Assignment 1 | Web Scraping

Lars Michael Alexander Tallqvist | 1901050

https://github.com/AlexanderTallqvist/AA-DATA-SCIENCE

Introduction

This assignment discusses the process of "Web Scraping" with the Python library Beautiful Soup. It also touches upon the presentation of the scraped data with various data analysis and manipulation tools such as Pandas and NumPy. In short, these are the tasks that were to be completed for this assignment:

1. Scrape the content of the website https://www.skinnytaste.com.
2. Filter interesting data from the scraped content.
3. Visualize the findings.
4. Further filter the data based on user selection.

Data collection

The data was scraped from the website https://www.skinnytaste.com using the Python library Beautiful Soup. The site contains various posts about food recipes and other food related topics that were to be collected and their data later analyzed. The website is built with a pagination system so that the front page contains the first nine (9) posts, the second page the following nine (9), and so on. Only food recipes were to be collected and analyzed, so posts that didn't contain a recipe were filtered out. In total, 30 pages (or about 250 recipes) were collected.

Data filtering

After the data had been collected it was time to filter out the relevant information. The HTML output from each page contained a total of 9 article tags with the class "teaser-post". I wrote a Python script that looped through these tags and collect all the relevant information. The information that was to be collected was: 1. The name of the food, 2. An image of the food, 3. The calories of the food, 4. The green point value of the food, 5. The blue point value of the food, 6. The purple point value of the food, 7. A recipe summary, 8. Various recipe keys (such as "Kid Friendly" or "Under 30 minute meal").

```
all_posts = []
source_string_base = "https://www.skinnytaste.com/page/"

for x in range(1, 30):

    source = requests.get(source_string_base + str(x)).text
    soup = BeautifulSoup(source, 'html.parser')

    for Post in soup.find_all('article', class_='teaser-post'):

        if Post.find('span', class_='icon-star'):

            entry = dict()
            entry['title'] = Post.h2.text
            entry['image'] = Post.img['src']
            entry['calories'] = Post.find('span', class_='icon-star').text

            if Post.find('span', class_='green'):
                entry['green_points'] = Post.find('span', class_='green').text
            else:
                entry['green_points'] = 0

            if Post.find('span', class_='purple'):
                entry['purple_points'] = Post.find('span', class_='purple').text
            else:
                entry['purple_points'] = 0

            if Post.find('span', class_='blue'):
                entry['blue_points'] = Post.find('span', class_='blue').text
            else:
                entry['blue_points'] = 0

            entry['summary'] = Post.find('p', class_='excerpt').text
            entry['keys'] = []

            for KeyImage in Post.find_all('img', class_='attachment-thumbnail'):
                if KeyImage['alt'] not in entry['keys']:
                    entry['keys'].append(KeyImage['alt'])

            all_posts.append(entry)


columns = ['title', 'image', 'calories', 'green_points', 'purple_points', 'blue_points', 'summary', 'keys']
dataFrame = pd.DataFrame(all_posts, columns=columns)
```

Image 1: Script for scraping data and filtering out relevant information.

	title	image	calories	green_points	purple_points	blue_points	summary	keys
0	Herb and Salt-Rubbed Dry Brine Turkey	https://www.skinnytaste.com/wp-content/uploads...	225	2	0	0	This Herb and Salt-Rubbed Dry Brined Turkey co...	[Gluten Free, Keto Recipes, Kid Friendly, Low ...
1	Turkey Pot Pie with Stuffing Crust	https://www.skinnytaste.com/wp-content/uploads...	390	8	4	4	Turkey Pot Pie with Stuffing Crust is a fun tw...	[]
2	Lightened Up Green Bean Casserole	https://www.skinnytaste.com/wp-content/uploads...	160	3	3	3	This holiday season enjoy a lighter, healthier...	[Kid Friendly, Vegetarian Meals]
3	Smashed Sweet Potatoes	https://www.skinnytaste.com/wp-content/uploads...	206	6	2	6	Smashed Sweet Potatoes, seasoned with thyme ar...	[Gluten Free, Kid Friendly, Vegetarian Meals]
4	Pumpkin Pistachio Energy Balls	https://www.skinnytaste.com/wp-content/uploads...	138	6	6	6	These no-bake Pumpkin Pistachio Energy Balls l...	[Dairy Free, Kid Friendly, Under 30 Minutes, V...
...	...	...	...	...	...	...	...	...
246	How To Make Perfect Hard Boiled Eggs	https://www.skinnytaste.com/wp-content/uploads...	77	2	0	0	Having hard boiled eggs on hand for quick brea...	[Dairy Free, Gluten Free, Keto Recipes, Kid Fr...
247	Swiss Chard Eggs Benedict	https://www.skinnytaste.com/wp-content/uploads...	244	3	3	4	I love this lighter take on Eggs Benedict made...	[Dairy Free, Gluten Free, Under 30 Minutes]
248	Shrimp, Peas and Rice	https://www.skinnytaste.com/wp-content/uploads...	346	8	3	8	This Shrimp, Peas and Rice dish is a family fa...	[Gluten Free, Kid Friendly, Under 30 Minutes]
249	Apricot-Rum Glazed Spiral Ham	https://www.skinnytaste.com/wp-content/uploads...	145	4	4	4	Apricot-Rum Glazed Spiral Ham is perfect for t...	[Dairy Free, Gluten Free, Kid Friendly]
250	Green Bean Salad	https://www.skinnytaste.com/wp-content/uploads...	176	3	3	5	One of my favorite ways to enjoy green beans– i...	[Dairy Free, Gluten Free, Keto Recipes, Low Ca...

Image 2: The collected and filtered data in a table.

Data visualization

After the data had been scraped and filtered, in was time to present it visually. This assignment had us visualize the data with the following sub-tasks:

1. Visualize the calorie distribution.
2. Visualize the point (green, blue, purple) distribution.
3. Visualize the recipe key distribution.

Calorie distribution:

The data contained 251 recipes that had calorie counts ranging from 15-587. I chose to group the calories in three (3) different groups ranging from low (0-250 calories), medium (250-500 calories) and high (500+ calories). The data was then presented in a pie diagram.

```
# Visualize calorie data
dataFrame.astype({'calories': 'float'}).describe()
```

	calories
count	251.00000
mean	234.80239
std	111.36243
min	15.00000
25%	151.50000
50%	228.00000
75%	299.50000
max	587.00000

Image 3: The data tables calorie column described.

```
# Visualize calorie data
dataFrame.astype({'calories': 'float'}).describe()
low = 0
medium = 0
high = 0

dataFrame['calories'] = dataFrame['calories'].astype(float)

for index, item in dataFrame.iterrows():
    cal = item['calories']
    if cal > 0 and cal < 250:
        low = low + 1
    if cal > 250 and cal < 500:
        medium = medium + 1
    if cal > 500:
        high = high + 1

low_label = 'Low (0-250) ' + ' (' + str(low) + ')'
medium_label = 'Medium (250-500) ' + ' (' + str(medium) + ')'
high_label = 'High (500+) ' + ' (' + str(high) + ')'


pointFrame = pd.DataFrame({'Calorie Groups': [low, medium, high]},
                          index=[low_label, medium_label, high_label])
plot = pointFrame.plot.pie(y='Calorie Groups', figsize=(15, 15), autopct='%1.1f%%')
```

Image 4: Python script for grouping calories in three (3) different groups.

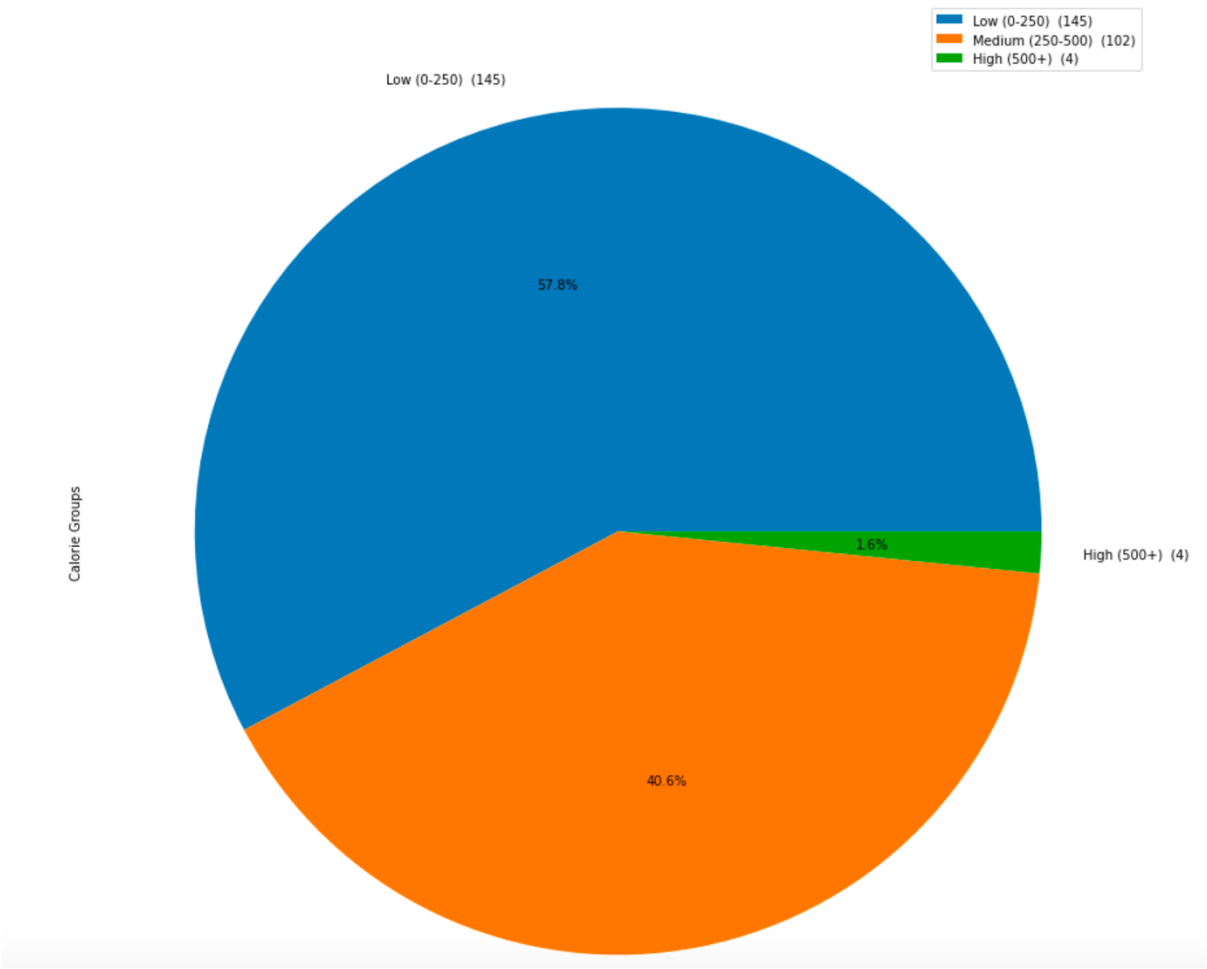

Image 5: Calorie distribution presented in a pie diagram. Low group = 145 entries and 57,8%, Medium group = 102 entries and 40,6%, High group = 4 entries and 1,6%.

Points distribution:

The visualization related to the points distribution was presented in a similar way to the calorie distribution. Out of the 251 recipes a total of 1266 blue points were counted, 1117 purple points, and 1523 green points. The data was presented in a pie diagram.

```
# Visualize points data
blue = dataFrame['blue_points'].astype(int).sum()
purple = dataFrame['purple_points'].astype(int).sum()
green= dataFrame['green_points'].astype(int).sum()

blue_label = 'Blue' + ' (' + str(blue) + ')'
purple_label = 'Purple' + ' (' + str(purple) + ')'
green_label = 'Green' + ' (' + str(green) + ')'


pointFrame = pd.DataFrame({'Points Distribution': [blue, purple, green]},
                          index=[blue_label, purple_label, green_label])
plot = pointFrame.plot.pie(y='Points Distribution', figsize=(15, 15), autopct='%1.1f%%')
```

Image 6: Python script for presenting point data in a pie diagram.

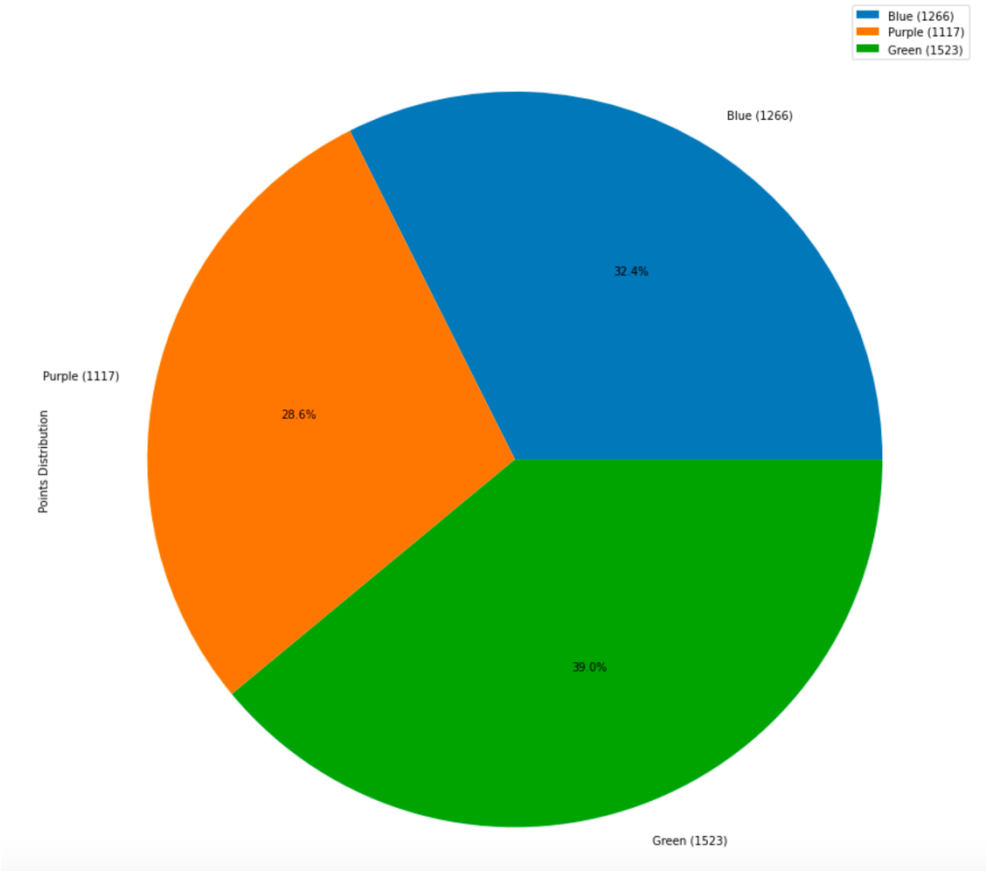

Image 7: Food point distribution presented in a pie diagram. Blue points = 1266 and 32,4%, Purple points = 1117 and 28,6%, Green points = 1523 and 39%.

Recipe key distribution:

The recipes could contain a "key" which described a certain property of the recipe (e.g. "Vegetarian Meal" or "Gluten free". These keys were collected, and the distribution of the keys was presented in a staple diagram. The y-axis contains the name of the key, while the x-axis represents the number of times a key occurred in a recipe. The most popular key turned out to be "Gluten free", and the most seldomly used key was "Slow Cooker Recipes".

```
# Visualize keys data
keyFrame = pd.DataFrame(all_posts, columns=['keys'])
key_occurance = Counter()

for keys in keyFrame['keys']:
    key_occurance += Counter(keys)

keys = key_occurance.keys()
values = key_occurance.values()

plt.figure(figsize=(15,10))
y_pos = np.arange(len(keys))
plt.barh(y_pos, values, align='center')
plt.yticks(y_pos, keys)
plt.xlabel('Occurrence')
plt.title('Meal Key Distibution')
plt.show()
```

Image 8: Python script for visualizing key distribution in a staple diagram.

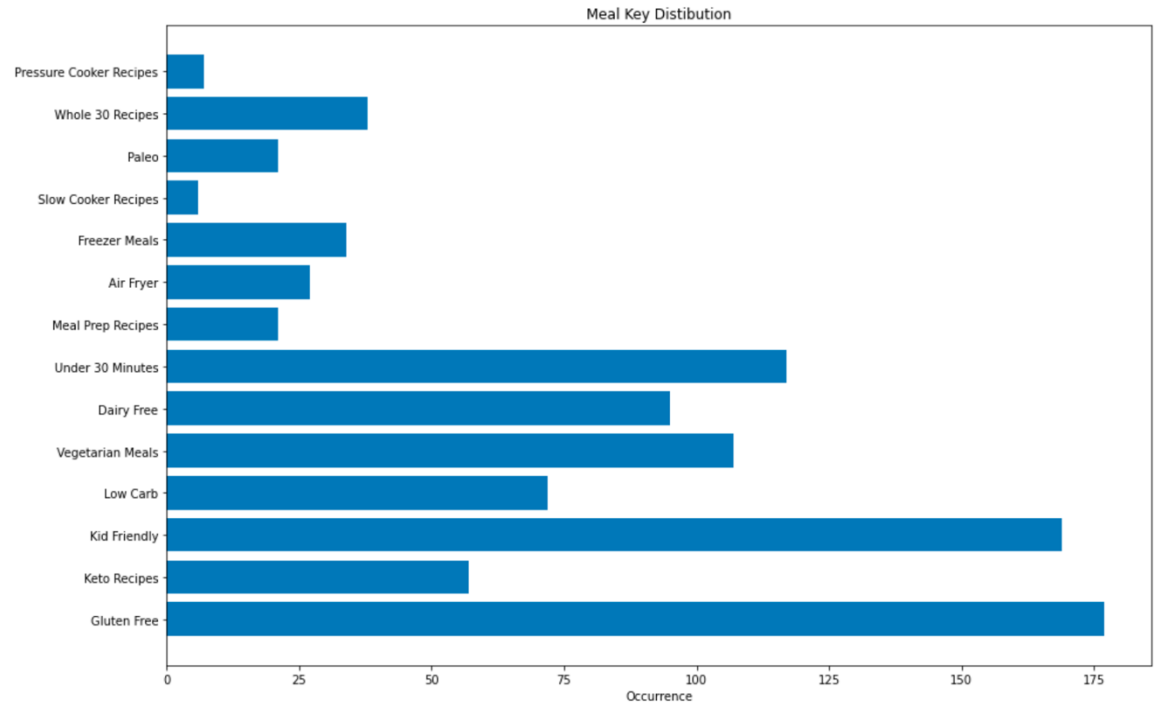

Image 9: Staple diagram of the key distribution.

User interactions with the data

The final task had us implement user interaction to the program we had created. The user had to be able to filter the data in the following ways:

1. Filter the data based on a calorie range.
2. Filter the data based on blue, green and purple point values.

After the filtering, a view containing five (5) results with at least the title, image and the description of the recipe were to be presented for the user, ordered based on calories. The program that I ended up creating asks the user for a MIN and MAX value for calories, blue points, green points and purple points.

Image 10: Filtered results showing the recipe title, description, image, calories, and point values.

```
# Show results based on user input
accepted_results = []
max_calories = int(input('Max Calories '))
min_calories = int(input('Min Calories '))
max_blue = int(input('Max Blue Points '))
min_blue = int(input('Min Blue Points '))
max_green = int(input('Max Green Points '))
min_green = int(input('Min Green Points '))
max_purple = int(input('Max Purple Points '))
min_purple = int(input('Min Purple Points '))

dataFrame['calories'] = dataFrame['calories'].astype(float)
sortedFrame = dataFrame.sort_values(by=['calories'], ascending=True)

for index, item in sortedFrame.iterrows():

    calories = int(item['calories']);
    bp = int(item['blue_points']);
    gp = int(item['green_points'])
    pp = int(item['purple_points'])

    if calories <= max_calories and calories >= min_calories:
        if bp <= max_blue and bp >= min_blue:
            if gp <= max_green and gp >= min_green:
                if pp <= max_purple and pp >= min_purple:
                    accepted_results.append(item)
                    if (len(accepted_results) >= 5):
                        break
print("")
print("RESULTS")
print("")
for item in accepted_results:
    print('Title: ' + item['title'])
    print('Summary: ' + item['summary'])
    display(Image(url=item['image']))
    print('Calories: ' + str(item['calories']))
    print('Blue points: ' + str(item['blue_points']))
    print('Purple points: ' + str(item['purple_points']))
    print('Green points: ' + str(item['green_points']))
    print("")
```

Image 11: Python script that enables users to filter the data based on calories and points.

Conclusion

Web scraping can be a powerful tool when used together with a data analysis and manipulation tools such as Pandas and NumPy. It can be hard to see all the different data visualization possibilities when simply browsing through a site such as skinnytaste.com. I can definitely say that I've learned a lot, and having the knowledge that I've acquired through this assignment has made me look at various data related websites in a different way, and I now see the opportunities that web scraping offers.