Setup

Compared to the previous exercise sheets, you should install `mlxtend`, `pyreadr`, and `rdata`. If you have used the `requirements.txt` from ILIAS to set up your environment, you already have these dependencies available. We will also work with `matplotlib` and `pandas` again.

Task: Association Rules

The aim of this exercise is to apply different approaches for frequent itemset mining and association rule mining. We work with the `Groceries` dataset, which contains real-world transaction data from a grocery outlet. You can download the dataset to your current directory with the script `prepare_groceries_dataset.py` provided on ILIAS. Just run `python prepare_groceries_dataset.py` after you have activated the environment for the exercises.

a) **[Transaction Data]** Load the dataset and bring it into a suitable form, e.g., a list of transactions, which are lists of items. Python's built-in `open()` routine in combination with some simple string operations should suffice.
How is the dataset structured? How many different items are there? How is the length of transactions distributed? How is the frequency of items distributed?

b) **[Frequent Itemset Mining]** Use `apriori()` from `mlxtend.frequent_patterns` to determine all frequent itemsets with a minimum support of 5%. `apriori()` requires the transaction data to be in a specially encoded `pandas.DataFrame`. You can use a `TransactionEncoder` from `mlxtend.preprocessing` for that purpose.
Are all of the frequent itemsets also maximal frequent? Why or why not?

c) **[Association Rules Mining]** Use functions `apriori()` and `association_rules()` from `mlxtend.frequent_patterns` to determine all association rules with a minimum support of 1% and a minimum confidence of 40%.
Which five rules have the highest confidence? Which rules contain *yogurt* (as one of the items) on the left-hand side and have a confidence greater than 50%?

d) **[Multi-Level Mining]** Use the mapping contained in `groceries_structure.csv` to aggregate the dataset to `level2`. In other words, the items in the original dataset have values from column `label`, which you should replace with their corresponding `level2` values now. Make sure to avoid duplicate items in each transaction. Extract all rules with a minimum support of 10% and a minimum confidence of 40%.
Why is it reasonable to use a higher support threshold than in the previous subtasks?

e) **[Level-Crossing Mining]** Create a level-crossing representation of the dataset, including the original `label` values besides their respective `level2` values. This means each transaction should contain two 'items' for each actual item now. Make sure this also is the case if there are naming clashes between the two levels. Extract association rules with the previous subtask's thresholds.
Which challenges do you encounter due to the level-crossing representation?