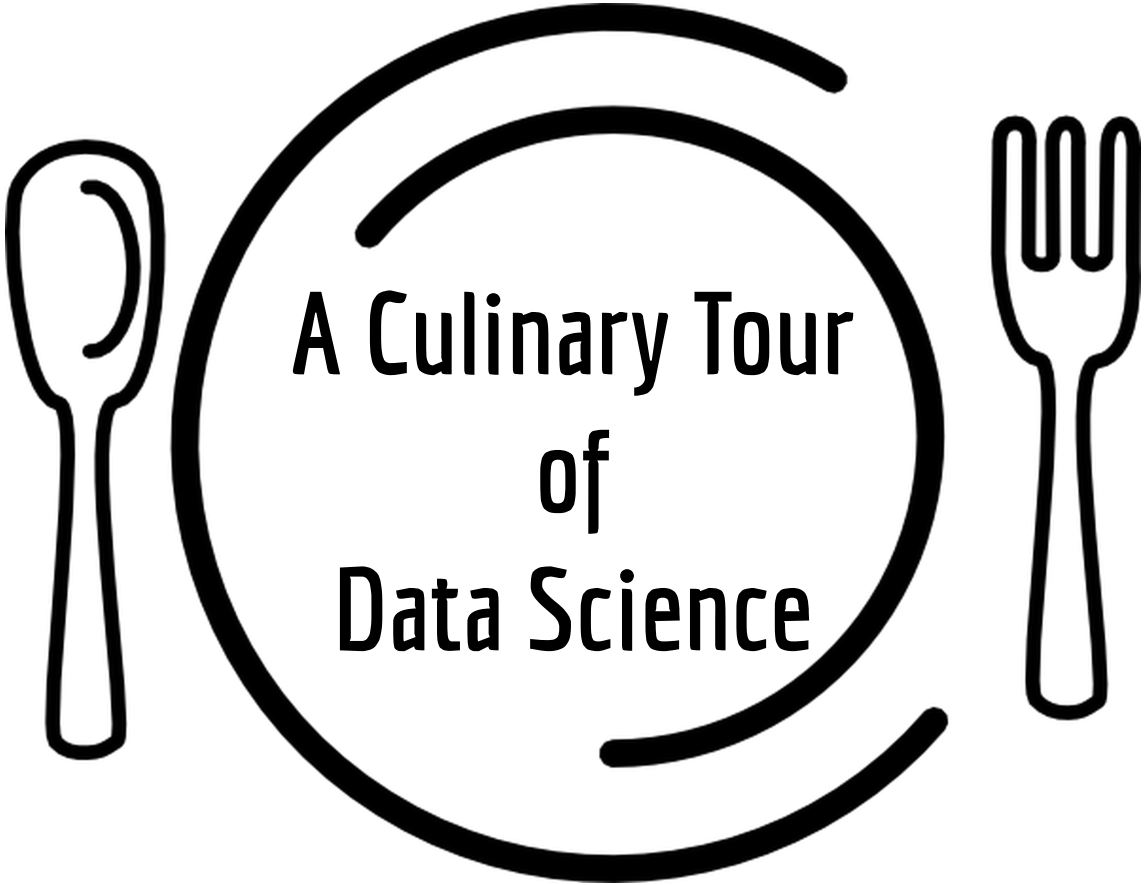
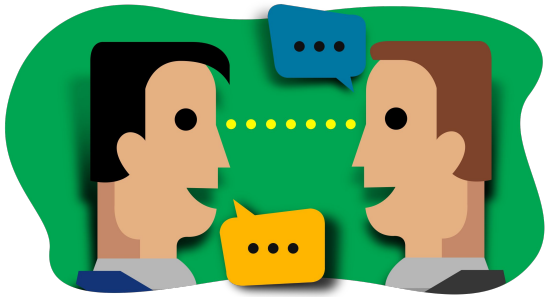


Team 36: Maria Katergi - Davit Martirosyan - Carla Ohanesian - Iuliana Voinea

Food

Social

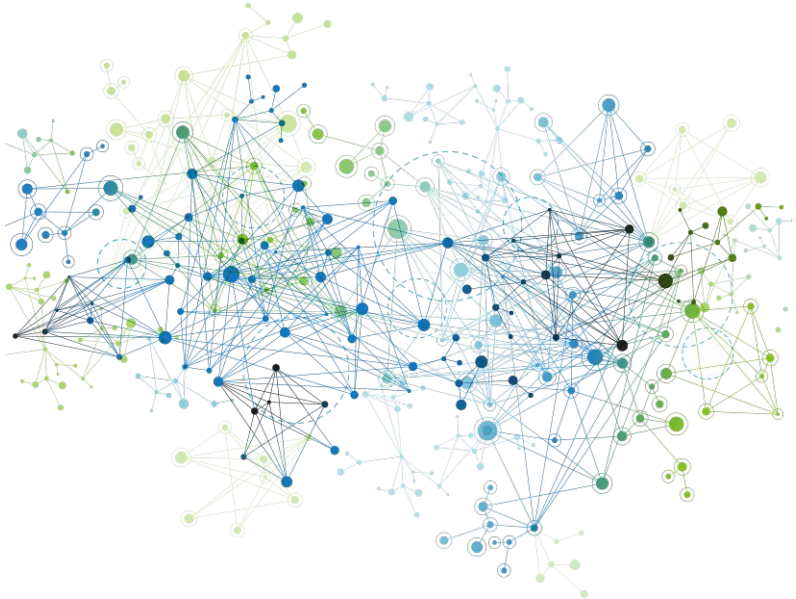


Physiological

Many recipes!

Goal

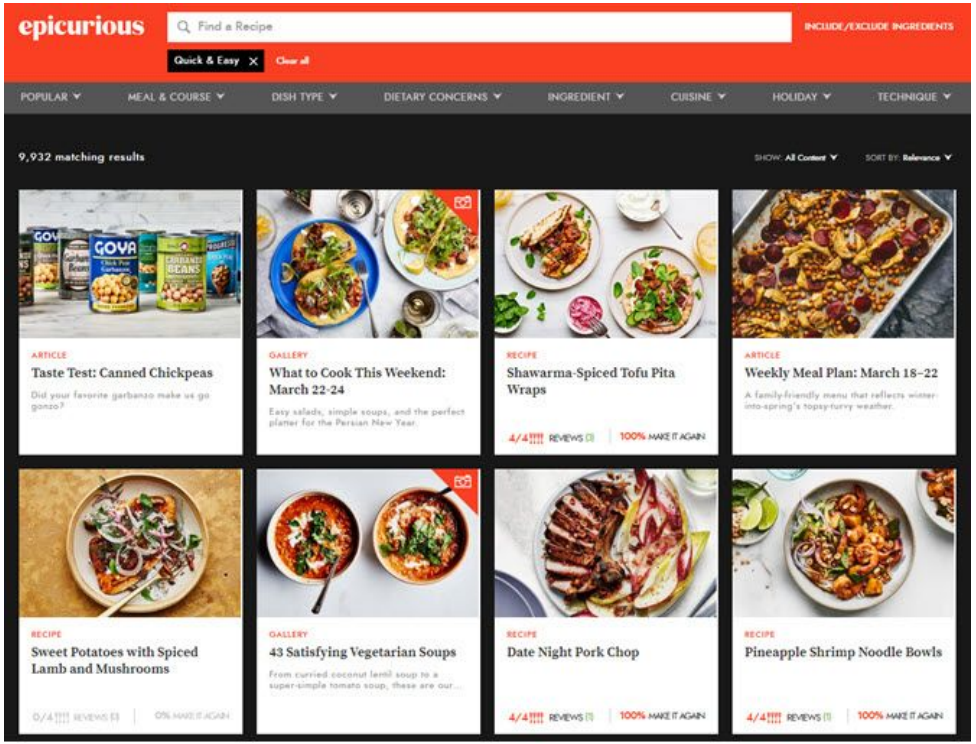
- Analyze several recipe features by creating a **dish network** in order to observe how different recipes relate to each other and how we can exploit these relationships to draw meaningful conclusions.



Data

More than **20K** recipes:

- title
- directions
- ingredients
- categories
- fat
- sodium
- protein
- calories
- rating



kaggle

Preprocessing

Drop useless columns → Drop imperfect entries → * Extract ingredients → **12,466 Recipes**

- ***desc*** column containing the recipe description

- Entries with ***rating* 0**
- Entries with **null** values
- **Duplicates**
- Entries with **outlier values** for:
  - calories
  - sodium
  - protein
  - fat

- The original ***ingredients*** column contained them listed together with *quantities*, *measures*, *stop words* and *utensils*
- **NLP** approach
- used **NLTK**

Building proper ingredient lists with NLP

```
['1 12-ounce package frozen spinach soufflé, thawed',
 '1/2 pound extra-wide egg noodles, freshly cooked',
 '1 cup sour cream',
 '2 tablespoons purchased pesto sauce',
 '1/4 teaspoon ground nutmeg',
 '1 cup grated sharp cheddar cheese']
```

```
['ground',
 'egg',
 'soufflé',
 'cheese',
 'spinach',
 'cream',
 'sauce',
 'noodle',
 'pesto',
 'cheddar']
```

Tokenize

Split the **free-text** from the original ingredients lists into tokens.

POS tagging

Keep only **Nouns.**

Remove unwanted words

From ***predefined list*** containing **quantities**, **measures**, **utensils**, **etc.**

Lemmatize

Map together different inflected and derived forms of the same ingredient.

Strip characters

Sometimes appear appended to ingredients (e.g. *).

Keep unique words

Ingredient lists can contain duplicate words.

Graph Construction

Subsampling: only 5000 recipes at random

Nodes: Recipes

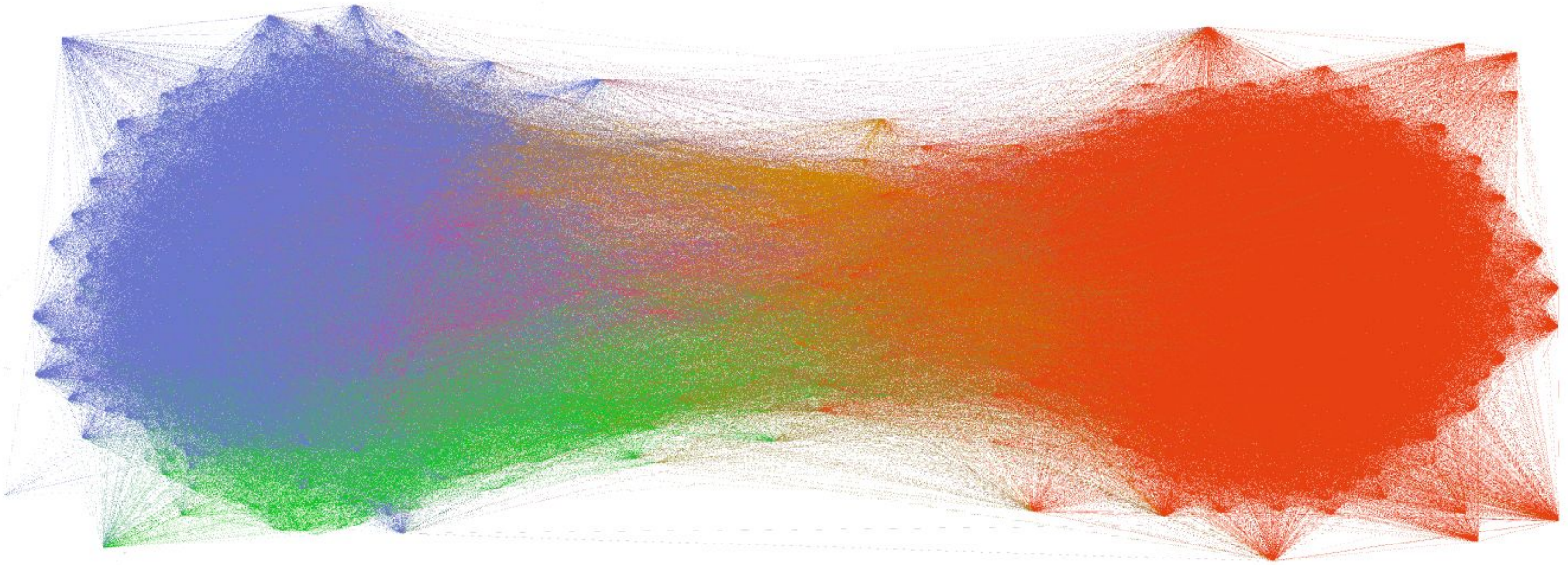
Edges: if common ingredients

Weights: Jaccard similarity $$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Thresholding: kept only **meaningful connections** at Jaccard similarity **>= 0.2**

-> reduced the number of edges from 8,685,453 to 230,352

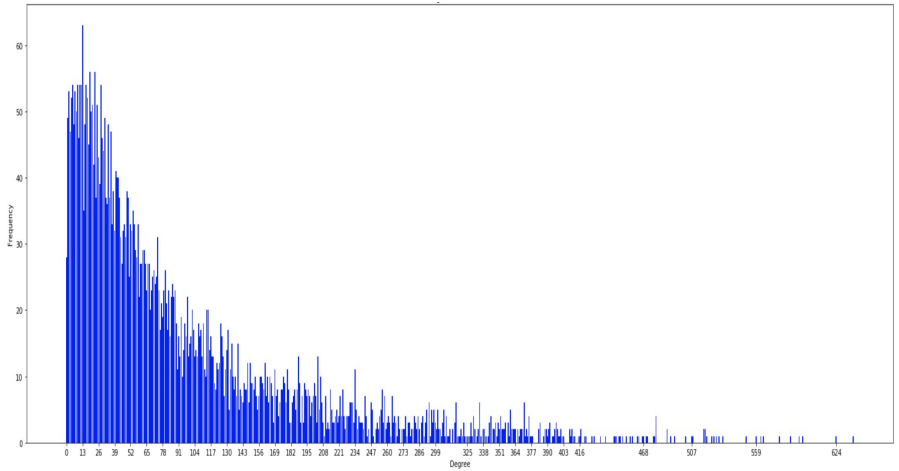
Gephi Visualization



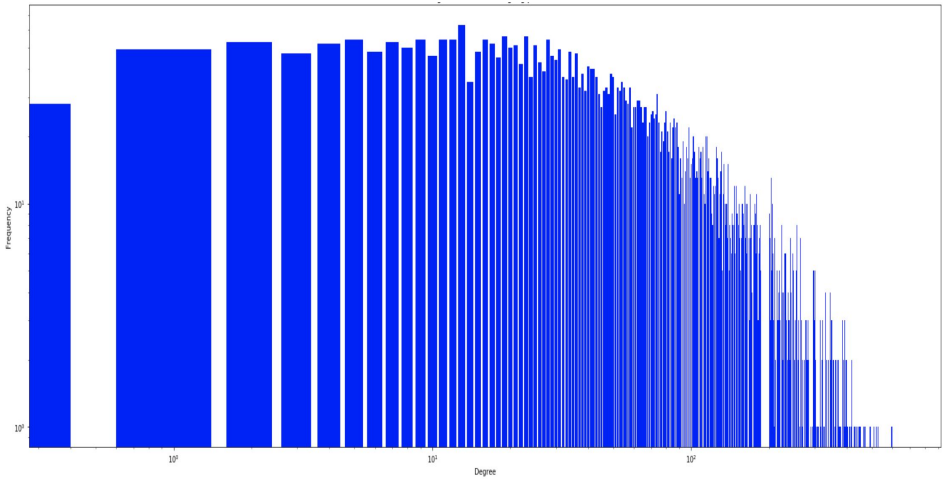
Graph Properties

Average degree	92.14
Sparsity	0.0184
Global clustering coefficient	0.3363
Average clustering coefficient	0.3152
Number of connected components	29
Percentage of nodes in largest component	99.44%
Diameter of largest component	7
Average shortest path length in largest component	2.7297
Degree distribution	Log-normal

Degree Distribution



Looks like a Power law



Log-log plot of degree distribution

⇒ **Log-normal** distribution

Community Detection - Clusters

Desserts
1017 nodes

Tarte Tatin, Pecan Shortbread Cookies, Apple-Raisin Bread Pudding

Spanish and Latin American Cuisine
1188 Dish

Spanish-Style Shrimp and Scallop Salad, Beef and Avocado Fajitas, Nicaraguan-Style Steak

European and Asian cuisine
1851 Dish

Melon and Prosciutto Risotto, Mussels alla Diavola, Duck Liver Pâté, Chinese Broccoli with Crabmeat, Coconut-Curry, Asian Noodle

Beverages & Ice Cream
417 Dish

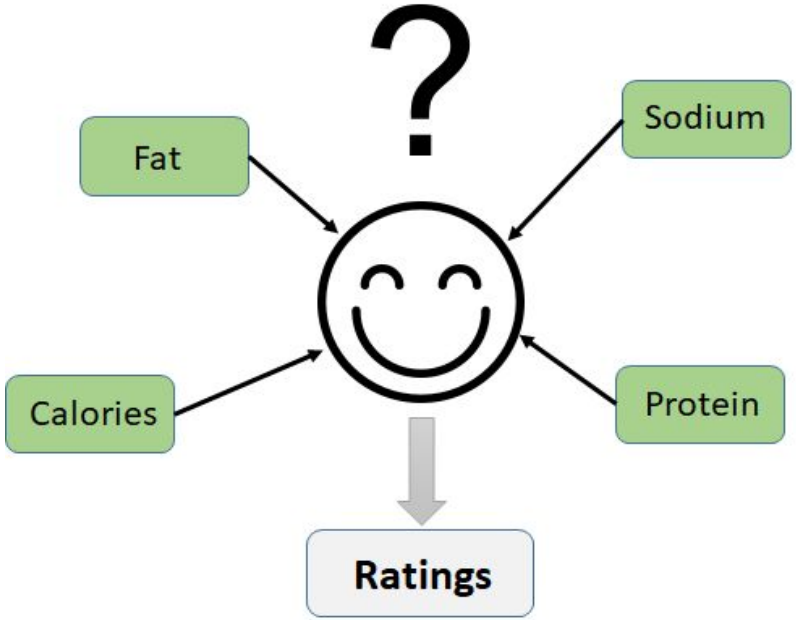
Strawberry & Cream Frappe, Earl Grey Rum Punch, Strawberry-Banana Smoothie, Homemade Ginger Ale

Simple Dishes mainly vegetarian
499 Dish

Mini Pizzas, Lettuce Soup, Mashed Potatoes with Kale, Potato Gratin with Goat Cheese and Garlic

Machine Learning with Graphs

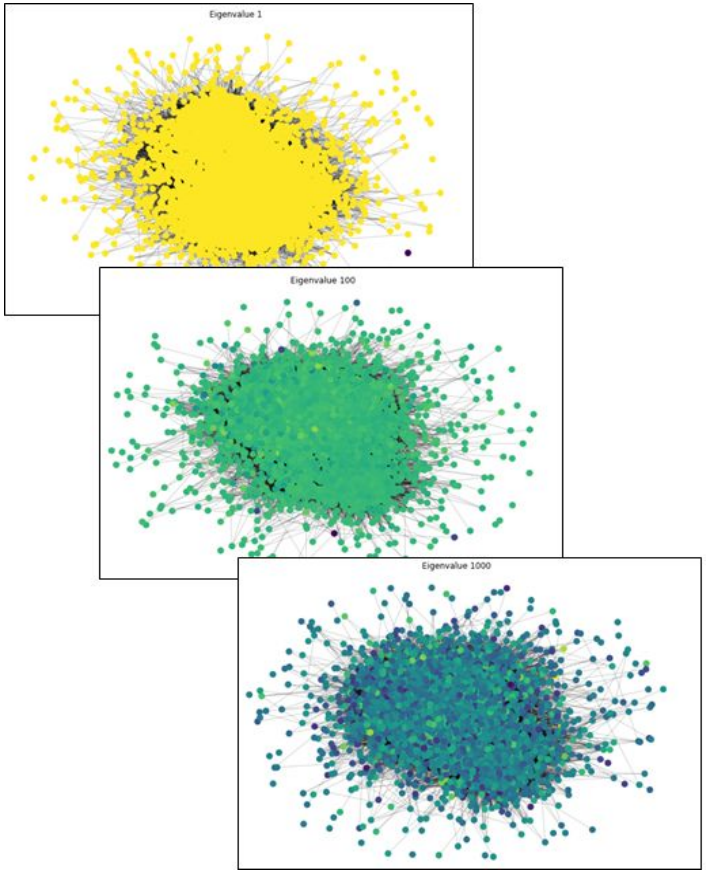
What nutritional factors affect people's culinary preferences?



Logistic Regression

Training accuracy: **0.592**
Validation accuracy: **0.590**
Test accuracy: **0.580**

Graph filtering



- Perform eigenvalue decomposition
- Plot eigenvalues as signals on graph

- Denoise features using filters
- Train new logistic regression model

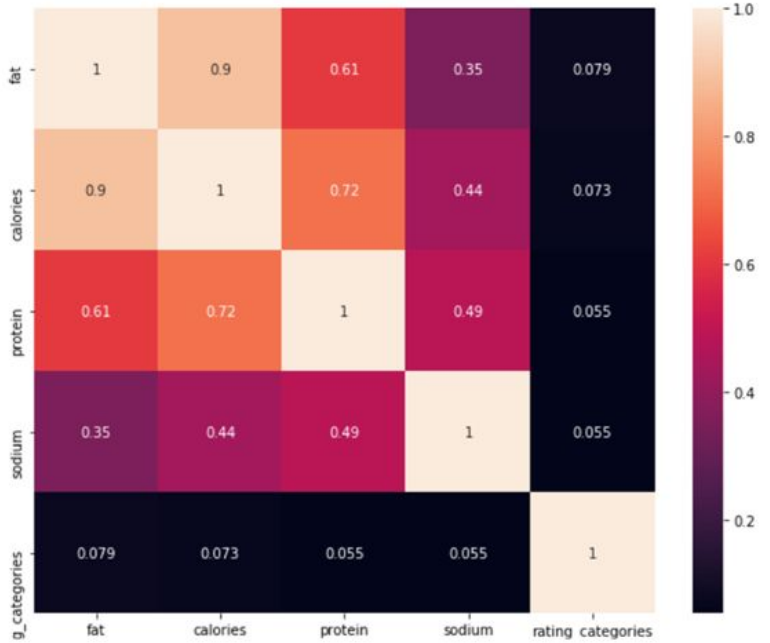
Training accuracy:	**0.5923**
Validation accuracy:	**0.5900**
Test accuracy:	**0.5820**

Reason 1: Imbalance in data

Class 0: Low rating	Class 1: Average rating	Class 2: Good rating
211	1836	2953

Imbalanced data leads to constant predictions for good ratings

Reason 2: Correlation



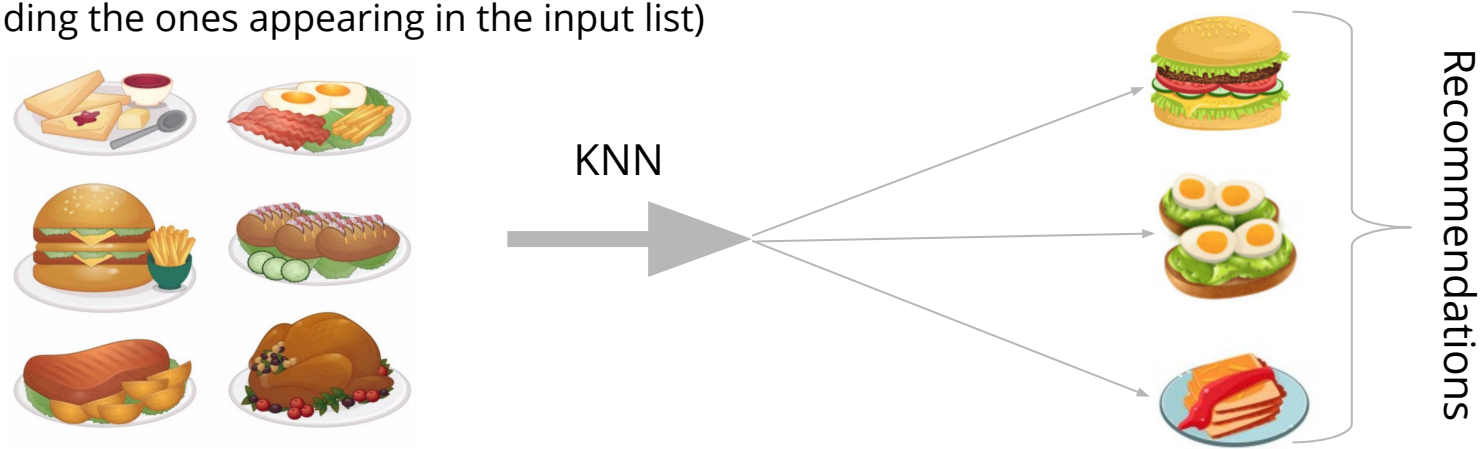
Very low correlation between features and ratings

KNN Recommender System

1. Take the k-highest rated recipes, K, from the given list of recipes
2. Iterate through all k chosen recipes giving a score to each connected recipe i as follows

$$score_i = \sum_{k \in K} jaccard\_sim(i, k)$$

3. Sort the recipes based on the scores in descending order and return the first n recipes (excluding the ones appearing in the input list)

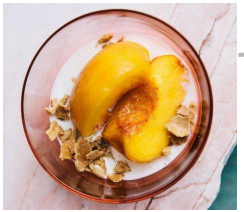


Top-5 Recommendation Example

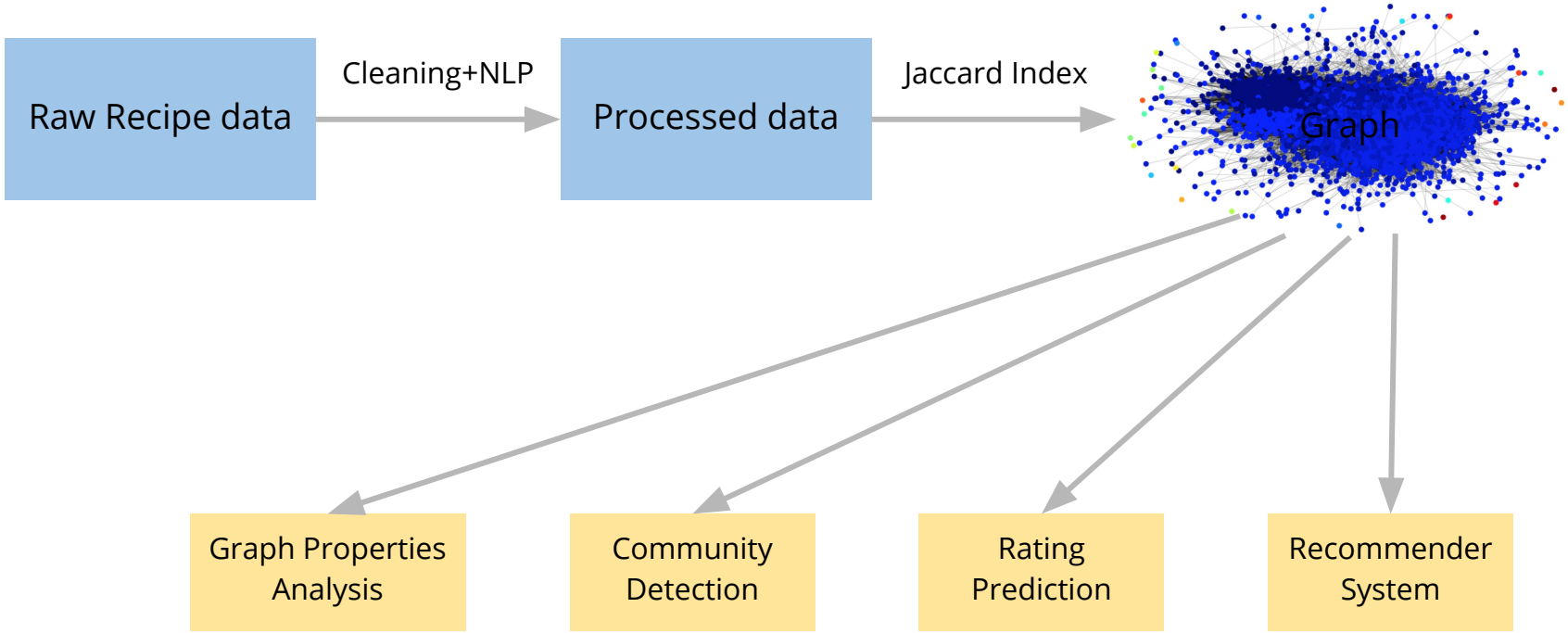
1. Crushed-Mint Lemonade
2. Ginger-Honey Lemonade
3. Papaya Smoothie
4. Oranges with Pomegranate Molasses and Honey
5. 'Pine Nut Brittle'
6. 'Sweet Avocado Mousse'
7. 'Gimlet'
8. 'Campari Citrus Cooler'
9. 'Tangerine Granita'
10. 'Chocolate, Cherry and Marsala Cassata'
11. 'Ginger Pudding'
12. 'Trifle with Strawberries and Caramel-Coated Bananas'
13. 'Thin Apple Tarts'
14. 'Shortbread Cookies'
15. 'Burnt-Caramel Custards'
16. 'Peach White-Wine Sangria'

KNN
(k=10)

1. Butter Pie Crust Dough
2. Classic Sour Cherry Pie with Lattice Crust
3. Florida Punch
4. Peaches in Ginger Syrup
5. Cantaloupe Granita



Conclusion


Raw Recipe data
Cleaning+NLP
Processed data
Jaccard Index
Graph
Graph Properties Analysis
Community Detection
Rating Prediction
Recommender System

Thank you for your attention!
Any Questions?