PRODUCT RELATIONSHIP ANALYSIS SYSTEM CONCEPT

Samuel Jendryke

XENOS Schutweg 5, 5145 NP Waalwijk

Luty, 2026

Introduction

The goal of this project is to create an analytical tool that automatically identifies groups of products frequently purchased together based on order history. Instead of relying on predefined product categories or intuitive assumptions, the system will use real sales data to detect relationships between items.

This project represents the first stage of a broader concept related to the future optimization of warehouse and logistics processes. At this stage, the focus is not yet on physical product placement in the warehouse or on simulating picking routes. The primary objective is to build a solid data foundation that reveals which products actually occur together in orders and what their real operational significance is. In addition to generating clusters, the system will also determine their relative importance based on historical order data, making the results easier to interpret and enabling their use in later project phases.

The system is intended to operate with minimal user interaction — the user initiates the analysis and receives results in the form of product-to-cluster assignments, optionally providing only a few basic parameters. Simplicity of use does not imply technological simplification. The project assumes the use of advanced graph analysis and clustering methods to ensure that the results are reliable and suitable for further applications.

Such solutions are neither experimental nor new. Analysis of products purchased together has been used for years in large e-commerce platforms and analytical systems. The reason is simple — product categories rarely reflect how customers actually shop. Being in the same category does not mean products are bought together, and the opposite is also true.

Why is this needed? With thousands of SKU, identifying recurring patterns purely through observation becomes extremely difficult. In practice, many warehouse decisions are still based mainly on intuition and assumptions because there is no tool that clearly shows real relationships between products. Automatic clustering and measuring their significance within real orders will not solve every problem, but it provides a more objective point of reference. As a result, future logistical and analytical decisions can be based on data rather than solely on experience.

.

1. Scope and Objectives of the Project

The scope of the project includes developing a tool that enables analysis of relationships between products based on order history and automatically creates groups of items frequently purchased together. An additional element of the analysis will be determining the relative importance of the resulting clusters based on real order data, allowing for a better understanding of their operational significance. The objective is to produce clear and repeatable results that can be used in future initiatives, such as warehouse optimization projects. The system is intended to be simple to use while remaining grounded in reliable data analysis and prepared for further development. From the outset, the project assumes an approach where quality, reliability, and stability of results take priority — the selection of methods and solutions will be driven not by ease of technical implementation, but by their real analytical value.

The following elements should be included within the scope of the project:

- use of real order data as the primary source of information about relationships between products,
- development of an automated method for retrieving input data through integration with one of the existing systems,
- creation of a single, consistent, and repeatable approach to grouping items instead of relying on intuition or manual decisions,
- determining the relative importance of clusters based on order history,
- use of the most suitable available methods and algorithms to ensure reliable and stable analytical results,
- maximizing simplicity of use — running the analysis should require only basic user actions,
- preparing the system to operate on different datasets in the future and to serve as a foundation for further projects,
- ensuring that results remain independent of existing product category structures,
- enabling comparison of results across different time periods,
- ensuring safe operation of the tool without impacting live operational systems,
- presenting results in a clear and easy-to-interpret manner,
- minimizing additional technological dependencies wherever possible.

2. Simplified System Workflow

Data

- retrieving order data from the selected time period in the form of rows: order_id, order_date, sku_id, qty (optional),
- preparing a list of products assigned to each order,
- initial data preprocessing, including excluding orders containing only a single SKU and optionally limiting very large orders to reduce noise.

Illustratively: list of orders where each order is treated as a set of products, e.g.

Order 1001 → [SKU_A, SKU_B, SKU_C].

Identifying Co-Occurrences

- identifying products that appeared within the same orders,
- counting all pairs of SKU that occur together.

Illustratively: table showing how many times two products appeared together, e.g.

SKU_A + SKU_B → 120 times,
SKU_A + SKU_C → 15 times.

Calculating Relationship Weights Between SKU

- calculating the strength of relationships between products using the Lift metric,
- filtering out weak or random relationships,
- preparing a product relationship graph.

Illustratively: a network of connections between products, where thicker lines represent products that are more frequently purchased together.

Cluster Creation

- grouping products based on the relationship graph using the Louvain algorithm,
- assigning each product to a specific cluster_id..

Illustratively: a list of product groups, e.g.

Cluster 1 → [SKU_A, SKU_B, SKU_D],
Cluster 2 → [SKU_X, SKU_Y].

Determining Cluster Importance

- analyzing orders to determine how often products belonging to specific clusters appear,
- assigning relative importance to clusters based on their share in orders..

Illustratively: a summary showing the significance of clusters, e.g.

Cluster 1 → 68% of orders,
Cluster 2 → 21% of orders,
Cluster 3 → 11% of orders.

3. Input Data

In order to create product clusters, it is necessary to obtain basic information about orders. The minimum data scope should remain as simple as possible to ensure that the analysis is stable and easy to maintain.

A key assumption of the project is to keep the process as simple as possible from the user's perspective. Manual data entry is not an option — with thousands of SKU and a large volume of orders, it would be impractical and prone to errors. Therefore, data should be provided automatically through integration with a system that already contains the required information.

Required Data

- order_id — order identifier,
- order_date — order date (used to define the analysis time range),
- sku_id — product identifier,
- qty (optional) — quantity of the product in the order.

At the current stage, it is not required for cluster creation; however, it is recommended to include it to maintain data completeness and enable future analyses.

Initial Data Preparation

- before starting the analysis, orders containing only a single product (1-SKU) should be filtered out, as they do not provide information about relationships between products and may distort statistical calculations,
- each order is treated as a set of unique SKU (without duplicates within the same order).

Possible Data Sources

- **Option A — Integration with Locus WMS (recommended)**
  The Locus system includes an archival LTT database that allows direct connection and retrieval of order data for a selected time range. This approach provides exactly the information required for analysis without additional intermediaries or the need to expand the existing infrastructure. Due to its implementation simplicity and immediate data availability, this is currently the preferred solution.
- **Option B — Integration with the ORP system**
  An alternative would be connecting the application to the ORP system. In this case, integration would need to be developed in cooperation with Wolfpack. This may involve additional costs, dependency on external work, and potentially longer project timelines.
- **Option C — Integration with other systems (e.g., Yellowfin)**
  It is theoretically possible to use data from other analytical platforms such as Yellowfin. However, this would require prior verification of available integration methods and confirmation that the data is accessible in a suitable format.

Recommended Approach

Since integration with the Locus LTT database provides direct access to the required data and does not require additional intermediary layers, it represents the most practical and fastest solution at the current stage of the project.

4. Identifying Product Co-Occurrences

After retrieving the order data, the next step is to determine which products actually appear together within the same orders. At this stage, the system does not yet analyze relationship strength or create clusters — its role is solely to identify all product co-occurrences.

Each order is treated as a list of SKU. For every order, a set of product pairs that appeared together is generated. In practice, this means that if an order contains three products, all their combinations are recorded as potential relationships.

Example:

Order 1001 → [SKU_A, SKU_B, SKU_C]

Based on this order, the following pairs are created:

SKU_A + SKU_B
SKU_A + SKU_C
SKU_B + SKU_C

Each subsequent order increases the occurrence count for these pairs. As a result, a simple table is created showing how many times specific products appeared together.

Illustratively:

Product pair	Number of co-occurrences
SKU_A + SKU_B	120
SKU_A + SKU_C	15
SKU_B + SKU_D	87

At this stage, no evaluation is made as to whether a relationship is strong or random — only factual co-occurrence data derived from orders is recorded. The resulting list of co-occurrences forms the foundation for the next step, where the strength of relationships between products will be calculated.

5. Calculating Relationship Strength Between Products

After determining which products appear together in orders, the next step is to assess how strong these relationships are. Simply counting co-occurrences is not sufficient — popular products may appear together with many others solely because they are frequently purchased. For this reason, the Lift metric is used to evaluate relationship quality, followed by relationship filtering to retain only reliable connections.

What is Lift

Lift is a metric that indicates whether two products are purchased together more often than would be expected by chance. It helps distinguish situations where products truly have a purchasing relationship from those where they simply appear frequently in orders because they are popular. In practice, Lift answers the question:

Does the presence of product A in an order increase the likelihood of product B appearing as well?

Formula:

$Lift(A,B) = P(A \cap B) / (P(A) * P(B))$

Practical Form:

$Lift(A,B) = (together\_count * total\_orders) / (count\_A * count\_B)$

Relationship Filtering

After calculating the Lift values, a filtering step is applied to remove random or overly weak relationships before building the product graph.

At this stage, the following criteria are considered in particular:

- a minimum number of co-occurrences for a product pair (min_edge_support),
- a minimum Lift threshold (lift_threshold).

Filtering applies to relationships between products rather than to the SKU themselves — this makes it possible to preserve seasonal or promotional clusters while limiting the influence of random co-occurrences resulting from a small number of orders.

Example relationships after calculating Lift:

SKU Pair	Together Count	Lift
SKU_A + SKU_B	120	2.4
SKU_A + SKU_C	3	5.8
SKU_D + SKU_E	18	1.3
SKU_X + SKU_Y	1	9.5

Assume the following thresholds:

- min_edge_support = 10
- lift_threshold = 1.2

After filtering, only relationships that meet both conditions remain:

SKU Pair	Together Count	Lift
SKU_A + SKU_B	120	2.4
SKU_D + SKU_E	18	1.3

Relationships such as SKU_A + SKU_C or SKU_X + SKU_Y are discarded — despite having a high Lift — because the number of co-occurrences is too low to be considered statistically stable.

Lift Value Interpretation

Lift ≈ 1 — no meaningful relationship,
Lift > 1 — products are purchased together more often than random chance,
Lift < 1 — products appear together less often than expected.

Example:
Number of orders: 10,000
SKU_A appears in 2,000 orders
SKU_B appears in 1,000 orders
SKU_A and SKU_B appeared together in 600 orders

Lift = (600 * 10,000) / (2,000 * 1,000)
Lift = 3

A value of 3 means that the products appear together three times more often than would be expected by chance.

The method used to calculate and filter relationships based on Lift has a direct impact on the quality of clusters in the next stage of the analysis. For this reason, it is necessary to test different parameter values — such as the Lift threshold or the minimum number of co-occurrences — and select the most optimal configuration. In some cases, Lift alone may not be sufficient; the analysis can then be extended with additional metrics, such as Confidence or the Jaccard index, to increase the stability and reliability of the results.

After filtering is applied, a set of relationships with sufficient strength and statistical stability is obtained. This set forms the foundation for building the product graph used in the next stage — cluster creation.

6. Product Cluster Creation (Louvain)

After calculating the strength of relationships between products, the next step is to group them into larger sets, referred to as clusters. The goal is not to create new categories or artificial classifications, but to identify natural product groupings that emerge directly from order data.

The **Louvain** algorithm is used for this task. It was developed for analyzing large relationship networks and is widely applied in data science to detect natural data communities.

In this case, the relationship network is formed by products, and the connections between them are derived from the previous calculation stage (**Lift**). Each product represents a node in the graph, and the strength of connections reflects how frequently products appear together in orders.

How the Algorithm Works

The algorithm analyzes the entire structure of relationships simultaneously and identifies groups of products where internal connections are stronger than connections with the rest of the network. The process is iterative:

- initially, each product forms its own group,
- the algorithm evaluates whether moving a product to another group improves the quality of the partition,
- products with strong relationships begin to form clusters,
- the structure is simplified and analyzed again,
- the process ends when further changes no longer improve the result.

The number of clusters is not defined manually — it emerges directly from the data.

Applications in Logistics and Warehouse Systems

An approach based on analyzing relationships between SKU is not new and is used in modern logistics systems, particularly in the areas of slotting optimization and order behavior analysis.

Examples of systems using similar concepts include:

- Manhattan Associates WMS — optimization of product placement based on order relationships,
- Blue Yonder (JDA) — SKU relationship analysis and warehouse layout recommendations,
- SAP EWM — advanced slotting strategies that consider product co-occurrence frequency,
- 3PL and e-commerce solutions using so-called order affinity or basket analysis.

In many warehouses, analyses of this type are embedded within systems as part of larger optimization modules. The proposed tool implements the same concept more directly — by creating an independent analytical layer based solely on order data.

Example Result

The result of the algorithm is the assignment of each product to a specific cluster:

Sku Id	Cluster Id
SKU_001	1
SKU_002	1
SKU_003	1
SKU_014	2
SKU_015	2
SKU_021	3

This can also be presented as a list of groups:

Klaster	Sku
Klaster 1	SKU_001, SKU_002, SKU_003
Klaster 2	SKU_014, SKU_015
Klaster 3	SKU_021

The result of this stage is an organized mapping of sku_id → cluster_id. The resulting clusters form a structured analytical layer describing real relationships between products and become the starting point for further analysis, such as determining the importance of individual clusters or leveraging the data in future projects.

7. Determining Cluster Importance

After clusters are created, their actual operational importance can be evaluated based on historical order lines. The goal of this stage is not to change the cluster structure, but to determine which product groups generate the highest workload in the picking process.

How the analysis works

- each order line is assigned to a cluster based on sku_id,
- the total number of order lines belonging to each cluster is counted,
- the share of each cluster's order lines relative to all order lines in the analyzed period is calculated,
- this share is treated as a measure of relative cluster importance — the higher the share, the more frequently the cluster participates in picking operations and the greater the operational load it generates.

Example

Order lines:

Order 1001 → SKU_A (Cluster 1), SKU_B (Cluster 1), SKU_X (Cluster 3)
Order 1002 → SKU_C (Cluster 1), SKU_Y (Cluster 2)

Line count:

Cluster 1 → 3 lines
Cluster 2 → 1 line
Cluster 3 → 1 line
Total → 5 lines

Cluster importance summary:

Cluster 1 → 60% share of order lines
Cluster 2 → 20%
Cluster 3 → 20%

Interpretation

- a higher percentage indicates greater real usage of the cluster in warehouse operations,
- the metric reflects picking activity frequency rather than simple cluster presence in orders,
- analyzing data at the order-line level reduces the influence of single highly popular SKUs on the final result.

At this stage, it is an additional analysis that does not affect the clustering process itself, but helps to better interpret the obtained results.

8. Possible Future Applications in the Warehouse

The current project concludes with the creation of product clusters and the determination of their relative importance. At this stage, the system does not interfere with the warehouse layout or ongoing operational processes. However, the data obtained provides a solid foundation for further analysis and potential improvements in the future.

Key potential directions for using the results:

- **Support for optimizing warehouse layout and picking routes**
  Clusters reveal which products actually appear together in orders, which can support decisions related to SKU placement, operational zone design, and reducing picker travel distance — both for single-order picking and batch picking.
- **Replenishment process optimization**
  Cluster analysis can help identify products that are frequently consumed together. This enables:
    - o planning replenishment for groups of products instead of individual SKU,
    - o reducing situations where pickers encounter multiple empty locations within the same route,
    - o better prioritization of replenishment based on cluster importance.
- **Data-driven operational decision-making**
  Instead of relying solely on intuition, planned changes can be evaluated against real data describing relationships between products.
- **Foundation for future simulations and optimization tools**
  Clusters and their importance can be used in future projects, such as simulating different warehouse layout variants or analyzing the impact of operational changes on the picking process.
- **Detection of potentially suboptimal layout decisions**
  Cluster analysis makes it possible to identify situations where products frequently purchased together are located far apart within the warehouse. This can signal the need to reassess SKU placement and support more informed layout decisions based on real operational data.

It is important to emphasize that these are potential applications resulting from the analytical capabilities of the project. Implementing specific changes within the warehouse environment would require a separate project phase and additional operational analysis.