

PROJECT DOCUMENTATION

-Grocery Analysis-

1. INTRODUCTION

1.1 Problem Statement: I noticed that I spend too much money on food. To better understand and optimize my expenses, I conducted a data-driven analysis.

1.2 Objective: Develop a workflow for capturing my purchases and subsequently analyzing them to identify potential savings.

2. METHODOLOGY

2.1 Data Capturing

To know which tables my database should have, I analyzed the information given on a receipt. Receipts usually contain the **store**, the **date** of the purchase, and products with **name**, **weight**, **amount** and **prize**. To divide the products into **categories**, i came up with the following categories: "Vegetables", "Fruits", "Milk products", "Cheese", "Bread", "Cleaning utensils", "Meat/Meat replacement" and "further".

Having collected over 70 receipts, i decided to develop a **Python application** to automatically extract data from receipts. My application uses **Tesseract** to extract text from receipts, reducing manual labour.

After small manual adjustments in the **applications UI** and categorizing the products with buttons, a click on "save" uploads each receipt to the **SQLite database**.

The screenshot shows a web-based application interface titled "Produktkorrektur". It features a form for entering receipt data. At the top, there are input fields for "Supermarkname:" and "Datum: DD.MM.YY". Below these is a table with columns: "Kategorie", "Gewicht", "Anzahl", "Produktname", and "Preis". The table contains multiple rows of product data, including categories like "Gemüse", "Obst", "Milchprodukte", "Käse", "Brot", "Haushaltsmittel", "Fleischersatz", and "Sonstiges". Each row has a "Löschen" button next to it. To the right of the table, there is a receipt preview for "EFFENBERGER VOLLKORN - BÄCKEREI". The receipt includes the store's address, contact information, and a list of items purchased, such as "Berliner Landbrot" for €4.90. At the bottom of the receipt preview, the total sum is shown as €4.90. Below the receipt preview, there is a "Speichern" button.

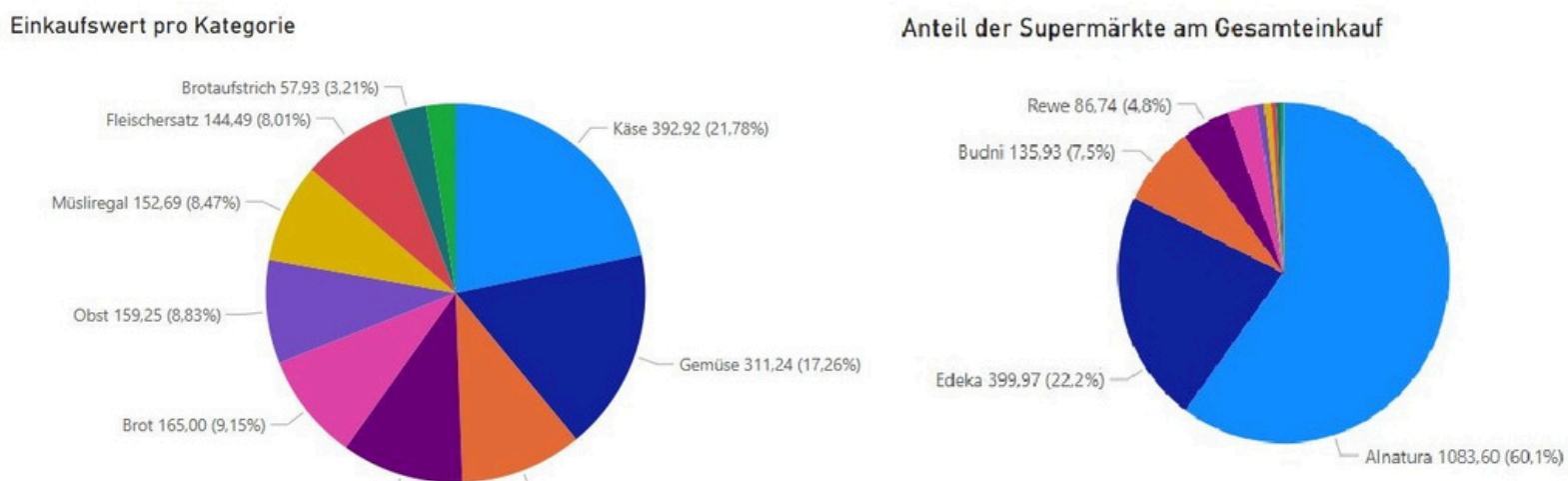
Img. 1: Python application UI

2.2 Data Processing & Analysis

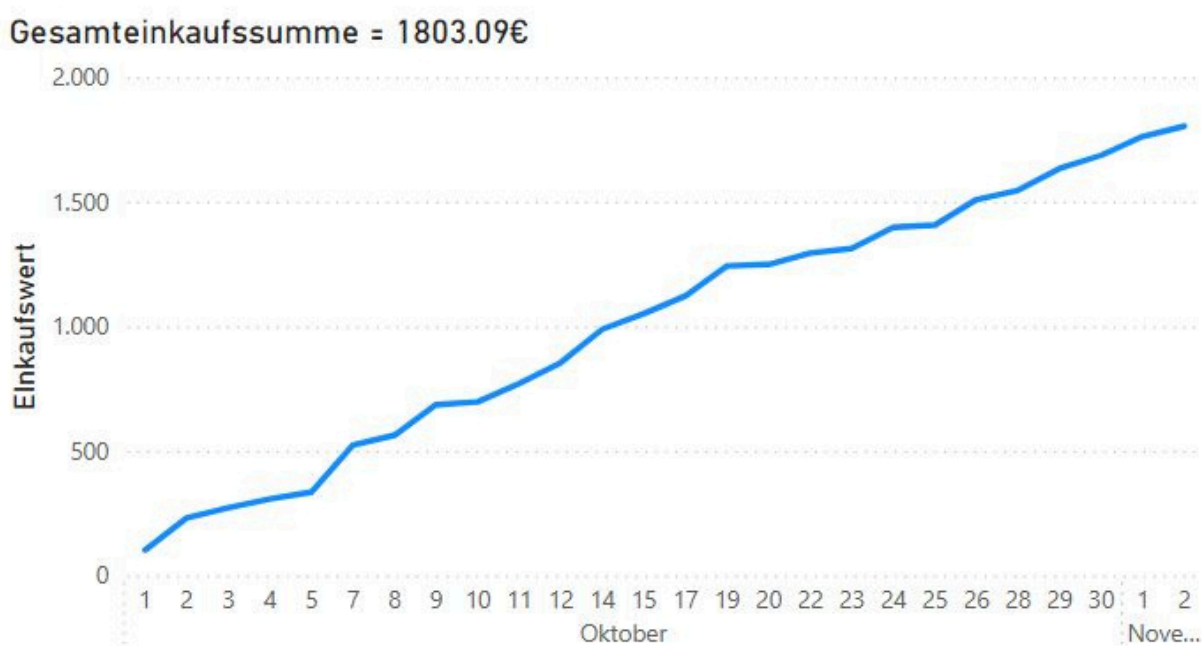
Once the receipt data has been successfully uploaded to the SQLite database, my next step was to export this data into a format suitable for analysis. I extracted the data from the database and converted it into a CSV file for easier manipulation and further analysis.

I used **python's pandas** library to find fraudulent data such as empty fields or numeral outliers to **clean** my **dataset**.

For the **analysis** and **visualization** of the data, I used **Power BI**. The CSV files containing the exported data were imported into Power BI, where I set up a dashboard to track and **visualize expenses** across the categories, time periods, and stores.



Img. 2: Analyzed Data displayed in 2 example graphs



Img. 3: Graph showing expenses over time

3. RESULTS

After visualizing the data, I gained an overview of my monthly food expenses. Creating multiple graphs with **different viewpoints**, I planned to **identify trends** and patterns in my spending habits. By separating products in categories, my spendings became more visibly accessible.

With my analyzed data, I was able to identify which **products**, **categories** and **stores** I spend the **most money** on.

4. NEXT STEPS

For my next steps, I intend to look for insights that could highlight potential saving opportunities. For example, I will explore whether shopping at certain stores leads to lower overall costs or if purchasing larger packaging sizes (**weight**) results in more economical options.

Additionally, comparing the **price per weight** of individual products bought **in different stores** could be very insightful.

With these steps, I am hoping to be able to manage my food budget more effectively.