Mini Project 1 – What should I cook tonight?

Tanjim Pranto
tanjim.pranto@abo.fi

1 Introduction

1.1 Problem Statement

In the present food and nutrition environment, access to a large body of recipes is highly essential for various purposes: from diet planning to machine learning food recommendation models. However, gathering such recipe data from online sources is a cumbersome process and requires huge amounts of time and manual effort. This project targets the automation of recipe data scraped from the "Skinnytaste" website, which holds a top position in popular sources that provide healthy recipes. The focus would be on scraping data about names, images, ingredients, instructions, and nutrition details for all recipes and storing it in a proper format for further analysis and re-usability.

1.2 Objectives

The major objectives of this project include the following:

- Automate Data Extraction: Construct a web scraper that will automatically retrieve data in detail about recipes from multiple pages of the website.
- Handle Variations in Data: Provide mechanisms to handle inconsistencies and variations in the HTML structure of the website while extracting only valid information.
- Ensure Data Integrity: Skip incomplete or problematic entries in order to maintain high data quality.
- Data Storage and Accessibility: The gathered data should be stored in tabular format, hence easier to access in further processing or analysis.

1.3 Motivation

There are two motivations behind this project. First, time and effort can be saved since the gathering of data is automated. For this reason, one can create a large-scale dataset that is impractical or too time-consuming to compile manually. The information collected can be used as a foundation to build other applications in the realms of dietary meal planning, recommendation systems for recipes, or anything related to nutritional analysis. With automation, we seek to present high-quality recipe data suitable for use in these various applications.

2 Data Collection

2.1 Methodology

The web scraping was carried out using the Selenium-based Python environment. The scraper was designed to navigate through a website, identify recipes on every page, and then extract relevant information from individual recipe pages. Selenium can effectively mimic user interactions, like dynamic content, which is the very reason it's necessary to scrape most modern web pages since most of them depend on JavaScript for rendering.

2.2 Step-by-Step Process:

Initialization: The very first thing the scraper does is instantiate a Selenium WebDriver by creating some options that will prevent it from being detected as a bot. It does this by running the browser in headless mode, using random delays between each action, or modifying browser fingerprints.

Handling Pagination: Scraper then iterates over the first 50 pages of the "Skinnytaste" website; each of them contains links to multiple recipes. The scraper, for each page, scrapes the links to all

the recipes listed and proceeds to visit each one individually.

Extracting a Recipe Page: On visiting a recipe page, the scraper attempts to extract the following:

- Recipe Title: Name of the recipe.
- Image URL: The URL of the main image representing the recipe.
- Ingredients: A list of ingredients needed to prepare the recipe.
- Instructions: Step-by-step directions for creating the dish.
- Nutritional Information: How many calories and, or other nutrition facts.
- Recipe key: Dietary or meal categories this recipe falls under; such as vegan, gluten-free, etc.
- Cooking Time: Total time taken to cook.
- Course: What type of course it is?
- Summary: Details summary about the recipe.
- WW Points/Personal Points: Rating given to the recipes.
- Page Categories/Tags: Tags given to the recipes.

Error Handling and Skipping: If the image URL/recipe key is not present or the important recipe page structure is broken, then the scraper will skip that page. This assures dataset integrity and quality.

Data Storage: The crawled data will be stored in comma-separated values format that can easily be imported into data analysis libraries such as Pandas. This CSV would include columns for all the extracted information regarding the title of the recipe, the link to the recipe, image URL, ingredients, instructions, and nutritional information.

2.3 Data Storage for Easy Access

The crawled information will be stored in structured CSV format. This is a good choice of format as it carries the following advantages:

- Compatibility: CSV works with most data processing and analysis tools, such as Excel, Pandas, and SQL databases.
- Readability: Due to the tabular nature of CSV, it becomes easy to visually inspect and verify if this is small enough.
- Scalability: CSV can easily handle a lot of records without noticeable performance degradation; hence, it is suitable for datasets that amount to thousands of records.

3 Data Scraping Challenges

Inconsistent HTML Structure:

One of the most challenging things in this project was the fact that the structure of the HTML of the different pages or meal plan pages was inconsistent. For certain recipes, some information fields like an image or nutritional data might not exist. Because of such inconsistency, flexible scraping logic with fallback mechanisms against possible missing data became required.

Solution:

- We utilized different CSS selectors for each piece of information with the assumption of possible differences.
- Added try-except blocks to skip recipes in cases where either their structure was different than what was assumed or important elements were missing.

Anti-Bot:

The most prevalent anti-bot measures on websites to avoid scraping by automation include things like CAPTCHAs, rate limits, and suspicious activity based on browsing pattern detection.

Solution:

- Random delays were added between actions such that every action seems more like human browsing.
- Rotated user-agent strings to simulate a variety of different browsers.
- Handled session cookies and reCAPTCHA challenges if there were any.

Large-Scale Data:

Handling large amounts of data over a number of pages requires efficient handling and storing of the data. It was necessary to ensure that no data collected was duplicated and that only complete and relevant records were stored.

Solution

- A set was used to track visited URLs to avoid duplication.
- Skipped recipes that didn't have complete data or had repeated titles and links.
- Store the data in .csv structured format, where each column is different data for the recipe.

4 Visualization of Data Analysis

Several visualizations had been created to better understand what was collected. The following charts were deemed necessary:

Calories Distribution: Histogram or box plot to show the distribution of calories across various recipe types. This helps in understanding the caloric range and identifying outliers. The histogram appears to have a bimodal distribution, with two distinct peaks around 200 and 400 calories. A high frequency of items in the 200 and 400 calorie range indicates that many recipes in the dataset have calories around this value. There is a noticeable gap between 300 to 350 calories, indicating fewer recipes that fall in that calorie range. This gap is interesting because it breaks up what might otherwise be a continuous increase. The bimodal peaks suggest that recipes tend to be either low-calorie or moderately high-calorie, which might reflect common dietary preferences. To understand why these peaks exist, it could be useful to examine the types of recipes that fall under each peak (e.g., are the 200-calorie recipes mainly salads or breakfast options, while the 400-calorie recipes are main courses.). This irregularity indicates diverse calorie distributions across the recipes, which could be influenced by different types of dietary categorizations or cooking styles represented in the data. The gap in frequency of around 300 calories could be a target for recipe development. There may be an opportunity to create recipes that fill this middle ground—ideal for users seeking a balance between light and heavy meals. It would be insightful to correlate these calorie counts with the recipe keys from the previous analysis (e.g., gluten-free, vegan, keto). This could help determine if certain dietary tags are associated with higher or lower calorie ranges. For instance, it would be interesting to see if gluten-free recipes tend to fall within the lower-calorie peak or whether certain dietary restrictions tend to have higher-calorie options.

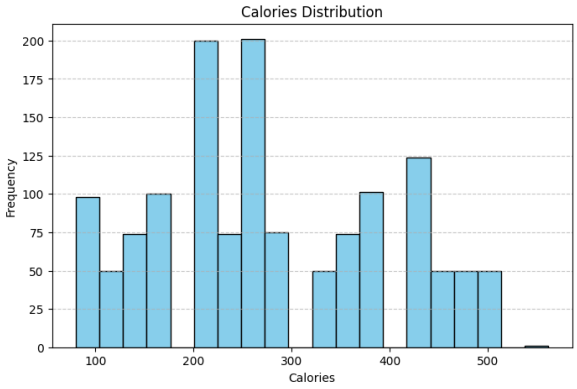

Figure 1: Calories Distribution Across Various Recipe Types

Recipe Key Distribution: A bar chart showing the frequency of various recipe keys like "vegan", "gluten-free" etc. The above histogram helps in understanding the diversity of recipes concerning their categorization. "GF" (Gluten-Free) has the highest frequency, with a count of approximately 1200. This indicates a high prevalence of gluten-free recipes, suggesting a focus on gluten intolerance or the popularity of gluten-free diets. "DF" (Dairy-Free) also appears prominently, with just under 900 occurrences. This category is another significant focus, possibly catering to those with lactose intolerance or dietary preferences avoiding dairy. The broad distribution across different keys indicates a diverse range of dietary accommodations. From gluten-free and dairy-free to high protein and keto-friendly, there seems to be a well-rounded selection of recipes aimed at various dietary needs. However, the dominance of gluten-free and dairy-free categories could imply a strong demand for these types of recipes, potentially reflecting popular dietary trends. The presence of vegan, low-carb, and whole-food recipes alongside keto and high-protein recipes suggests an approach to catering to both plant-based and meat-based diets, covering a wide spectrum of dietary habits. Considering the prominence of air fryer (AF) and keto-friendly (KF) recipes, adding more

recipes that combine these categories could be beneficial, such as "Keto-Friendly Air Fryer" recipes, as there appears to be strong interest in both functional kitchen tools and specific diet plans. The relatively lower frequency of vegan recipes suggests a gap that could be filled, especially if dietary shifts continue to lean toward more plant-based eating. Expanding this collection could attract users seeking more vegan options.

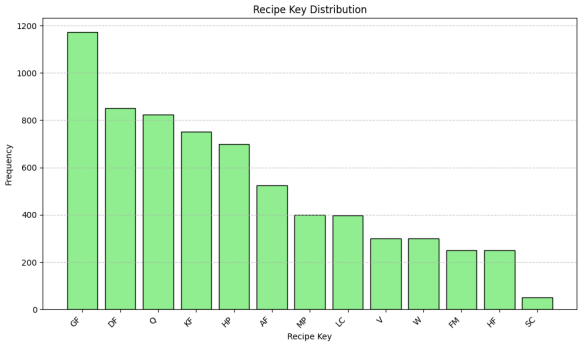

Figure 2: Recipe Key Distribution Across Various Recipe Types

Personal Points Distribution: A distribution of Weight Watchers points per recipe. This can give a feel for which different recipes might be more or less suitable depending on one's diet goals. The histogram shows a dominant peak at around 6 points, with a frequency of over 400. This suggests that a large number of recipes are assigned 6 personal points, which could indicate a typical or balanced level of calories and nutrients that many recipes fall into. Specifically, the 0, 3, and 4-point ranges have considerable frequency, suggesting that a reasonable number of recipes are quite diet-friendly, possibly suitable for those looking to minimize their personal points. The distribution tapers off significantly beyond 6 points, with relatively few recipes exceeding 8 personal points. The highest point values (around 12 points) have very few recipes, which indicates that there are not many high-point recipes in this dataset, making it easier for those watching their point intake to avoid highly scoring items. There are noticeable groupings at 0-4 and a significant spike at 6 points. These clusters indicate different types of recipes—those likely categorized as lower in calories (such as salads or vegetable-heavy dishes) and others that might be more substantial but still not extreme in terms of their point value.

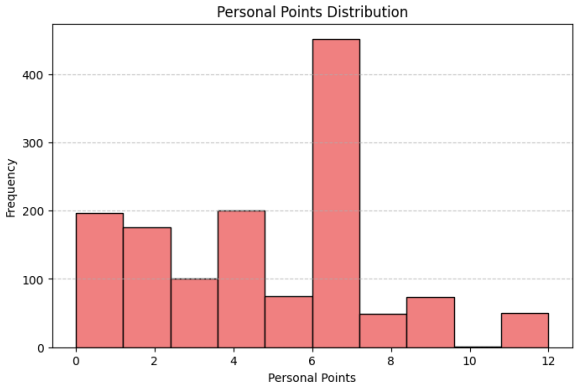

Figure 3: Personal Points Distribution Across Various Recipe Types

4.1 Observations

Calorie Content Variability: Calories across the recipes were indeed normally distributed with very wide dispersion, suggesting that the website has great variation in the types of recipes. Some were very low in calories, meant for persons on very restrictive diets, while others were really indulgent.

Category Overlap: Most recipes overlap into several categories, which serves to illustrate the way recipes can flexibly serve different diets.

Missing Information: Some recipes/pages did not have complete nutritional information and images, which could limit usability for certain applications, such as diet planning or visual recognition models.

5 Conclusion

5.1 Bottlenecks and How They Were Overcome

HTML Structure Variability: One big bottleneck was the inconsistency in the structure of HTML. These were overcome by the introduction of fallback mechanisms and the usage of multiple CSS selectors for data extraction.

Anti-bot detection: It allowed the implementation to scrape without getting blocked by applying a pattern of browsing more human-like, including error handling. It was helpful in this respect because of the randomized delays, rotation of user agents, and handling of CAPTCHAs.

Data quality: Skipped those recipes where information was incomplete or missing to ensure that data is of good quality. In such a process, the integrity of the dataset would have been maintained,

but this also resulted in a considerable reduction in the number of recipes collected.

5.2 Future Directions

Expand the dataset: Perform the scraper on other websites for recipes in order to build more comprehensive datasets.

Improve the data quality: We can perform OCR on text extraction from images or use NLP models to infer a lot of information that is missing.

Exploit the Dataset by Applications: The dataset shall be used for applications such as recommendation systems, dietary analysis tools, and machine learning models for culinary purposes.