The Home Cooks Guide to Ingredients and Spices

Final Project: Data and Analytics Bootcamp - Team 8

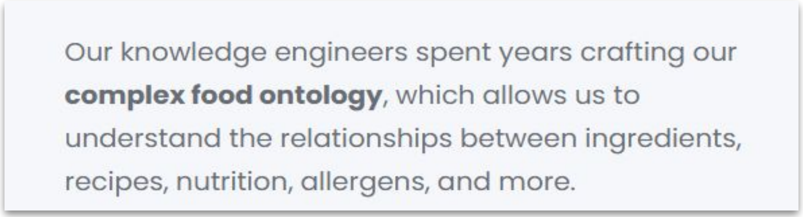
Charlie Willmore, Brittany Garza and Lauren Neidhardt

Project Selection

Charlie had an idea for the average at home chef to be able to have way to know which herbs and spices go well together.

We all love to cook and we all have many random, unused spices in our cabinets so these questions seem like a natural fit.

Project Data: Spoonacular



Our knowledge engineers spent years crafting our **complex food ontology**, which allows us to understand the relationships between ingredients, recipes, nutrition, allergens, and more.

The Spoonacular API is an online resource containing:

- Ingredients
- Recipes
- Product Information
- Menu Items (restaurants)

menu items
products
recipes

Questions we hope to answer with the data:

1st Question:
Which ingredients occur in recipes of the same cuisine most frequently?

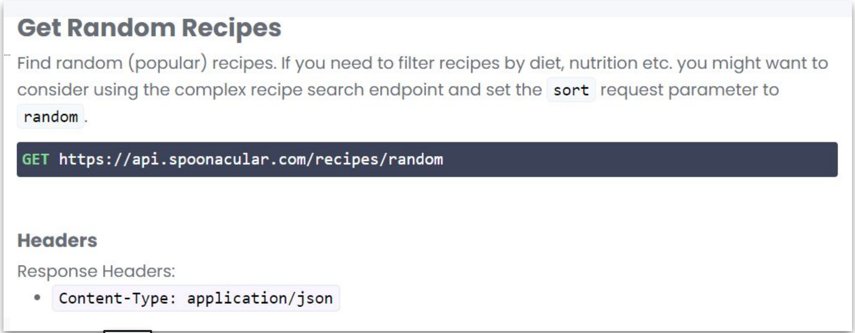
2nd Question:
Can we use them as a predictor of cuisine?

3rd Question:
Which spices are frequently used together in recipes?

Data Exploration

Initial Data Processing:

- Data downloaded via Spoonacular API
- Performing a random recipe search, 100 random recipes were downloaded in each call.
- Data pulled from JSON to limit attributes to recipe, cuisine and ingredient names
- Created a for loop to pull 10k recipes and then merged them all into single dataframe.



Get Random Recipes

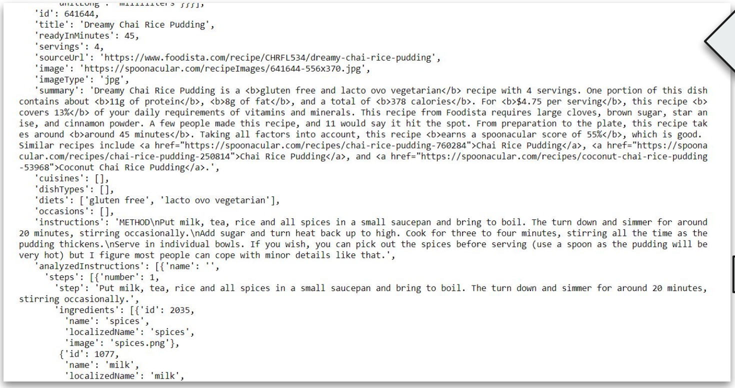
Find random (popular) recipes. If you need to filter recipes by diet, nutrition etc. you might want to consider using the complex recipe search endpoint and set the `sort` request parameter to `random`.

```
GET https://api.spoonacular.com/recipes/random
```

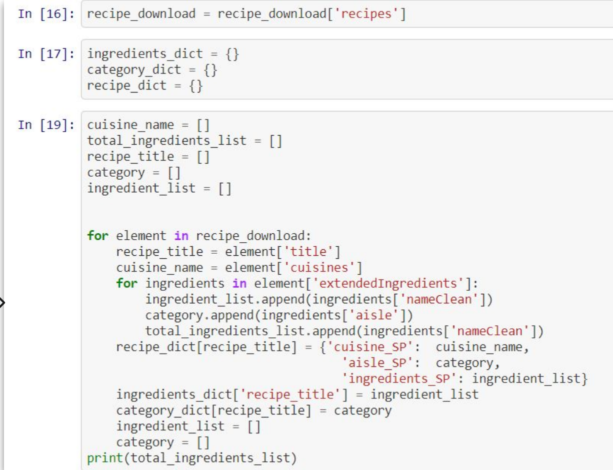
Headers

Response Headers:

- `Content-Type: application/json`



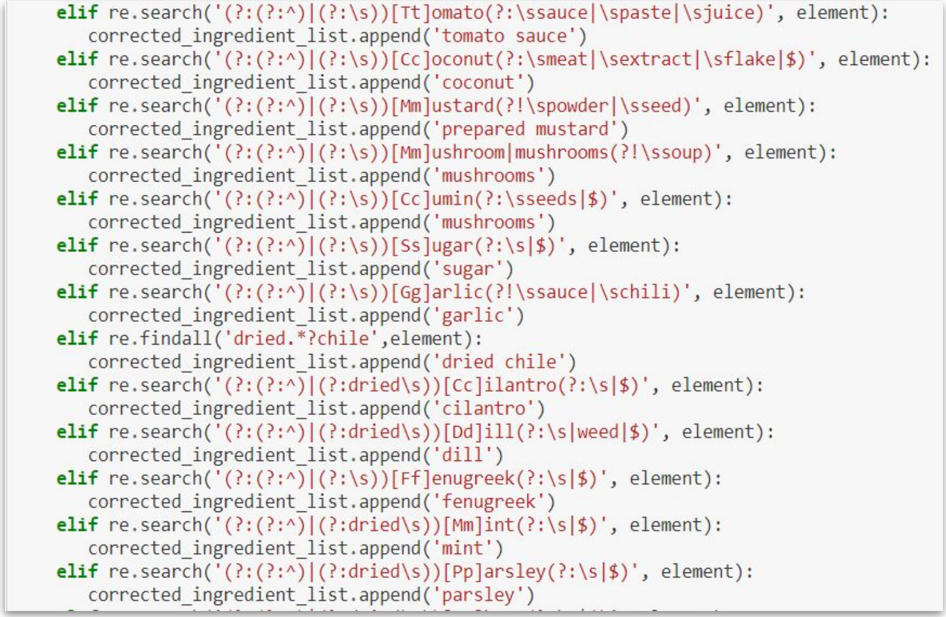
```
         'unitLong': 'milliliters'}}}],
   'id': 641644,
   'title': 'Dreamy Chai Rice Pudding',
   'readyInMinutes': 45,
   'servings': 4,
   'sourceUrl': 'https://www.foodista.com/recipe/CHRFL534/dreamy-chai-rice-pudding',
   'image': 'https://spoonacular.com/recipeImages/641644-556x370.jpg',
   'imageType': 'jpg',
   'summary': 'Dreamy Chai Rice Pudding is a <b>gluten free and lacto ovo vegetarian</b> recipe with 4 servings. One portion of this dish
contains about <b>11g of protein</b>, <b>8g of fat</b>, and a total of <b>378 calories</b>. For <b>$4.75 per serving</b>, this recipe <b>
covers 13%</b> of your daily requirements of vitamins and minerals. This recipe from Foodista requires large cloves, brown sugar, star an
ise, and cinnamon powder. A few people made this recipe, and 11 would say it hit the spot. From preparation to the plate, this recipe tak
es around <b>around 45 minutes</b>. Taking all factors into account, this recipe <b>earns a spoonacular score of 55%</b>, which is good.
Similar recipes include <a href="https://spoonacular.com/recipes/chai-rice-pudding-760284">Chai Rice Pudding</a>, <a href="https://spoona
cular.com/recipes/chai-rice-pudding-250814">Chai Rice Pudding</a>, and <a href="https://spoonacular.com/recipes/coconut-chai-rice-pudding
-53968">Coconut Chai Rice Pudding</a>.',
   'cuisines': [],
   'dishTypes': [],
   'diets': ['gluten free', 'lacto ovo vegetarian'],
   'occasions': [],
   'instructions': 'METHOD\nPut milk, tea, rice and all spices in a small saucepan and bring to boil. The turn down and simmer for around
20 minutes, stirring occasionally.\nAdd sugar and turn heat back up to high. Cook for three to four minutes, stirring all the time as the
pudding thickens.\nServe in individual bowls. If you wish, you can pick out the spices before serving (use a spoon as the pudding will be
very hot) but I figure most people can cope with minor details like that.',
   'analyzedInstructions': [{'name': '',
     'steps': [{'number': 1,
       'step': 'Put milk, tea, rice and all spices in a small saucepan and bring to boil. The turn down and simmer for around 20 minutes,
stirring occasionally.',
       'ingredients': [{'id': 2035,
         'name': 'spices',
         'localizedName': 'spices',
         'image': 'spices.png'},
        {'id': 1077,
         'name': 'milk',
         'localizedName': 'milk',
```



```
In [16]: recipe_download = recipe_download['recipes']

In [17]: ingredients_dict = {}
         category_dict = {}
         recipe_dict = {}

In [19]: cuisine_name = []
         total_ingredients_list = []
         recipe_title = []
         category = []
         ingredient_list = []


         for element in recipe_download:
             recipe_title = element['title']
             cuisine_name = element['cuisines']
             for ingredients in element['extendedIngredients']:
                 ingredient_list.append(ingredients['nameClean'])
                 category.append(ingredients['aisle'])
                 total_ingredients_list.append(ingredients['nameClean'])
             recipe_dict[recipe_title] = {'cuisine_SP':  cuisine_name,
                                          'aisle_SP':  category,
                                          'ingredients_SP': ingredient_list}
             ingredients_dict['recipe_title'] = ingredient_list
             category_dict[recipe_title] = category
             ingredient_list = []
             category = []
         print(total_ingredients_list)
```

Data Exploration

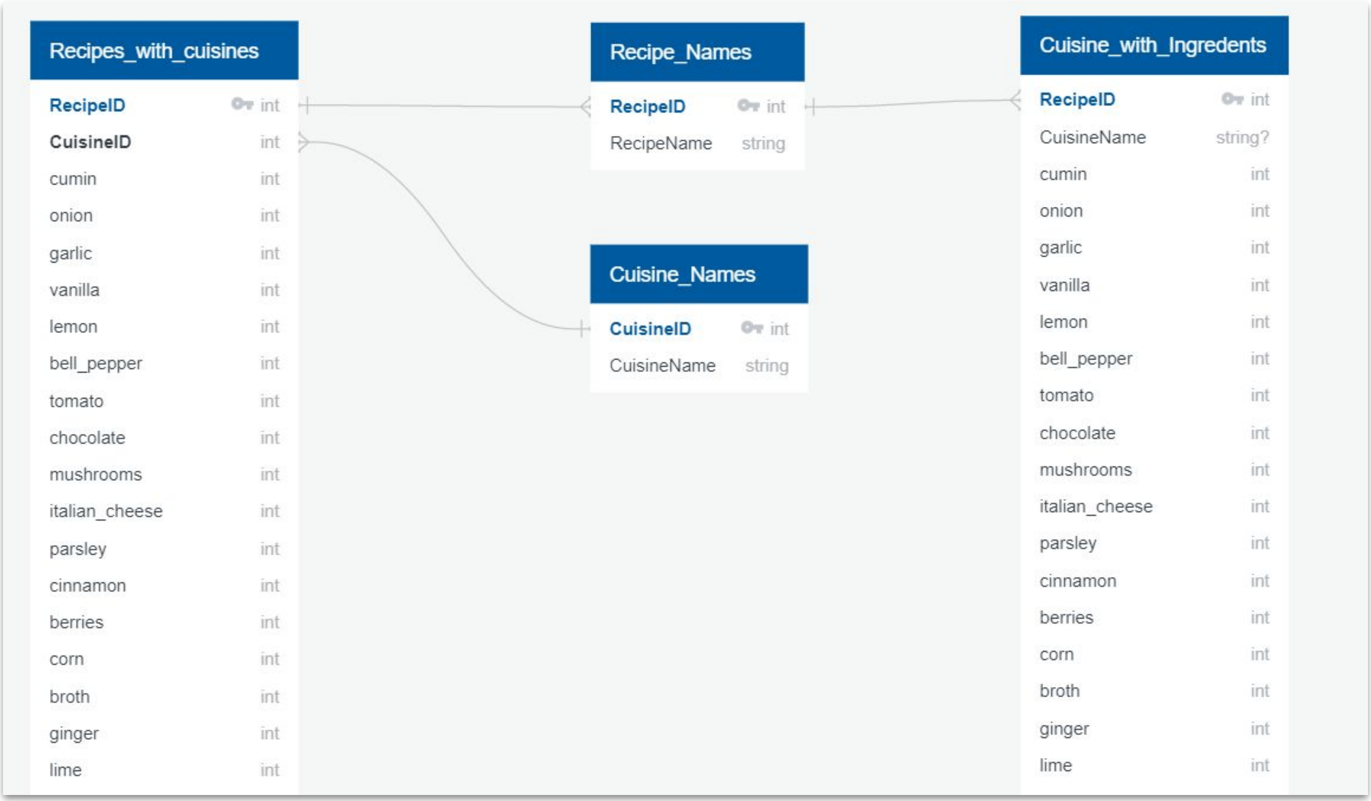
- In Pandas, data converted from string to a list
- Used Regex to create a consistent, clean list of ingredients
- Drop overly common items like salt and pepper, oil



```
elif re.search('(?:(?:^)|(?:\s))[Tt]omato(?:\ssauce|\spaste|\sjuice)', element):
   corrected_ingredient_list.append('tomato sauce')
elif re.search('(?:(?:^)|(?:\s))[Cc]oconut(?:\smeat|\sextract|\sflake|$)', element):
   corrected_ingredient_list.append('coconut')
elif re.search('(?:(?:^)|(?:\s))[Mm]ustard(?!\spowder|\sseed)', element):
   corrected_ingredient_list.append('prepared mustard')
elif re.search('(?:(?:^)|(?:\s))[Mm]ushroom|mushrooms(?!\ssoup)', element):
   corrected_ingredient_list.append('mushrooms')
elif re.search('(?:(?:^)|(?:\s))[Cc]umin(?:\sseeds|$)', element):
   corrected_ingredient_list.append('mushrooms')
elif re.search('(?:(?:^)|(?:\s))[Ss]ugar(?:\s|$)', element):
   corrected_ingredient_list.append('sugar')
elif re.search('(?:(?:^)|(?:\s))[Gg]arlic(?!\ssauce|\schili)', element):
   corrected_ingredient_list.append('garlic')
elif re.findall('dried.*?chile',element):
   corrected_ingredient_list.append('dried chile')
elif re.search('(?:(?:^)|(?:dried\s))[Cc]ilantro(?:\s|$)', element):
   corrected_ingredient_list.append('cilantro')
elif re.search('(?:(?:^)|(?:dried\s))[Dd]ill(?:\s|weed|$)', element):
   corrected_ingredient_list.append('dill')
elif re.search('(?:(?:^)|(?:\s))[Ff]enugreek(?:\s|$)', element):
   corrected_ingredient_list.append('fenugreek')
elif re.search('(?:(?:^)|(?:dried\s))[Mm]int(?:\s|$)', element):
   corrected_ingredient_list.append('mint')
elif re.search('(?:(?:^)|(?:dried\s))[Pp]arsley(?:\s|$)', element):
   corrected_ingredient_list.append('parsley')
```

Project Database

ERD showing table relationships



Recipes_with_cuisines	
RecipeID	🔑 int
CuisineID	int
cumin	int
onion	int
garlic	int
vanilla	int
lemon	int
bell_pepper	int
tomato	int
chocolate	int
mushrooms	int
italian_cheese	int
parsley	int
cinnamon	int
berries	int
corn	int
broth	int
ginger	int
lime	int

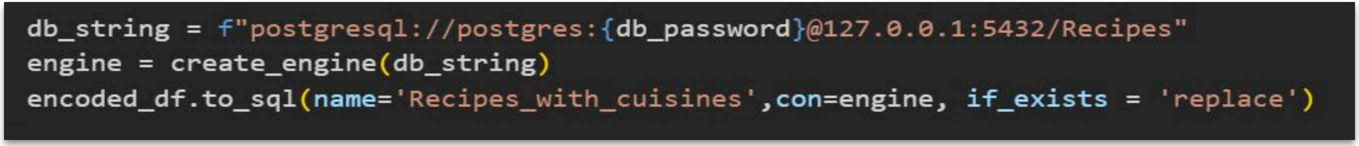
Recipe_Names	
RecipeID	🔑 int
RecipeName	string

Cuisine_Names	
CuisineID	🔑 int
CuisineName	string

Cuisine_with_Ingredents	
RecipeID	🔑 int
CuisineName	string?
cumin	int
onion	int
garlic	int
vanilla	int
lemon	int
bell_pepper	int
tomato	int
chocolate	int
mushrooms	int
italian_cheese	int
parsley	int
cinnamon	int
berries	int
corn	int
broth	int
ginger	int
lime	int

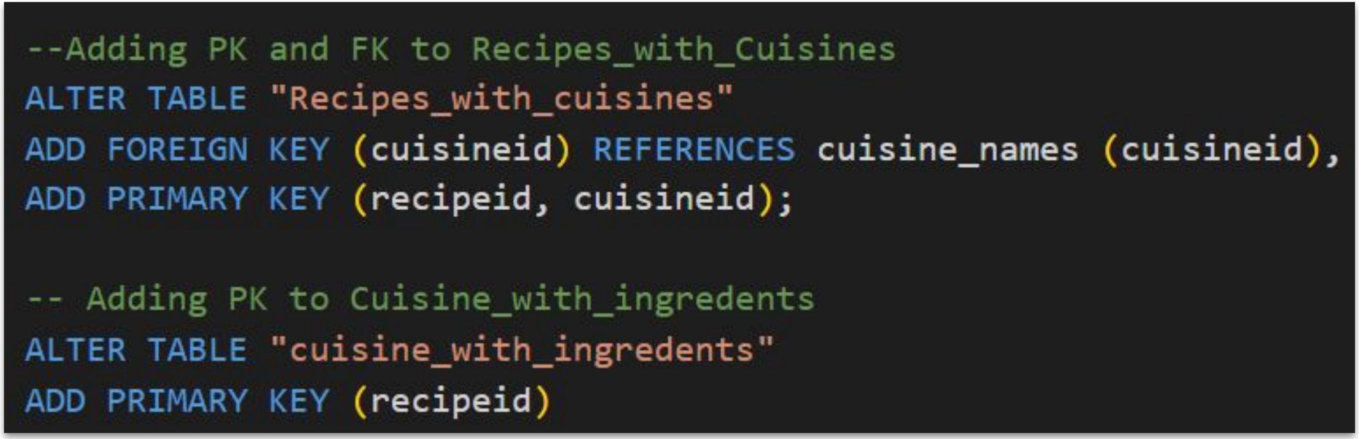
Importing tables to Postgres

- Used SQLAlchemy to import tables with 300+ columns



```
db_string = f"postgresql://postgres:{db_password}@127.0.0.1:5432/Recipes"
engine = create_engine(db_string)
encoded_df.to_sql(name='Recipes_with_cuisines',con=engine, if_exists = 'replace')
```

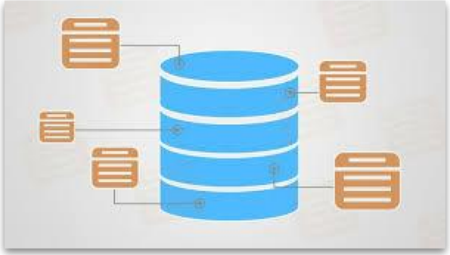
- Declared the primary and foreign keys within Postgres

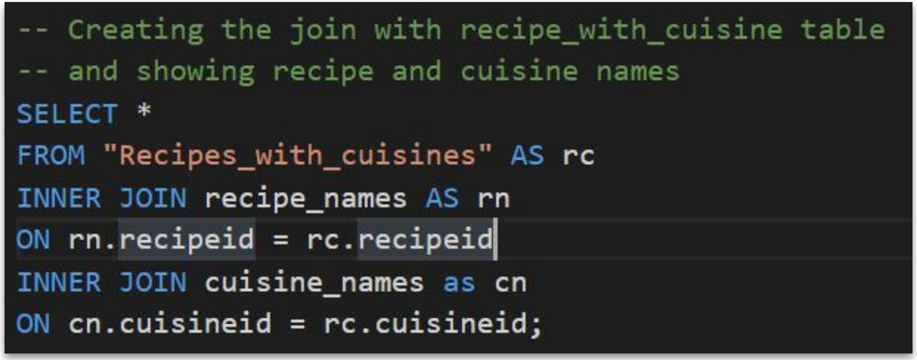


```
--Adding PK and FK to Recipes_with_Cuisines
ALTER TABLE "Recipes_with_cuisines"
ADD FOREIGN KEY (cuisineid) REFERENCES cuisine_names (cuisineid),
ADD PRIMARY KEY (recipeid, cuisineid);

-- Adding PK to Cuisine_with_ingredents
ALTER TABLE "cuisine_with_ingredents"
ADD PRIMARY KEY (recipeid)
```

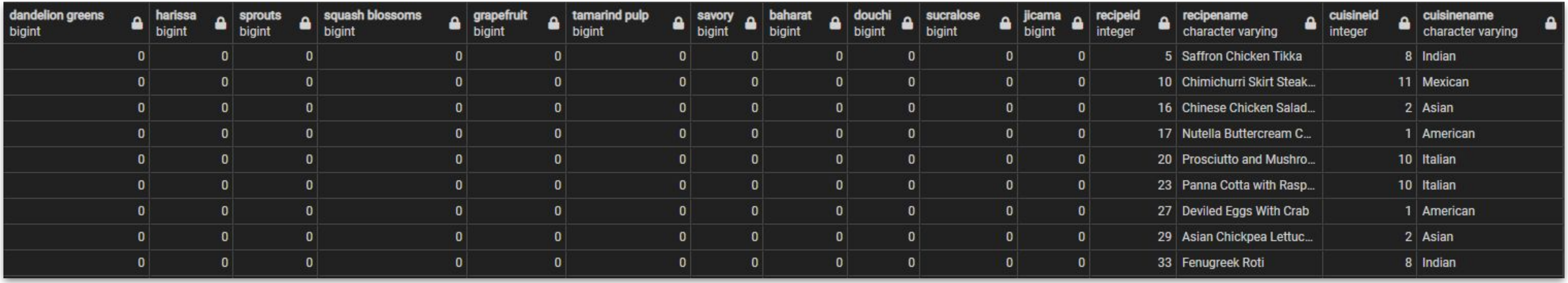
Joining the tables in SQL

- Created join to display the relationship of cuisine and recipe





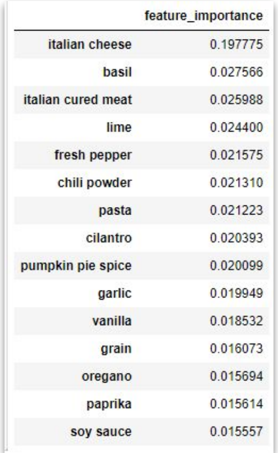
```sql
-- Creating the join with recipe_with_cuisine table
-- and showing recipe and cuisine names
SELECT *
FROM "Recipes_with_cuisines" AS rc
INNER JOIN recipe_names AS rn
ON rn.recipeid = rc.recipeid
INNER JOIN cuisine_names as cn
ON cn.cuisineid = rc.cuisineid;
```



dandelion greens bigint	harissa bigint	sprouts bigint	squash blossoms bigint	grapefruit bigint	tamarind pulp bigint	savory bigint	baharat bigint	douchi bigint	sucralose bigint	jicama bigint	recipeid integer	recipename character varying	cuisineid integer	cuisinename character varying
0	0	0	0	0	0	0	0	0	0	0	5	Saffron Chicken Tikka	8	Indian
0	0	0	0	0	0	0	0	0	0	0	10	Chimichurri Skirt Steak…	11	Mexican
0	0	0	0	0	0	0	0	0	0	0	16	Chinese Chicken Salad…	2	Asian
0	0	0	0	0	0	0	0	0	0	0	17	Nutella Buttercream C…	1	American
0	0	0	0	0	0	0	0	0	0	0	20	Prosciutto and Mushro…	10	Italian
0	0	0	0	0	0	0	0	0	0	0	23	Panna Cotta with Rasp…	10	Italian
0	0	0	0	0	0	0	0	0	0	0	27	Deviled Eggs With Crab	1	American
0	0	0	0	0	0	0	0	0	0	0	29	Asian Chickpea Lettuc…	2	Asian
0	0	0	0	0	0	0	0	0	0	0	33	Fenugreek Roti	8	Indian

What are the principal ingredients in a cuisine?

Method 1:
Feature importances from random forest classifier

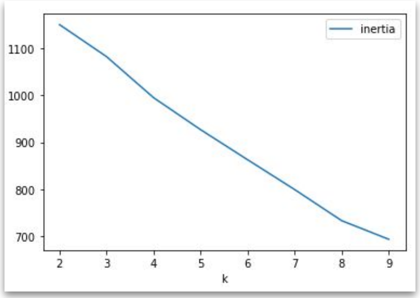


	feature_importance
italian cheese	0.197775
basil	0.027566
italian cured meat	0.025988
lime	0.024400
fresh pepper	0.021575
chili powder	0.021310
pasta	0.021223
cilantro	0.020393
pumpkin pie spice	0.020099
garlic	0.019949
vanilla	0.018532
grain	0.016073
oregano	0.015694
paprika	0.015614
soy sauce	0.015557

**Identifies ingredients critical to separate cuisines, not the principal ingredients of a cuisine*

Method 2:
PCA and KMeans

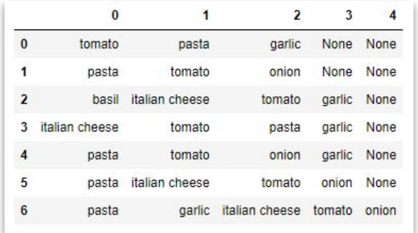
Recipes with similar ingredients should naturally cluster together. Result should be sets of ingredients within a cuisine that go together- based on being together in recipes



***Elbow plots suggest a lack of natural grouping.*

Method 3:
Market Basket Analysis- statistics rather than ML

Group ingredients based on cutoffs:
Support: % of recipes that contain A
10%
Confidence: % of recipes that contain A and B relative to % that contain A
30%
Lift: Confidence A given B / support B
3.0



	0	1	2	3	4
0	tomato	pasta	garlic	None	None
1	pasta	tomato	onion	None	None
2	basil	italian cheese	tomato	garlic	None
3	italian cheese	tomato	pasta	garlic	None
4	pasta	tomato	onion	garlic	None
5	pasta	italian cheese	tomato	onion	None
6	pasta	garlic	italian cheese	tomato	onion

****For the Italian recipes, 7 relationships*
For Asian recipes, 29 relationships

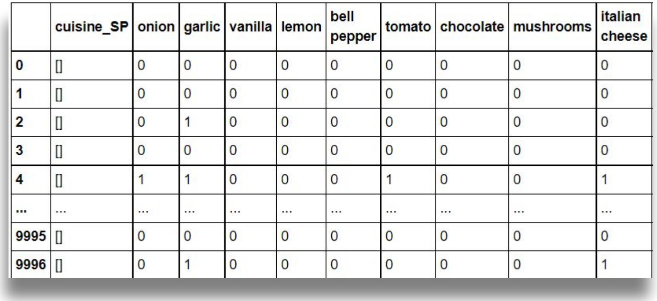
How do you interpret the results?

Random Forest

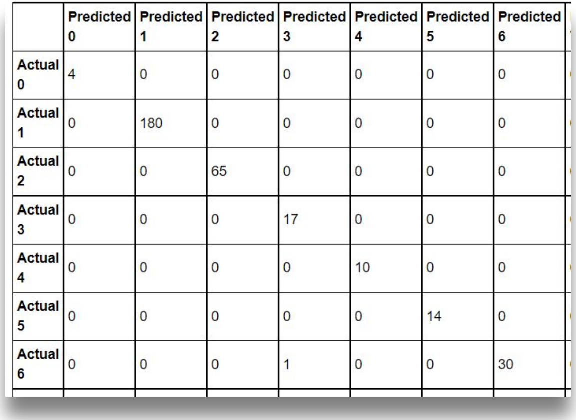
Method I:

Random Forest was the machine learning model we chose to classify known ingredients by cuisine and then predict null cuisines in the data. About 7000 recipes had a null value for cuisine, 3000 had a populated value

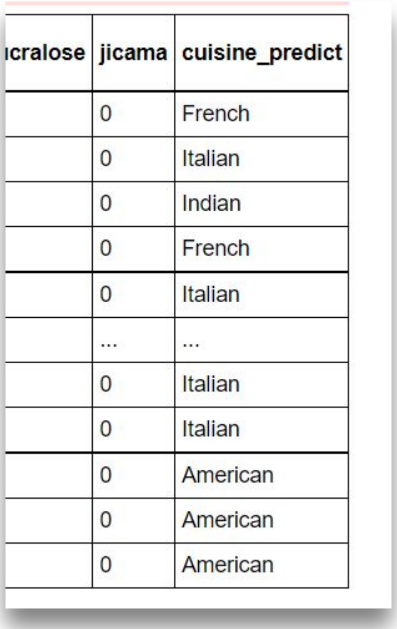
- Ingredients are given 1 or 0 if they occur in a recipe
- Cuisine names were normalized
- Label Encoder used to encode unique cuisine names into numbers
- Split the data between known and unknown cuisine type
- Created a confusion matrix
- Performed cross validation and hyper parameter tuning using randomized search CV



	cuisine_SP	onion	garlic	vanilla	lemon	bell pepper	tomato	chocolate	mushrooms	italian cheese
0	[]	0	0	0	0	0	0	0	0	0
1	[]	0	0	0	0	0	0	0	0	0
2	[]	0	1	0	0	0	0	0	0	0
3	[]	0	0	0	0	0	0	0	0	0
4	[]	1	1	0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...	...
9995	[]	0	0	0	0	0	0	0	0	0
9996	[]	0	1	0	0	0	0	0	0	1

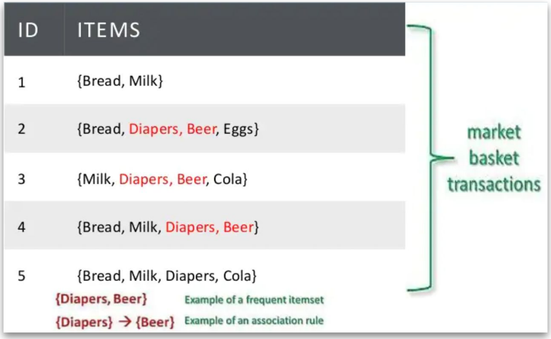


	Predicted 0	Predicted 1	Predicted 2	Predicted 3	Predicted 4	Predicted 5	Predicted 6
Actual 0	4	0	0	0	0	0	0
Actual 1	0	180	0	0	0	0	0
Actual 2	0	0	65	0	0	0	0
Actual 3	0	0	0	17	0	0	0
Actual 4	0	0	0	0	10	0	0
Actual 5	0	0	0	0	0	14	0
Actual 6	0	0	0	1	0	0	30



icralose	jicama	cuisine_predict
	0	French
	0	Italian
	0	Indian
	0	French
	0	Italian
	...	...
	0	Italian
	0	Italian
	0	American
	0	American
	0	American

Market Basket Analysis

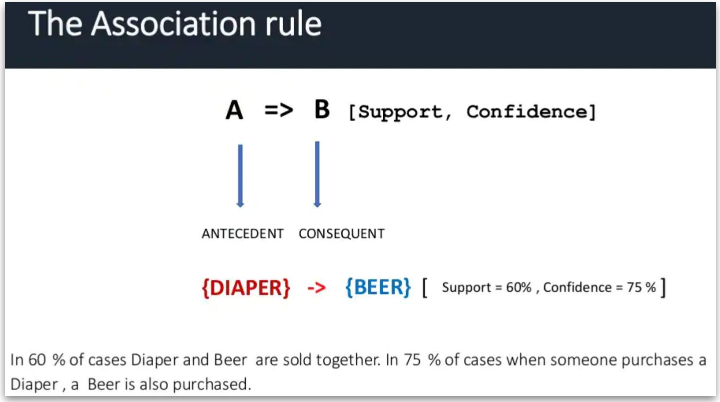


ID	ITEMS
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

market basket transactions

{Diapers, Beer} Example of a frequent itemset

{Diapers} → {Beer} Example of an association rule



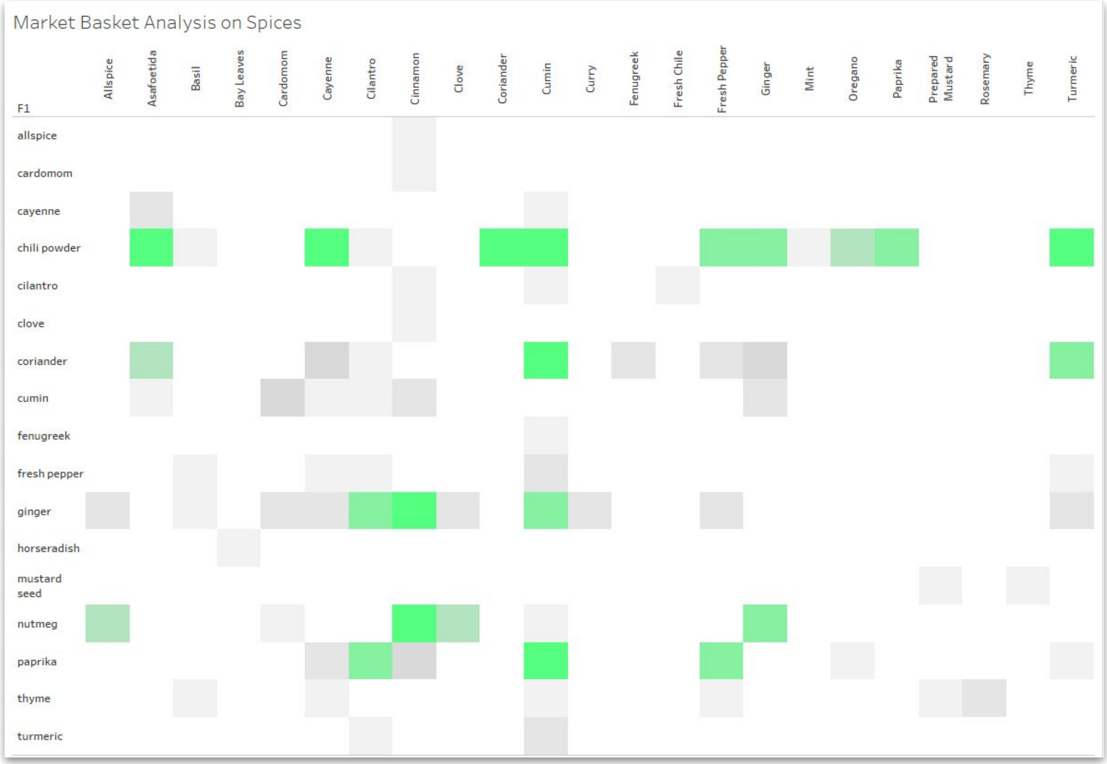
The Association rule

A => B [Support, Confidence]

ANTECEDENT CONSEQUENT

{DIAPER} -> {BEER} [Support = 60% , Confidence = 75 %]

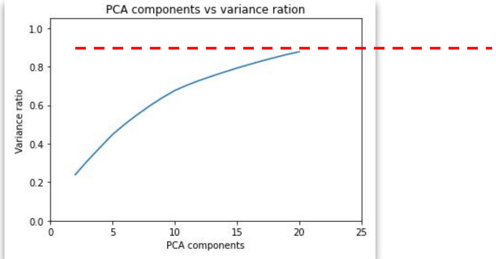
In 60 % of cases Diaper and Beer are sold together. In 75 % of cases when someone purchases a Diaper , a Beer is also purchased.



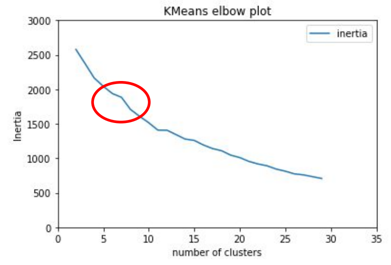
Understanding Consumer Behavior using Market Basket Analysis (Association Rule Mining). Sandeep Prasad

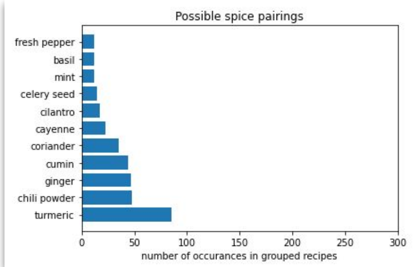
What spices occur together in recipes?

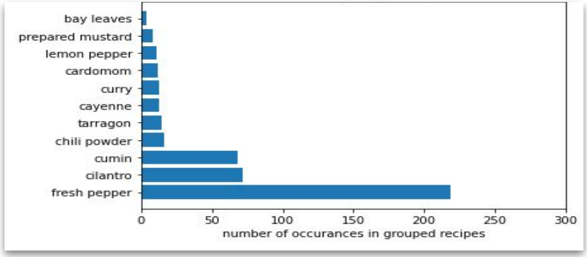
Possible Workflow:



Identify number of principal components to capture 90% of variability in dataset.

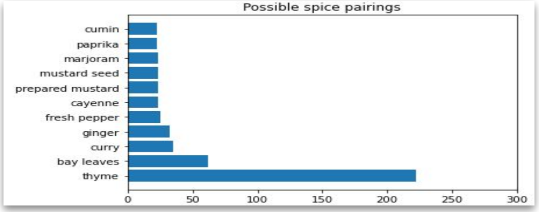


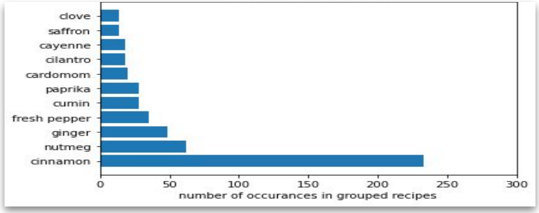


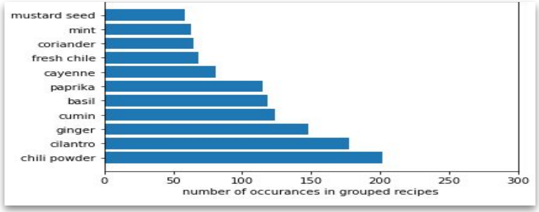


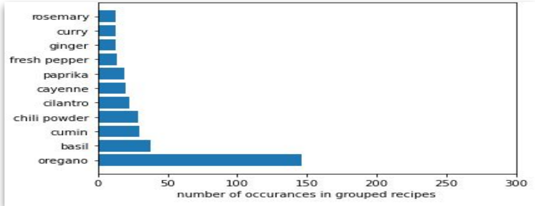
Run KMeans and plot the data.

Use elbow plot to determine characteristic number of clusters for KMeans.
What is the right number?









Dashboard Demonstration

Project Takeaways

Recommendations for Future Projects:

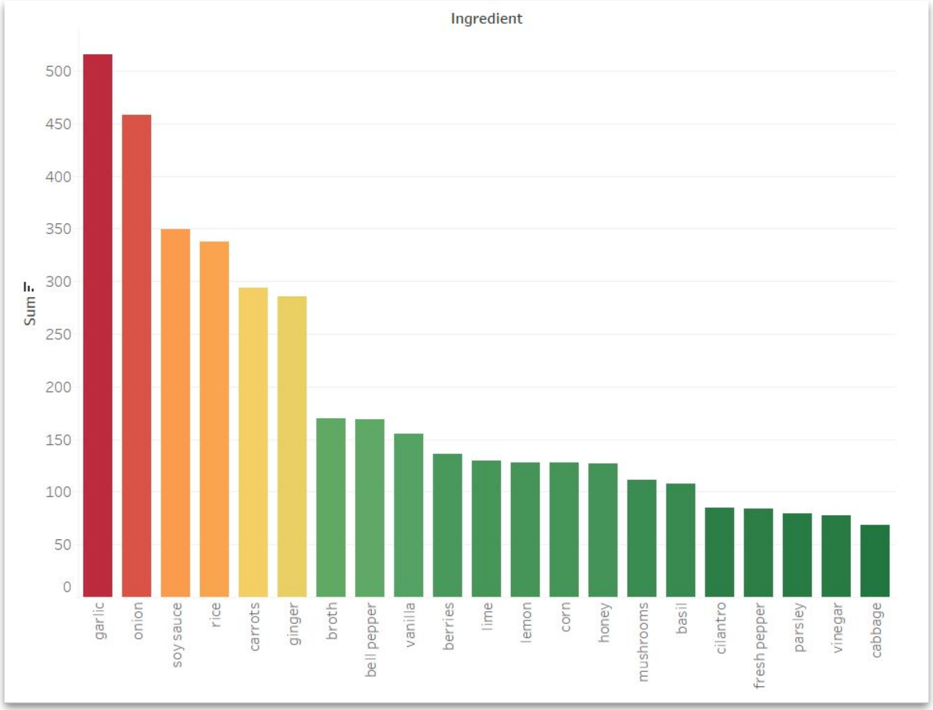
- A project like this, lends itself to Market Basket Analysis, but a deeper understanding of how to interpret the results is needed
- With more time this project would be useful to being able to select ingredients and match for recipes
- Additional data that is available pertaining to diets, nutrition, etc would be well positioned for additional analysis

Things to do differently:

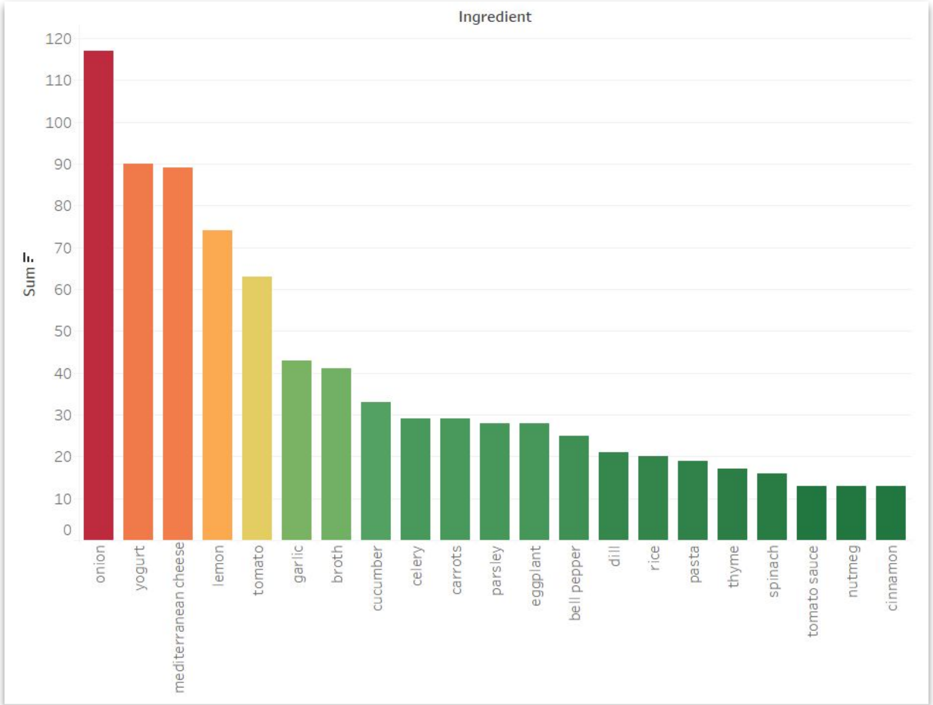
- With more time, more classification
- Decision tree models
- Explore different clean up methods that could possibly take some of the time out spent on Regex

Results

Asian

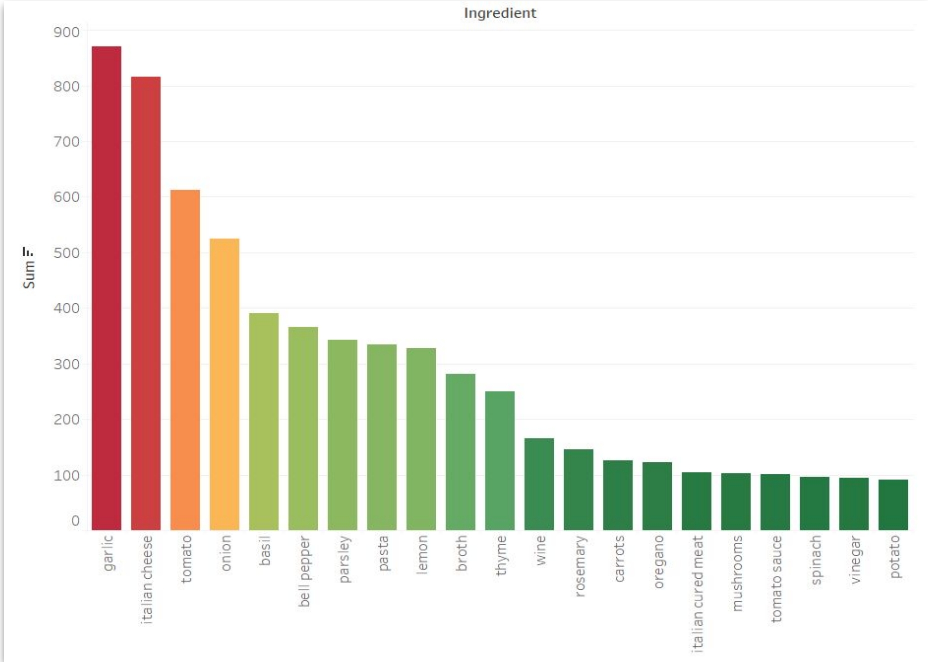


Greek

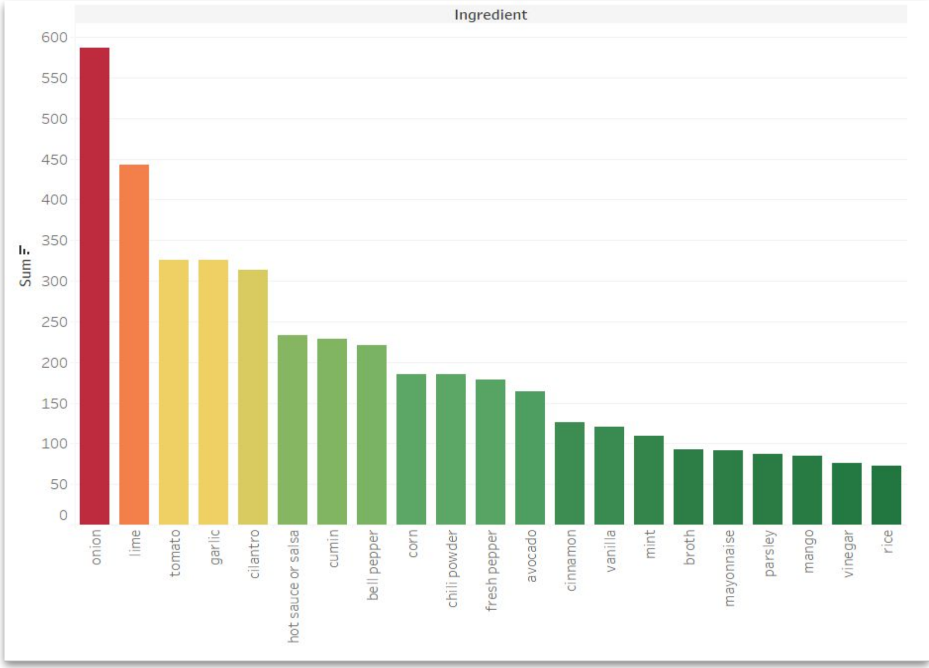


Results

Italian



Mexican



Appendix