“What’s Cooking?” Recipe Ingredients inform Cuisine of Origin

Mingyu Song
Neuroscience
Princeton University
mingyus@princeton.edu

Lili Cai
Neuroscience
Princeton University
lxcai@princeton.edu

Christopher Criscitiello
Math
Princeton University
cc26@princeton.edu

Collins Metto
CS
Princeton University
ckmetto@princeton.edu

Abstract

Food is a delicious and integral part of lives, and we’re interested in the similarities and differences in recipe ingredients across cultures world wide. The Kaggle “What’s Cooking” dataset has 39.5k recipes from 20 cuisines. Each recipe has a varying list of 1 to 65 ingredients, of 6,714 possible ingredients. We use unsupervised learning methods including SVD, K-Means, PCA to find relationships between ingredients and cuisines. We find that cuisines are correlated by geography and political influence, salt and onions are the most common ingredients to cluster recipes, and each cuisine has key signature ingredients. Latent topic models such as LDA and BMF were able extract latent cuisines and popular combinations of ingredients. We also use supervised learning to classify what cuisine a recipe belongs to based on the ingredients (the original Kaggle challenge). Though we use a smaller 80% of Kaggle’s training set, our logisitic regression achieves the benchmark accuracy on the original challenge and the RandomForest method is the best performing tree-based method.

1 Introduction

We’re interested in a dataset that has opportunity for multi-class classification, unsupervised clustering and similarity analysis. The Kaggle “What’s Cooking” dataset [1] provides an opportunity for all of these machine learning techniques. We’re particularly interested in seeing if unsupervised learning will cluster recipes based on their ingredient distributions, or their culture of interest. For example, will LDA latent topics reveal cultural background or dish category (vegetarian, meat, dessert), while PCA picks up important ingredient types (starch, fruit, spices)?

The “What’s Cooking” dataset has 39,774 recipes, each containing 1-65 ingredients out of 6,714 possible ingredients (for example recipe, see Supplemental Materials 7.1). Each recipe is labeled with a culture of origin, with 20 possible labels such as Southern US, Chinese and Moroccan. We will use only the training data from the Kaggle challenge (their test data is unlabeled), and use the same 80/20 training/testing split when evaluating our models. As such, we keep in mind that our training data is smaller than the one used by Kaggle’s challenge when comparing our results to their leaderboard.

2 Related Work

The 2015 Kaggle challenge asked coders to classify these 20 cuisines based on their ingredients. Baseline classifier performance with logistic regression had a score of .77, with each individual cuisine performing better than chance, ranging from .4 to .9. Despite biases in recipe distribution (ie. 8k Italian recipes vs. 800 Morrocan recipes), the cuisines with highest accuracy, precision and recall were: Brazilian, Mexican, Morrocan, British, Greek and Thai [2]. The leaderboard score was .83 [5], and top scorers that explained their code used combinations of TF-IDF preprocessing, XGBoost, fine tuning parameters, neural networks and stacking/blending models [3].

Despite the popular challenge, very little public work applied unsupervised learning methods to the data set [4]. We aim to reproduce the supervised learning methods within reason, in addition to exploring clustering based on unsupervised learning.

3 Methods

3.1 Data Preprocessing

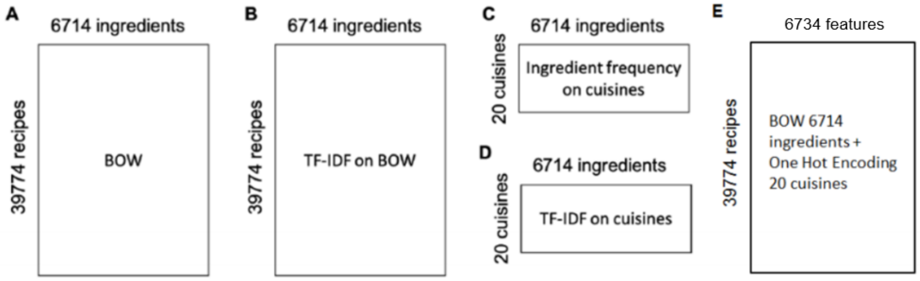


Figure 1: **Five data representations after pre-processing.**

Since we want to answer different questions with our unsupervised learning models, we pre-process the data into the following representations (Figure 1):

A. Bag-of-Word (BOW): BOW is a binary matrix of 39k recipes (samples) × 6714 ingredients (features), with each entry indicating whether a certain ingredient appears in a certain recipe.

B. Term-Frequency-Inverse-Document-Frequnecy (TF-IDF) on BOW: We adopt the following definition of TF-IDF [16] in this project:

$$\mathrm{tf}(t,d) = \frac{1 \text{ if } t \text{ occurs in } d \text{ and } 0 \text{ otherwise}}{\text{number of words in } d},$$
$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|},$$
$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D),$$

where t is a term, d is a document, D is the set of all documents, and N is the total number of documents.

For TF-IDF on BOW, recipes correspond to documents, and ingredients are terms. TF-IDF on BOW is thus also a 39k × 6714 matrix. Each ingredient is effectively normalized by the total number of ingredients in the recipe (an ingredient in a recipe with many ingredients will have a lower score), and the frequency of this ingredient in all recipes (an ingredient that is common to all recipes will have a lower score).

C. Ingredient frequency on cuisines: We combined all ingredients from all recipes within a cuisine together in one conglomerate "recipe", condensing 39k recipes into 20 recipes, effectively 1 umbrella recipe for each cuisine. In each umbrella recipe for each cuisine, ingredients could appear more than once. We count the sum of all ingredients in each umbrella recipe, or cuisine. This yields a matrix of 20 cuisines (documents) × 6714 ingredients (features).

D. TF-IDF on cuisines: We take the TF-IDF of Ingredient Frequency on cuisines (C). This will determine which signature ingredients are most common to each cuisine, but not universally common across cuisines. This yields a matrix of 20 cuisines (documents) × 6714 ingredients (features).

E. BOW + one hot coding of 20 cuisines. We take BOW from (A) and extend 20 features by one hot encoding the 20 cuisines.

Ingredient frequency and TF-IDF on cuisines (C and D) are used in data exploration and description. BOW, TF-IDF on BOW and BOW plus cuisines (A, B and E) are used for unsupervised learning. BOW (A) and reduced features on (A) is used for supervised learning. This yields a matrix of 39775 recipes and 6734 features.

Since all features are in the same arbitrary units, and due to their binary nature, we chose not to standardize the data. (We ran PCA, Truncated SVD and logistic regression with standardized BOW, but saw better results when they were not standardized; we reason this is due to the binary nature of the data.)

3.2 Machine Learning Methods

Unless otherwise indicated, we used Python's scikit learn toolbox [20][21] to implement methods.

3.2.1 Supervised Learning

We perform 3-fold cross validation using these classifiers:

- **Logistic Regression:** C=1, L2-penalty
- **Random Forest:** N_estimators=10, 100 depth=30, 100
- **XGBoost:** N_estimators=200, depth=20; model did not finish running after 2 days
- **Gradient Boost:** N_estimators=100, depth=10; model did not finish running after 2 days

3.2.2 Unsupervised Learning

For Data Description and Exploration:

- **Agglomerative Clustering:** Agglomerative Clustering is a method that outputs a hierarchiy of clusters. We use this method as an alternative to K-means. It starts with each sample as its own cluster, and then group clusters based on their similarity. We use the "Ward" linkage for group similarity We implement Agglomerative Clustering using package scipy.cluster.hierarchy.

For Relationship Between Ingredients and Cuisines:

- **K-means:** K-means is an iterative algorithm which finds k centers (points in the feature space), and assigns each data point to one of these centers. This partitions the data into k clusters. We apply k-means with Euclidean distance to the BOW representation. We examine the results for $k = 2, 3$ and 20.
- **Truncated Singular Value Decomposition (SVD):** Like PCA, truncated SVD performs a linear dimensionality reduction on the data. Truncated SVD is very fast for large sparse matrices, like our data. We implement it with sklearn, with all parameters set to default. We use the first two components to visualize the dimension-reduced data. We also use truncated SVD with 1000 components to reduce the dimension of the data prior to supervised learning. These 1000 components explain 90% of the variance in the data.
- **Market Basket Analysis:** Market basket analysis finds conjunctive rules (A AND B) prevalent in the data, and frames them as association rules ($A \implies B$). We use the Apriori algorithm, with minimum support of 0.01, to find association rules with the greatest confidence and lift. The confidence of a rule $A \implies B$ is an estimate of the probability the consequent B occurs given A, $\Pr(B|A)$. The lift of a rule $A \implies B$ is an estimate of $\Pr(A \text{ AND B})/(\Pr(A)\Pr(B))$. If A and B are independent, they have lift 1. A lift much greater than 1 indicates A and B are dependent. Market basket analysis is implemented the Python package mlxtend.frequentpatterns [19].

For Latent Cuisine Extraction:

- **Latent Dirichlet Allocation (LDA) on TF-IDF on BOW:** We perform LDA analysis on the TF-IDF on BOW representation, using the LDA function in sklearn package, with various number of components (from 2 to 20).
- **Binary Matrix Factorization (BMF) [17] on BOW:** BMF is a special case of non-negative matrix factorization. It aims to factorize binary matrix X into two matrices H and W:

$$X \approx HW,$$

where $X \in \mathbb{R}^{n \times p}$, $H \in \mathbb{R}^{n \times r}$ and $W \in \mathbb{R}^{r \times p}$. Because the BOW matrix is binary, BMF is well suited to extract latent structure from it. It can be seen as finding r latent recipes, with basis matrix W describing their tendency to have each ingredient, and mixture matrix H quantifying how each sample (recipe) can be seen as a weighted combination of the latent recipes. We used the `Bmf` method in python library `Nimfa` [18], setting maximum number of iterations to 100 (to make sure the method converges), and keeping all other parameters to their default values. Consistent with LDA analysis, we test for $r = 2$ to 20.

- **Principal Component Analysis (PCA) on TF-IDF on BOW:** We run PCA on the TF-IDF on BOW representation, using the PCA function in sklearn package, with all parameters set to default values. To investigate the meaning of the first two principal components (PC1 and PC2), we project every recipe to the directions of PC1 and PC2, and plot the average projection for each cuisine, which results in 20 blobs on the PC1-PC2 space.

- **K-Means:** K-means was also able to extract some latent cuisines along at higher cluster numbers, but mostly just ingredients.

3.3 Metrics for Evaluating Performance

For unsupervised learning comparison between models, we decided not to use typical metrics such as log likelihood and BIC, and instead focus on the interpretation of the extracted features and results. We did this for two reasons: (1) each method (ie. Market Basket vs. PCA vs. LDA) yielded different interpretations of the data structure, so we reasoned typical metrics of evaluation would not be an appropriate means to understanding the analysis. (2) Relevant preprocessing for each model makes comparisons between models difficult. Ultimately, it wasn't about which unsupervised model was "the best," but more about what each method could teach us about ingredients and culture. For within a model (ie. LDA), we used LL and BIC to compare which number of components was best fit.

For supervised learning, we reasoned since classifying cuisines is a non-life threatening question, we care equally about false negatives as well as false positives. Thus, we report the confusion matrix and values derived from the confusion matrix: overall accuracy, cuisine accuracy and cuisine f1-score. Since there is an unequal distribution in the number of recipes per cuisine, we reason that cuisines with more recipes will also have higher accuracy overall. To counter this, we use the f1-score, which is the harmonic mean of precision and recall (true positive rate), it punishes extreme values and accounts for the unbalanced classification problem [9]. A higher f1-score indicates better classifier performance and ability to balance between both precision and recall.

4 Results

4.1 Data Description and Exploration

Characterizing the Cuisines

To characterize the cuisines, we list both the most common ingredients (based on ingredient frequency matrix in Figure 1C) and the most "signature" ingredients (based on TF-IDF on cuisine matrix in Figure 1D) for each cuisine (Supplementary Maerials 7.3, Table 7). While the most common ingredients in many cuisines consists of overlapping ingredients like salt, onions and olive oil, the most signature ingredients contains exclusively cuisine-specific ingredients. Some examples of the most signature list include: greek: feta cheese crumbles, dried oregano, Greek seasoning; southern_us: grits, collard greens, buttermilk; Indian: garam masala, curry leaves, paneer.

Both lists are useful: we wondered if unsupervised learning would extract common or signature features, or a combination of both. We will compare features and latent clusters discovered by unsupervised learning methods (e.g. LDA) to the most common/signature ingredients of each cuisine to see if the latent recipes represents any cuisine; we also expect the features with the highest weight/predictiveness in supervised learning would be among the most signature ingredients.

Relationship Between Cuisines

We use hierarchical clustering to further characterize the similarity between the 20 cuisines based on their ingredient frequency (Figure 1C) and TF-IDF on cuisines (Figure 1D), by looking at the Pearson correlation coefficient between each cuisine (2).

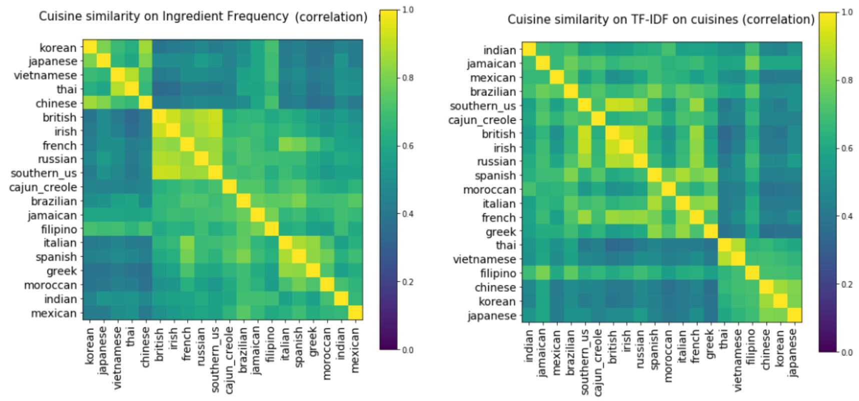


Figure 2: **Cuisine Similarity.** The colors represent the Pearson correlation coefficients of the ingredient frequency vector between each pair of cuisines, and the cuisines are reordered based on hierarchical clustering. Left: Clustering on Ingredient Frequency. Right: Clustering on TF-IDF on Cuisines. [15] (using package `scipy.cluster.hierarchy`).

We find that cuisine similarity reflects geographic proximity, in addition to history of colonization and cultural influence. For example, the high correlation between Filipino cuisine with European cuisines could reflect colonization by Britain and Spain, and correlation between Vietnamese-Brazilian could reflect Brazil's contribution to American troupes during the Vietnam war (interestingly, this correlation is higher than Vietnamese-French correlation, despite heavy French colonization in Vietnam).

Using ingredient frequency for clustering extracts the following groups (Figure 2, Left):

- Asian (Korean, Japanese, Vietnamese, Thai and Chinese)
- North European and Southern US (British, Irish, Southern US, Russian and French)
- South European and North African (Italian, Spanish, Greek, Moroccan)

This reordering method is not perfect, for example, it seems that Filipino should also be clustered with Asian cuisines; French cuisine, though belonging to the cluster with British, Irish, Southern-US and Russian, is also highly correlated with the Italian, Spanish, Greek and Moroccan cluster. If we repeat the clustering with TF-IDF processing on cuisines, the correlation between cuisines is similar, while clustering order is different: for example, Filipino cuisine is now grouped with the other Asian cuisines (Figure 2, Right). Both methods give a coarse sense of how different cuisines are similar to each other. Later, we will analyze the prediction results of supervised learning models to see if errors/confusions occur more frequently between similar cuisines.

4.2 Supervised Learning

Since this was a high dimensional dataset of over 6k features, we reasoned the family of tree searches would perform most efficiently and accurately. Thus, we used logistic regression (as a baseline) in addition to three tree-based methods: random forest, gradient boost and XGBoost, the later of which was the most popular in winning challenge kernels.

We ran the models on the whole feature set of size 6714, and tried several methods for feature reduction: truncated SVD extracted 1000 features, and using an individual model's feature selection on the whole BOW using `sklearn.feature_selection.SelectFromModel`. Both the whole and the reduced feature set on log regression achieved baseline accuracy of 0.77 (Figure 3). RFC took several hours to run, and GBC, and XGBoost took over 2 days so we stopped those models prematurely.

Our assumption was that cuisines with more recipes such as Italian will have higher accuracy overall: and indeed cuisines such as Mexican and Indian at 6438 and 3003 recipes respectively have high f1 scores. However, Brazilian cuisine with the least number of recipes at 467, had the best prediction

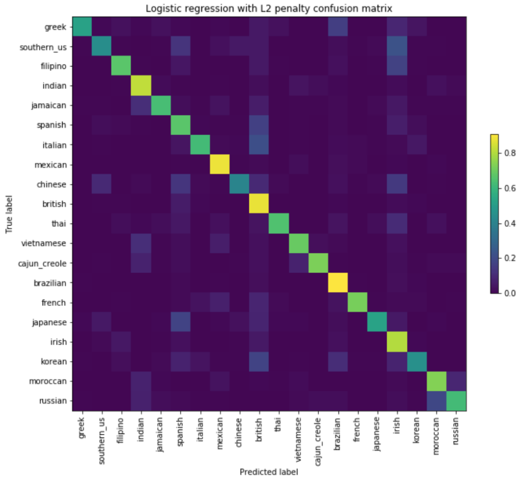


Figure 3: Training for log regression was done on 1000 features extracted by truncatedSVD. Chinese was least performing cuisine with a F_1 score of 0.50 and Brazilian was the best performing at 0.90 F_score.

Model	No. of features	Accuracy	F1_Score	Hyperparameters
LogR w/o feature_selection	6714	**0.77**	0.77	C = 1, L2 penalty
LogR w/ feature_selection	2309	0.76	0.77	C = 1, L2 penalty
LogR with truncatedSVD	1000	**0.77**	0.77	C = 1, L2 penalty
RandomForest 1	987	0.57	0.53	n_estimators = 10, max_depth = 30
RandomForest 2	987	**0.69**	0.68	n_estimators = 100, max_depth = 100
GradientBoosting	-	-	-	n_estimators = 100, max_depth = 10
XGBoost	962	-	-	n_estimators = 200, max_depth = 20
XGBoost test	962	0.40	0.39	n_estimators = 1, max_depth = 2

Table 1: Logistic regression accuracy shows little variation on accuracy with different feature set sizes. See Figure 3 for the confusion matrix. Increasing the depth and number of estimators for RF improves performance. We do not report on Gradient Boosting since we stopped running after > 48 hours. Similarly, we stopped XGBoost after > 48 hours, and scaled down the hyperparameters to observe performance as in XGBoost test.

(accuracy of .9 and f1-score of .8), and so was British at 804 recipes, suggesting the model is not just biased by unequal recipe distribution (see performance for individual cuisines in Supplementary Materials 12). In terms of performance of the models, logistic regression was the best performing model in terms of speed and accuracy and RF was the second best at 0.69 accuracy. XGBoost, designed to improve performance and speed of gradient boosting models [8], was trained on 962 features selected from the logistic regression model. To minimize running time, we tested XGB's performance on a single tree and a maximum depth of 2, but this resulted in a decrease in accuracy, as expected.

Feature Selection. Using the chi square scoring function, we selected the top 20 features and noted that these features also appear in our signature ingredients listed in table 6. Examples of top features include ”cumin seed” which also appears as a signature ingredient for Indian cuisine. ”Gochujang base” corresponds to Korean cuisine and ”Cathaca” to Brazilian. More examples are listed in Table 11.

4.3 Unsupervised Learning

K-means and SVD on BOW provides information about common ingredients

For small values of k, k-means partitions the recipes by the presence or absence of common ingredients in the recipes. For example, for $k = 2$, the first cluster contains all recipes with salt, and the second cluster contains all recipes without salt. For $k = 3$, the clusters are determined by the presence or absence of onions and salt in the recipes. 99.98% of recipes in the first cluster contain

Table 2: Top rules based on “confidence” in Market Basket Analysis

Antecedents	Consequents	Confidence
Onions, carrots, pepper	Salt	1
Baking powder, white sugar, eggs	All-purpose flour	.98
Baking powder, white sugar, all-purpose flour	Eggs	.94

no onions and no salt; 99.96% of recipes in the second cluster contain no onions and salt; 100% of the recipes in the third cluster contain onions (and of these 55% contain salt). For $k = 4$, k-means divides the clusters based on salt, onions and olive oil. For $k = 5$, k-means partitions the clusters based on salt, onions, olive oil and soy sauce.

Given the recipes, k-means was only partially able to recover the cuisines. To see if k-means favored 20 clusters, corresponding to the cuisines, we performed 5-fold cross validation using the k-means objective function for differing values of k (Supplementary Materials Figure 6). There is no clear kink in these plots, indicating no preference for $k = 20$ clusters. However, further examination revealed that k-means was able to recover some cuisine information. For example, several clusters consisted mostly of one cuisine (Indian or Mexican), one of the clusters consisted of mostly Asian cuisines, and another cluster consisted of mostly British, Irish, Southern US and Russian cuisines (Supplementary Materials Figure 8).

As an alternative to K-means, We also performed agglomerative clustering on the BOW representation. However, agglomerative clustering did not strongly partition the groups by ingredient or cuisine (Figure 9).

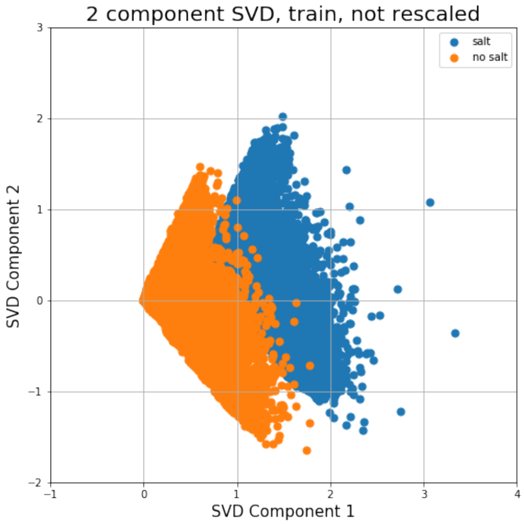


Figure 4: Training for log regression was done on 1000 features extracted by truncatedSVD. Chinese was least performing cuisine with a F_1 score of 0.50 and Brazilian was the best performing at 0.90 F_score.

Mapping the data onto the first two components of truncated SVD, there is also a clear distinction between recipes with and without salt (Figure 4). We suspect this is because salt is the most common ingredient, and thus explains a significant portion of variance in the data set. Indeed, rescaling before performing SVD eliminates this distinction between salt and no salt (Figure 7 in the Appendix).

Unlike the components of unstandardized SVD, the components of truncated SVD with standardization each correspond approximately to a cuisine. Identifying the elements of each component with largest magnitude, we see, for example, that component 5 corresponds with Indoan cuisine, 6 with Mexican, and 12 with Thai (Table 9).

Market Basket Analysis on BOW: relationship between ingredients

We performed market basket analysis on the BOW representation to uncover association rules among the ingredients. The rules with the highest confidence tended to have salt as a consequent. The rules with the lowest confidence tended to have salt as an antecedent. This is most likely because salt is a very common ingredient. Market basket analysis uncovered other interesting relationships, for example see Table 2.

We also ranked association rules by lift, in addition to confidence. The rules with the highest lift did not include salt but instead involved ingredients common in baking (e.g., yeast, flour, sugar, eggs), indicating that these ingredients are highly dependent. In fact, the rules with lowest lift usually included salt. This is because the lift of rule $A \implies B$ is symmetric in the antecedent and consequent, while confidence is directional. That is, for $A \implies B$ to have a high lift, the

Table 3: Top rules based on “lift” in Market Basket Analysis

Antecedents/Consequents	Antecedents/Consequents	Lift
active dry yeast	warm water	39.1
clove	cinnamon sticks	22.9
coriander seeds	cumin seed	22.2
garlic powder	onion powder	20.8

Table 4: Top rules based on “confidence” with cuisines included as features in Market Basket Analysis.

Antecedents	Consequents	Confidence
Corn tortillas	Mexican	0.98
Salsa	Mexican	.95
Garam Masala	Indian	.93

confidence of both $A \implies B$ and $B \implies A$ should not be too small; as discussed, this is not the case for rules involving salt. Some of the rules with highest lift are listed in Table 3.

Market Basket Analysis on BOW+Cuisines: Relationship between ingredients and cuisines

Interestingly, market basket analysis naturally associates certain ingredients with cuisines. After one hot-encoding the cuisines and appending these 20 extra features to the other 6714 features (data preprocessing from Figure 1E), the rules with the highest confidence include cuisines as consequents, for example, salsa and corn tortillas is associated with Mexican while garam masala with Indian. (Table 4).

Discovering latent recipes with BMF and LDA

Both LDA (on TF-IDF on cuisines) and BMF (on BOW) analyses aim to identify latent recipes. We present their results together, as they are largely similar, but also show interesting distinctions.

Setting the number of latent recipes to around 6 seem to yield the best results in both analyses (Table 5). By comparing the top ingredients for the latent recipes to the common and signature ingredients for each cuisine in Table 7, we find that many latent recipes represent cuisines. For BMF on BOW, latent recipe 1 corresponds to Southern-US, 2 corresponds to Mexican, 3 corresponds to Italian, and 6 corresponds to Chinese. For LDA on TF-IDF on BOW, latent recipe 1 also corresponds to Southern-US, 2 corresponds to Italian, 3 corresponds to Chinese, 4 corresponds to Greek (maybe also Italian), and both 5 and 6 correspond to Mexican. Interestingly, these cuisines that are picked out by unsupervised learning methods are also the most represented ones in this dataset (the ones with the most recipes). Besides picking out cuisines, other latent recipes (4 and 5) in the BMF analysis are a collection of commonly used ingredients.

If the number of latent recipes is too small (for example, n=2; see Supplementary Materials Table 8), both models just pick out sets of common ingredients, unable to discover anything interesting.

If we increase the number of latent recipes (for example, n=15; see Supplementary Materials Table 8), neither models continue to discover more cuisines other than the major ones (e.g. Italian, Southern-US, Mexican and Chinese). For BMF and BOW, the rest of the latent recipes are very sparse with only one or a couple universally common ingredients. For LDA on TF-IDF on cuisines, we see replicates among latent recipes (e.g. 6 and 15 both represent Southern-US/British/Irish, and 12 and 13 both represent Italian); some latent recipes seem to represent certain types of cooking (e.g. 11 seems to represent dessert), but it is hard to extract meaning for most of them.

PCA

Using PCA, we were able to recover similarity and geographic proximity between cuisines. Figure 5A shows the average projection of recipes within each cuisine to the first two principal components. On the PC1-PC2 plane, cuisines form clusters: on the top-left are Asian cuisines, with two subclusters of East-Asian (Korean, Chinese and Japanese) and South-Asian (Thai, Vietnamese and Filipino); in the middle-left are Indian, Jamaican, Brazilian; on the lower-left are Italian/Spanish-related cuisines (Mexican, Moroccan, Cajun&Creole, Spanish, Greek, and Italian); on the right

Table 5: Top ingredients in the six latent recipes discovered by BMF and LDA analyses. Only ingredients that have the top 0.1% feature weights (feature loading in basis matrix W for BMF, and normalized pseudocounts for LDA) are shown.

BMF on BOW	LDA on TF-IDF on cuisines
Latent recipe 1 : sugar, all-purpose flour, large eggs, unsalted butter, butter, baking powder, milk	Latent recipe 1 : sugar, all-purpose flour, baking powder, vanilla extract, unsalted butter, butter, milk
Latent recipe 2 : onions, garlic, tomatoes, ground cumin, chili powder, carrots, vegetable oil	Latent recipe 2 : olive oil, ground black pepper, garlic cloves, butter, extra-virgin olive oil, salt, grated parmesan cheese
Latent recipe 3 : olive oil, garlic cloves, ground black pepper, kosher salt, extra-virgin olive oil, grated parmesan cheese, purple onion	Latent recipe 3 : soy sauce, sesame oil, ginger, water, garlic, rice vinegar, oil
Latent recipe 4 : salt, pepper, butter	Latent recipe 4 : extra-virgin olive oil, fresh lemon juice, olive oil, garlic cloves, purple onion, ground cumin, ground black pepper
Latent recipe 5 : water	Latent recipe 5 : jalapeno chilies, avocado, olive oil, garlic, shredded mozzarella cheese, corn tortillas, grated parmesan cheese
Latent recipe 6 : soy sauce, sesame oil, green onions, garlic, sugar, vegetable oil, scallions	Latent recipe 6 : salsa, shredded cheddar cheese, sour cream, flour tortillas, chili powder, black beans, ground beef

are British-related cuisine (British, Irish, Russian, Southern-US and French). Thus, PC1 nicely separates British-related cuisines from others; while PC2 further separates Asian, central-American and South-European-related cuisines.

The PCA results (more specifically, the projections to the first two PCs) form clusters that are very similar to the cuisine correlation analysis in Figure 2: cuisines that are correlated with appear close to each other in the PC1-PC2 plane. In fact, there is a strong negative relationship between the Euclidean distance on the PC1-PC2 plane and cuisine correlation coefficients (Figure 5B). This indicates that dimension reduction with PCA preserves the similarities between cuisines.

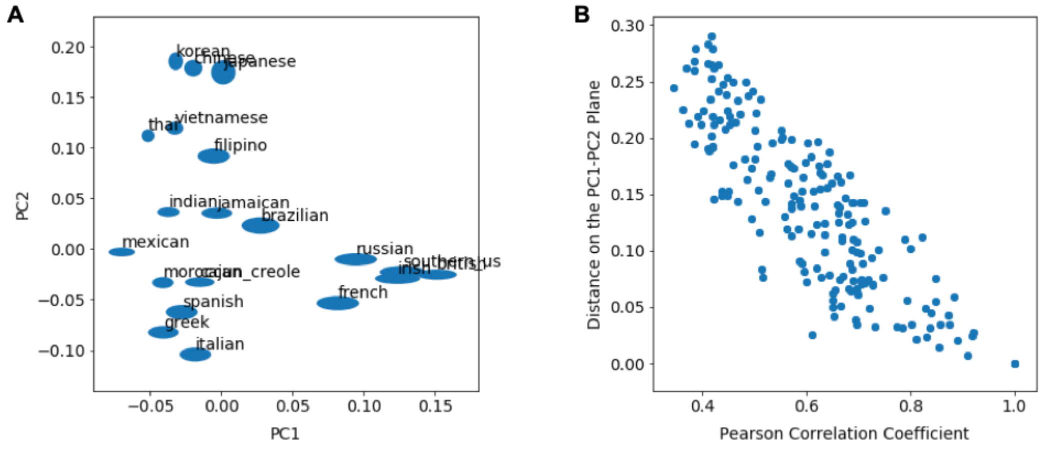


Figure 5: **PCA results.** **(A)** The average projection of recipes within each cuisine to the first two principal components (PC1 and PC2). The center of the ellipses are the average projection across recipes within each cuisine; the widths and heights indicate the standard deviation across recipes (scaled by 0.1 for better visualization). **(B)** The Euclidean distance between two cuisines on the PC1-PC2 plane against the Pearson correlation coefficients between them (values taken from the correlation matrix in Figure 2). Each dot is one pair of cuisines.

5 Discussion and Conclusion

Our unsupervised learning exploration of the "What's Cooking" dataset has several takeaways regarding our knowledge of ingredients, cuisines, and classification. Across all cultures, we learned that salt and onions are the most common ingredients that can separate recipes. Next, we learned (unsurprisingly) that similarity between cuisines are correlated by geography. Finally, we learn that latent topic models can extract latent cuisines. We didn't find latent representation or clustering of a specific types of ingredients, such as starch, protein, vegetarian, or dessert dishes. On ingredient data, SVD and Kmeans are able to extract ingredients but less so cuisines, while latent models LDA and BMF were able to extract cuisines. On cuisine data, PCA and heiarchical clustering applied to cuisines yielded correlation matrix for the cuisines. Delightfully, the correlation from hierarchical clustering of cuisines matched the distance of cuisines from PCA results on ingredients.

From supervised learning, we find that Brazilian and Mexican cuisines yielded highest accuracy and f1 scores, probably due to combination of low ingredient count and high uniqueness of ingredients.

Given more time, we could explore several methods for improving classification based on our unsupervised learning methods. (1) Have a 2-step classification process: first classify whether a recipe is in a particular cuisine: European, Asian, Other. Next, from these sub groups, train on more specific cuisine. (2) Decrease the feature set by extracting top 100 ingredients for each cuisine based on the signature ingredients from TF-IDF of cuisines.

Futhermore, we anticipate that a richer dataset (ie. one that includes the amount of each ingredient) would give more interesting supervised and unsupervised learning results. Since our raw data was in binary format, SVD and PCA could not take advantage of the scale of a feature. Additionally, supervised learning might be able to better classify recipes with ingredient amounts, as seen by examples in the literature [6].

Most relevantly, with globalization, we anticipate more cross-cultural recipes. Perhaps instead of classification, one could tag the top 3 cultures and their percentage of influence over a recipe. With the increase of online cataloging of recipes, machine learning could be an effective tool to track the evolution of recipes over time, which could be a reflection of the political climate and interplay between cultures.

Acknowledgments

The code is adapted from code the authors used in homework assignments 1, 2 and 3. As such, the code is also based on references cited in those assignments. We thank Prof. Barbara Engelhardt and the cos424 TAs for comments on the poster and throughout the class!

References

[1] Kaggle "What's Cooking" Dataset. https://www.kaggle.com/c/whats-cooking/data

[2] Kaggle What's Cooking Challenge Walk Through. https://flothesof.github.io/kaggle-whats-cooking-machine-learning.html

[3] Wen, Jeff. What's Cooking? http://jeffwen.com/2015/12/19/whats_cooking

[4] Levy, Alona. Cultural Diffusion by Recipe https://www.kaggle.com/alonalevy/cultural-diffusion-by-recipes

[5] Kaggle What's Cooking Leaderboard https://www.kaggle.com/c/whats-cooking/leaderboard

[6] Zhao, Alice. Cupcake vs. Muffin https://github.com/adashofdata

[7] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge Massachusetts, 2012.

[8] Tianqi Chen and Carlos Guestrin. *XGBoost: A Scalable Tree Boosting System*. CoRR, abs/1603.02754, 2016. http://arxiv.org/abs/1603.02754.

[9] Beyond Accuracy: Precision and Recall https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

[10] Lettier. Your Easy Guide to Latent Dirichlet Allocation https://medium.com/@lettier/how-does-lda-work-ill-explain-using-emoji-108abf40fa7d

[11] Ruozzi, Nicholas. LDA Presentation. https://www.utdallas.edu/ nrr150130/cs7301/2016fa/lects/Lecture_20_LDA.pdf

[12] CS424, Lecture 18: LDA. PDF located on Piazza website.

[13] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825?2830.

[14] cos424 Precept code: SampleAnalysis.ipynb

[15] https://github.com/TheLoneNut/CorrelationMatrixClustering/blob/master/CorrelationMatrixClustering.ipynb

[16] https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[17] Zhang, Zhongyuan, et al. "Binary matrix factorization with applications." *Seventh IEEE International Conference on Data Mining (ICDM 2007).* IEEE, 2007.

[18] Marinka Zitnik and Blaz Zupan. Nimfa: A python library for nonnegative matrix factorization. *Journal of Machine Learning Research*, 13:849853, 2012.

[19] mlxtend. 2019.http://rasbt.github.io/mlxtend/

[20] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, Vol 12, pgs 2825–2830, 2011.

[21] Lars Buitinck and Gilles Louppe and Mathieu Blondel and Fabian Pedregosa and Andreas Mueller and Olivier Grisel and Vlad Niculae and Peter Prettenhofer and Alexandre Gramfort and Jaques Grobler and Robert Layton and Jake VanderPlas and Arnaud Joly and Brian Holt and Gaël Varoquaux.API design for machine learning software: experiences from the scikit-learn project. ECML PKDD Workshop: Languages for Data Mining and Machine Learning. Pgs 108-122. 2013.

6 References for Code Used

7 Supplementary Materials

7.1 An example sample (recipe) in the “What’s cooking” dataset

{
“id”: 24717,
“cuisine”: ”indian”,
“ingredients”: [“tumeric”, “vegetable stock”, “tomatoes”, “garam masala”, “naan”, “red lentils”,
“red chili peppers”, “onions”, “spinach”, “sweet potatoes”]
}

7.2 Number of recipes in each cuisine

Table 6: Number of recipes in each cuisine.

Cuisine	Number of recipes
italian	7838
mexican	6438
southern_us	4320
indian	3003
chinese	2673
french	2646
cajun_creole	1546
thai	1539
japanses	1423
greek	1175
spanish	989
korean	830
vietnamese	825
moroccan	821
british	804
filipino	755
irish	667
jamaican	526
russian	489
brazilian	467

7.3 The ten most common and most signature ingredients in each cuisine

Table 7: The ten most common and most “signature” ingredients in each cuisine. The most common ingredients are the ones with the highest values on ingredient frequency, while the most signature ingredients are those with the highest values on TF-IDF on cuisines.

Cuisine	Ten Most Common Ingredients	Ten Most Signature Ingredients
greek	salt, olive oil, dried oregano, garlic cloves, feta cheese crumbles, extra-virgin olive oil, fresh lemon juice, ground black pepper, garlic, pepper	feta cheese crumbles, feta cheese, dried oregano, greek seasoning, pitted kalamata olives, kalamata, fresh oregano, phyllo dough, ground lamb, grape leaves
southern_us	salt, butter, all-purpose flour, sugar, large eggs, baking powder, water, unsalted butter, milk, buttermilk	grits, collard greens, buttermilk, bourbon whiskey, quickcooking grits, yellow corn meal, white cornmeal, chopped pecans, black-eyed peas, cajun seasoning
filipino	salt, garlic, onions, water, soy sauce, pepper, oil, sugar, carrots, ground black pepper	fish sauce, calamansi juice, lumpia wrappers, calamansi, lumpia skins, oyster sauce, thai chile, shrimp paste, pork belly, fried garlic

indian	salt, onions, garam masala, water, ground turmeric, garlic, cumin seed, ground cumin, vegetable oil, oil	garam masala, curry leaves, paneer, ghee, coriander powder, cumin seed, asafoetida, urad dal, black mustard seeds, green chilies
jamaican	salt, onions, water, garlic, ground allspice, pepper, scallions, dried thyme, black pepper, garlic cloves	scotch bonnet chile, jamaican jerk season, ackee, callaloo, jerk seasoning, ground allspice, thyme, dark rum, allspice, jerk sauce
spanish	salt, olive oil, garlic cloves, extra-virgin olive oil, onions, water, tomatoes, ground black pepper, red bell pepper, pepper	saffron threads, chorizo sausage, spanish chorizo, serrano ham, manchego cheese, spanish paprika, sherry vinegar, roasted red peppers, arborio rice, chorizo
italian	salt, olive oil, garlic cloves, grated parmesan cheese, garlic, ground black pepper, extra-virgin olive oil, onions, water, butter	lasagna noodles, ricotta cheese, arborio rice, prosciutto, marinara sauce, fresh parmesan cheese, pasta sauce, parmigiano reggiano cheese, italian sausage, spaghetti
mexican	salt, onions, ground cumin, garlic, olive oil, chili powder, jalapeno chilies, sour cream, avocado, corn tortillas	refried beans, enchilada sauce, taco seasoning mix, taco seasoning, corn tortillas, tomatillos, salsa, tortilla chips, shredded Monterey Jack cheese, cotija
chinese	soy sauce, sesame oil, salt, corn starch, sugar, garlic, water, green onions, vegetable oil, scallions	Shaoxing wine, oyster sauce, sesame oil, hoisin sauce, dark soy sauce, light soy sauce, chinese rice wine, chinese five-spice powder, rice vinegar, rice wine
british	salt, all-purpose flour, butter, milk, eggs, unsalted butter, sugar, onions, baking powder, large eggs	stilton cheese, suet, beef drippings, stilton, golden syrup, dried currants, marmite, mincemeat, raspberry jam, beef kidney
thai	fish sauce, garlic, salt, coconut milk, vegetable oil, soy sauce, sugar, water, garlic cloves, fresh lime juice	fish sauce, Thai red curry paste, red curry paste, kaffir lime leaves, beansprouts, lemongrass, galangal, thai basil, rice noodles, thai green curry paste
vietnamese	fish sauce, sugar, salt, garlic, water, carrots, soy sauce, shallots, garlic cloves, vegetable oil	fish sauce, beansprouts, rice paper, rice noodles, thai basil, rice vermicelli, lemongrass, thai chile, daikon, hoisin sauce
cajun_creole	salt, onions, garlic, green bell pepper, butter, olive oil, cayenne pepper, cajun seasoning, all-purpose flour, water	cajun seasoning, andouille sausage, creole seasoning, file powder, crawfish, creole mustard, smoked sausage, okra, red beans, dried oregano
brazilian	salt, onions, olive oil, lime, water, garlic cloves, garlic, cachaca, sugar, tomatoes	cachaca, aai, manioc flour, palm oil, chocolate sprinkles, dried black beans, frozen banana, granola, dende oil, chia seeds
french	salt, sugar, all-purpose flour, unsalted butter, olive oil, butter, water, large eggs, garlic cloves, ground black pepper	gruyere cheese, grated Gruyre cheese, chopped fresh thyme, fresh tarragon, Nioise olives, herbes de provence, calvados, semisweet chocolate, thyme sprigs, capers
japanese	soy sauce, salt, mirin, sugar, water, sake, rice vinegar, vegetable oil, scallions, ginger	mirin, sake, dashi, nori, konbu, sushi rice, dried bonito flakes, rice vinegar, wasabi paste, bonito flakes
irish	salt, all-purpose flour, butter, onions, potatoes, sugar, baking soda, baking powder, milk, carrots	Irish whiskey, Guinness Beer, irish cream liqueur, corned beef, irish bacon, Baileys Irish Cream Liqueur, buttermilk, soda bread, stout, low-fat buttermilk
korean	soy sauce, sesame oil, garlic, green onions, sugar, salt, water, sesame seeds, onions, scallions	Gochujang base, kimchi, sesame oil, gochugaru, toasted sesame seeds, rice wine, asian pear, mirin, toasted sesame oil, rice cakes

moroccan	salt, olive oil, ground cumin, onions, garlic cloves, ground cinnamon, water, ground ginger, carrots, paprika	couscous, ras el hanout, preserved lemon, saffron threads, harissa, chickpeas, harissa paste, dried apricot, green olives, lamb shoulder
russian	salt, sugar, onions, all-purpose flour, sour cream, eggs, water, butter, unsalted butter, large eggs	sauerkraut, buckwheat flour, pierogi, dill, fresh dill, farmer cheese, beets, cottage cheese, sour cream, pickled beets

7.4 Latent recipe results

Table 8: Top ingredients in latent recipes discovered by LDA and BMF analyses, with number of latent recipes set to 2 or 15. Only ingredients with the top 0.1% feature weights are shown.

Number of latent recipes	**BMF on BOW**	**LDA on TF-IDF on BOW**
2	1: salt, sugar, all-purpose flour, butter, large eggs, eggs, unsalted butter	1: garlic, onions, water, soy sauce, vegetable oil, green onions, salt
	2: salt, onions, olive oil, garlic, garlic cloves, water, pepper	2: butter, all-purpose flour, olive oil, sugar, salt, unsalted butter, extra-virgin olive oil
15	1: sugar	1: cumin seed, garam masala, ground turmeric, green chilies, oil, tumeric, ghee
	2: garlic, onions	2: salsa, sour cream, flour tortillas, shredded cheddar cheese, black beans, chili powder, corn tortillas
	3: olive oil	3: quickcooking grits, 2% reduced-fat milk, condensed cream of mushroom soup, processed cheese, swiss cheese, butter, milk
	4: butter, milk	4: dry white wine, olive oil, garlic cloves, shallots, extra-virgin olive oil, unsalted butter, flat leaf parsley
	5: water	5: avocado, jalapeno chilies, fresh lime juice, chopped cilantro fresh, purple onion, lime, white onion
	6: soy sauce, sesame oil, green onions, corn starch, scallions, rice vinegar, ginger	6: vanilla extract, butter, sugar, all-purpose flour, eggs, white sugar, egg yolks
	7: vegetable oil	7: potatoes, meat, beets, cabbage, plain flour, pepper, onions
	8: garlic cloves, onions	8: soy sauce, sesame oil, fish sauce, rice vinegar, scallions, green onions, sugar
	9: ground cumin, chili powder, tomatoes, jalapeno chilies, chopped cilantro fresh, sour cream, avocado	9: ice cubes, cucumber, fresh dill, salmon fillets, greek yogurt, nori, plain yogurt
	10: salt	10: ground cumin, curry powder, ground coriander, ground cinnamon, chickpeas, ground ginger, olive oil
	11: pepper	11: sugar, large egg yolks, strawberries, whipping cream, unsalted butter, powdered sugar, whole milk
	12: onions, eggs, oil, milk, salt, flour	12: extra-virgin olive oil, olive oil, fresh lemon juice, balsamic vinegar, garlic cloves, red wine vinegar, capers
	13: black pepper	13: grated parmesan cheese, olive oil, shredded mozzarella cheese, mozzarella cheese, dried basil, parmesan cheese, italian seasoning

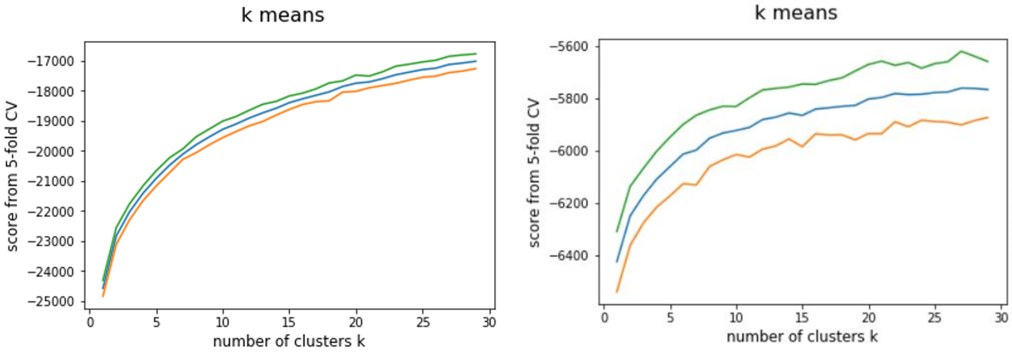


Figure 6: These plots show the score versus number of clusters k. The score function for is the negative of the k-means objective function (the sum of squares of the data points from their cluster centers). The size of the data was too large to perform multiple iterations of k-means, so we performed k-means on a reduced feature set (61 features, Left). We also performed k-means on the entire feature set, using 10% of the samples (Right). The green and yellow curves indicate two standard deviations from the mean (blue)

	14: ground black pepper, kosher salt, extra-virgin olive oil	14: onions, worcestershire sauce, green bell pepper, cajun seasoning, mayonaise, pepper, hot sauce
	15: all-purpose flour, large eggs, unsalted butter, baking powder, baking soda, buttermilk, vanilla extract	15: all-purpose flour, buttermilk, baking powder, milk, warm water, eggs, baking soda

7.5 k-means and SVD Results

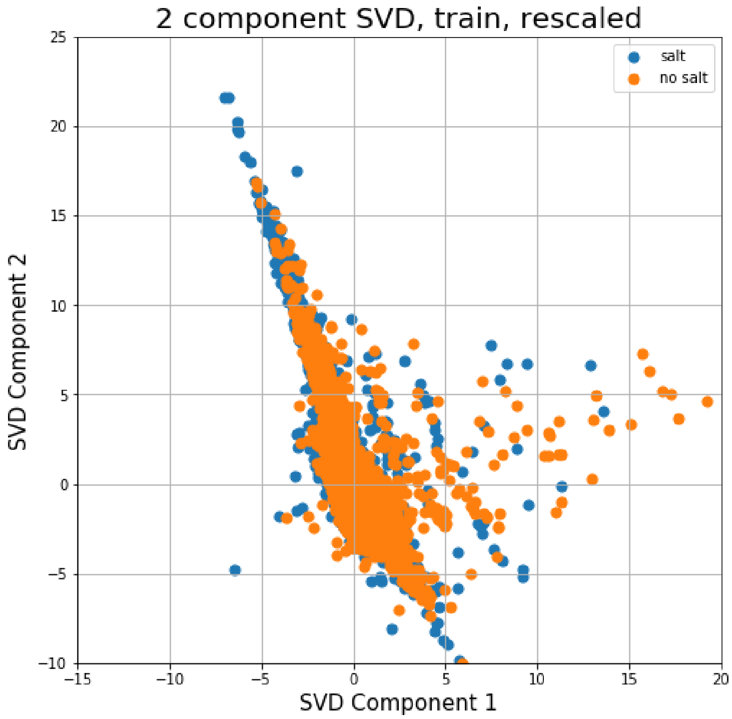


Figure 7: If the data is standardized, there is no clear distinction between salt and no salt for 2-component truncated SVD.

Table 9: Example components from SVD analysis reveals cuisine information.

component number	ingredients	approximate cuisine
5	’clove’ ’coriander powder’ ’cumin seed’ ’garam masala’ ’garlic paste’ ’green cardamom’ ’green chilies’ ’ground turmeric’ ’red chili powder’	Indian
6	’avocado’ ’black beans’ ’chili powder’ ’corn tortillas’ ’cumin’ ’dry white wine’ ’flour tortillas’ ’jalapeno chilies’ ’olive oil’ ’salsa’ ’shredded cheddar cheese’ ’sour cream’	Mexican
12	’coconut milk’ ’fish sauce’ ’galangal’ ’kaffir lime leaves’ ’lemongrass’ ’mirin’ ’palm sugar’ ’shrimp paste’	Thai

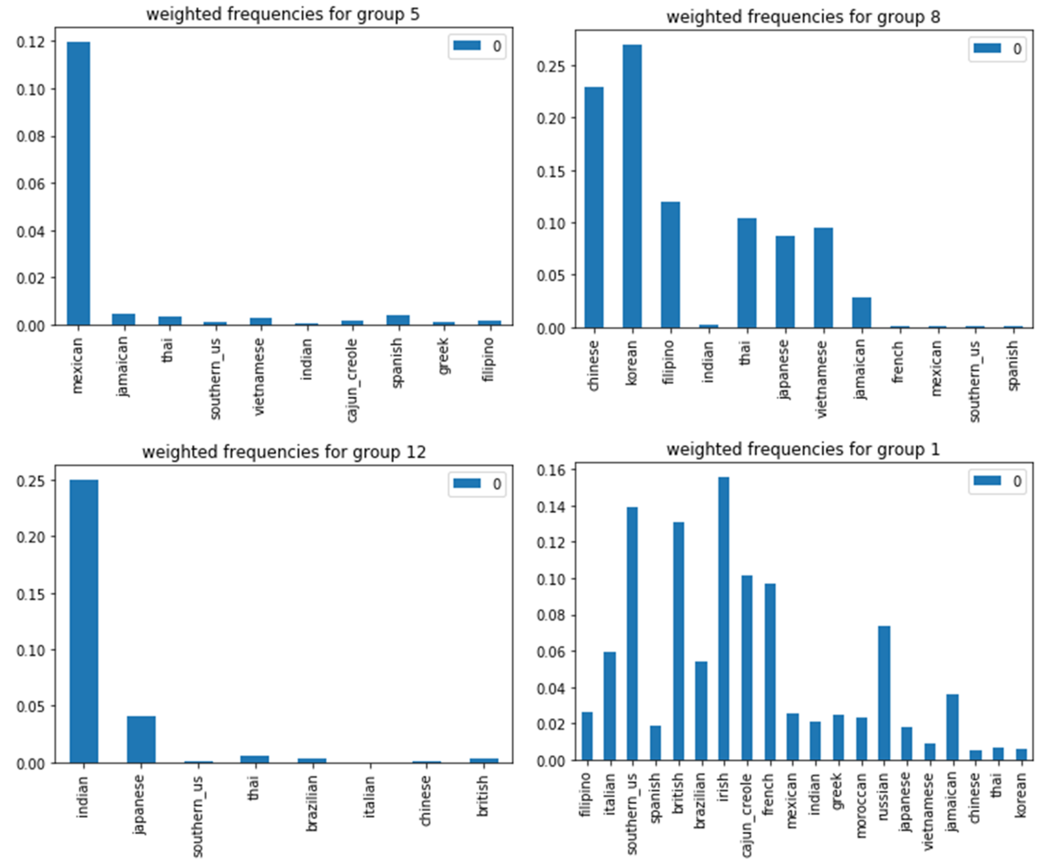


Figure 8: Weighted frequencies for four groups from k-means with $k = 20$. Weighted frequency is frequency of the cuisine in the group divided by the total number of recipes of that cuisine.

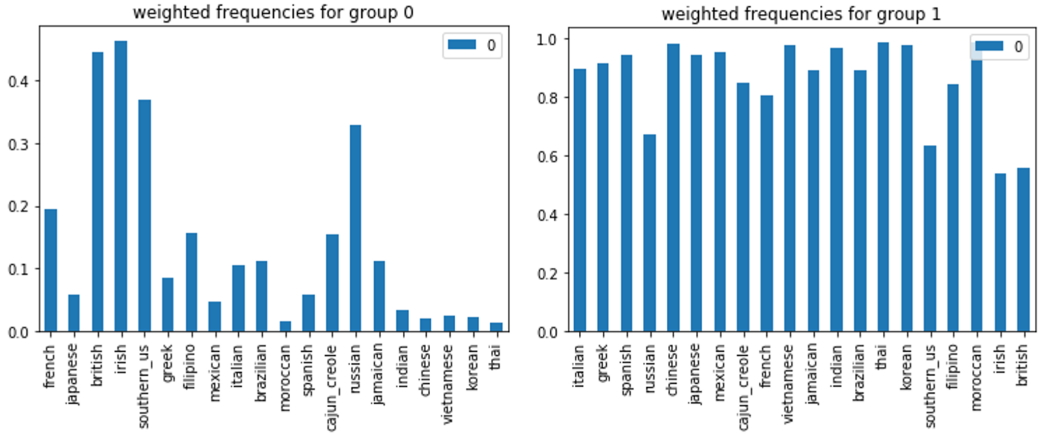


Figure 9: Weighted frequencies of cuisines in each group produced by agglomerative clustering for two clusters.

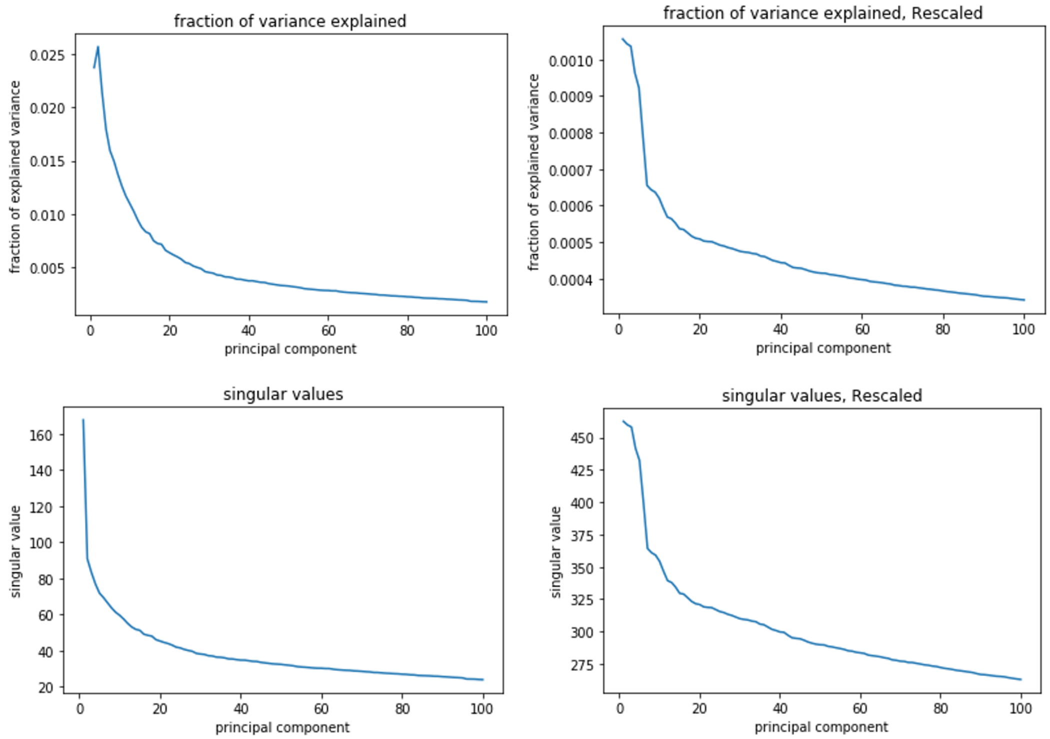
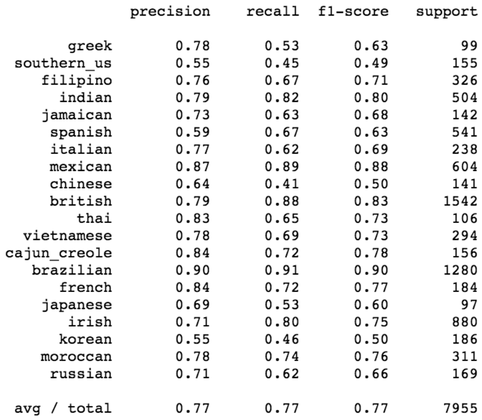


Figure 10: Explained variance and eigenvalues vs component for truncated SVD with (left) and without (right) standardization.

Top Features	**Matching Cuisine**
Gochujang base	Korean
Cachaca	Brazilian
Cajun seasoning	Cajun_creole
Coconut milk	Indian
Corn starch	Chinese
Corn tortillas	Mexican
Couscous	Moroccan
Cumin seed	Indian
Feta cheese crumbles	Greek
Fish sauce	Thai
Garam masala	Indian
Grated parmesan cheese	Italian

Figure 11: Top features selected using chi2 scoring.



```
              precision    recall  f1-score   support

       greek       0.78      0.53      0.63        99
 southern_us       0.55      0.45      0.49       155
    filipino       0.76      0.67      0.71       326
      indian       0.79      0.82      0.80       504
    jamaican       0.73      0.63      0.68       142
     spanish       0.59      0.67      0.63       541
     italian       0.77      0.62      0.69       238
     mexican       0.87      0.89      0.88       604
     chinese       0.64      0.41      0.50       141
     british       0.79      0.88      0.83      1542
        thai       0.83      0.65      0.73       106
  vietnamese       0.78      0.69      0.73       294
cajun_creole       0.84      0.72      0.78       156
   brazilian       0.90      0.91      0.90      1280
      french       0.84      0.72      0.77       184
    japanese       0.69      0.53      0.60        97
       irish       0.71      0.80      0.75       880
      korean       0.55      0.46      0.50       186
    moroccan       0.78      0.74      0.76       311
     russian       0.71      0.62      0.66       169

 avg / total       0.77      0.77      0.77      7955
```

Figure 12: Top features selected using chi2 scoring.