Current Topics in Computer Science

Topic 2: Product Mix and Clustering

Final Project Report

October 27, 2025

Contents

Abstract

This project investigates **product mix optimization using clustering techniques**. By analyzing transaction data, we aim to identify natural groupings of products that can support marketing strategies, inventory planning, and sales forecasting. A business intelligence (BI) system with a data warehouse, ETL pipeline, and dashboard was developed to demonstrate insights for decision-making.

Chapter 1

Introduction

1.1 Topic investigation and data source analysis

1.1.1 Topic investigation

The selected topic, Product Mix and Clustering, focuses on identifying optimal sets of products (SKUs) for each store or customer cluster within CompanyX's sales ecosystem. The goal is to reduce redundant SKUs while maintaining key business performance indicators such as revenue, profit, and sales quantity.

In retail and distribution, the product mix (also called the product assortment) refers to the combination of products a company offers to the market. It is often described by four dimensions:

Width: The number of product lines offered.

Length: The total number of SKUs across all lines.

Depth: The number of variants per product line.

Consistency: The degree of similarity among product lines.

1.1.2 Real-world example

In the context of consumer goods and retail, companies often face challenges related to excessive product mix complexity, redundant SKUs, and fluctuating market demands. Several multinational corporations have implemented data-driven product mix optimization and clustering approaches to streamline operations, reduce costs, and improve decision-making. The following three case studies illustrate these practices.

Case 1: Unilever – SKU Rationalization to Streamline Portfolio

Problem: Unilever faced excessive SKU proliferation across its global brands such as Dove, Lifebuoy, and Knorr. Many product variants contributed minimally to sales but significantly increased manufacturing and distribution complexity. The company realized that maintaining too many low-selling SKUs led to inefficiencies such as longer production runs, warehouse congestion, and higher transportation costs.

Analytical Approach: Unilever adopted a data-driven SKU rationalization framework. By analyzing historical sales, revenue, and margin data, it clustered SKUs based on performance (e.g., high-revenue, high-margin vs. low-performing SKUs). Advanced analytics identified redundant variants that overlapped in consumer need but added little incremental revenue.

Solution & Outcome: Through data segmentation and optimization, Unilever **eliminated around 20–25% of SKUs in specific categories while maintaining over 95% of total sales revenue**. The rationalization also reduced logistics costs by 15% and improved production efficiency without sacrificing market coverage.

Case 2: Coca-Cola – Managing Product Width and Depth

Problem: Coca-Cola's global portfolio includes hundreds of beverage brands and flavor variations. However, an overly broad product mix caused brand overlap and internal competition, especially in markets saturated with multiple soft drink flavors and pack sizes. The company needed to simplify its portfolio while protecting high-revenue products and maintaining consumer choice.

Analytical Approach: Coca-Cola applied product mix and profitability clustering, evaluating SKUs based on attributes such as sales volume, profit contribution, and regional preference. The analysis considered product width (number of product lines) and depth (number of variants per line). K-means clustering identified clusters of underperforming SKUs with low profit margins and overlapping target audiences.

Solution & Outcome: The company discontinued low-margin or low-demand SKUs and consolidated certain flavor variants, achieving a **12% reduction in product lines and 10% reduction in depth. Despite the cuts, Coca-Cola retained over 90% of sales revenue**.

Case 3: Procter & Gamble (P&G) – Data-Driven Product Line Simplification

Problem: Procter & Gamble (P&G), managing brands such as Tide and Pantene, faced challenges in balancing innovation with operational complexity. Over time, product variants increased rapidly due to local marketing strategies, resulting in overlapping SKUs and declining supply chain efficiency. Managers struggled to decide which variants to retain, reformulate, or discontinue.

Analytical Approach: P&G implemented a hierarchical clustering approach using product-level KPIs—revenue, profit margin, and seasonality index. The analysis revealed segments of SKUs that had either strong customer loyalty or seasonal relevance, versus those contributing little incremental revenue.

Solution & Outcome: By integrating data mining results into its product lifecycle management (PLM) system, P&G eliminated nonessential SKUs and focused on high-performing, regionally relevant variants. **The initiative reduced SKU count by 8% while maintaining 97% of total revenue**

Reference:

https://www.nasdaq.com/articles/findings-unilever-illustrate-why-sku-rationalization-
https://thesupplychainlink.com/coca-cola-supply-chain-and-case-study/ https:
//ivypanda.com/essays/procter-and-gamble-improving-customer-value-through-process-re

1.1.3 Data Source Analysis

The primary data source is the **CompanyX transactional database**, which follows a normalized schema similar to AdventureWorks. Key schemas and tables are summarized in Table 1.1.

These tables form the foundation of the **data warehouse design** through a staging process. Data from these operational sources will be extracted, transformed, and loaded (ETL) into dimensional structures such as:

- **DimProduct**: Contains detailed, transactional sales data (revenue, profit, quantity, etc.) at the grain of Product, Market, and Date. This is your primary source for performance calculations.
- **FactProductMix**: Contains detailed, transactional sales data (revenue, profit, quantity, etc.) at the grain of Product, Market, and Date. This is your primary source for performance calculations.
- **FactMarketAnalytics**: Contains aggregated, market-level performance measures (SUM(TotalRevenue), SUM(TotalProfit), SUM(TotalQuantity)), calculated by summarizing FactProductMix (or the original source). This table was used as the input for your Market K-Means clustering.
- **DimDate**:A standard calendar dimension containing date attributes (Year, Month, Quarter, Day) for each DateKey.
- **DimMarket**: Contains descriptive attributes (like name, territory, region) for each unique MarketKey

1.2 Problem Statement

The objective of this project is to identify an optimal set of SKUs for each market cluster that retains at least 93 percent of total revenue, 90 percent of overall profit, and 88 percent of sales quantity, while achieving reductions of approximately 18 to 22 percent in product total SKUs, 12 to 17 percent in number of product lines, and 8 to 12 percent in variants per product line with prioritizing the removal of products that negatively impact operational efficiency by targeting products with a high SeasonalityIndex and a low ConsistencyIndex.

Table 1.1: Summary of Data Sources in CompanyX Database

Source Schema	**Key Tables**	**Description / Role**
Production	`Product`, `ProductCategory`, `ProductSubcategory`	Contains product details, categories, and attributes such as color, size, and standard cost.
Sales	`Store`, `SalesTerritory` ,`SalesOrderDetail` ,`SalesOrderHeader`	Stores transactional data, including order quantity, unit price, and revenue per store or customer.

Chapter 2

Methodology

2.1 Grain define

-**FactProductMix** grain → one row = one product in one market in one month

-Each record in the FactProductMix table represents a single product (SKU) sold within a specific market during a specific month.Metrics such as **TotalRevenue, TotalProfit, TotalQuantity, SKUCount, VariantCount, CategoryCount, ConsistencyIndex, and SeasonalityIndex** are aggregated and stored at this level of detail.

-**FactMarketAnalytics** grain → one row = one market in one market cluster

-**DimDate** grain → month-level (e.g., 2025-10-01 as the representative date for October 2025)

2.2 Data Warehouse Design

- Fact table: FactProductMix,FactMarketAnalytics, DimMarketClusterABC
- Dimension table: DimProduct, DimMarket
- Analysis/Derived table: DimSeasonality, DimMarketCluster

2.3 ETL Pipeline

2.3.1 Overview

This report describes the SSIS (SQL Server Integration Services) package designed to load and maintain the `ProductMixDW` data warehouse. The pipeline extracts data primarily from the `CompanyX` source database and populates various dimension and fact tables within the data warehouse. The pipeline is designed to run on a **monthly schedule**, loading `FactProductMix` incrementally based on the last load date.

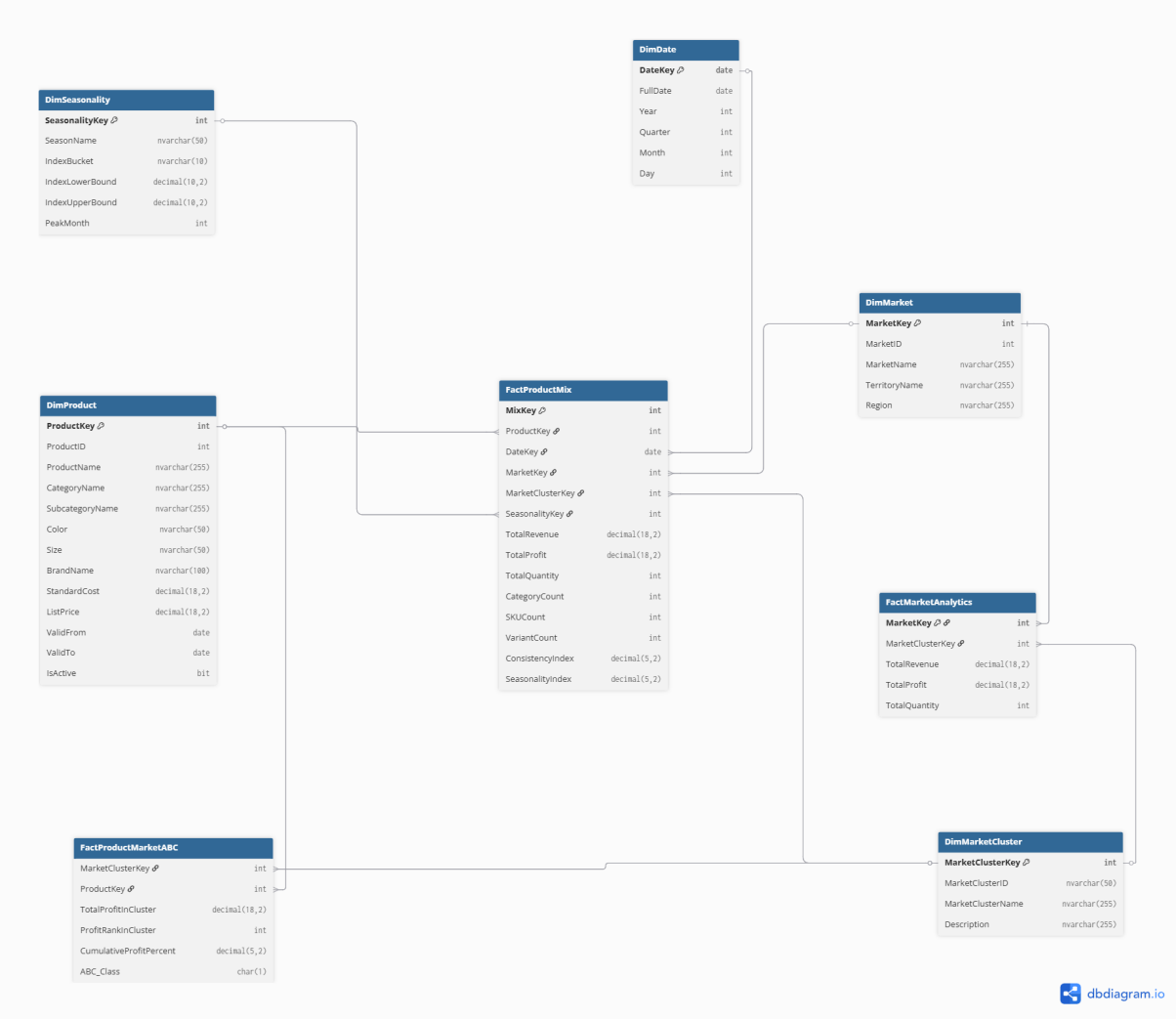

Figure 2.1: Star Schema

2.3.2 Pipeline Structure & Workflow

The main ETL process is organized within a single SSIS package (`Package.dtsx`) using Sequence Containers to group logical steps:

Phase 1: Load Dimensions

- **Load Dimensions:** This phase executes processes to populate or update dimension tables. Parallel execution within this container is possible and recommended.
  - `Load DimProduct (SCD2)`: This Data Flow loads product information using Slowly Changing Dimension Type 2 logic.

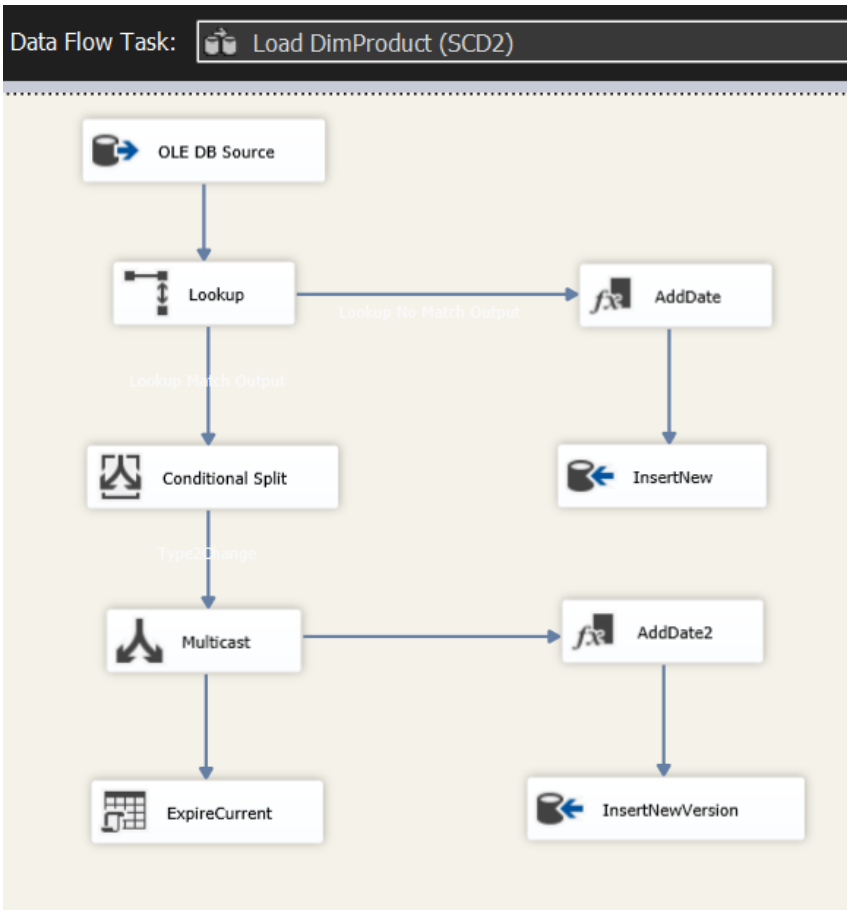

Figure 2.2: Data Flow: Load DimProduct (SCD2)

  * Reads source product data and calculates hash value from product attributes.
  * Checks if the product exists in `DimProduct` and retrieves the current active record's `RowHash` (`Lookup`).
  * If the product is new (`Lookup No Match Output`), adds current date (`AddDate`) and inserts the record (`InsertNew`).
  * If the product exists (`Lookup Match Output`), checks if the source `RowHash` differs from the dimension's `RowHash` (`Conditional Split`).
  * If the hash differs (`Type2Change` output), uses `Multicast` to:

- Expire the current record in `DimProduct` (likely updates `IsActive=0` and `ValidTo` via `ExpireCurrent`).
- Add current date (`AddDate2`) and insert the new version of the product record (`InsertNewVersion`).

– `Load DimMarket`: This Data Flow loads market information, likely using Type 1 (overwrite) or Insert/Update logic.

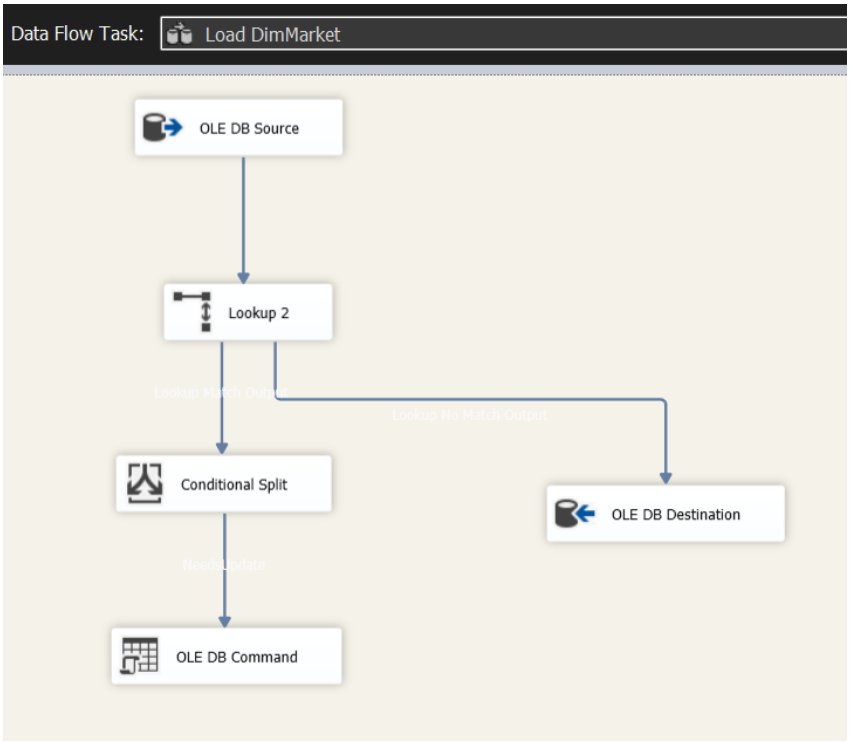

Figure 2.3: Data Flow: Load DimMarket

* Reads source market data.
* Checks if the market exists in `DimMarket` (`Lookup 2`).
* If the market is new (`Lookup No Match Output`), inserts the record to DWH.
* If the market exists (`Lookup Match Output`), checks if any attributes need updating (`Conditional Split`).
* If updates are needed (`Need Update` output), updates the existing record in `DimMarket`.

– `Init DimDate`: This Execute SQL populates the date dimension table.

– `Init DimSeasonality`: This Execute SQL populates the seasonality dimension table with predefined buckets (Low, Medium, High).

– `Init DimMarketCluster`: This process populates the market cluster dimension table.

Phase 2: Load Cluster Information

- **Load Cluster Info:** This phase updates market cluster assignments.

- `Clear Cluster Map`: Clears previous market cluster mapping data from a staging area.
- `Load Cluster Map to Staging`: Loads a CSV file, containing `MarketKey` to `MarketClusterKey` mappings, into a staging table. This file is generated after data mining processes, so this step ensures any new or updated mappings are incorporated.

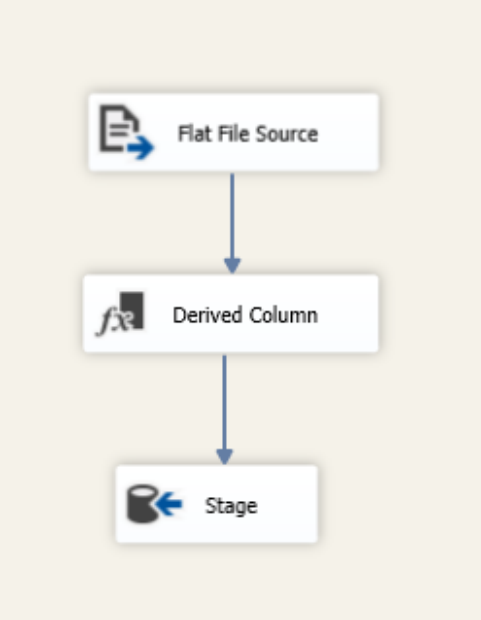

Figure 2.4: Data Flow: Load Cluster Map to Staging

- `Update MarketClusterKey`: Updates target tables (e.g., `DimMarket`, `FactProductMix`) using the staging table.

Phase 3: Load Fact Tables

- **Load FactTables:** This phase handles the loading and post-processing of fact tables. The branches for the different fact tables could potentially run in parallel.
  - **FactProductMix Branch (Incremental):**
    - `Incremental Loading`: This Execute SQL retrieves the high-watermark using `SELECT ISNULL(MAX(DateKey), CAST('19000101' AS DATE)) FROM dbo.FactProductMix` and stores it in an SSIS variable.
    - `Store Cache`: Load dimension data (ProductID, MonthStart, ProductKey) into a Cache Transform/Connection Manager for faster lookups during fact processing.
    - `Load FactProductMix Monthly`: Extracts source data from `CompanyX` where order date , aggregates data, performs lookups (using the cache and other lookups), and inserts into `dbo.FactProductMix`.
    - `Backfill Mix Counts`: Calculate and update `CategoryCount` and `SKUCount` in `FactProductMix` for the newly loaded data.
    - `Update SeasonalityKey`: This Execute SQL calculates `SeasonalityIndex` and updates `SeasonalityKey` in `FactProductMix`.

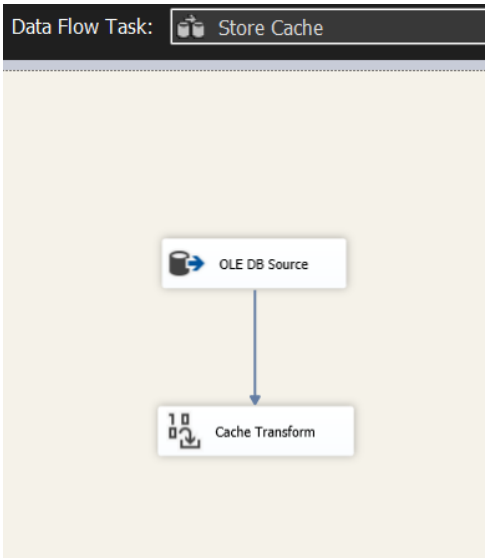

Figure 2.5: Data Flow: Store Cache

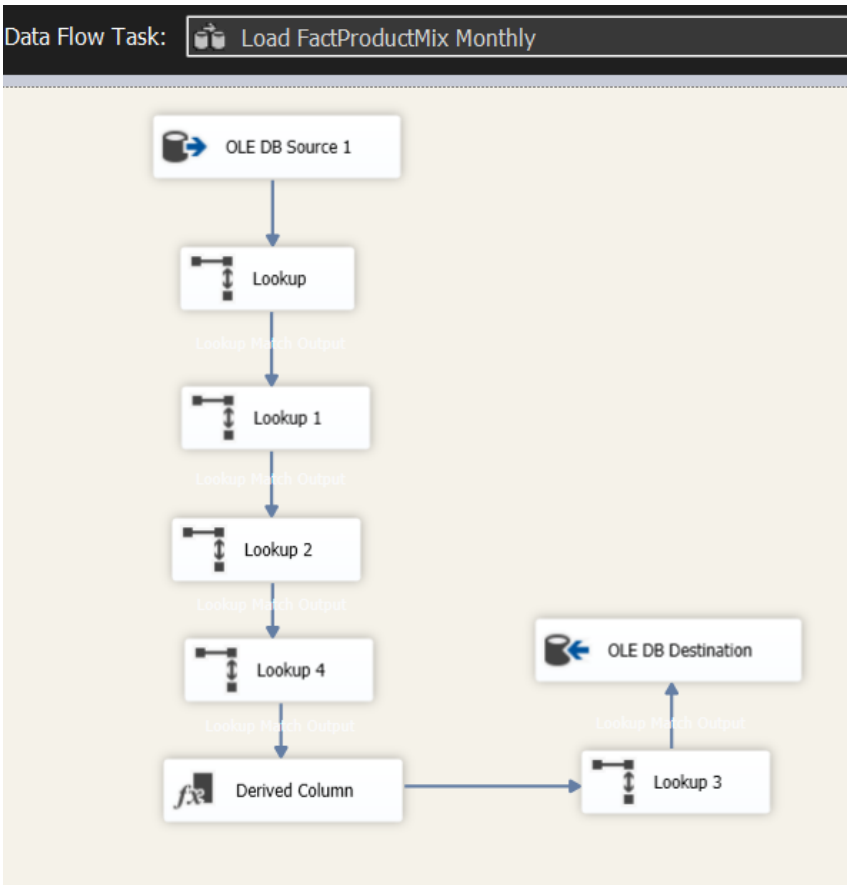

Figure 2.6: Data Flow: Load FactProductMix Monthly

– **FactMarketAnalytics Branch (Full Refresh):**
  * `Truncate FactMarketAnalytics`: Clears the table.
  * `Load FactMarketAnalytics`: Recalculates total aggregates per market from source data and loads the table.

- **FactProductMarketABC Branch (Full Refresh):**
  - `Truncate FactProductMarketABC`: Clears the table.
  - `Load FactProductMarketABC`: Performs ABC analysis based on profit, assigns class, and loads the table.

2.4 Scheduling (SQL Server Agent)

The ETL process is automated using a SQL Server Agent Job named `Load ProductMix DWH Monthly`.

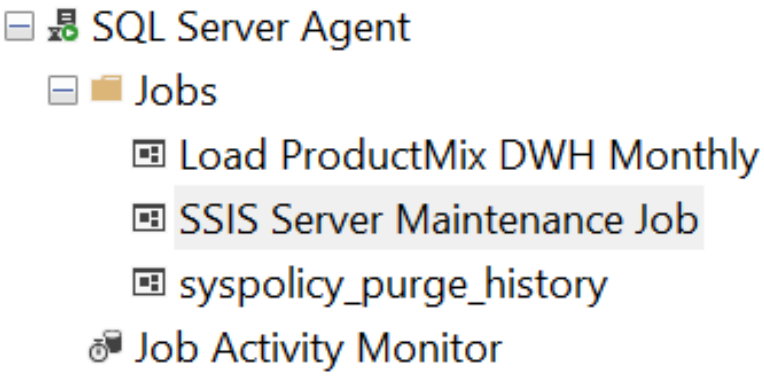

Figure 2.7: SQL Server Agent Job

- **Package Execution:** The job step `Run_SSIS_Load_FactProductMix` executes the SSIS package `Monthly_Load.dtsx` located on the **File system**

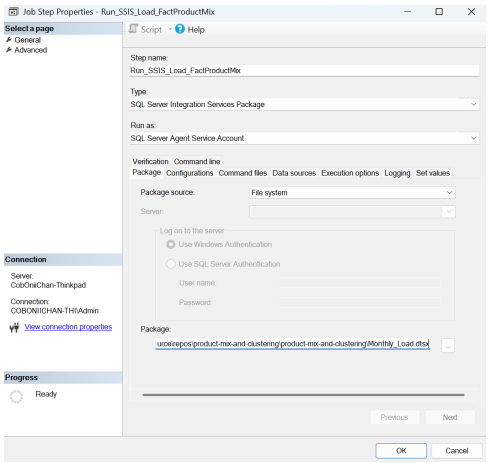

Figure 2.8: Job Step Properties

- **Schedule:** The job uses a schedule named `Run on monthly`.
  - **Frequency:** Occurs on **Day 1** of every 1 month.
  - **Time:** Occurs once at **12:00:00 AM**.
  - **Duration:** The schedule is set to start on **November 1, 2025**, with **no end date**.

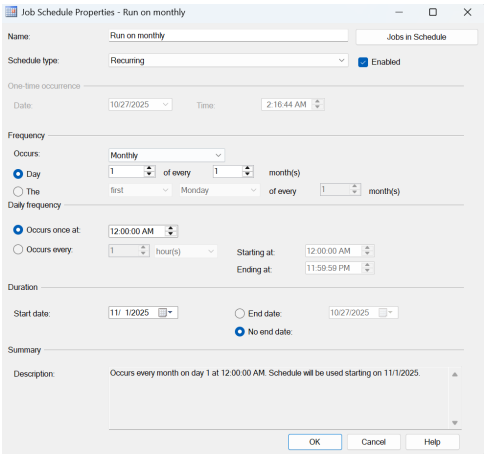

Figure 2.9: Job Schedule Properties

- **Note:** Using the `SSIS Catalog` as the package source is generally recommended over the `File system` for better management and security, however, the current configuration uses the File system due to encountered bugs preventing deployment to the SSIS Catalog.

2.5 Use-case Analysis

Scenario: A manager responsible for markets within the "Problem Market (Large)" cluster (Market Cluster 5) needs to implement the company's SKU reduction strategy: retain 90% Profit, 93% Revenue, 88% Quantity, while achieving 18 to 22% SKU reduction, 8 to 12% Variant reduction, and 12 to 17% Product Line reduction.

How the Manager Uses the Dashboard:

1.Selects Market Context: The manager opens the Power BI dashboard and selects "Problem Market (Large)" from the Market Cluster slicer. All KPIs instantly update to show the 100% baseline for this specific cluster, and the ABC Analysis Table dynamically displays the A, B, and C products ranked by profit contribution specifically for Cluster 5.

2.Applies Primary Cuts (ABC Classification): Following the ABC methodology, the manager uses the ABC Class slicer to un-check Class 'C' products, which represent the bottom 5% of profit contribution.

3.Monitors Impact on KPIs: The manager immediately observes all six KPI Gauges/Cards:

4.Refines Cuts Iteratively (Targeting 'B' and Risk): Since cutting only 'C' products is likely insufficient to meet the 18 to 22% SKU reduction, 8 to 12% Variant reduction, and 12 to 17% Product Line reduction., the manager proceeds iteratively:

5.They might un-check Class 'B' products in the ABC slicer. They re-check the gauges. If all retention goals (Profit, Revenue, Quantity) are still met, this might be the solution.

6.However, if cutting all 'B' products causes Revenue or Quantity to drop below their targets, the manager rechecks Class 'B'. They then use the SeasonalityIndex slider (lowering the maximum) and ConsistencyIndex slider (raising the minimum) to make more targeted cuts within the 'B' and 'C' classes, removing the riskiest and least efficient products first.

They continuously adjust these sliders, watching all six KPIs, until they find a balance satisfying all targets simultaneously.

Generates Actionable List: Once a satisfactory balance is achieved across all six goals, the manager reviews the filtered ABC Analysis Table (or a separate "SKUs to Keep" table). This list now represents the optimized product assortment for the "Problem Market (Large)" cluster, derived using the ABC framework and refined by supply chain risk factors. They can export this list for implementation.

Chapter 3

Results

3.1 Cluster Analysis

3.1.1 Market Clusters

Markets were segmented using the K-Means clustering algorithm based on their aggregate financial performance. The features used for this analysis were: SUM(TotalRevenue), SUM(TotalProfit), SUM(TotalQuantity) for each `MarketKey`, sourced from the `FactMarketAnalytics` table.

This clustering approach groups markets according to their overall economic size, profitability, and sales volume. It enables the identification of distinct strategic market segments, such as:

- **High-Volume Giants**
- **Profit Stars**
- **Problem Markets**

MarketClusterKey	MarketClusterID	MarketClusterName	Description
0	M001	High-Volume Giant	Highest revenue and sales volume, but with moderate profit margins.
1	M002	Small & Efficient	Lowest revenue and volume, but maintains a solid, efficient profit margin.
2	M003	Large & Healthy	A large, balanced market with high revenue, high volume, and strong profit.
3	M004	Profit Star	The most profitable market, driven by high-margin sales, not high volume.
4	M005	Problem Market (Small)	A smaller market that is actively losing money (negative profit).
5	M006	Problem Market (Large)	A large, high-volume market that is unprofitable (breaks even or loses money).

Figure 3.1: Market Clusters identified using K-Means analysis.

3.1.2 ABC-Ranking Analysis

After identifying *where* to focus (i.e., the market clusters), the next step is determining *what* to optimize or reduce.

The features for this analysis were sourced from the `FactProductMix` table and included: TotalRevenue, TotalProfit, SKUCount, VariantCount, CategoryCount

This cluster-specific ranking provides managers with a clear, data-driven framework to make strategic SKU rationalization decisions:

Identify Core Products (Class A): The table immediately highlights the 'A' products – the vital few SKUs that generate the top 80 percent of the selected cluster's total profit. These are typically protected during reduction efforts to ensure the 90 percent profit retention goal is met.

Target Low Contributors (Class C): The 'C' products, contributing only the bottom 5 percent of the cluster's profit, are clearly identified as the primary candidates for removal. Managers can see exactly how many SKUs fall into this category for their cluster, helping them estimate the impact of cutting these items towards the 20 percent SKU reduction target.

Evaluate the Middle Ground (Class B): The 'B' products (contributing the next 15 percent of profit) can be reviewed individually. If further SKU reduction is needed after cutting 'C' products, the manager can use this table – potentially alongside SeasonalityIndex and ConsistencyIndex data – to strategically select the lowest-ranked or highest-risk 'B' products for removal, while carefully monitoring the impact on the overall profit and revenue retention goals displayed in the dashboard's KPI gauges.

ClusterKey	ClusterID	ClusterName	Description
0	C001	High-Complexity / Problem	Loses money on average and has very high operational complexity (high SKU and variant counts).
1	C002	Low-Value / Simple	Standard low-value, low-profit products with relatively simple variant/SKU profiles.
2	C003	Star Products	The best products. Highest revenue and profit, combined with the lowest operational complexity.
3	C004	High-Variant / Problem	Loses money. The high variant count (11.7) makes these products especially complex and costly.
4	C005	Commodity / High-Volume	High-volume, high-quantity "commodity" items. They sell a lot but bring in very little revenue or profit.
5	C006	High-Complexity / Low-Value	Low-profit products that have a high operational complexity (high SKU count) without a high return.
6	C007	High-Revenue / High-Complexity	Very successful products (high revenue/profit) that are operationally complex. Potential for optimization.
7	C008	Niche / Simple-Variant	Very low revenue. Simple (1 variant) but part of a very wide (high SKU) product family.
8	C009	High-Revenue / Problem (Loss Leader)	These products sell well (high revenue) but lose a significant amount of money. Likely priced too low.

Figure 3.2: Sub-Product Clusters supporting SKU reduction analysis.

3.1.3 Seasonality Index and Consistency index

We will use SeasonalityIndex and ConsistencyIndex as a "supply chain risk score" to prioritize which SKUs to cut within the clusters we've already identified.

This gives us a clear action plan: We first identify a "Rank B Cluster" (or Rank C Cluster in some case) . We then rank all products inside that cluster by their individual risk scores. The products with the highest seasonality and lowest consistency will be the first ones we cut, as they are not only unprofitable but also the most expensive and riskiest to hold in inventory.

3.2 BI Dashboard

Dashboard Overview:

The dashboard, shown in the Figure, is designed for intuitive interaction and clear visualization of key performance indicators (KPIs) against strategic goals. It comprises three main sections:

Filter Pane (Left): This section contains the primary controls for the simulation:

Figure 3.3: Dashboard BI

Market Cluster Slicer: Allows the manager to select their specific market cluster(s) (e.g., "Profit Star," "Problem Market - Small"), setting the context for the analysis.

ABC Class Slicer: Enables broad, profit-based cuts by allowing the manager to include or exclude products classified as 'A', 'B', or 'C' based on the cluster-specific ABC analysis.

Profit Rank Slicer: Provides fine-grained control by allowing the manager to keep only products ranked above a certain profit threshold (e.g., keep the top 50 most profitable products).

Risk Sliders (Seasonality and Consistency): Allow managers to implement the supply chain risk strategy by filtering out products with high seasonality or low sales consistency.

KPI Gauges (Top Center): Six prominent gauge visuals provide immediate feedback on the impact of the selected filters against the project's targets:

Analysis Area (Bottom Center and Right): This section provides detailed views to support decision-making:

ABC Analysis Table: Displays the products relevant to the selected filters, ranked by profit within the chosen market cluster. It includes columns for ABCClass, ProfitRankInCluster, key performance metrics (TotalProfitInCluster, TotalRevenueInCluster), and risk/complexity indicators (AverageSeasonalityIndex, AverageConsistencyIndex, AverageVariantCount), allowing managers to scrutinize individual products.

Scatter Plot (Profit vs. Revenue): Visually represents the product portfolio, colored by ABCClass.

Chapter 4

Conclusion

This project successfully addressed the complex challenge of SKU rationalization by developing a data-driven methodology and an interactive decision-support tool. By employing K-Means clustering, markets were segmented into distinct strategic groups based on their aggregate performance profiles (e.g., "Profit Star," "Problem Market"). Subsequently, an ABC analysis was performed within each market cluster, ranking products based on their specific profit contribution to that segment.