# **Description of the Datasets**

The study was conducted using four datasets related to supermarket transactions: **Item**, **Promotion**, **Sales**, and **Supermarkets**, each provided in a .csv format. The problem description included brief overviews of each dataset but did not provide detailed descriptions of the variables within them. To conduct a thorough analysis, it is crucial to have an in-depth understanding of each variable. Therefore, **general knowledge** and the **OpenAI ChatGPT-4o** model were utilized to gain a more comprehensive description of each variable. The detailed descriptions of these variables are as follows.

**Item Dataset**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Data Type** |
| **code** | Unique code for identify item. | int64 |
| **description** | Description of the item. | object |
| **type** | Item type (Type 1, Type 2, Type 3, Type 4) | object |
| **brand** | Brand name of the item | object |
| **size** | Weight of the item in Oz or LB | object |

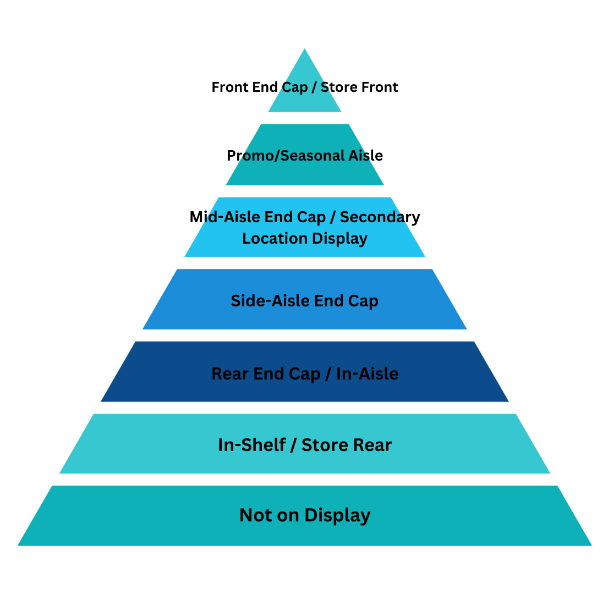
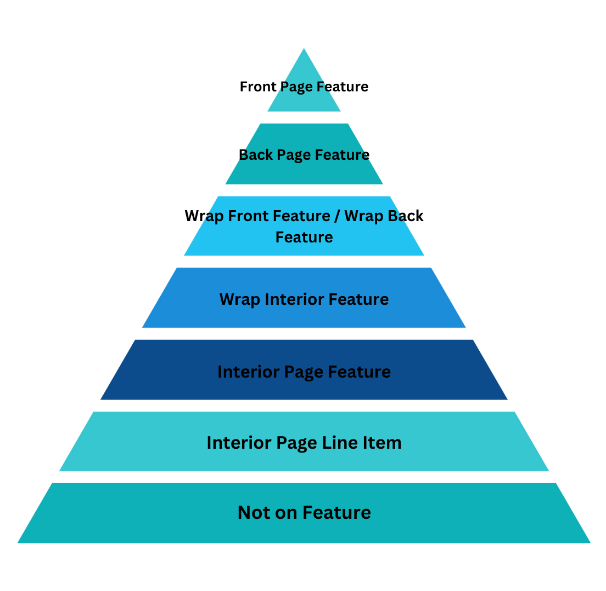
**Promotion Dataset**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Data Type** |
| **code** | Unique code for identify item | int64 |
| **supermarkets** | Unique code for identify supermarket | int64 |
| **week** | Week number of the promotion happen | int64 |
| **feature** | Visibility that products receive in promotional catalog | object |
| **display** | Display locations of products within the store | object |
| **province** | Province of the supermarket (1, 2) | int64 |

The “**Feature”** column represents various types of promotional placements or visibility that products may receive in marketing materials, such as promotional catalogs. In this dataset, there are 8 distinct types of features. Based on the research conducted, a detailed description of each feature type is provided below.

* **Not on Feature:** Product is not in the promotional catalog.
* **Interior Page Feature:** Product is featured inside the promotional catalog.
* **Wrap Interior Feature:** Product is highlighted on the interior wraparound page of the promotional catalog.
* **Wrap Back Feature:** Product is featured on the back of a wraparound page of the promotional catalog.
* **Interior Page Line Item:** Product is listed as a line item within the interior pages of the promotional catalog.
* **Wrap Front Feature:** Product is highlighted on the front of a wraparound page of the promotional catalog.
* **Front Page Feature:** Product is prominently displayed on the front page of the promotional catalog.
* **Back Page Feature:** Product is featured on the back page of a promotional catalog.

The **Display** column provides information about the physical placement or display locations of products within the store. This dataset includes 11 distinct display values, each corresponding to a specific location where a product can be positioned to enhance visibility or serve promotional purposes. A detailed explanation of each display type is provided below.

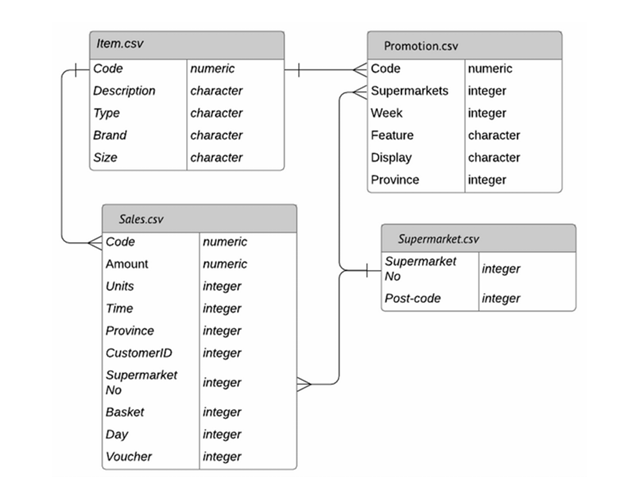
* **Mid-Aisle End Cap:** Display located at the end of a store aisle, in the middle of the store.
* **Not on Display:** Product is not display in the store.
* **Rear End Cap:** Product is displayed on an end cap at the back of the store.
* **Store Rear:** Product is placed towards the back of the store.
* **Front End Cap:** Display at the front of an aisle near the store's entrance or main walkways.
* **In-Shelf:** Product is located on a regular shelf, alongside other products.
* **Store Front:** Product is placed near the front of the store.
* **Secondary Location Display:** Product is placed in an additional display location beyond its regular shelf spot.
* **In-Aisle:** Product is displayed directly within an aisle.
* **Promo/Seasonal Aisle:** Product is featured in a dedicated aisle for promotional or seasonal items.
* ******Side-Aisle End Cap:** Refers to an end cap on the side aisles.

**Sales Dataset**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Data Type** |
| **code** | Unique code for identify item | int64 |
| **amount** | Total amount that customer pay | float64 |
| **units** | Number of units that customer buy | int64 |
| **time** | Purchase time of the order | int64 |
| **province** | Province of the supermarket (1, 2) | int64 |
| **week** | Week number that purchase happens | int64 |
| **customerId** | Unique code for identify Customer | int64 |
| **supermarket** | Unique code for identify supermarket | int64 |
| **basket** | Unique code for identify items that purchased together by customer | int64 |
| **day** | Day number that purchase happens | int64 |
| **voucher** | Any discount applied or not (0,1) | int64 |

**Supermarket Dataset**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Data Type** |
| **supermarket\_No** | Unique code for identify supermarket | int64 |
| **postal-code** | Postal code of region where supermarket located | int64 |

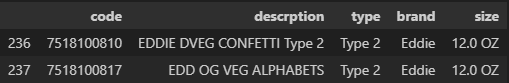
Accordingto the problem statement relationships among the datasets represented as follows.

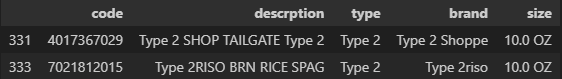
# **Data Cleaning and Pre processing**

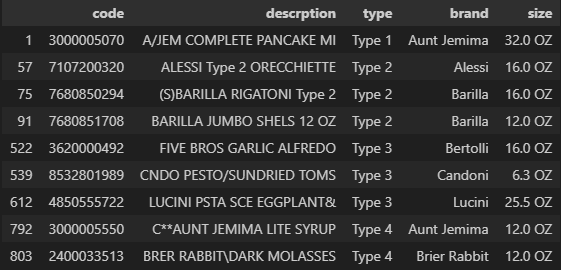
There are no missing or duplicate values in the datasets. However, unrealistic values, duplicate classes, and incorrect data types used in certain variables were identified in some instances. These issues are addressed as follows.

### **Cleaning in Item dataset**

First, attention is directed to the *size* variable. The expected format of this variable is **"[number]<space> [unit]"**. However, it has been observed that 138 instances deviate from this expected format. Consequently, these instances must be corrected to adhere to the proper structure. Given the variations in the formats of these values, manual adjustments are required.

Next, the focus shifts to the *brand* variable, which contains 131 unique brand names within the dataset. An examination was conducted to determine whether the same brand name appeared in different formats, resulting in the identification of one such instance.

Consequently, "Edd Og" is amended to "Eddie." Additionally, it was observed that several brand names were prefixed with "Type 2." This discrepancy is clearly a data entry error, as these values pertain to the *type* variable located adjacent to the *brand* variable. Therefore, these entries have been rectified accordingly.

Subsequently, the focus shifts to the *description* variable. It is observed that the typical structure of the description follows the format: **"<Company Name><Product Name>"**. However, several descriptions deviate from this pattern. To maintain a consistent structure, these descriptions are adjusted accordingly.

It was observed that certain item codes share identical descriptions, as outlined below.

While the item codes and sizes may differ, the codes are assigned uniquely to distinguish between them. A closer examination of the sizes reveals that they are often equivalent, but represented in different units. For instance, in the first row above, 32 ounces (Oz) is equivalent to 2 pounds (LB). This suggests that these entries represent the same item but were recorded under different codes. This conclusion is further supported by analyzing the prices, which appear to be identical on specific days, confirming the initial assumption.

It is important to note **that some items share the same description but differ in size**. Therefore, it is crucial to accurately distinguish these items and avoid making any changes to them. Therefore, the initial step involves converting all values from pounds (LB) to ounces (OZ). Subsequently, similar items are identified and consolidated under a single code. (code which first occurred) Then it is necessary to update these codes in both the *sales* and *promotion* datasets, as the *code* variable is present in both datasets.

### **Cleaning in Promotion dataset**

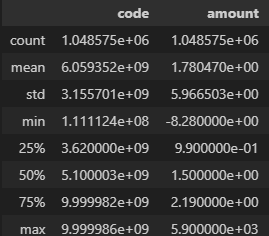
The *promotion* dataset does not require any cleaning. However, it is essential to verify whether there are any values for the *code* variable that are not present in the *item* dataset. Upon conducting this check, it was determined that no such discrepancies exist.

### **Cleaning in Supermarket dataset**

The *supermarket* dataset does not require any cleaning.

### **Cleaning in Sales dataset**

Initially, a verification was conducted to determine whether any *code* values existed in the dataset that were absent from the *item* dataset, as well as whether any *supermarket* values were missing from the *supermarkets* dataset. The results indicated that no such discrepancies were present.

Next, an observation was made regarding the presence of negative values in the *amount* variable. Given that this variable represents currency, negative values are not permissible. Consequently, it is necessary to impute these values to ensure accuracy and consistency within the dataset.

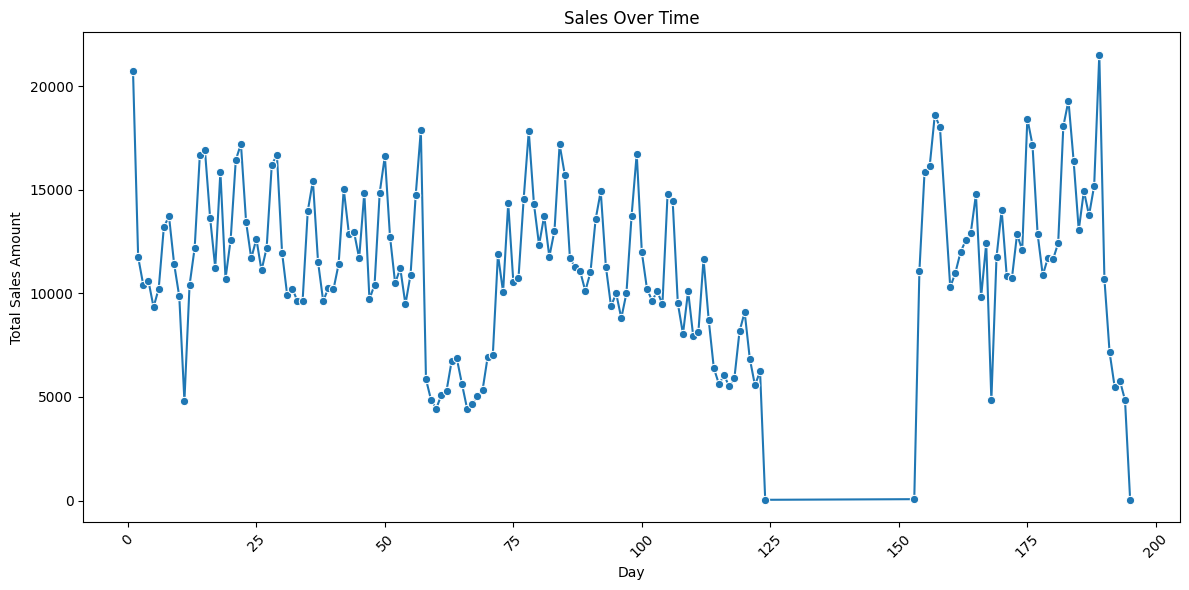
There are 2,426 instances with negative amount values. It is observed that the amount values can vary due to the influence of factors such as day, province, and voucher variables.

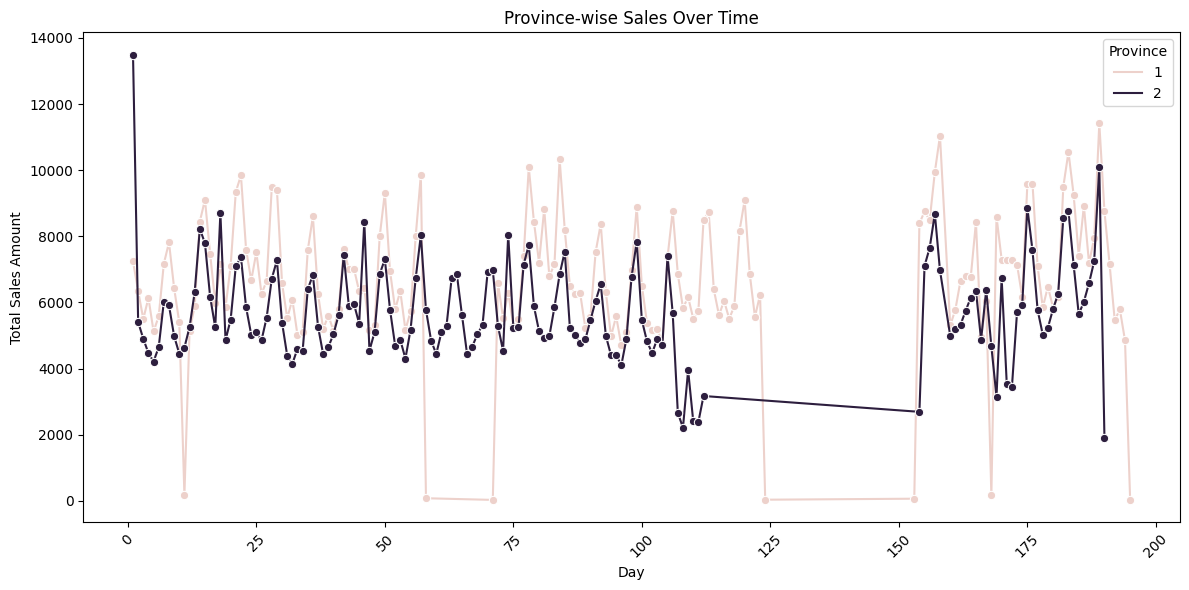
Therefore, when selecting a replacement for these negative values, a systematic approach was employed: first, the unit amounts for a given item were identified based on the specific day, province, and the presence or absence of a voucher. The mode of these values was then selected and multiplied by the number of units to replace the negative values. If no suitable replacement value satisfying the above conditions could be identified, the negative value was replaced with NaN.

Ultimately, it was determined that there was insufficient information to impute 471 values, representing 0.05% of the entire dataset. As a result, it was decided to remove these instances.

# **Business-valued solutions**

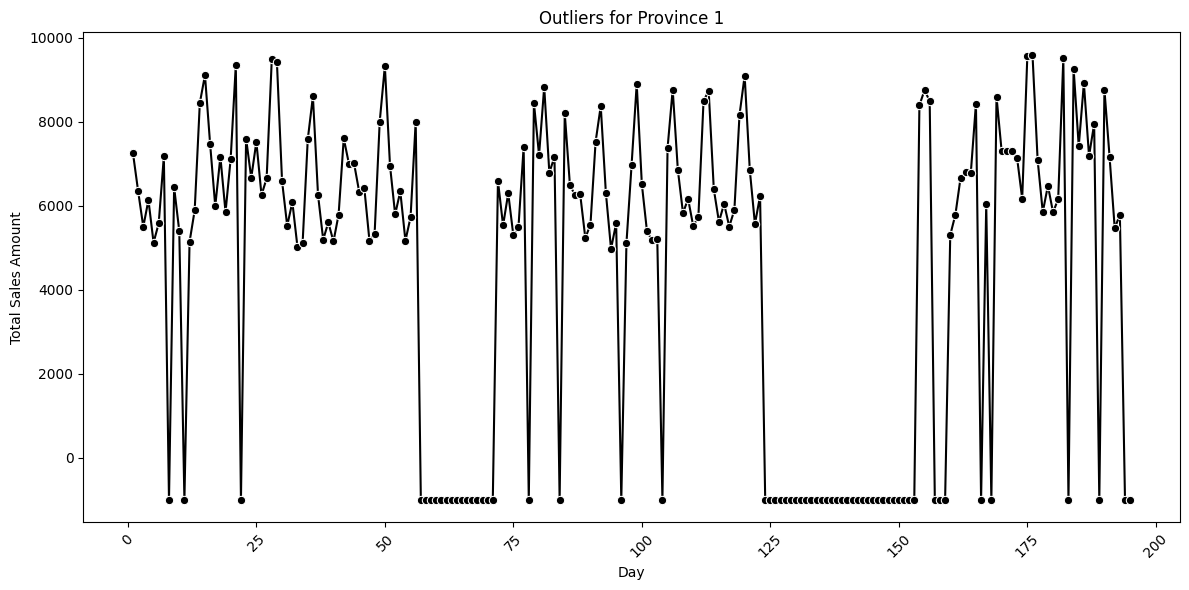
## **Sales Forecasting Model**





We can observe that for some days we don’t have any data for sales. Since this may affect for our prediction model it is important to impute those values. Also we can identify some clear outliers in data. Hence identify other possible outlers and handle them is also very important.

When identifieng outlier we use isolation forest methode. So bellow you can see identified outlers as values 0.



For handle those outliers we use 2 techiniques.

1. Use time-based interpolation for handle isolated outliers
2. Use ARIMA model for handle grouped outliers.

