# Frequent Pattern Mining

Niels Van den Broeck

March 21, 2025

## 1 Introduction

In this assignment, I created a Jupyter Notebook to analyze transactional data from an e-commerce store using association rule mining techniques. This report explains the different implementation decisions, and results analysis. The Jupyter Notebook contains more explanations of the code and allows for experiments to be executed in different ways.

## 2 Data Inspection and Preparation

### 2.1 Truncation

The dataset is fairly large, making it computationally hard to apply the Apriori algorithm. So, I created a truncation function to reduce the dataset size without losing meaningful patterns

You can choose to:

- Keep the first N occuring days of the transactions, based on the InvoiceDate
- Keep the first N occuring customers in the dataset, based on CustomerID
- Keep the first N occuring transactions, based on Invoice

For testing purposes, I added a parameter "random" which if enabled, it will not take the first N occuring entries, but select them randomly over the dataset, ensuring a more representative dataset.

### 2.2 Missing Values

Some missing values can simply be replaced by a default value like the description becoming "No description available", or country becoming "Unknown". Although, this might affect some results when apriori is ran on that column, for example when checking rules between countries and products, apriori will interpret the "Unknown" values all as one country, giving it a high score for some products/rules. Therefore, we carefully adjust the dataset depending on the type of rules being generated, by for example removing all transactions with missing countries when searching for rules on countries. Another way would be to simply ignore all rules where "Unknown" is present.

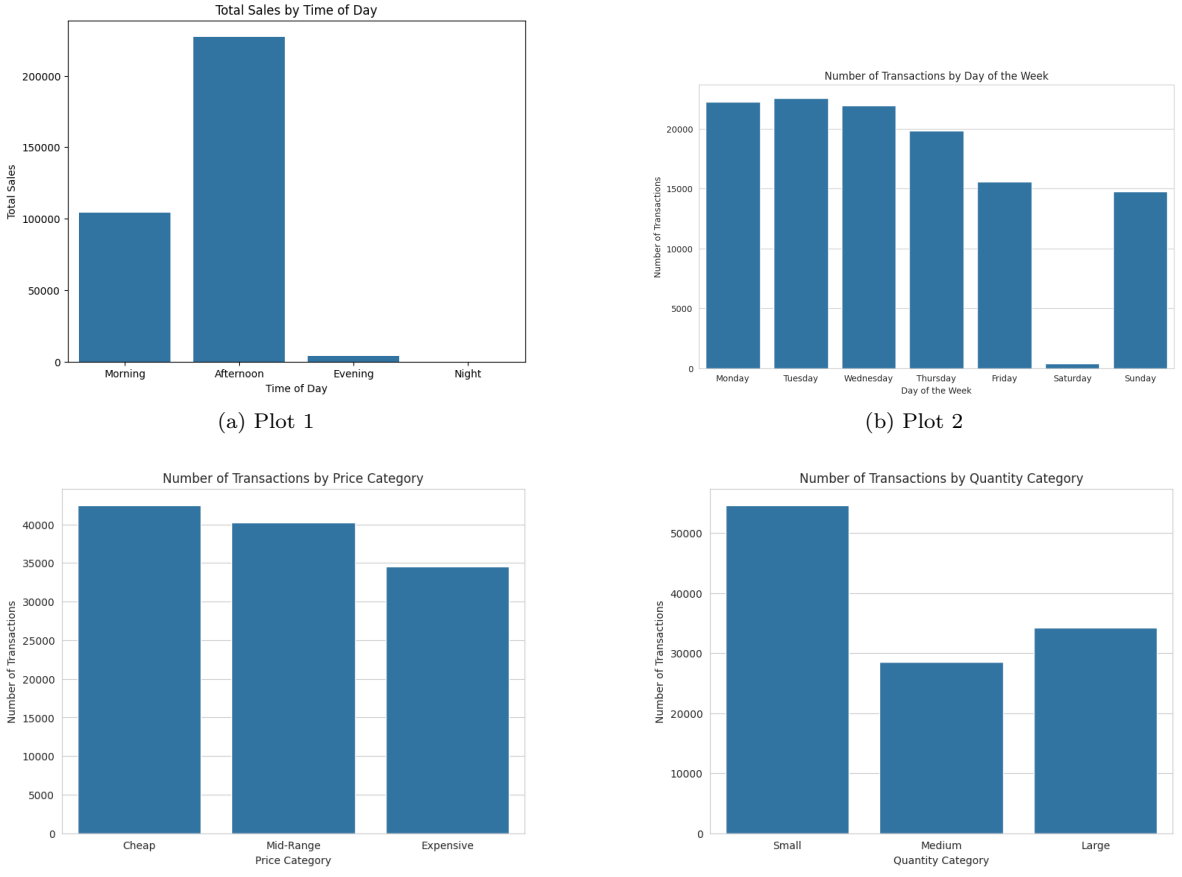
Two important values that must be present all the time are Invoice and StockCode, since these are necessary for creating most rules. If the Invoice is missing, we cannot link the product to a transaction and thus it is impossible to make rules with other products in the transaction. If the StockCode is missing, we simply don't know where to look for the same products. In this case, the description is compared with other descriptions, in case of a match, the StockCode is copied. In all other cases, the entries are discarded.

If the price is missing, we check if the StockCode is already present and copy its price. Otherwise, the price is kept at NaN. Additionally, if the quantity is negative (or zero), it is assumed that the products are returned and thus can be ignored and removed for this project.

## 2.3 Categorization

To create rules in the next part, the dataset is split up in different categories.

- TimeOfDay: The transactions are split up in 4 groups: 'Morning' when purchased between 6am and 12pm, 'Afternoon' until 6pm, 'Evening' until 12am and 'Night' for the remaining hours. This is a logic way to split up the time of day for a e-commerce dataset, since there is a clear distinction between the groups, although we assume that most transaction will take place in the Morning and Afternoon.
- DayOfWeek: Based on the date, the entries are split up in "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" and "Sunday". After generating a plot of the amount transactions for each day, it is visible that there are close to none transactions on Saturday (See Figure **??**)
- PriceCategory and QuantityCategory: These categories are calculated using the qcut from pandas. Qcut splits a column into equal-sized quantiles and assigns each value to a category based on its rank, ensuring that each category has approximately the same number of entries. For price, it is split up in "Cheap", "Mid-Range" and "Expensive". Note that the ranges are calculated based on unique products and not the prices of all transactions, This creates a fair distribution of all products that occur in the dataset. The ranges fall between €0 → €1.65 → €3.9 → €647.19. For quantity, it is split up in "Small", "Medium", "Large". These range between 1 → 2 → 6 → 19152. in figure **??**, you can see some of the categorizations.




(a) Plot 1

(b) Plot 2

(c) Plot 3

(d) Plot 4

Figure 1: Four Plots

# 3 Mining Association Rules

In this section, I will experiment with different features and apriori configurations. The features are the different columns in the dataset. The minimum support is the is minimum percentage of occurrence of a certain relation. For example, when there is a rule between product A and B, the support is equal to all the transactions that include A and B divided by all the transactions. The minimum confidence is the minimum percentage of times that a rule holds true when the left side of the rule is present. Using previous example, For rule A→B, it is all the transactions that include A and B divided by all the transactions that include A. This can also be seen as "If A is bought, there is a x% (confidence) chance that B will also be bought." The minimum lift is the minimum value of how likely two items are to be bought together compared to being bought independently. If the value is greater than 1, they are positively correlated. When the value is close to 1, they are independent, and otherwise negatively correlated. I choose a minimum lift of 1.1 since i am already able to experiment with the minimum support and confidence.

## 3.1 Product-Product Rules

Firstly, we will look at the relation between products themselves.

**Rule 1:** *RED HANGING HEART T-LIGHT HOLDER → WHITE HANGING HEART T-LIGHT HOLDER*

| Support: | Confidence: | Lift: |
|---|---|---|
| 0.0505 | 0.7994 | 4.6266 |

This rule indicates that if you buy a red hanging heart T-light holder, you will most likely buy a white hanging heart T-light holder as well. The support of 0.0505 indicates that out of all transactions, 5.05% include these 2 products. The Confidence says that if you buy the red holder, there is 79.94% chance of also buying the white one.

**Rule 2:** *BLUE FELT EASTER EGG BASKET → PINK FELT EASTER EGG BASKET*

| Support: | Confidence: | Lift: |
|---|---|---|
| 0.0167 | 0.8737 | 35.2741 |

*PINK FELT EASTER EGG BASKET → BLUE FELT EASTER EGG BASKET*

| Support: | Confidence: | Lift: |
|---|---|---|
| 0.0167 | 0.6748 | 35.2741 |

Here you can see that rules often go in both ways. The support will obviously stay the same but the confidence might differ a lot. In this case, there are fewer blue felt Easter egg basket products in a transaction that do not contain the pink one, than there are pink Easter egg basket products that do not contain the blue one. Hence the difference in confidence.

**Rule 3:** *60 TEATIME FAIRY CAKE CASES, PACK OF 60 PINK PAISLEY CAKE CASES → PACK OF 72 RETRO SPOT CAKE CASES*

| Support: | Confidence: | Lift: |
|---|---|---|
| 0.0207 | 0.7687 | 8.7751 |

It is also possible to have a rule between more than 2 items. In this rule, buying the fairy cake cases, as well as the paisley cake cases, will result in a high chance of also buying the retro spot cake cases.

All the above rules are expected since they highly correlate with each other. It makes sense that blue Easter egg baskets and pink Easter egg baskets are often bought together, for example in a kindergarten for an activity. In Figure 2, all product rules are plotted with support on the x-axis, confidence on the y-axis, and lift is represented by the color and size of the markers.

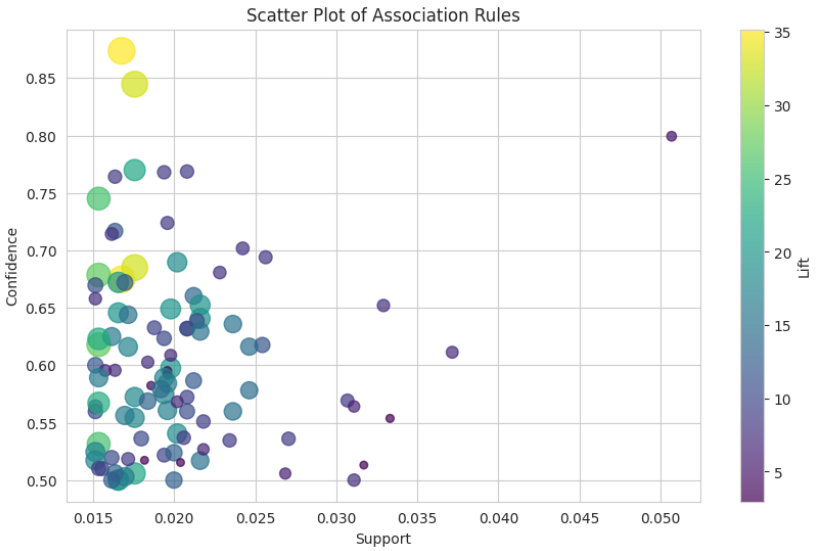



Figure 2: Product Rules

## 3.2 Extended Rules

Next, we will examine more complex rules based on the categories we introduced earlier.

**Rule 1: Time of Day Based** *CHOCOLATE HOT WATER BOTTLE, Afternoon → HOT WATER BOTTLE TEA AND SYMPATHY*

| Support: | Confidence: | Lift: |
|---|---|---|
| 0.0202 | 0.5405 | 7.8513 |

In this rule, we still look at the products, but we introduced a new feature namely the time of day category. We can see that if the chocolate hot water bottle is bought, and it is afternoon, that it is likely that the hot water bottle tea and sympathy is also bought. Although that the confidence is not that high, it is still a great insight to for example, show these products in the afternoon.

**Rule 2: Price Category Based** *17589.0 → Cheap*

| Support: | Confidence: | Lift: |
|---|---|---|
| 0.0010 | 0.7724 | 2.1316 |

Now, we will get rid of the products and look at the customers and their buying behavior. In this rule, you can see that customer with id 17589.0 mostly buys cheap products. Even though the support is not high (meaning that the customer has not that much transactions compared to the entire dataset), the confidence shows that this is a solid rule.

**Rule 3: Price-Quantity Relation** *Expensive → Small*

| Support: | Confidence: | Lift: |
|---|---|---|
| 0.2017 | 0.6851 | 1.4717 |

Lastly, I wanted to check whether there is a clear relation between the quantity in a transaction and the price of the product. It is logical that expensive products are often bought in small quantities. With this rule, we can say that when expensive products are bought, there is a 68.51% chance that it will be bought in small quantities. Other rules with these relations can be viewed in Figure 3 Note that products in this dataset are classified expensive from €3.9 and above, while transactions are classified small between 1 and 2 items. Indicating the smaller than expected confidence.

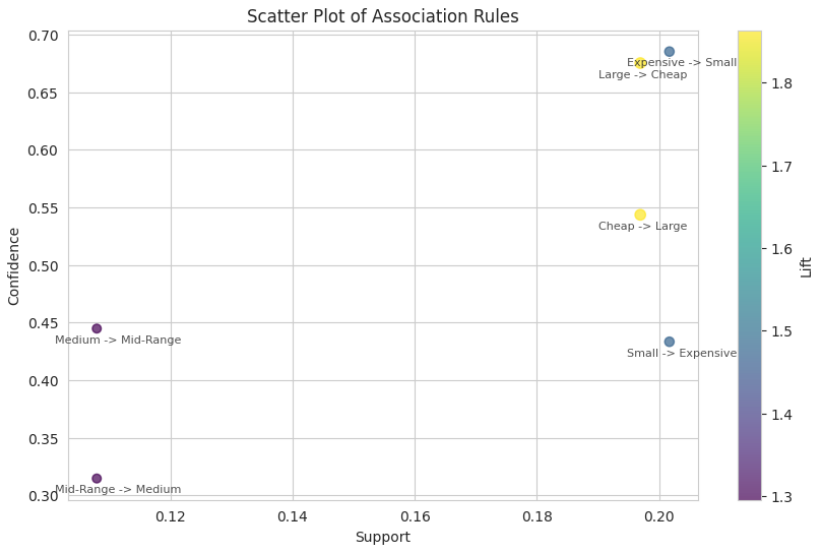



Figure 3: Price-Quantity Rules