UNIVERSITÀ DEGLI STUDI DI
MILANO-BICOCCA

Master degree in Data Science

Data Management project

Authors:
Andrea Cardinali, Adonis Kingsley Granita, Matteo Simeoni

2023/2024

Contents

1 Abstract

The Italian cuisine is one of the most, if not the most, loved type of cooking in the whole world. This project aims to collect many Italian recipes, together with a list of available products in a grocery store. The two data are then combined together in such a way that each recipe has a list of the necessary ingredients with their unitary cost, portions and an estimate of the calories.

2 Introduction

This project aims to create a collection of Italian cooking recipes that estimates how much each recipe costs to make. The idea behind this was that students living on a budget or people that love doing budgeting may find this helpful for different reasons. Additional data like the portions and an estimate of the calories of each recipe were included to have more useful information and choose the right recipe each time. This could suits athletes that need to be careful how much calories they can take. Students may find this particularly helpful as they are, more often than not, busy studying or going to lectures and do not have much time to spend on cooking. Having a list of easy and fairly cheap dishes to make is really a life saviour for them. For example, students living on their own that have only an hour break for lunch can easily search for a recipe that is ready in a few minutes but able to fullfil the daily calories intake.

The recipes are obtained by scraping the GialloZafferano website [3] while the list of products available to buy is from the Carrefour website [1]. After scraping both websites, the data was cleaned of any errors and the quantities needed were transformed (with some approximations) to have the same unit of measurement. The result is then saved as JSON file and stored in a MongoDB database, a NoSQL database structure which can be queried to extract different information about the recipes, such as the cost. In this case a NoSQL database like MongoDB was preferable to use instead of a simple MySQL database since the structure of the data, which has a variable number of ingredients, was more naturally suited to the flexible schema design of a NoSQL database.

3 Data Acquisition

The set of recipes was obtained by using dynamic web scraping on GialloZafferano [3]. It was necessary to use the Selenium library to dynamically scrape the website as the recipes are divided into different pages. The program was able to access one recipe at the time from the list in each page, save its name, the required ingredients and the necessary quantity. It was then able to go back to the list to access the next recipe. The process went on until it reached the last recipe on the last page. The entire program was able to run and same most of the pages but it failed to save more or less ten recipes, perhaps due to unknown characters in the name of the recipes. However, the resulting data consists of 4772 rows and 3 columns, as shown in Figure 1 below, which can be considered quite enough.

	titolo	categoria	ingredienti_dict
0	Budino al gianduia con salsa all'arancia	DOLCI	{'Panna fresca liquida': '150 g', 'Amido di ma...
1	Sfogliatine di pere con Nutella®	DOLCI	{'Nutella®': '75 g', 'Pere Williams': '1', 'Pa...
2	Monodose natalizia	DOLCI	{'Burro a temperatura ambiente': '100 g', 'Zuc...
3	Polpettine mostruose	SECONDI PIATTI	{'Ricotta di vaccina': '175 gr', 'Parmigiano R...
4	Friselle con cotechino e lenticchie	ANTIPASTI	{'Friselle da 30 grammi ciascuna': '4', 'Cotec...
...	...	...	...
4767	Capesante su crema di carciofi ed erbe aromatiche	ANTIPASTI	{'Capesante': '4', 'Cuori di carciofo (surgela...
4768	Pavesini mummia	DOLCI	{'Pavesini classici (3 pacchetti)': '33', 'Pav...
4769	Girelle alla marmellata	DOLCI	{'Uova (circa 3 medie)': '150 g', 'Zucchero': ...
4770	Cake pops fantasmini	DOLCI	{'Farina 00': '200 g', 'Latte condensato': '23...
4771	Casarecce di legumi con gamberi, spinaci e limone	PRIMI PIATTI	{'Casarecce di ceci': '250 g', 'Gamberi argent...

Figure 1: Dataset of recipes

The third column was then transformed into a dictionary that takes each ingredient as a key and the necessary quantity as its value. This was done mainly for two reasons: the first because it was easier to associate the ingredients to the products from the supermarket, and secondly to improve interpretability when looking at the raw data. So, for example, instead of having 'capesante 4', it would be "capesante": "4".

After a first scraping process, the data acquired was not enough so additional data was scraped to save, for each recipe, the number of portions, the necessary time to cook and an estimation of the calories. In Figure 2 there is a view of the new additional data.

The dataset of the recipes was then enriched with data of basic ingredients obtained from carrefour [1]. This time it was used a static web scraping to get the information needed. Static web scraping was more than enough because the carrefour web page lists all the products (and the information needed) in the same, long page. In this case the program was able to get the data by scraping the same

	Recipe Name	Portions	Cooking Time (min)	Calories (Kcal)
0	Insalata di riso	2	35	660.0
1	Chili con carne	2	135	562.0
2	Gateau di patate	2	100	317.0
3	Paella de marisco	3	95	617.0
4	Piadina romagnola fatta in casa	2	34	488.0
...	...	...	...	...
500	Hosomaki vegetariano	4	50	274.0
501	Cornbread burger con porcino	3	100	322.0
502	Maki di kiwi e salmone	3	42	306.0
503	Demi baguette al salmone	2	35	728.0
504	Gateau di patate con zucca e taleggio	3	110	NaN

Figure 2: Dataset of recipes

one webpage. Initially the program scrolled to the bottom of the page and waited a few seconds to let it load properly. Then, for each category of food (i.e. 'Pasta', 'Carne', 'Verdura', etc.) it captured the name of the product, the price for Kg (or L) and the belonging category.

	product_name	brand	price	source_file
0	Carrefour Classic Burger di Prosciutto Cotto 2...	Carrefour	15,93	GASTRONOMIA
1	Carrefour Veg Medaglioni Bulgur, broccoli e po...	Carrefour Veg	14,39	GASTRONOMIA
2	Carrefour Veg Medaglioni Melanzane, farro e mi...	Carrefour Veg	14,39	GASTRONOMIA
3	Carrefour Extra Cannelloni Ricotta e Spinaci 3...	Carrefour	9,97	GASTRONOMIA
4	Carrefour Sensation Vegetal Cous Cous Aromatic...	Carrefour	15,95	GASTRONOMIA
...	...	...	...	...
7718	Gamberetti Boreali in salamoia	Polar Seafood	47,92	PESCE
7719	Bubble Tea Black Forest - Gusto ribes nero 450 ml	Flavour Drink	10,00	PESCE
7720	Bubble Tea Tropical Jungle - Gusto ibisco 450 ml	Flavour Drink	10,00	PESCE
7721	Bubble Tea Wild Straberry - Gusto fragola 450 ml	Flavour Drink	10,00	PESCE
7722	Bubble Tea California Peach - Gusto pesca 450 ml	Flavour Drink	10,00	PESCE

Figure 3: Dataset of ingredients

In Figure 3 there are some rows of the ingredient database. The total number of products obtained by web scraping is 7723. Such a "big" dataset ensures that the majority of recipes has most of the ingredients available to buy at Carrefour.

Out of all the recipes gathered, it was chosen to select only the recipes belonging to the category of "Piatti unici". It was best to do this because the following phases showed better results when using a smaller dataset, rather than initial one.

4 Data Cleaning

Both datasets needed to be cleaned after the scraping to remove any errors and wrongly formatted data. The first thing to fix in the recipes dataset was the apostrophe in the word "Olio extravergine d'oliva", which was the only one that caused errors in the names of the recipes, and transformed the entire word into "Olio extravergine di oliva". There was also a problem with the encoding of the symbol $\frac{1}{2}$ for some ingredients. This was solved by replacing $\frac{1}{2}$ with 0.5.
Other transformations needed were the units of measurement and quantities such as "cucchiaio", "cucchiaino", "foglia", "q.b." and so on. First of all the quantities are transformed to take all the same unit (gr) so that it'll be easier to compute the price. The equivalence is not exactly correct but in this case the difference would be negligible.
For not precise quantities such as "cucchiaio", a reference table online [2] was used while for quantities such as "q.b.", "foglia" and so on approximate values were assigned, for example "q.b." was associated with "3 g". To perform this step it was used a function that checked if the quantity was in the correct format, and if they were not it extracted the unique quantities. In Figure 4 there are the quantities that are not "correct".

A conversion table for these values was then created with the correct values, while words such as "abbattuto", "secco", "Modena" and so on are removed.

The ingredients were initially saved as a string for each row and they needed to be transformed into a dictionary so that each ingredient is associated with a quantity, to have a result like "Panna fresca liquida":"150 g". To do this process it was necessary to first divide the string into two pieces: the name of the ingredient and the quantity with the unit of measurement in case. Then the two pieces were finally converted in a dictionary with the name as the key and the quantity as the value.

The cleaning process for the products of the supermarket was performed merely for matching them with the ingredients, rather than to simply make the data clearer. First of all, many products had the brand in the field of the name (like "Kikkoman Salsa di Soia" where Kikkoman is the brand of that soy sauce), so it had to be removed. After removing the name of the brand, any eventual information between parentheses was removed, every word was then transformed in lowercase and prepositions and words such as "Filiera" were temporary removed. With the help of ChatGPT it was easy to find such words that bring no relevant information and may have caused problems in the matching phase.

```
['(abbattuto',
 '(secco',
 '0.5"',
 '1"',
 '100',
 '2"',
 '4"',
 '5"',
 'Modena',
 'abbattuto',
 'bacca',
 'bicchiere',
 'ciuffo',
 'costa',
 'coste',
 'cucchiai',
 'cucchiai"',
 'cucchiaini',
 'cucchiaino',
 'cucchiaino"',
 'cucchiaio',
 'fette',
 'fette"',
 'filetti"',
 'filetto"',
...
 'rametto',
 'salmone',
 'secco',
 'spicchi',
 'spicchio']
```

Figure 4: Quantities

Finally, the last cleaning process was also applied to the ingredients before starting to match them with the products, so to increase the probability to find a good match.

5 Data Exploration

To asses the quality of the data it was used the ProfileReport function built in Python, which was able to perform some basic exploratory data analysis. The function was first applied to the dataset of the recipes and the output showed that out of 238 recipes, there were no missing values nor redundant recipes. The function also showed the most frequent words founded in the recipes name and ingredients as in Figure 5 and Figure 6.

Figure 5 shows that, apart from "di", "con" and "e", the majority of the

Figure 5: Frequent recipes

recipes are about cooking rice, vegetables, piadine, burgers, tuna, sandwiches, ham, potatoes and so on.

Figure 6: Most frequent ingredients

Figure 6 shows the most frequent ingredients are: olives, olive oil, salt, pepper, tomatoes, rice, flour, etc.

The data of the calories and portions has 505 rows, but there are some missing data regarding the calories as not all recipes have it on the website. Regarding the number of portions and cooking time, there are no missing value and most recipes are meant for two portions. The recipes take on average an hour to cook but a quarter of them is ready in half an hour, perfect time for people in a rush.

The function was then applied to the dataset of the products and the result,

as expected, was that there are no missing data and no duplicates, like the dataset of the recipes. Figure 7 shows the most frequent brand of the products such as "Carrefour", "Lavazza", "Bonduelle", "Mulino", "Barilla" and so on.

Figure 7: Most frequent brands

It's also interesting to look at how many products were available for each category. Figure 8 shows that sweets is the category with the highest number of products available, followed by seasoning products and the by cheese.

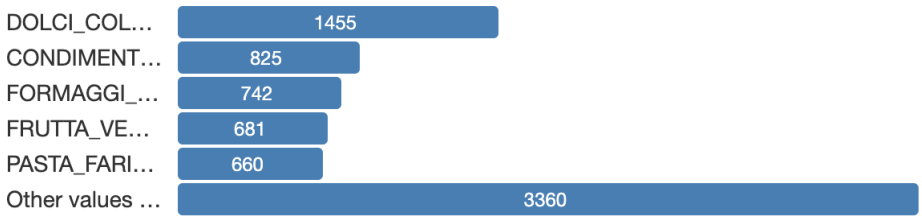

Figure 8: Categories

6 Data Integration and Enrichment

6.1 Data Integration

In this management project, data integration has been performed to ensure that information from various sources was easily accessible. The integration process involved several phases. First, a matching of the ingredients of the recipes and the supermarket products was performed using the cosine similarity index.

Next, the maximum and minimum prices for each ingredient available at the supermarket were extracted. By determining a price range instead of using the average price, a more detailed understanding of recipe costs was provided in Figure 9. This approach enabled users to see how much they would spend if they opted for the most economical choices versus purchasing higher-priced, potentially higher-quality branded ingredients.

	ingrediente	price_max	price_min
0	"Acciughe sottolio	49.83	49.08
1	"Acciughe sottolio filetti	47.60	38.40
2	"Burrata (4 da 125 g luna)	20.72	13.52
3	"Cosciotto dagnello	3.99	3.99
4	"Filetto di salmone da 200 g luno	74.50	17.45
...	...	...	...
858	macinato grossolanamente)	NaN	NaN
859	o di pesce volante)	69.75	6.64
860	pulito e rifilato	19.90	19.90
861	pulito e rifilato)	19.90	19.90
862	sgocciolato	NaN	NaN

863 rows × 3 columns

Figure 9: ingredients and their prices range

Finally, a new JSON file was created containing the ingredients, the price of each ingredient, and the total price per recipe. This file served as a source of information, making it easier to access and use the data in MongoDB.

6.2 Data Enrichment

In order to have a better understanding of the cost of each portion of the dataset, the time to make it and the calories that the recipe would have for people that were interested on it, it was decided to enrich the JSON file with data on calories, portions, and preparation time for each recipe. The final Document obtained was a JSON file with the recipes, its ingredients, its prices, the portion, its calories and the time to make it as in Figure 10.


```
_id: ObjectId('6687c53758a2f4995962f923')
Calories : 801
Cooking Time (min) : 40
Portions : 3
categoria : "PIATTI UNICI"
▸ ingredienti : Object
titolo : "Bacalhau à Braz"
tot_price_max : 21.011000000000006
tot_price_min : 7.740100000000001
```


Figure 10: ingredients and their prices range

7 Data Quality

7.1 Matching data Quality

The matching of the ingredients of the recipes and the supermarket products was performed using a similarity index. The similarity index was computed by first transforming the text into numerical vectors using the TF-IDF method, which highlights important words and downplays common ones, and then calculating the cosine similarity between the vectors. To ensure the accuracy of the matching, for the calculation of the similarity index, irrelevant words that could artificially inflate the similarity index have been removed, such as brand names, articles, conjunctions, and other unnecessary terms. It was then set a similarity threshold of > 0.7. This allowed to create a comprehensive dataset containing ingredients and their different prices at the supermarket 11.

During the matching process, many null values were encountered. To address this, different words were reassigned to the ingredients and products, including some previously excluded terms such as brand names that could represent the product itself. For these null values, the similarity threshold was lowered to > 0.4 and considered only the matches with the maximum similarity index obtained for each product. Despite these efforts, 38 null values remained, as their similarity indices were too low to establish sufficiently reliable matches.

	ingrediente	product_name	price	source_file
0	Ceci precotti (peso sgocciolato)	Valfrutta Ceci 360 g	5,17	CONDIMENTI_CONSERVE
1	Ceci precotti (peso sgocciolato)	Valfrutta Ceci 360 g	5,17	FRUTTA_VERDURA
2	Ceci precotti (peso sgocciolato)	Carrefour Ceci cotti al Vapore 3 x 150 g	4,74	CONDIMENTI_CONSERVE
3	Ceci precotti (peso sgocciolato)	Carrefour Ceci cotti al Vapore 3 x 150 g	4,74	FRUTTA_VERDURA
4	Ceci precotti (peso sgocciolato)	Carrefour Ceci 330 g	5,07	CONDIMENTI_CONSERVE
...	...	...	...	...
4476	Mele Golden	Mele Golden 900 g	1,99	FRUTTA_VERDURA
4477	Mele Golden	Mele Golden 3 kg	1,33	FRUTTA_VERDURA
4478	Mele Golden	Mele golden Melinda 3 kg	1,79	FRUTTA_VERDURA
4479	Mele Golden	Mele Golden	0,98	FRUTTA_VERDURA
4480	Mele Golden	Mele Golden Bio	3,11	FRUTTA_VERDURA

4481 rows × 4 columns

Figure 11: matched products and ingredients

However, several incorrect matches were also obtained. These incorrect matches were retained to maintain a more complete dataset. It was opted to prioritize completeness over accuracy, beliving that a larger and more thorough dataset can offer a richer and more valuable basis for analysis.

8 Storage

The data obtained from the acquisition, cleaning and matching phases had to be saved in a database to be queried. To represent recipes with the different prices, the best choice was to rely on MongoDB. Being a document oriented database, this was suited more to manage data saved as documents, like a JSON and a CSV in this case.

8.1 Queries

Using MongoDB Compass, various queries have been executed to analyze the recipes and their prices available in the JSON file.

1. The 5 cheapest per portion recipes with low cost products

Figure 12: cheaper recipes per portion

2. The 5 most caloric recipes per portion.

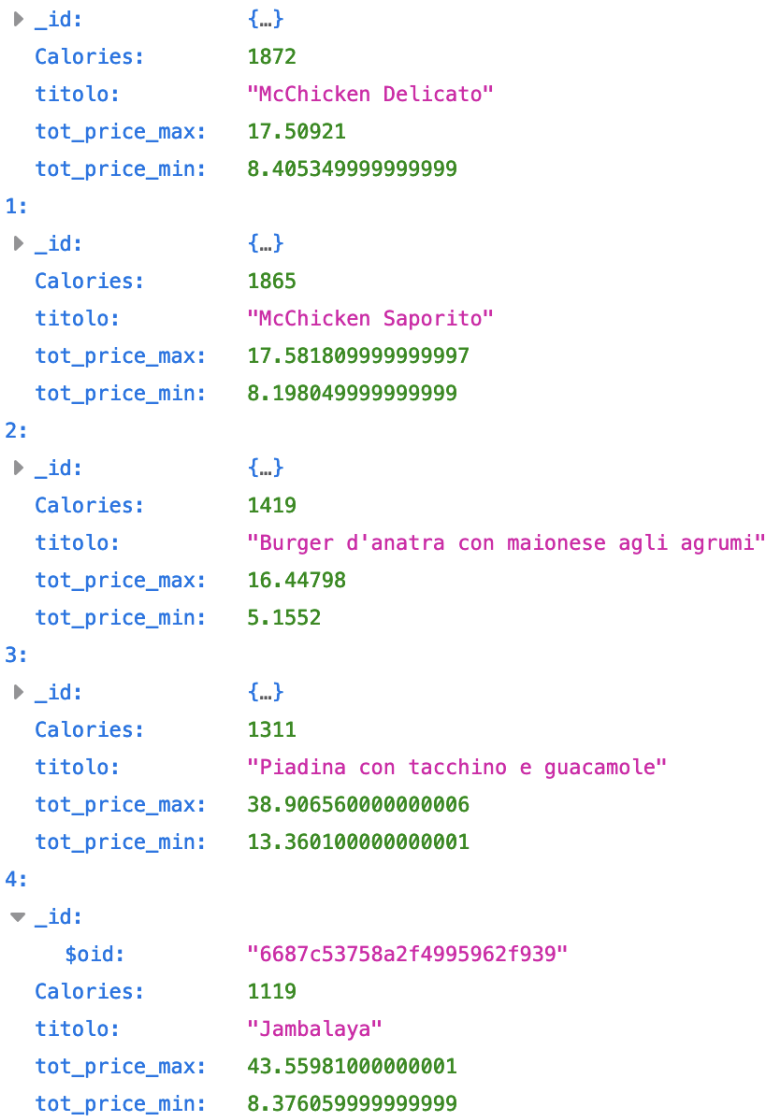

Figure 13: recipes calories per portion

3. The 5 recipes with the highest difference between the price for buying expensive vs cheaper ingredients.

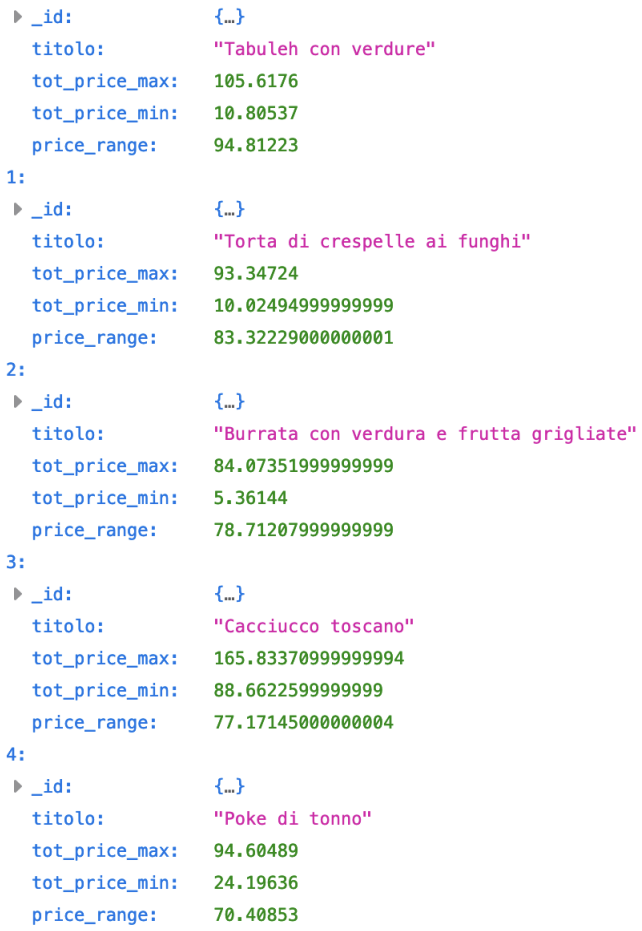

Figure 14: high range recipes

4. The 5 recipes with cooking time < 20 min and with the cheapest price per portion.


```
▸ _id:                  {…}
  Cooking Time (min):   18
  titolo:               "Polenta taragna al gorgonzola"
  tot_price_min:        3.3587399999999996
  price_per_portion:    1.6793699999999998
1:
▸ _id:                  {…}
  Cooking Time (min):   10
  titolo:               "Temaki"
  tot_price_min:        6.839279999999995
  price_per_portion:    1.7098199999999988
2:
▸ _id:                  {…}
  Cooking Time (min):   10
  titolo:               "Carpaccio di zucchine con tonno"
  tot_price_min:        4.7299299999999995
  price_per_portion:    2.3649649999999998
3:
▸ _id:                  {…}
  Cooking Time (min):   17
  titolo:               "Wrap con hummus piccante e verdure"
  tot_price_min:        6.866429999999984
  price_per_portion:    3.433214999999992
4:
▸ _id:                  {…}
  Cooking Time (min):   12
  titolo:               "Piadina con crudo, brie, insalata e salsa cocktail"
  tot_price_min:        6.909399999999965
  price_per_portion:    3.4546999999999826
```


Figure 15: low cooking time and price per portion

9 Conclusions

The development of this project was made easier thanks to online resources like ChatGPT and Copilot, that helped to write the code, and resources from the lectures which were enough to understand how the different instruments worked. Initially the idea was to use of APIs since the data wouldn't need an extensive cleaning process. However, no APIs with Italian recipes were found, so there was no other choice but to perform web scraping. The process of scraping GialloZafferano took several hours, since going from one recipe to another took a very long time, but scraping the Carrefour website was significantly faster.

The data cleaning and matching were the two phases that took more time to develop. It was tricky to find and correct every error. Also the matching phase had some issues: there were some wrongly matching ingredients with products, or no matching at all. This was resolved partially by choosing only the recipes belonging to the category "Piatti unici". Thanks to this, the matching quality increases so to have good results.

The integration of the data into MongoDB was relatively straightforward, no problems were found and the queries were obtained with little to no error.

Surely this project could be improved to have a more extensive coverage of the cooking recipes. Additional data like recipes in English, data about the macro nutrients found in each food, a better calories estimation and a diet guideline, or even new data from multiple grocery stores could be of better use when integrating the data and choosing what to cook.

References

[1] Carrefour. *carrefour spesa online*. 2024. URL: https://www.carrefour.it/spesa-online/.

[2] Cooker.NET. *CUP&SPOON*. 2024. URL: http://www.iispareto.it/wp/wp-content/uploads/2014/04/Misure_in_cucina.pdf.

[3] GialloZafferano. *ricette GialloZafferano*. 2024. URL: https://www.giallozafferano.it/ricette-cat/.