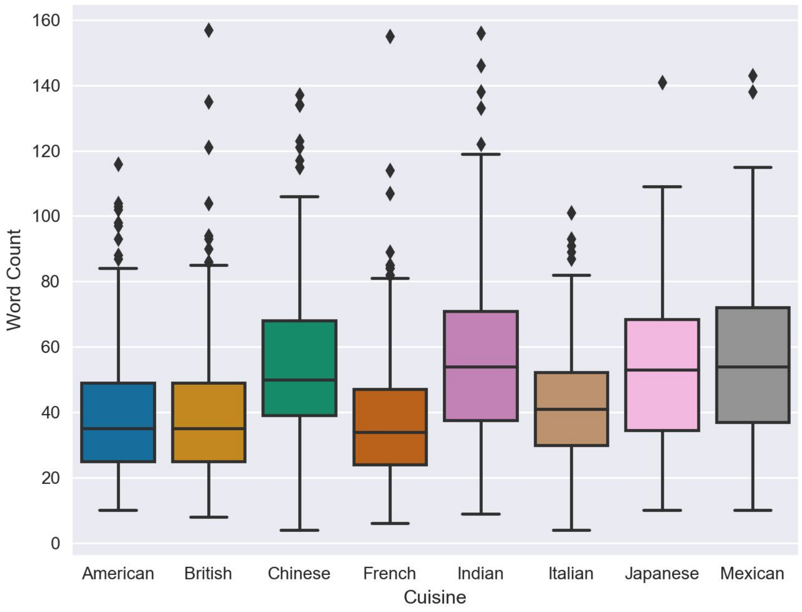
Recipe classification

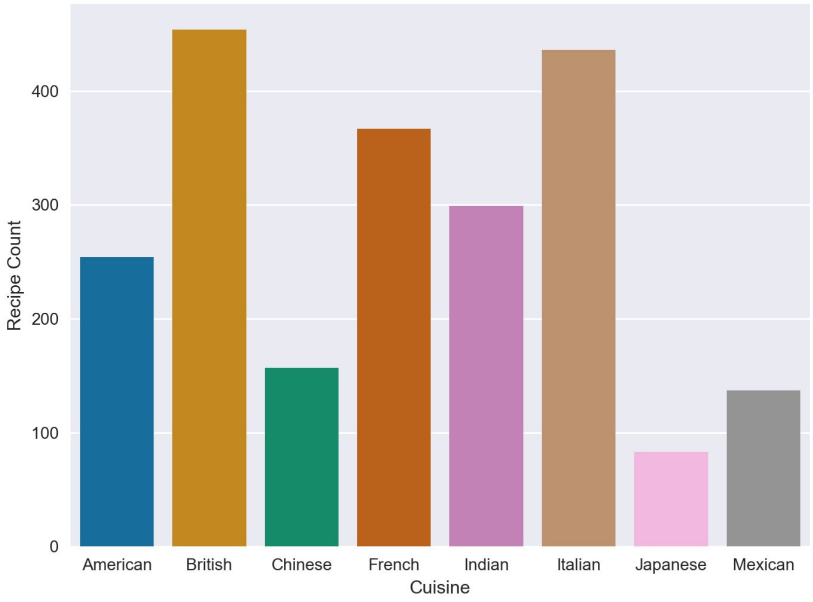
Hope you had lunch today!

Introduction and objectives

- The goal of this project was to classify various **recipes** by **cuisine**
- We scraped our data from **Allrecipes** and **BBC Food**, obtained over **2,000 observations** in **13 cuisine types**
- To classify our recipes we used **SVM classifier**, which improved after feature engineering
- To check to for general tendencies we also used **unsupervised clustering**
- Finally, we built a **Markov chain** to generate new recipes

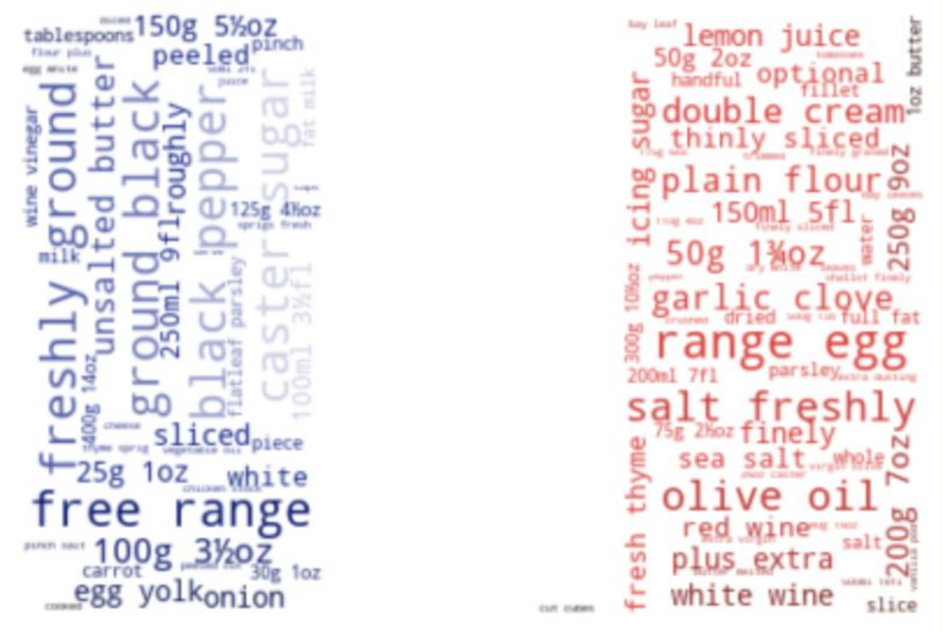
EDA. Class imbalance



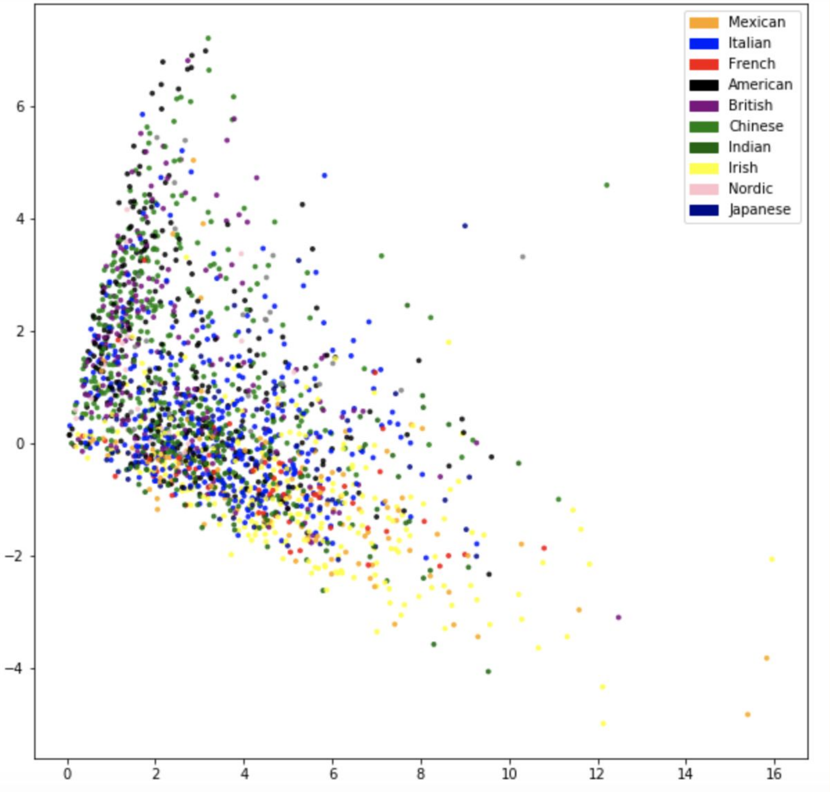


Word Clouds: Italian vs Japanese

Word Clouds: French vs Indian



Dimensionality reduction & Features



We figured out that getting rid of the **underrepresented** classes, such as **Nordic** and **Pakistani** cuisines, would improve our classification models.

Carefully examining this scatter plot also made us realize that **American cuisine** (black dots) imitates all others too much, so it would be smart to examine the data without US recipes, too.

Classifying: SVM with SDG training

accuracy 0.7227036395147314

	precision	recall	f1-score	support
Mexican	0.82	0.88	0.85	32
Italian	0.69	0.87	0.77	114
African	0.96	0.73	0.83	30
French	0.45	0.41	0.43	71
American	0.85	0.42	0.57	66
British	0.66	0.71	0.68	124
Chinese	0.77	0.97	0.86	31
Indian	0.86	0.97	0.91	77
Irish	1.00	0.14	0.25	7
Japanese	0.89	0.68	0.77	25
accuracy			0.72	577
macro avg	0.80	0.68	0.69	577
weighted avg	0.73	0.72	0.71	577

After removing underrepresented classes and US-recipes we tried several classifiers: **Naive Bayes**, **Logistic Regression** (generalized for multi class classification), and **Support Vector Machine** with **Stochastic Gradient Descent.**

SVM both improved the most and showed the best accuracy, as well as precision and recall for most classes.

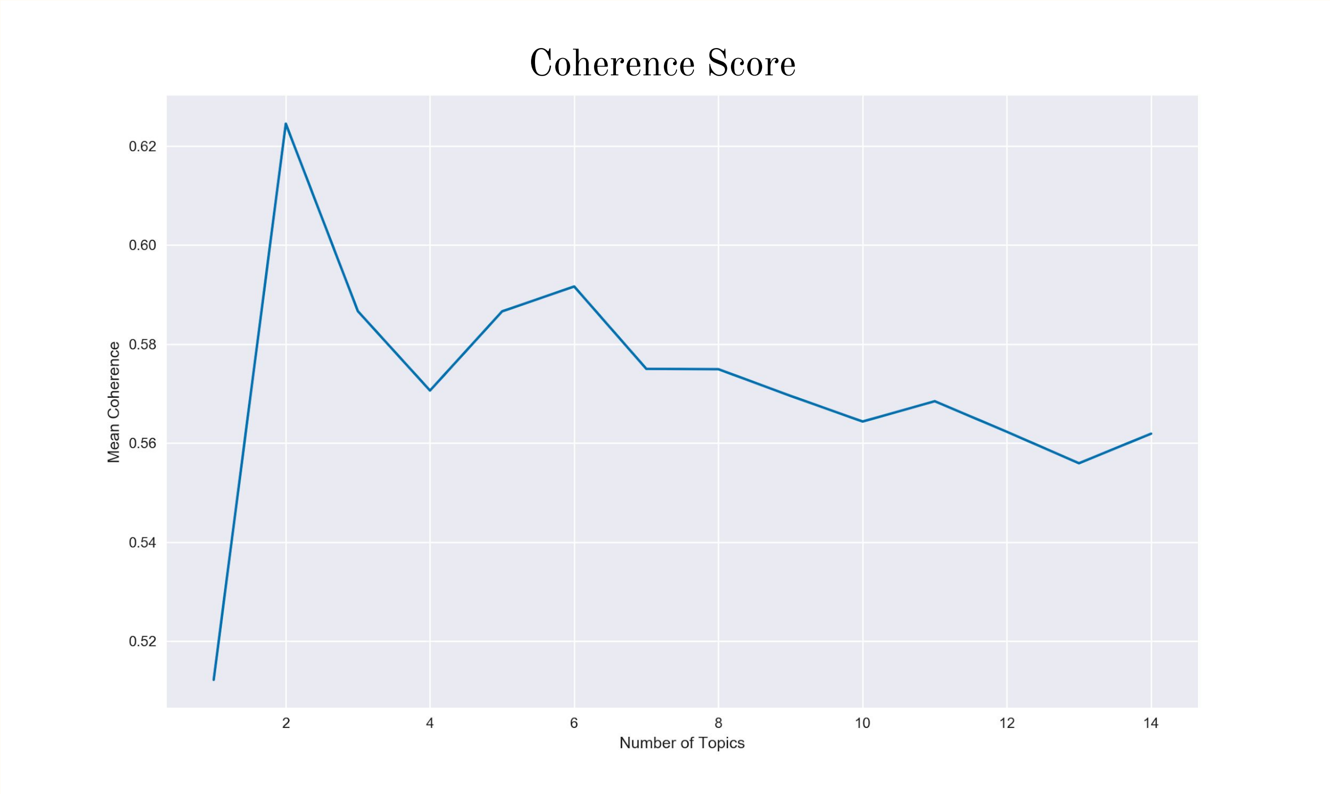
Clustering

Motivation: Understand why our model is plateauing around 72% accuracy, gain insight into natural groupings of our data

Modeling: Using Latent Dirichlet Allocation combined with pyLDAvis

Evaluation: Look for optimal number of clusters using basic Coherence Metric

Evaluating Clusters and Coherence

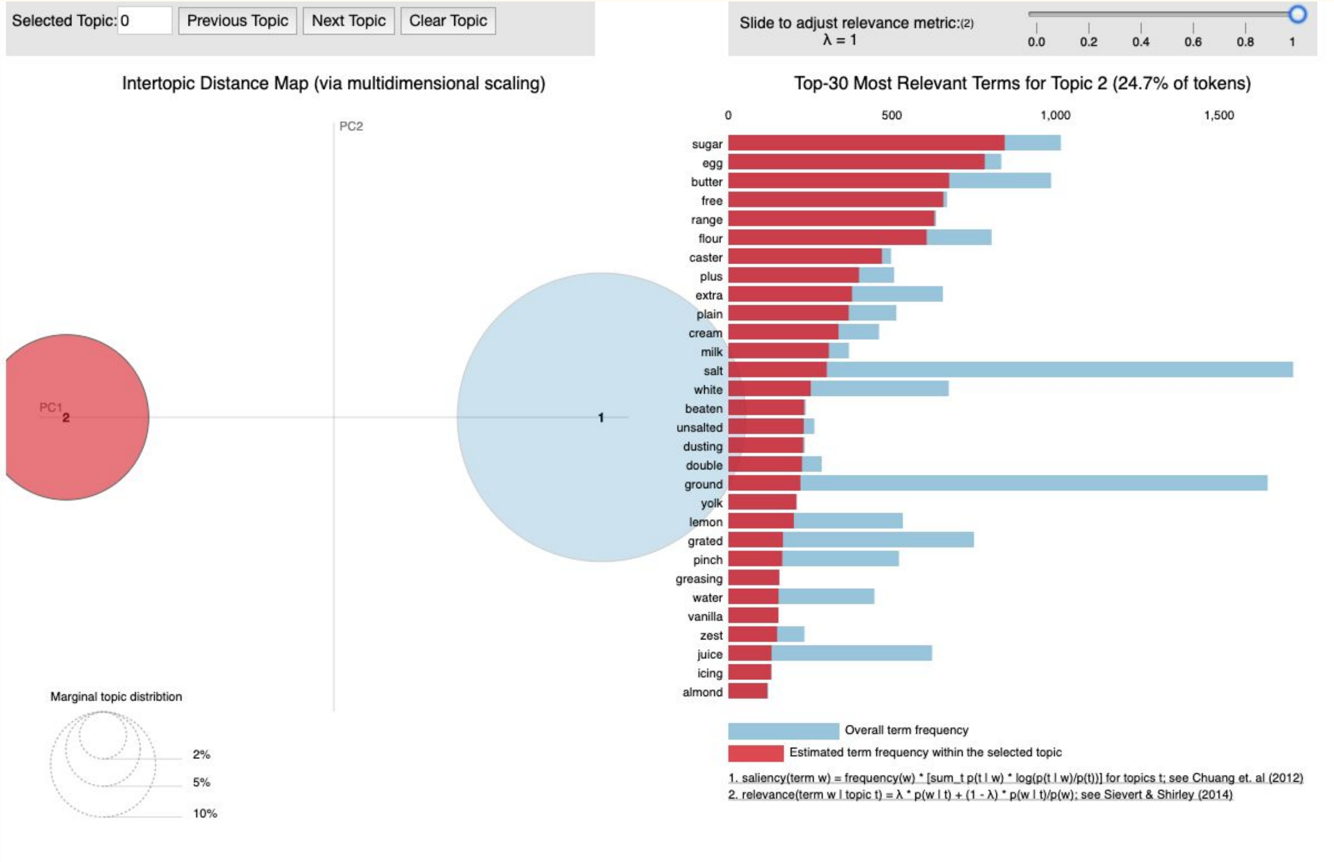


Coherence Score: A metric that evaluates a cluster topic by measuring the degree of semantic similarity between high scoring words in the topic.

Notable

- n=2
- n=6

Discovering Dessert: n=2



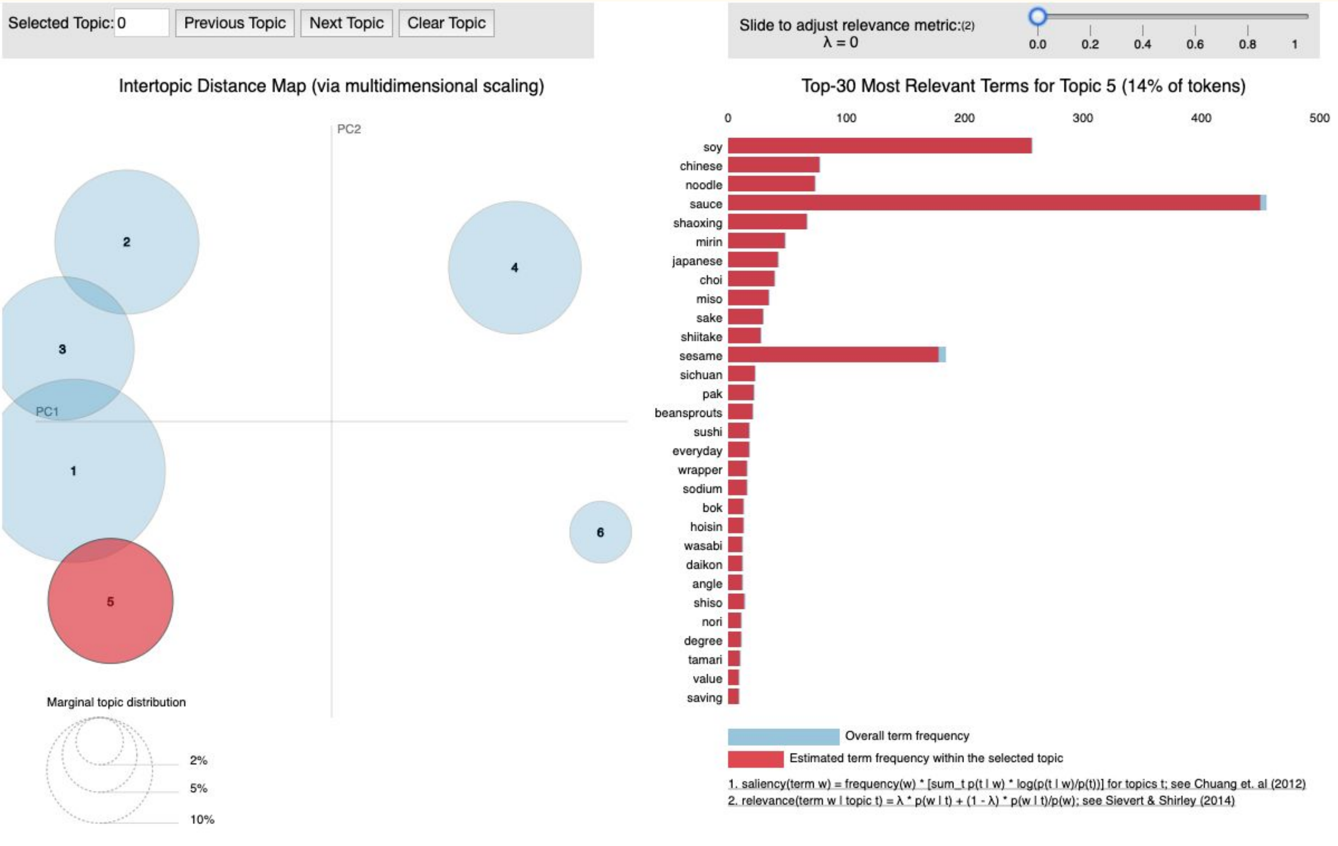
- Strongest separation between desserts and non-desserts.
- Desserts share common base ingredients across cuisines
  - Sugar
  - Egg
  - Butter
  - Flour
- Unpredicted class that is confusing the model

Uncovering Cuisines: n=6

Clusters Are Roughly:

1. Indian/Mexican
2. Italian
3. French
4. Desserts
5. Japanese/Chinese
6. Drinks

- Start to see actual cuisine types emerge
- But where is British Food??



Markov Chain and NLG

Finally, we took a chance to explore how **Markov Chain Neural Networks** can be useful for **Natural Language Generators**.

By training on the entire dataset, the NN ‘learns’ the probabilities of one word following the other, and executes a function that generates somewhat meaningful recipes.

Cookbook by Neural Net:

'2 loins of lamb or mutton neck fillet, diced, 3 tbsp tomato purée, 1 tbsp dill seeds, 1 tsp sea salt, plus extra leaves to serve, salt and pepper'

'2 tbsp crème fraîche, 300g/10½ oz fettucine or tagliatelle, cooked according to packet instructions'

'200g/7oz white sweet potatoes, 12-10 inch flour tortillas, 9 ounces shredded Cheddar cheese'

'2 Japanese aubergine cut into very small florets, 1 garlic clove roughly chopped to garnish’

Conclusions

- **Desserts** are actually a very **distinct category** of food, at least from the language perspective
- **American** cuisine imitates European and Asian, **confusing** the model
- In order to generate more **meaningful recipes** a neural net should probably train on a more **homogeneous** dataset
- Bon Appetit!