Cuisine Classification by Ingredients

Ronak Desai, Shidhesh Supekar, Kalven Bonin

Overview - *What's Cooking*

Given a dataset with list of recipes and their ingredients classified by a cuisine type (e.g. “Greek”) can one build a model to predict the cuisine type of a recipe give a list of its ingredients.

Summary of Dataset

- Has ~40,000 Recipes
  - List of Ingredients
  - 1 of 20 Cuisine Types

	ingredients	cuisine_name
0	[romaine lettuce, black olives, grape tomatoes...	greek
1	[plain flour, ground pepper, salt, tomatoes, g...	southern_us
2	[eggs, pepper, salt, mayonaise, cooking oil, g...	filipino
3	[water, vegetable oil, wheat, salt]	indian
4	[black pepper, shallots, cornflour, cayenne pe...	indian
5	[plain flour, sugar, butter, eggs, fresh ginge...	jamaican
6	[olive oil, salt, medium shrimp, pepper, garli...	spanish
7	[sugar, pistachio nuts, white almond bark, flo...	italian
8	[olive oil, purple onion, fresh pineapple, por...	mexican
9	[chopped tomatoes, fresh basil, garlic, extra-...	italian
10	[pimentos, sweet pepper, dried oregano, olive ...	italian
11	[low sodium soy sauce, fresh ginger, dry musta...	chinese
12	[Italian parsley leaves, walnuts, hot red pepp...	italian

Attempted Models

1. Baseline
2. Bag of Words
3. TF-IDF

Baseline Model

- Most popular cuisine type is Italian Food
- Naively Predict every cuisine as Italian
- 19.7% accuracy within Testing/Training Set

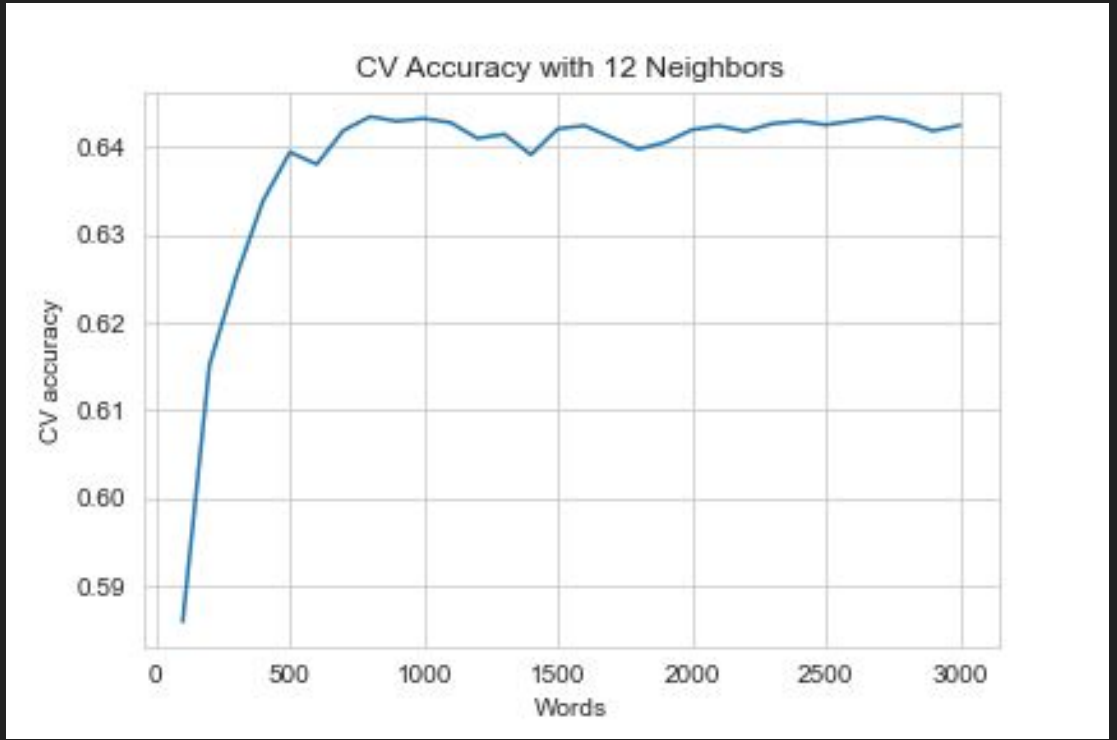
Bag of Words Model

- Filtered out ‘stop words’)
- Porter-Stemmer
- One-Hot Encoded Data Frame with counts from each recipe
- 2649 unique words

cuisine_name	pepper	salt	oil	onion	garlic	ground	fresh	sauc
greek	1	0	0	1	1	0	0	0
southern_us	1	1	1	0	0	1	0	0
filipino	1	1	1	1	1	0	0	1
indian	0	1	1	0	0	0	0	0
indian	1	1	1	1	1	1	0	0
...	...	...	...	...	...	...	...	...
irish	0	1	0	0	0	0	0	0
italian	1	0	0	1	0	0	0	0
irish	0	1	0	0	0	1	0	0
chinese	0	1	1	0	1	0	1	1
mexican	1	1	0	1	1	1	1	0

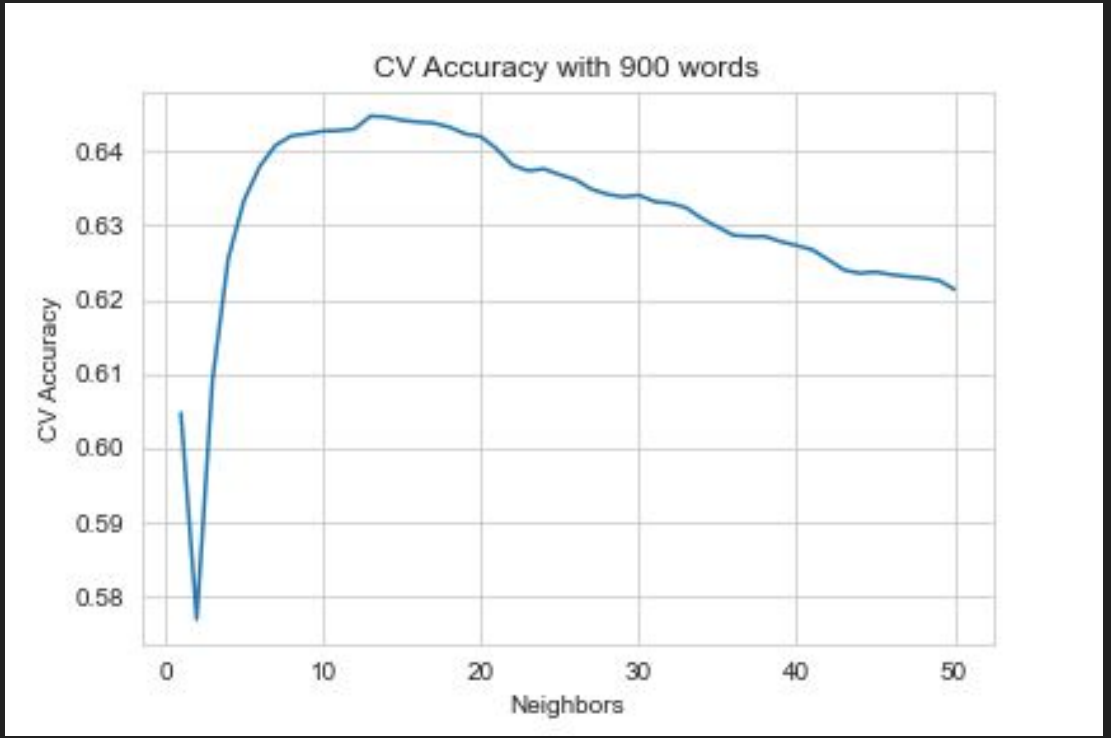
KNN-Classifier

- Trained Data Using 5-fold CV while varying number of neighbors and words used
- Found k = 12 neighbors to work best.
- Used 900 (highest counts) out of the ~3000 words to avoid overfitting on the very uncommon words



KNN-Classifier

- Trained Data Using 5-fold CV while varying number of neighbors and words used
- Found k = 12 neighbors to work best.
- Used 900 (highest counts) out of the ~3000 words to avoid overfitting on the very uncommon words



KNN Classifier Accuracy

- Model
  - 12 Neighbors
  - 900 Most Frequent Words
- Cross-Validation Accuracy: 65.2%
  - Mean Precision: 63.1%
  - Mean Recall: 50.7%
- Testing Accuracy with this Model: 66.3%
  - Precision: 64.0%
  - Recall: 51.6%

TF-IDF(Model)

- Use term frequency to construct a 20 (number of cuisines) dimensional vector for each ingredient.

- Weight (Coefficient) associated to ingredient [i] and cuisine [c] is given by

$$\frac{\{\text{Number of occurrence of [i] in cuisine [c]}\}}{\{\text{Total cuisine [c]-recipes}\} \times \{\text{Total number of occurrence of ingredient [i]}\}}$$

TF-IDF…

- Example of few of such vectors

	brazilian	british	cajun_creole	chinese	filipino	french	greek	indian	irish	italian	jamaican	japanese	korean	mexican	morocca
salt	0.000029	0.000034	0.000033	0.000024	0.000039	0.000031	0.000034	0.000045	0.000039	0.000031	0.000043	0.000020	0.000022	0.000029	0.00003
olive oil	0.000038	0.000011	0.000031	0.000008	0.000011	0.000031	0.000068	0.000019	0.000011	0.000063	0.000019	0.000006	0.000007	0.000032	0.00007
onions	0.000044	0.000028	0.000053	0.000018	0.000066	0.000022	0.000023	0.000062	0.000032	0.000025	0.000052	0.000016	0.000039	0.000036	0.00005
water	0.000033	0.000023	0.000030	0.000049	0.000068	0.000027	0.000020	0.000046	0.000026	0.000023	0.000048	0.000046	0.000048	0.000024	0.00003
garlic	0.000031	0.000008	0.000040	0.000049	0.000075	0.000014	0.000031	0.000041	0.000012	0.000032	0.000046	0.000019	0.000063	0.000035	0.00003

TF-IDF (Prediction)

- To predict the cuisine
  - We add the vectors associated to each of the ingredients
  - Find the direction with maximum weight
  - Declare cuisine associated to that direction as our prediction.
- Example

	Brazilian	British
Salt	0.01	0.02
Pepper	0.02	0.04

  - We predict that the recipe with ingredient {Salt, Pepper} is British since 0.06>0.03

TF-IDF (Metrics)

- Training set
  - Accuracy 69%
  - Precision 61%
  - Recall 74%

	precision	recall	f1-score	support
brazilian	0.37	0.72	0.49	374
british	0.35	0.78	0.48	643
cajun_creole	0.44	0.87	0.59	1237
chinese	0.88	0.78	0.83	2138
filipino	0.63	0.72	0.67	604
french	0.68	0.55	0.61	2117
greek	0.52	0.80	0.63	940
indian	0.87	0.82	0.84	2402
irish	0.43	0.75	0.55	534
italian	0.97	0.56	0.71	6270
jamaican	0.37	0.89	0.52	421
japanese	0.91	0.72	0.80	1139
korean	0.59	0.90	0.71	664
mexican	0.97	0.76	0.85	5150
moroccan	0.37	0.91	0.53	657
russian	0.29	0.78	0.42	391
southern_us	0.86	0.47	0.61	3456
spanish	0.45	0.57	0.50	791
thai	0.74	0.77	0.75	1231
vietnamese	0.53	0.78	0.63	660
accuracy			0.69	31819
macro avg	0.61	0.74	0.64	31819
weighted avg	0.79	0.69	0.70	31819

TF-IDF (Metrics)

- Test Set
  - Accuracy 63%
  - Precision 55%
  - Recall 66%

	precision	recall	f1-score	support
brazilian	0.24	0.47	0.32	93
british	0.21	0.52	0.30	161
cajun_creole	0.42	0.83	0.56	309
chinese	0.84	0.72	0.78	535
filipino	0.50	0.60	0.55	151
french	0.64	0.45	0.53	529
greek	0.44	0.72	0.55	235
indian	0.88	0.78	0.82	601
irish	0.34	0.56	0.42	133
italian	0.95	0.53	0.68	1568
jamaican	0.30	0.87	0.44	105
japanese	0.85	0.69	0.76	284
korean	0.52	0.89	0.65	166
mexican	0.97	0.73	0.83	1288
moroccan	0.34	0.90	0.49	164
russian	0.23	0.62	0.34	98
southern_us	0.82	0.41	0.54	864
spanish	0.35	0.44	0.39	198
thai	0.69	0.71	0.70	308
vietnamese	0.46	0.73	0.57	165
accuracy			0.63	7955
macro avg	0.55	0.66	0.56	7955
weighted avg	0.75	0.63	0.65	7955

Conclusion (Test Set Metrics)

- TF-IDF
  - Accuracy: 63%
  - Precision: 55%
  - Recall: 66%
- Bag of Words
  - Accuracy: 66.3%
  - Precision: 64.0%
  - Recall: 51.6%

- Accuracies are similar for both models
- Precision is higher in Bag of Words, Recall is higher in TF-IDF

Analysis for Further Work

- Bag of Words Model
  - Can include bi-grams (groupings of two words)
- Add filtering and use the Porter-Stemmer algorithm in the TF-IDF model
- Precision/Recall values can be used to further adjust the weights.
  - High precision + Low recall —>Increase the weights (Ex: Italian)
  - Low precision + High recall —>Reduce the weights (Ex: Jamaican)