

Project Proposal: supermarket receipt visualizer

Input: scanned receipt from supermarket

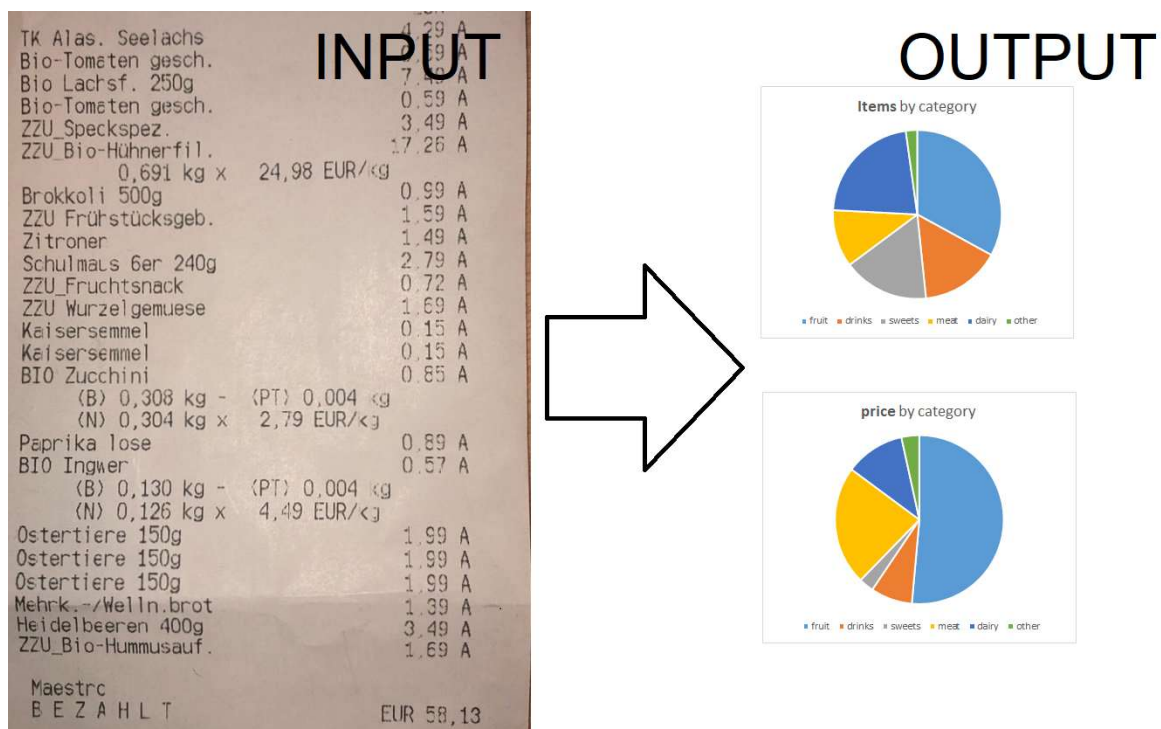
Output: pie chart to see distribution of items categories (fruit/veggy, dairy products, sweets, drinks, etc)

Database in the background will be fed to collect, item, category and **price**.

With NLP I am planning to identify the categories.

With pytesseract I will extract text from receipt image

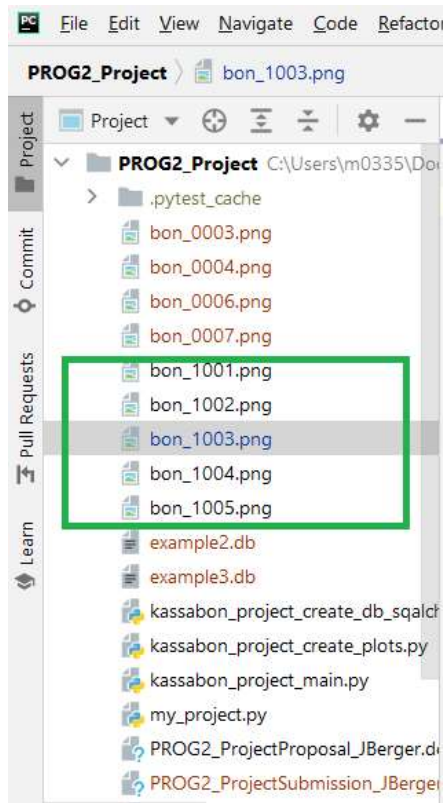
With matplotlib I will do visualisation



I deem this project useful to have an overview on shopping behaviour in terms of preferences on certain categories. We can assume a relationship between shopping behaviour and consumption. I hope to create awareness on shopping and hence consumption behaviour to support creating a healthier lifestyle by adapting groceries.

Project Delivery: supermarket receipt visualizer

Input: scanned receipt from supermarkets (bon_1001.png, bon_1002.png, etc)



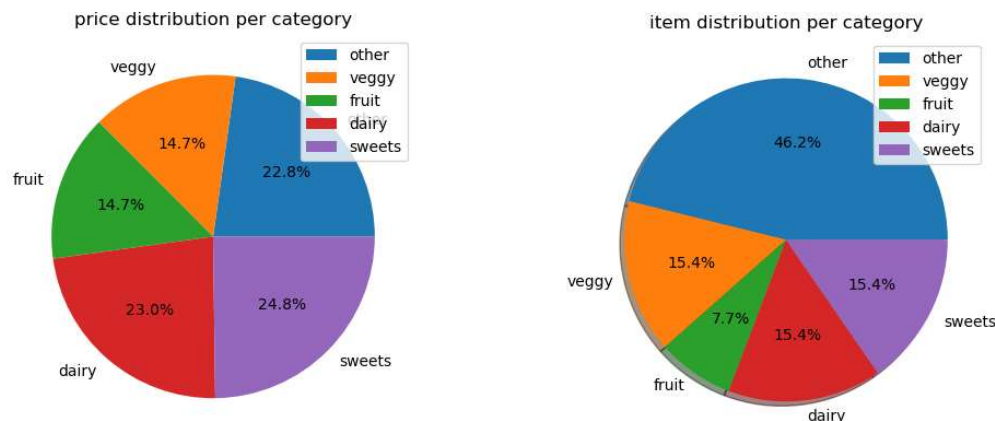
Files:

kassabon_project_create_db_sqlalchemy.py creates database entries

The screenshot shows a DB Browser for SQLite window. The table 'Einkauf' is displayed with the following data:

	id	date	supermarket	item	price	category
1	3276b8f3-c2e4-11eb-b1b9-bce92f82cf3c	2021-05-25	EUROSPAR	PIKEPLACE	7.78	other
2	32791b4f-c2e4-11eb-a454-bce92f82cf3c	2021-05-25	EUROSPAR	STARE	3.89	other
3	327dffc-c2e4-11eb-aa2a-bce92f82cf3c	2021-05-25	EUROSPAR	HASELNUSSE	2.39	other
4	3283a3cf-c2e4-11eb-83ee-bce92f82cf3c	2021-05-25	EUROSPAR	BASMATIREIS	2.49	other
5	328605f7-c2e4-11eb-b23f-bce92f82cf3c	2021-05-25	EUROSPAR	SANDWICH	1.85	other
6	32886895-c2e4-11eb-8a43-bce92f82cf3c	2021-05-25	EUROSPAR	ASIA	1.99	other
7	328acac9-c2e4-11eb-a47d-bce92f82cf3c	2021-05-25	EUROSPAR	PAPRIKA	0.99	veggy
8	328e2b80-c2e4-11eb-baa7-bce92f82cf3c	2021-05-25	EUROSPAR	BROCCOLI	1.49	veggy
9	32908e5a-c2e4-11eb-bb6d-bce92f82cf3c	2021-05-25	EUROSPAR	BANANEN	1.96	fruit
10	3292f0a0-c2e4-11eb-99cd-bce92f82cf3c	2021-05-25	EUROSPAR	ETER	2.99	dairy
11	3295530c-c2e4-11eb-9ba4-bce92f82cf3c	2021-05-25	EUROSPAR	FRISCHKAESE	0.89	dairy
12	329a17bb-c2e4-11eb-965d-bce92f82cf3c	2021-05-25	EUROSPAR	WAFFELN	1.49	sweets
13	329d7861-c2e4-11eb-b00d-bce92f82cf3c	2021-05-25	EUROSPAR	DUPLO	2.69	sweets
14	f245de2a-c2e8-11eb-a6b6-bce92f82cf3c	2021-05-14	SPAR	MUFFINS	1.49	sweets
15	f24a5d6c-c2e8-11eb-a33d-bce92f82cf3c	2021-05-14	SPAR	MILKA	1.69	sweets
16	f24cbe3d-c2e8-11eb-ab4d-bce92f82cf3c	2021-05-14	SPAR	CHOCO	1.89	sweets
17	f24dbde4-c2e8-11eb-bfbd-bce92f82cf3c	2021-05-14	SPAR	MILCHRETS	1.78	sweets

kassabon_project_create_plots.py generates 2 pie plots. The first plot illustrates the distribution of categories over items. The second plot shows the distribution of categories on the prices.



kassabon_project_main.py running all functions like:

1. select_image()
2. extract_text_from_image()
3. define_categories()
4. extract_date()
5. extract_supermarket()
6. confirm_prices()
7. assign_cat_to_item()
8. count_items_per_cat()
9. price_per_category()

my_project.py runs the program addressing the other modules mentioned earlier

```

1  import kassabon_project_main as kassa
2  from nltk.tokenize import word_tokenize
3  from kassabon_project_create_db_sqlalchemy import *
4  from kassabon_project_create_plots import *
5
6
7  text = kassa.extract_text_from_image(kassa.select_image())
8  words = word_tokenize(text)
9
10 #print(text)
11 #print(words)
12 #exit()
13
14 #creates a dictionary with categories as keys and list of tuples as values, where each tuple has item and price
15 dict1 = kassa.define_categories()
16
17 # words getting distributed into the dict1 for each category
18 kassa.assign_cat_to_item(dict1, words, text)
19
20 #supermarket = kassa.extract_supermarket(words)
21 #date = kassa.extract_date(text)
22
23 #populate database with data from dictionary
24 enter_data(dict1, kassa.extract_date(text), kassa.extract_supermarket(words))
25 print(dict1)
26 #create pie plots
27 create_plot_item_pie(kassa.count_items_per_cat(dict1), dict1.keys())
28 create_plot_price_pie(kassa.price_per_category(dict1), dict1.keys())

```

Test_kassabon_project_main.py to test the functions:

1. `extract_date()`
2. `extract_supermarket()`
3. `confirm_prices()`
4. `assign_cat_to_item()`
5. `count_items_per_cat()`
6. `price_per_category()`