Data Mining Assignment 1 – Frequent Pattern Mining

Problem Setting

You are provided with a dataset derived from a market analysis, which contains transactional information from an online grocery retailer. The dataset includes details about customer orders and the products purchased in each order, as well as additional metadata such as product categories and order timing.

Your task is to analyze this dataset to uncover purchasing patterns and relationships between products 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.

Dataset Description

The dataset is distributed as five CSV files:

- **order_products.csv**
  Contains product-level information for past orders.
- **orders.csv**
  Contains metadata about each order.
- **products.csv**
  Maps products to aisles and departments.
- **aisles.csv**, **departments.csv**
  Provide higher-level product groupings.

You are free to decide which files and attributes to use.

Task 1: Data Inspection and Preparation

a) Understanding and Pruning the Data

Before conducting any analysis, carefully review the dataset and its columns. Ensure you fully understand the data before proceeding.

The provided dataset contains over three million transactions. 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. There are multiple valid ways to do this (e.g. sampling orders, users, or items). Explain the steps you took and describe one alternative you considered. However, ensure that the reduced dataset is still large enough to uncover meaningful and relevant patterns.

b) Constructing Transactions

Before mining association rules, you must transform the raw data into a suitable transactional format. Clearly describe how you defined a transaction, how you defined items, and which preprocessing steps you applied to obtain a transactional dataset suitable for frequent pattern mining.

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 encouraged 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 and why 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 product_id or aisle_id (e.g., “Item A is often purchased with Item B”).
- Purchase Timing: Extract patterns involving order_dow or order_hour_of_day (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. Describe how a store manager can use these rules in practice.

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 the report as a PDF via Blackboard by **06/03/2026**.
- **Only the PDF** should be uploaded to Blackboard. Your source code must be made available via a **Git repository** (e.g., GitHub) and the link to this repository must be included in the report.

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

Notes on the Data

To avoid confusion here is some additional explanation:

- In order_products.csv the column add_to_cart_order refers to the place of that product in the order
- In orders.csv the column order_dow refers to the day of the week the order was taken in. 0 corresponds to Sunday, 1–6 to Monday–Saturday.

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 illogical 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_confiden ce' 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, rest of 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