space dispersion, and global semantic deviation using three independent metrics: (1) Word Complexity Score, (2) Embedding Dispersion Score, and (3) Cluster Transition Difficulty Score. All computations follow the Python code described in Appendix X.

9.1 Input Data

Two data sources are used:

- **unique_processes_raw.txt**: a plain-text list of 270 cleaned and standardized process terms.
- **unique_processes_clusters.csv**: Word2Vec embeddings (100-dimensional) and cluster labels produced using KMeans, Agglomerative/MBKMeans, DBSCAN, and GMM.

All analyses in this section utilize the **KMeans cluster assignments**.

9.2 Word Complexity Score (WCS)

For each process term p, a linguistic complexity score is computed based on token length, morphological suffixes, and lexical rarity. If a process contains tokens $t_1, t_2, \ldots, t_n$, we compute:

$$\mathbf{WCS}(p) = 0.4 \sum_{i=1}^{n} len(t_i) + 0.3n + 0.3 \left(\sum_{i=1}^{n} \frac{1}{1 + freq(t_i)} + suffix\_count(p) \right)$$

Suffixes such as *-ize*, *-ate*, *-ifying*, *-ation* increment the score. This captures linguistic and morphological indicators of technical or industrial complexity.

9.3 Embedding Dispersion Score (EDS)

Embedding Dispersion uses the 100-dimensional Word2Vec vector $\mathbf{e}_i$ of each process.

$$EDS_i = \|\mathbf{e}_i\|_2 \; + \; \mathrm{Var}(\mathbf{e}_i)$$

This captures both vector magnitude (semantic intensity) and within-vector variance (semantic spread).

9.4 Cluster Transition Difficulty Score (CTDS)

Let $\mathbf{g}$ be the global embedding centroid:

$$\mathbf{g} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{e}_i$$

CTDS measures how far a process lies from the center of the embedding space:

$$CTDS_i = \|\mathbf{e}_i - \mathbf{g}\|_2$$

Higher values indicate that the process is semantically specialized, unusual, or far from the "average" cooking action.

9.5 Normalization

Each metric $M \in \{WCS, EDS, CTDS\}$ is min–max normalized:

$$M_i^* = \frac{M_i - \min(M)}{\max(M) - \min(M)}$$

ensuring that all three metrics lie in the range $[0, 1]$.

9.6 Final Processing Intensity Score (PIS)

The final unified score is a weighted combination:

$$PIS_i = 0.33 \cdot WCS_i^* \; + \; 0.33 \cdot EDS_i^* \; + \; 0.34 \cdot CTDS_i^*$$

The weights (0.33/0.33/0.34) reflect equal importance of complexity and dispersion, with a slight emphasis on global semantic deviation.

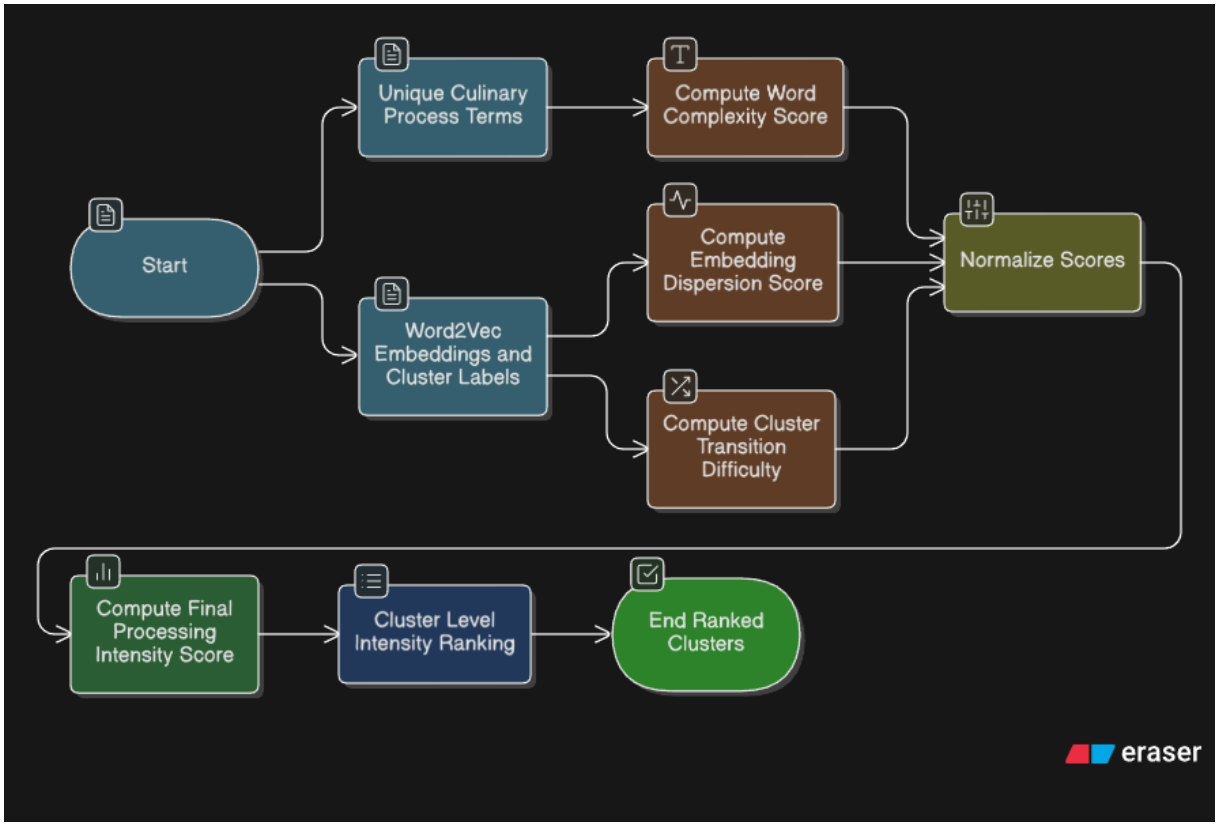

Figure 9.1: Architecture

9.7 Cluster-Level Intensity Ranking

For KMeans cluster k, the cluster-level intensity is defined as:

$$C_k = \frac{1}{|k|} \sum_{i \in k} PIS_i$$

This produces an interpretable ranking that indicates which clusters represent more complex, more industrial, or semantically heavier processes.

9.8 Processing Intensity Scoring Results

The proposed Processing Intensity Score (PIS) was computed for all 270 culinary processes using the weighted combination of Word Complexity Score (WCS), Embedding Dispersion Score (EDS), and Cluster Transition Difficulty Score (CTDS). Table 9.1 presents the top five most intense processes.

Table 9.1: Top five processes ranked by Processing Intensity Score (PIS).

Process	**Cluster**	**WCS**	**EDS**	**CTDS**	**PIS**
splutter	1	0.513	0.945	0.972	0.812
deflate	3	0.410	0.903	0.922	0.747
stream	7	0.321	0.910	0.988	0.742
crimp	6	0.218	1.000	1.000	0.742
floured	3	0.423	0.846	0.883	0.719

9.9 Cluster-Level Processing Intensity

The average PIS for each KMeans cluster is reported in Table 9.2. Higher values indicate clusters consisting of more complex or effort-intensive cooking processes.

Table 9.2: Cluster-level mean Processing Intensity Score (PIS).

Cluster	**Mean PIS**
3	0.4575
4	0.4446
6	0.4359
7	0.4047
0	0.4044
1	0.3854
2	0.3392
5	0.3276

Cluster 3 is identified as the most processing-intensive group, while Cluster 5 represents the least intensive group.

Bibliography

[1] Arora, N., Bhagat, S., & Dhama, R. (2024). *Machine Learning and Natural Language Processing Models to Predict the Extent of Food Processing*. arXiv:2412.17217. Available at: `https://arxiv.org/pdf/2412.17217v1`

[2] Kaggle (2017). *Recipe Ingredients Dataset*. Available at: `https://www.kaggle.com/datasets/kaggle/recipe-ingredients-dataset` (Accessed: February 2025).

[3] CoSyLab, IIIT-Delhi (2020). *RecipeDB: A structured compendium of world recipes and their ingredients*. Available at: `https://cosylab.iiitd.edu.in/recipedb/` (Accessed: February 2025).

[4] USDA Agricultural Research Service (2009). *Food and Nutrient Database for Dietary Studies (FNDDS) 2009–2010*. Available at: `https://www.ars.usda.gov/.../fndds-download-databases/` (Accessed: February 2025).

[5] da Costa Louzada, M. L. & Gabe, K. T. (2025). *Nova food classification system: a contribution from Brazilian epidemiology*. Revista Brasileira de Epidemiologia, 28, e250027. Available at: `https://pmc.ncbi.nlm.nih.gov/articles/PMC12129239/`.

[6] Lloyd, S. (1982). *Least squares quantization in PCM*. IEEE Transactions on Information Theory, 28(2).

[7] Ward, J. H. (1963). *Hierarchical grouping to optimize an objective function*. Journal of the American Statistical Association, 58(301). (Agglomerative Clustering)

[8] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). *A density-based algorithm for discovering clusters in large spatial databases*. Proceedings of the 2nd ACM SIGKDD Conference. (DBSCAN)

[9] McLachlan, G., & Peel, D. (2000). *Finite Mixture Models*. Wiley. (Gaussian Mixture Models)

[10] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. arXiv:1301.3781. Available at: `https://arxiv.org/abs/1301.3781`

[11] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression*. Wiley.

[12] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

[13] Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5–32.

[14] Geurts, P., Ernst, D., & Wehenkel, L. (2006). *Extremely randomized trees*. Machine Learning, 63(1).

[15] Friedman, J. H. (2001). *Greedy function approximation: A gradient boosting machine*. Annals of Statistics, 29(5).

[16] Freund, Y., & Schapire, R. (1997). *A decision-theoretic generalization of on-line learning*. Journal of Computer and System Sciences.

[17] Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. KDD Conference. Available at: `https://arxiv.org/abs/1603.02754`

[18] Ke, G. et al. (2017). *LightGBM: A highly efficient gradient boosting decision tree*. NeurIPS. Available at: `https://papers.nips.cc/.../Paper.pdf`

[19] Cortes, C., & Vapnik, V. (1995). *Support-vector networks*. Machine Learning, 20(3).

[20] Cover, T., & Hart, P. (1967). *Nearest neighbor pattern classification*. IEEE Transactions on Information Theory.

[21] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

[22] Haykin, S. (2008). *Neural Networks and Learning Machines*. Prentice Hall.

[23] Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation*. IJCAI.

[24] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

[25] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.

Clustered Protocols

=== KMeans ===

Cluster 0:
absorbed, boil, braise, bubble, bubbling, check, curdle, deglaze, dissolve, evaporate, evaporated, foam, heat, meld, raise, reduce, roux, scald, scorching, settle, simmer, sit, skim, stand, steep, stew, stir, stirring, thicken, uncover, uncovered, wilt

Cluster 1:
burn, caramelize, cook, fry, pan-fry, pop, saute, scramble, sizzle, smoke, smoking, soften, splutter, start, stir-fry, stop, sweat, swirl, wait, wok

Cluster 2:
arrange, assemble, bake, carve, coat, dash, decorate, dip, distribute, divide, dollop, dress, dressing, drizzle, fill, fold, garnish, ladle, moisten, mound, place, pour, presentation, repeat, scatter, season, seasoning, serve, slice, smear, splash, spread, sprinkle, stack, style, taste, top, toss

Cluster 3:
deflate, dust, flatten, floured, grease, knead, mould, press, punch, rise, rising, roll, shape, square, stretch, unfold, unroll, wet, wrap

Cluster 4:
barbecue, baste, broil, brush, caramelized, char, charred, drip, flip, foil, glaze, grate, griddle, invert, marinade, marinate, move, preheat, roast, rotate, sear, slather, spray, thread, tilt, toast, turn, unmold, unwrap

Cluster 5:
add, blanch, blot, break, butter, chop, clean, cool, cover, crockpot, crumble, defrost, dice, drain, dripping, drop, dry, dump, flake, freezing, handle, hold, immerse, lard, measure, melt, mince, minced, moist, note, open, overcook, pack, parboil, peel, pick, poach, pre-heat, prepare, pressure, push, put, reheat, remove, replace, reserve, rest, rinse, save, scoop, scrape, select, separate, set, shred, smash, snap, soak, sort, split, sprout, squash, squeeze, steam, sterilize, submerge, take, tear, test, thaw, throw, touch, transfer, warm, wipe, yield

Cluster 6:
crimp, cut, devein, dredge, insert, massage, pat, pierce, poke, prick, pull, rub, scrub, seal, skin, slash, slit, strip, stuff, tie, trim, tuck, twist, wash

Cluster 7:
beat, blend, blitz, chill, combine, condensed, cream, crush, flavoring, frost, grind, ice, mash, mix, muddle, overmix, powdered, process, puree, refrigerate, shake, sieve, sift, smooth, store, strain, stream, sweeten, whip, whirl, whisk, zest

=== Agglomerative / MBK ===

Cluster 0:
absorbed, boil, braise, bubble, bubbling, burn, caramelize, caramelized, carve, check, cook, cover, curdle, deglaze, evaporate, evaporated, foam, fry, glaze, griddle, heat, invert, meld, overcook, overmix, pan-fry, poach, pop, pre-heat, raise, reduce, replace, roast, rotate, roux, saute, scorching, scramble, sear, settle, simmer, sizzle, skim, smoke, smoking, soften, splutter, start, steam, stir-fry, stop, sweat, swirl, test, thicken, tilt, toast, uncover, uncovered, unmold, wait, wilt, wok

Cluster 1:
blanch, blot, break, chop, crumble, crush, cut, devein, dice, drain, flake, grind, handle, immerse, mash, mince, minced, parboil, pat, peel, rinse, scrub, shred, slice, slit, smash, submerge, tear, trim, wash

Cluster 2:
add, beat, blend, blitz, butter, chill, combine, condensed, cream, dash, dissolve, dust, flavoring, frost, ice, lard, melt, mix, muddle, powdered, process, puree, refrigerate, scald, season, seasoning, sieve, sift, sit, smooth, soak, splash, stand, steep, sterilize, stew, stir, stirring, store, strain, stream, sweeten, taste, twist, whip, whirl, whisk, zest

Cluster 3:
bake, barbecue, baste, broil, brush, char, charred, dip, dredge, drip, grate, insert, marinade, marinate, massage, pierce, poke, prick, rub, shake, slash, slather, smear, thread

Cluster 4:
deflate, floured, knead, punch, rise, rising

Cluster 5:
arrange, coat, decorate, distribute, divide, dollop, dress, dressing, drizzle, garnish, ladle, mound, presentation, scatter, serve, spread, sprinkle, top, toss

Cluster 6:
crimp, fill, flatten, flip, foil, fold, grease, moisten, mould, pack, preheat, press, repeat, roll, seal, shape, spray, square, stack, stretch, strip, stuff, tie, tuck, unfold, unroll, unwrap, wrap

Cluster 7:
assemble, clean, cool, crockpot, defrost, dripping, drop, dry, dump, freezing, hold, measure, moist, move, note, open, pick, place, pour, prepare, pressure, pull, push, put, reheat, remove, reserve, rest, save, scoop, scrape, select, separate, set, skin, snap, sort, split, sprout, squash, squeeze, style, take, thaw, throw, touch, transfer, turn, warm, wet, wipe, yield

=== DBSCAN ===

Cluster -1:
absorbed, add, arrange, assemble, bake, barbecue, baste, beat, blanch, blend, blitz, blot, boil, braise, break, broil, brush, bubble, bubbling, burn, butter, caramelize, caramelized, carve, char, charred, check, chill, chop, clean, coat, combine, condensed, cook, cool, cover, cream, crimp, crockpot, crumble, crush, curdle, cut, dash, decorate, deflate, defrost, deglaze, devein, dice, dip, dissolve, distribute, divide, dollop, drain, dredge, dress, dressing, drip, dripping, drizzle, drop, dry, dump, dust, evaporate, evaporated, fill, flake, flatten, flavoring, flip, floured, foam, foil, fold, freezing, frost, fry, garnish, glaze, grate, grease, griddle, grind, handle, heat, hold, ice, immerse, insert, invert, knead, ladle, lard, marinade, marinate, mash, massage, measure, meld, melt, mince, minced, mix, moist, moisten, mould, mound, move, muddle, note, open, overcook, overmix, pack, pan-fry, parboil, pat, peel, pick, pierce, place, poach, poke, pop, pour, powdered, pre-heat, preheat, prepare, presentation, press, pressure, prick, process, pull, punch, puree, push, put, raise, reduce, refrigerate, reheat, remove, repeat, replace, reserve, rest, rinse, rise, rising, roast, roll, rotate, roux, rub, saute, save, scald, scatter, scoop, scorching, scramble, scrape, scrub, seal, sear, season, seasoning, select, separate, serve, set, settle, shake, shape, shred, sieve, sift, simmer, sit, sizzle, skim, skin, slash, slather, slice, slit, smash, smear, smoke, smoking, smooth, snap, soak, soften, sort ...

=== GMM ===

Cluster 0:
absorbed, boil, braise, bubble, bubbling, check, curdle, deglaze, dissolve, evaporate, evaporated, foam, heat, meld, raise, reduce, roux, scald, scorching, settle, simmer, sit, skim, stand, steep, stew, stirring, thicken, uncover, uncovered, wilt

Cluster 1:
burn, caramelize, cook, fry, pan-fry, pop, saute, scramble, sizzle, smoke, smoking, soften, splutter, start, stir, stir-fry, stop, sweat, swirl, wait, wok

Cluster 2:
arrange, assemble, bake, carve, coat, dash, decorate, dip, distribute, divide, dollop, dress, dressing, drizzle, fill, fold, garnish, ladle, moisten, mound, place, pour, presentation, repeat, scatter, season, seasoning, serve, slice, smear, splash, spread, sprinkle, stack, style, taste, top, toss

Cluster 3:
deflate, dust, flatten, floured, grease, knead, mould, press, punch, rise, rising, roll, shape, square, stretch, unfold, unroll, wet, wrap

Cluster 4:
barbecue, baste, broil, brush, caramelized, char, charred, drip, flip, foil, glaze, grate, griddle, invert, marinade, marinate, move, preheat, roast, rotate, sear, slather, spray, thread, tilt, toast, turn, unmold, unwrap

Cluster 5:
add, blanch, blot, break, butter, chop, clean, cool, cover, crockpot, crumble, defrost, dice, drain, dripping, drop, dry, dump, flake, freezing, handle, hold, immerse, lard, measure, melt, mince, minced, moist, note, open, overcook, pack, parboil, peel, pick, poach, pre-heat, prepare, pressure, push, put, reheat, remove, replace, reserve, rest, rinse, save, scoop, scrape, select, separate, set, shred, smash, snap, soak, sort, split, sprout, squash, squeeze, steam, sterilize, submerge, take, tear, test, thaw, throw, touch, transfer, warm, wipe, yield

Cluster 6:
crimp, cut, devein, dredge, insert, massage, pat, pierce, poke, prick, pull, rub, scrub, seal, skin, slash, slit, strip, stuff, tie, trim, tuck, twist, wash

Cluster 7:
beat, blend, blitz, chill, combine, condensed, cream, crush, flavoring, frost, grind, ice, mash, mix, muddle, overmix, powdered, process, puree, refrigerate, shake, sieve, sift, smooth, store, strain, stream, sweeten, whip, whirl, whisk, zest

Unique Cooking and Food-Processing Processes

absorbed add arrange assemble bake barbecue baste beat
blanch blend blitz blot boil braise break broil
brush bubble bubbling burn butter caramelize caramelized carve
char charred check chill chop clean coat combine
condensed cook cool cover cream crimp crockpot crumble
crush curdle cut dash decorate deflate defrost deglaze
devein dice dip dissolve distribute divide dollop drain
dredge dress dressing drip dripping drizzle drop dry
dump dust evaporate evaporated fill flake flatten flavoring
flip floured foam foil fold freezing frost fry
garnish glaze grate grease griddle grind handle heat
hold ice immerse insert invert knead ladle lard
marinade marinate mash massage measure meld melt mince
minced mix moist moisten mould mound move muddle
note open overcook overmix pack pan-fry parboil pat
peel pick pierce place poach poke pop pour
powdered pre-heat preheat prepare presentation press pressure prick
process pull punch puree push put raise reduce
refrigerate reheat remove repeat replace reserve rest rinse
rise rising roast roll rotate roux rub saute
save scald scatter scoop scorching scramble scrape scrub
seal sear season seasoning select separate serve set
settle shake shape shred sieve sift simmer sit
sizzle skim skin slash slather slice slit smash
smear smoke smoking smooth snap soak soften sort
splash split splutter spray spread sprinkle sprout square
squash squeeze stack stand start steam steep sterilize
stew stir stir-fry stirring stop store strain stream
stretch strip stuff style submerge sweat sweeten swirl
take taste tear test thaw thicken thread throw
tie tilt toast top toss touch transfer trim
tuck turn twist uncover uncovered unfold unmold unroll
unwrap wait warm wash wet whip whirl whisk
wilt wipe wok wrap yield zest

Table 9.3: Complete list of 270 unique cooking and food-processing actions.