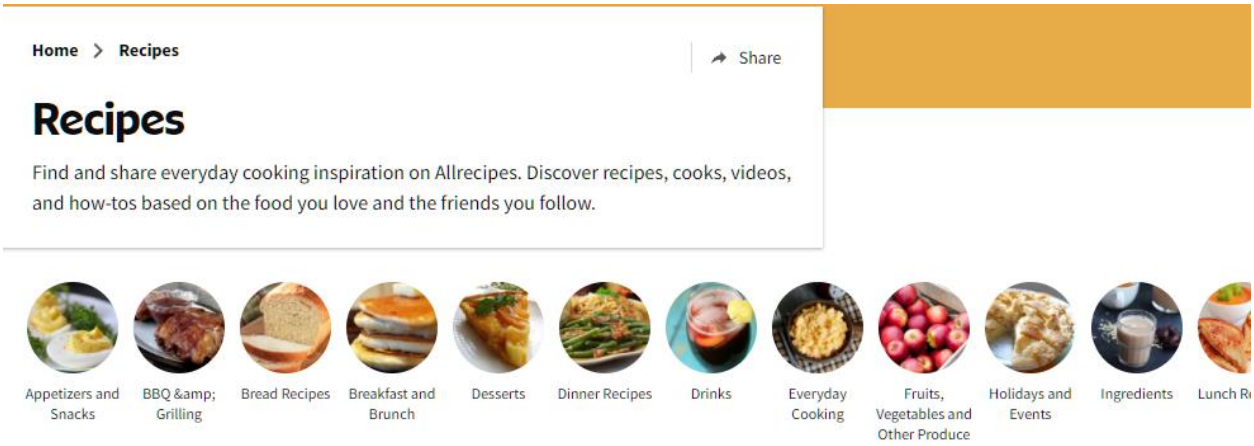
Project Documentation:

Video presentation can be found at https://uofi.box.com/s/2dplypc4hu89k6ytgg23u6ar6i0sztjr

The code was all written in a Jupyter Notebook. To run the code, the following Python modules will be required:

- BeautifulSoup: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- Selenium: https://www.selenium.dev/selenium/docs/api/py/
- Pandas: https://pandas.pydata.org/docs/index.html
- Numpy: https://numpy.org/

Most of the code for this project involves scraping the necessary data from allrecipes.com. The recipes are organized by category, so the **scrape_categories** function pulls the name of each category and the url for each category https://www.allrecipes.com/recipes/. The following image shows the recipes page and the recipe categories. Each of the category thumbnail images has and imbedded url that can be followed when clicked.

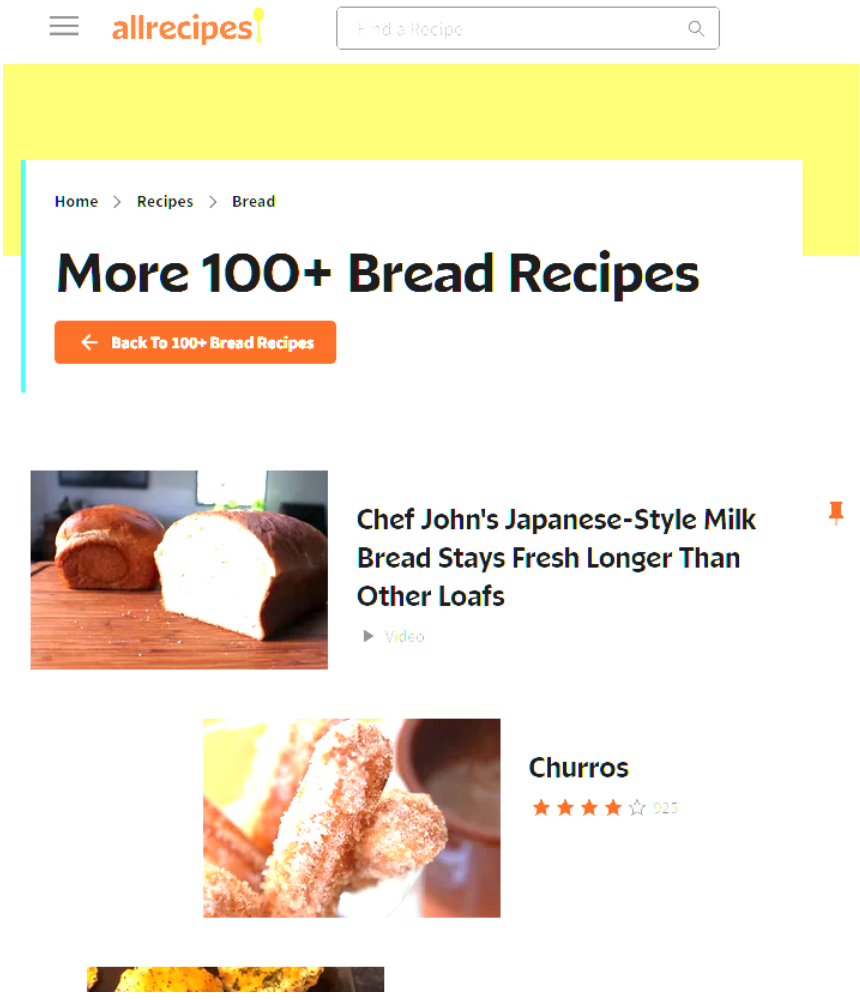


The following shows a pandas DataFrame created from running the scrape_categories function.

	category	url
0	Appetizers and Snacks	https://www.allrecipes.com/recipes/76/appetize...
1	BBQ & Grilling	https://www.allrecipes.com/recipes/88/bbq-gril...
2	Bread Recipes	https://www.allrecipes.com/recipes/156/bread/
3	Breakfast and Brunch	https://www.allrecipes.com/recipes/78/breakfas...
4	Desserts	https://www.allrecipes.com/recipes/79/desserts/
5	Dinner Recipes	https://www.allrecipes.com/recipes/17562/dinner/
6	Drinks	https://www.allrecipes.com/recipes/77/drinks/
7	Everyday Cooking	https://www.allrecipes.com/recipes/1642/everyd...
8	Fruits, Vegetables and Other Produce	https://www.allrecipes.com/recipes/1116/fruits...

The **scrape_recipes** function takes as an argument the URL to the recipes page for each category and pulls the recipe names and the URLs for those recipes. The URL for the categories leads to a webpage that uses an infinite scroll function with a button to load more recipes. I found that instead of using the infinite scroll function, you can add "?page=" plus a page number to get the list of recipes, e.g., "?page=2" for the second page of recipes and "?page=3" for the third page of recipes. In the scrape_recipes function is a loop that increments that page number and gathers all the recipes on each page. There is an argument for this function to control how many pages of recipes to scrape.

The following image shows the webpage displaying all the recipes for the "Bread Recipes" category. The scrape_recipes function will gather the names of each of these recipes and the URL for the recipe.



The table below is a pandas DataFrame showing the outcome of running the scrape_recipes function on each URL from the categories DataFrame.

	recipe_name	recipe_url
0	Herbed Pomegranate Salsa	https://www.allrecipes.com//recipe/38034/herbe...
1	Deer Jerky	https://www.allrecipes.com//recipe/46324/deer-...
2	Superb Sauteed Mushrooms	https://www.allrecipes.com//recipe/222795/supe...
3	Easy Apple Strudel	https://www.allrecipes.com//recipe/47821/easy-...
4	Seven Layer Taco Dip	https://www.allrecipes.com//recipe/19673/seven...
...	...	...
18211	Black Bean and Rice Enchiladas	https://www.allrecipes.com//recipe/222598/blac...
18213	Real Chiles Rellenos	https://www.allrecipes.com//recipe/214088/real...
18215	Slovak Stuffed Cabbage	https://www.allrecipes.com//recipe/14597/slova...
18219	Authentic Mexican Picadillo	https://www.allrecipes.com//recipe/267628/auth...
18228	Crispy Orange Beef	https://www.allrecipes.com//recipe/57966/crisp...

The **scrape_ingredients** function takes a URL for a recipe as an argument and appends to a list a dictionary containing the recipe name, the URL for the recipe, the rating of the recipe, and a list of ingredients along with each ingredient's associated quantity and units for that quantity. This function is then run on each URL from the recipes DataFrame to create a dictionary with an entry for each recipe. An example of what this looks like when this dictionary is converted to a pandas DataFrame is shown below.

	rating	ingredient	quantity	unit	recipe	url
0	4.74	sprig fresh mint	1.50	sprigs	Herbed Pomegranate Salsa	https://www.allrecipes.com//recipe/38034/herbe...
1	4.74	bunch cilantro	1.50	bunches	Herbed Pomegranate Salsa	https://www.allrecipes.com//recipe/38034/herbe...
2	4.74	Italian flat leaf parsley	1.50	bunches	Herbed Pomegranate Salsa	https://www.allrecipes.com//recipe/38034/herbe...
3	4.74	red onion	1.00	small	Herbed Pomegranate Salsa	https://www.allrecipes.com//recipe/38034/herbe...
4	4.74	large Pomegranates, raw	1.00	NaN	Herbed Pomegranate Salsa	https://www.allrecipes.com//recipe/38034/herbe...
...	...	...	...	...	...	...
17324	4.27	eggs	4.00	NaN	Almond Flour Bread	https://www.allrecipes.com//recipe/246002/almo...
17325	4.27	Almond milk	0.25	cup	Almond Flour Bread	https://www.allrecipes.com//recipe/246002/almo...
17326	4.27	(12 ounce) bottle olive oil	2.00	tablespoons	Almond Flour Bread	https://www.allrecipes.com//recipe/246002/almo...
17327	4.27	baking powder	2.00	teaspoons	Almond Flour Bread	https://www.allrecipes.com//recipe/246002/almo...
17328	4.27	pinch salt	0.50	teaspoon	Almond Flour Bread	https://www.allrecipes.com//recipe/246002/almo...

The following cleanup of the ingredient names is then performed:

- Remove added information about the ingredients that is listed in parentheses on some ingredients, for example, "gochujang (Korean hot pepper paste)" and "lemon (for zesting)" become “gochujang” and “lemon”.

- Remove the word "sprig", "pinch", "jar", "stick", "bunch", and "jigger" from the start of ingredients. For example, "pinch salt" and "sprig fresh mint" become “salt and "fresh mint".
- Remove the words “cooked” and “raw” from the end of ingredient names.

The **get_recommendation** function takes a list of ingredients from a user and a DataFrame of ingredients scraped from all recipes (as described above), and outputs a DataFrame containing the recipe name, rating, and URL for recipes that a user can make based on the ingredients they provided. The recipes in this returned DataFrame are sorted in descending order based on the recipe's rating.

For the purposes of testing, I have provided the following:

- **recipes.csv** contains recipe data scraped from each category This was created by scraping 40 pages of recipes from each recipe category, which resulted in about 18000 recipes, which included about 500 duplicate recipes. These duplicates have been removed from recipes.csv.
- **scraped_ingredients.csv** contains ingredients scraped from each recipe in recipes.csv.
- **recipe_ingredients.csv** is the same as scraped_ingredients.csv except the ingredient names have been cleaned using the cleaning steps described above.
- **pantry.csv** is a sample list of ingredients to mimic what a user may have. This was created by taking a random sample of ingredients from recipe_ingredients.csv.