# Data Mining Assignment 1 – Frequent Pattern Mining

## Problem Setting:

You are provided with the dataset **retail.csv**, which contains transactional data from an international e-commerce store. This dataset includes details about customer purchases, such as the country of purchase, quantity of products bought, product category, unit price, and purchase date. Your task is to analyze this dataset to uncover purchasing patterns and relationships between various attributes using frequent pattern mining techniques.

In this assignment, you will:

1. Prepare and preprocess the dataset for analysis.
2. Apply association rule mining algorithms to extract meaningful patterns.
3. Interpret the results to provide actionable business insights.

## Task 1: Data Inspection and Preparation

### a) Understanding and Pruning the Data:

Before conducting any analysis, carefully review the dataset and its columns. A description of the columns is provided later in this document. Ensure you fully understand the data before proceeding.

The provided dataset contains over one million transactions recorded between 01/12/2009 and 09/12/2011. Working with such a large dataset may pose challenges when applying algorithms like Apriori, as they can be computationally intensive. To make the dataset more manageable, you can reduce its size. A common approach is to focus on the most recent transactions, such as the last thirty days available in the dataset. However, ensure that the reduced dataset is still large enough to uncover meaningful and relevant patterns. Explain In your report the steps you took to reduce the dataset.

### b) Handling Missing Values:

The dataset contains some missing or erroneous values. Write code to handle these missing values appropriately. Justify your approach and discuss how it may affect your results in your report.

## c) Data Categorization:

Some features in the dataset, such as **InvoiceDate** (date and time of purchase), **UnitPrice**, and **Quantity**, contain continuous or highly detailed values. To enhance the frequent pattern mining process, it is necessary to create meaningful categories or bins for these features.

For instance, you can group InvoiceDate into time-based categories like “Morning”, “Noon” and “Evening”. For example, a timestamp such as “14/01/2024 18:08:00” could be categorized as “Evening”. This categorization will simplify the data, making it easier to identify patterns when applying frequent pattern mining algorithms.

Specifically, perform the following:

1. Create categories for **InvoiceDate**, **UnitPrice**, and **Quantity** based on logical groupings.
2. Document your choices, explaining the reasoning behind your selected categories. For example:
    - Why did you choose specific time intervals for InvoiceDate?
    - How did you determine ranges for UnitPrice or Quantity?
3. Highlight any potential disadvantages or limitations of your approach.

Note that there is no single "correct" way to create these categories. The key is to justify your decisions and demonstrate an awareness of how they may impact the analysis.

# Task 2: Mining Association Rules

## a) Exploring the Dataset:

Now you will explore the preprocessed data using association rule mining algorithms, such as **Apriori**, to extract frequent itemsets and generate association rules. You can implement an algorithm yourself, it is however incurred to use existing libraries like **apyori** in Python (https://pypi.org/project/apyori/).

Experiment with various **features**, **minimum support** and **minimum confidence** thresholds, and describe how these values affect the number and nature of the rules generated. Detail your findings in the report.

## b) Identifying Market Insights:

Analyze and interpret generated rules to address the following objectives:

- Product Categories: Extract rules involving **StockCode** or **Description** (e.g., “Item A is often purchased with Item B”).
- Purchase Timing: Extract patterns involving **InvoiceDate** (e.g., “Purchases made in the morning are associated with certain product categories”).

For each of these categories, identify **at least three distinct rules** with high support or confidence, and describe the patterns. Reflect on whether the results align with your expectations and how informative support and confidence are in your analysis.

# Additional Notes on Code and Submission:

- Implement your code in any language (Python recommended). Use libraries like:
    - **pandas** for data manipulation
    - **apyori** or **mlxtend** for mining association rules
    - **matplotlib** or **seaborn** for visualizations
- Include your findings in a concise report of 2–3 pages.
- Submit both the code and the report as a **.zip** file via Blackboard by **21/03/2025**.
- The ZIP file should include your full name like, **Lastname_Firstname-studentNumber.zip.**

Questions: In case you have any questions specific to the assignment, please send an email to nick.wils@uantwerpen.be

# Data Description:

The dataset contains the following columns:

- **InvoiceNo**: Unique identifier for each transaction.
- **StockCode**: Unique product code.
- **Description**: Description of the product purchased.
- **Quantity**: Number of units purchased in a single transaction.
- **InvoiceDate**: The date and time of the transaction.
- **UnitPrice**: Price per unit of the product.
- **CustomerID**: Unique ID assigned to each customer.
- **Country**: The country from which the customer made the purchase.

# Evaluation Criteria

Your submission will be evaluated based on the following grading rubric.

| | **Less than 7** | **7 to 10** | **10 to 13** | **14 to 17** | **18 to 20** |
|---|---|---|---|---|---|
| **Writing Style** | The report is very confusing; the writing style is below average. | At places the report is not very clear; there is a lack of clear structure. | The text is overall clear although some parts could be improved. | Most of the text is easy to follow, findings are explained in a clear way. | The text is very clearly and concisely written, illustrative examples and figures are not overused, but added where needed. |
| **Task 1** | Task was not completed. | Data was preprocessed in an unlogical way, no clear motivation behind preprocessing choices given. | Data was preprocessed in an okay way, but motivation behind choices is missing/not very clear. | Preprocessing was done well and motivation (as well as possible disadvantages of the chosen approach) are well explained. | Preprocessing was done well and motivation behind choices is excellent (e.g. backed up by statistical measures/figures). |
| **Task 2 a)** | Task was not completed. | There are some errors in the implementation, not many observations are made about the effect of 'min_support' and 'min_ confidence' on the generated rules. | The algorithm has been implemented correctly, some basic observations about the effect 'min_support'/' min _confidence' are given. | Correct algorithm implementation, the student gives interesting observations about the effect 'min_support'/' min_confidence' and shows clear understanding about why effects occur. | Correct algorithm implementation; analysis of results are excellent and also some Figures and illustrative Examples are also provided. |
| **Task 2 b)** | Task was not completed. | There are some mistakes in the execution of the task, analysis of the results is minimal. | Task has been executed correctly. Basic analysis of the results is given. | Task was executed correctly, the student gives a good motivation for their selection of 'interesting' rules, and the rest of the analysis is interesting as well. | Task was executed correctly, and the analysis of results is excellent. The discussion even goes beyond the questions that were asked in the assignment. |
| **Code** | Code raises many errors. | Code raises some errors or is very unclear. | Code runs but lacks clear structure and readability, only little documentation is given. | Code is readable and sufficiently documented. | Code is very readable and well documented. It is structured in a way that only by (un)commenting single lines, the code for the different tasks can be run. |