Electronics & Appliance Sales Analysis

Power BI Project Documentation

Author: Michał Kruk

Email: michal.kruk.3410@gmail.com

LinkedIn: linkedin.com/in/michalkruk88/

GitHub: github.com/MichalKruk3410

Executive Summary

This project analyzes multi-year sales performance (2023–2025) for a synthetic consumer electronics & appliances retailer. The report consolidates transactions across products, stores, and sales representatives to surface actionable insights for staffing, inventory, and marketing decisions.

The report is structured into five interactive dashboards: Landing Page, Overview, Products, Stores, and Customers.

Key insights include:

- Top category by sales volume: **TV**
- Best-selling product: **PlayStation 5**
- Highest-revenue product: **LG OLED C2 55"**
- Total revenue: **PLN 46 637 479,65**
- Products sold: **25 627**
- Average order value: **PLN 3,109.17**
- Most used payment method: **BLIK**
- Peak sales month: **July**

HOW TO USE REPORT

- Use slicers on each page to filter by time, region, store, category, and customer attributes.
- Navigate via the left-hand menu buttons or page tabs to switch between dashboards.
- Use cross-highlighting on visuals and drillthrough where available to pivot from a summary to details.

REPORT OBJECTIVE

The “Electronics & Appliance Sales & Performance Report” was built to demonstrate end-to-end BI skills for recruitment and portfolio purposes. It showcases data modeling, DAX, UX design, and storytelling in Power BI, enabling business users to answer questions such as:

- Which products and categories generate the highest revenue and volume?
- Who are the top-performing stores and sales representatives (dealers)?
- What customer segments (age, gender) drive demand?
- Which payment methods dominate and how do preferences vary over time and by region?

Data & Preparation

Datasets

- Fact_Sales_Transactions (transaction-level fact table)
- Dim_Products (product attributes)
- Dim_Salesreps (dealers / sales reps)
- Dim_Stores (store attributes)
- Dim_Dates (calendar dimension created in Power Query)

Scope & Granularity: Transaction-level records spanning 2023–2025 (≈15k rows). Frequency is static for portfolio purposes.

Provenance: Synthetic, generated data for demonstration.

File Format & Quality Notes:

- CSV files, comma-separated with headers; inconsistent date formats across sources.
- Presence of duplicates and mixed typing fixed during transformation.

Key Columns (selected):

- Sales_Transactions: TransactionID, TransactionDate, StoreID, StoreCity_Canonical, CustomerID, CustomerAge, CustomerGender, ProductID, ProductCategory, ProductName, IdDealer, Dealer, BasePricePLN, Discount%, UnitPrice, Quantity, TotalAmount, PaymentMethod
- Dim_Products: ProductID, ProductCategory, ProductName, BasePricePLN
- Dim_Salesreps: IdDealer, Dealer, StoreID, Gender
- Dim_Stores: StoreID, StoreCity, Region
- Dim_Dates: Date, Year, Month, MonthName, Day, DayName, Quarter, WeekOfYear

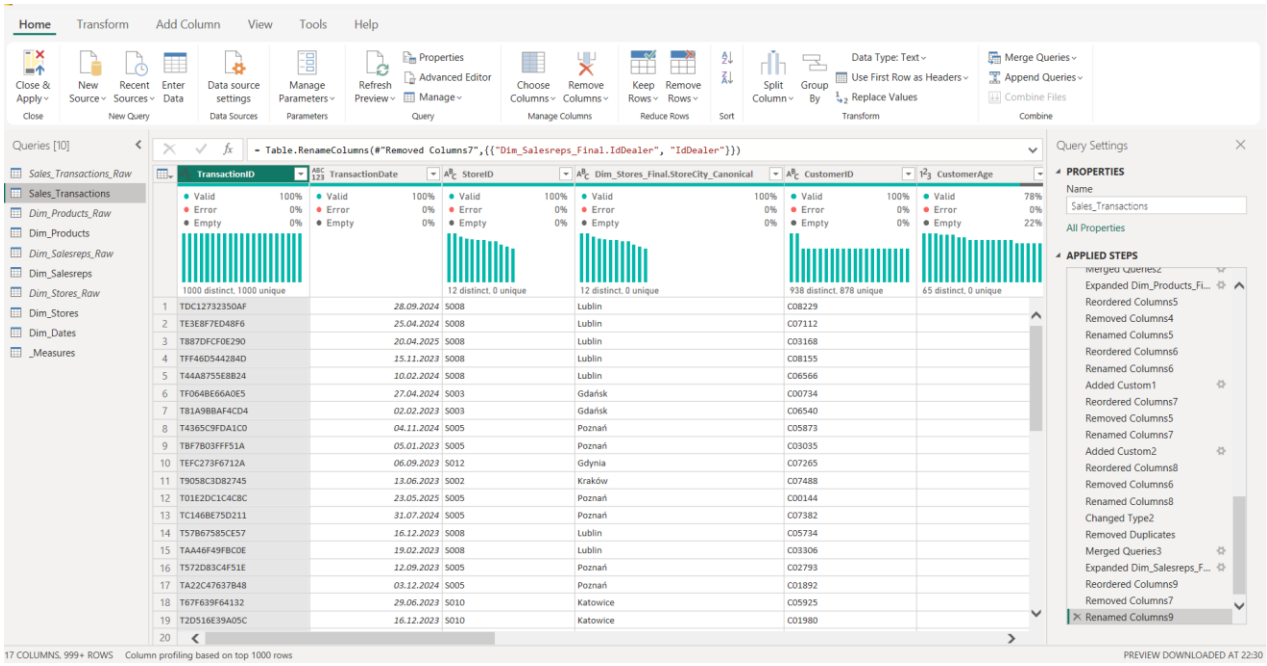
Power Query (ETL)

Two-layer query pattern used for each source: a Raw staging query (Enable Load = Off) and a Final reference query (Enable Load = On).

Main transformations included:

- Type conversions; trimming and cleaning text; handling nulls and errors.
- Removing duplicates and obvious outliers; standardizing date formats.
- Merges/joins to conform keys; column splitting/combining; conditional columns.
- Creation of calendar (Dim_Dates).

Notes: No parameters/functions were required; query folding not applicable for CSV; error handling minimal.



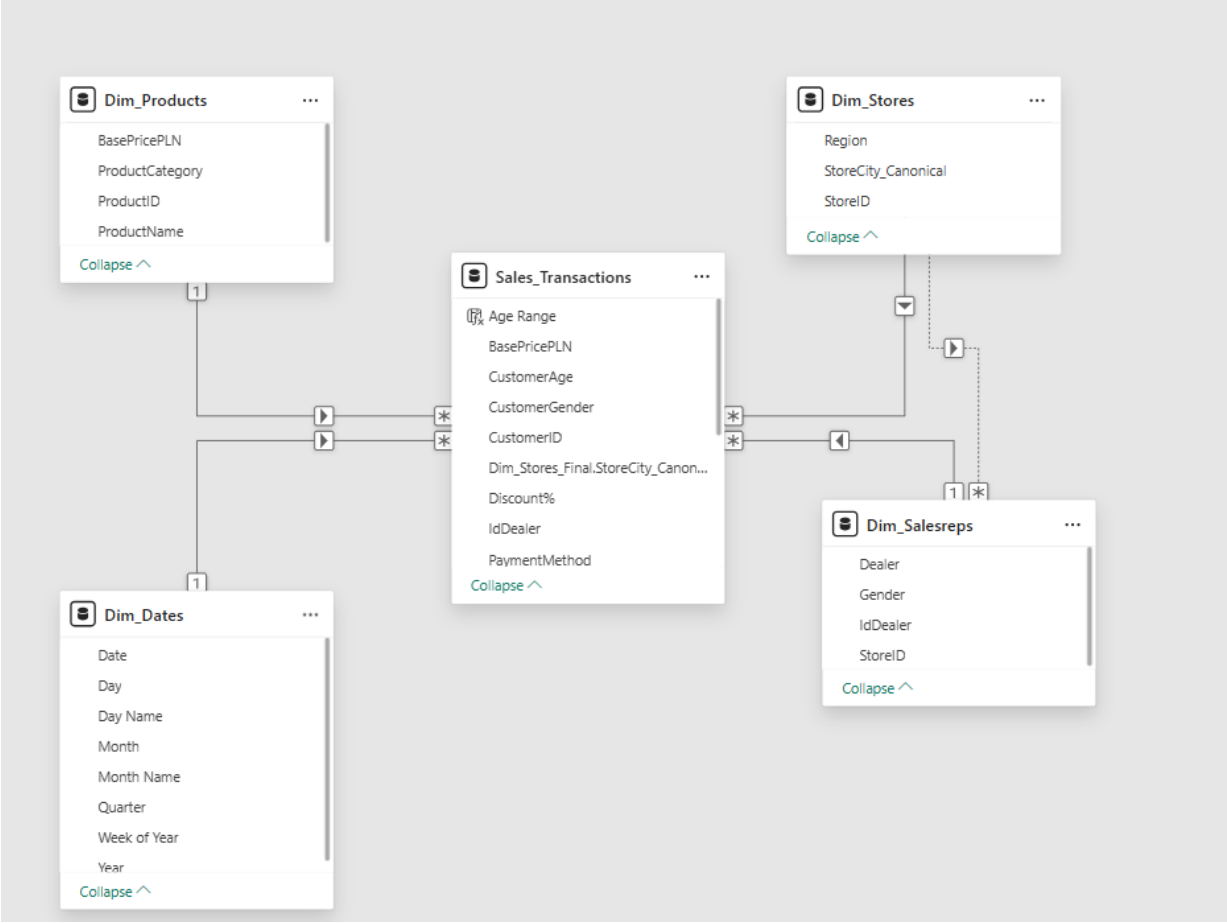
Data Model & KPIs

Model Architecture: Star schema with one fact table and four dimensions.

- Fact: Sales_Transactions
- Dimensions: Dim_Products, Dim_Salesreps, Dim_Stores, Dim_Dates
- Relationships: Simple 1-to-many from dimensions into fact (single direction).
- Date Table: Custom calendar (role: primary date for time analysis).
- RLS: Not implemented (portfolio scope).

KPIs (business definitions):

- Total Sales – sum of TotalAmount.
- Products Sold – sum of Quantity.
- Total Transactions – count of TransactionID.
- Average Order Value – Total Sales / Total Transactions.
- Top-performing Category / Product – based on sales quantity or revenue (contextual).
- Top-performing Store / Dealer – based on revenue.
- Customer KPIs – transactions by gender; average customer age.



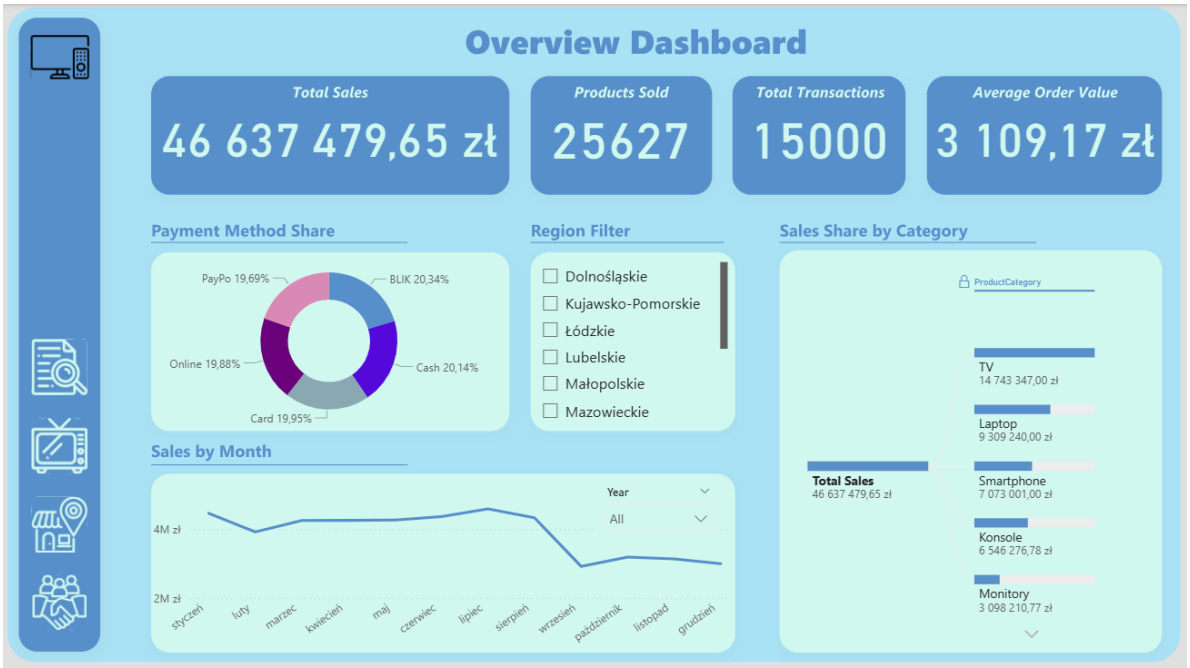
Dashboard Overview

Landing Page

Navigation entry point with report description and quick links to sections.

Overview Dashboard

Headline KPIs (Total Sales, Products Sold, Total Transactions, Average Order Value), sales share by category, monthly trend, payment method share, and region filter.

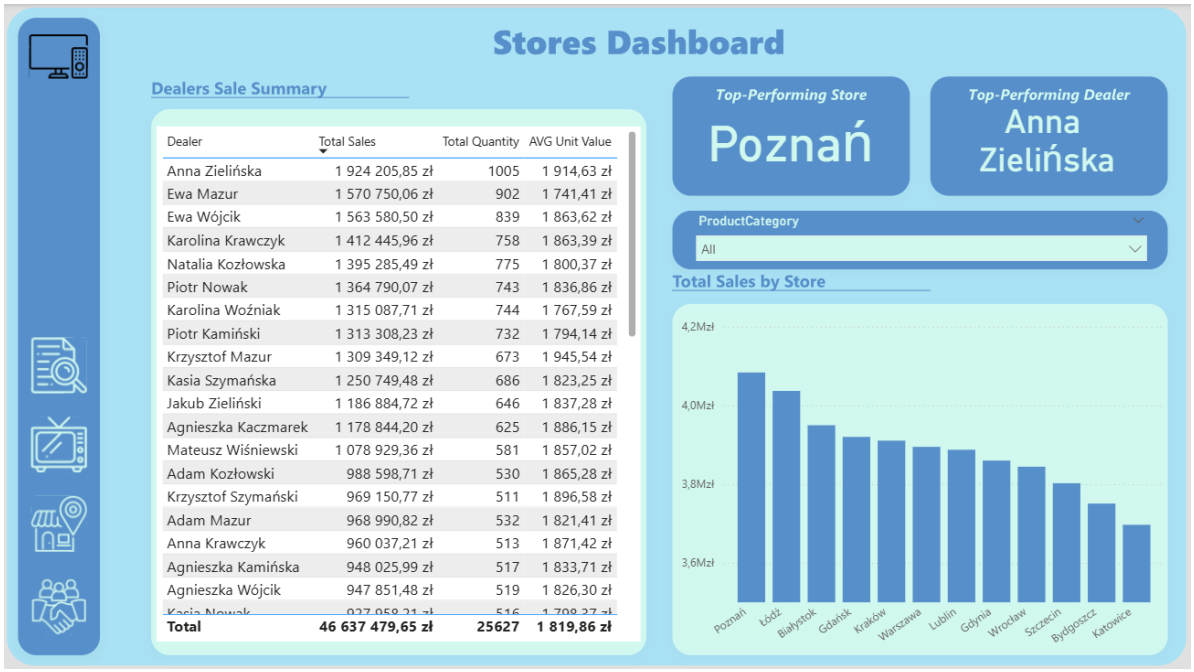


Products Dashboard

KPIs for Top-performing Category and Best-selling Product, Top 10 revenue products, Top 10 selling products, and a category & product summary table.

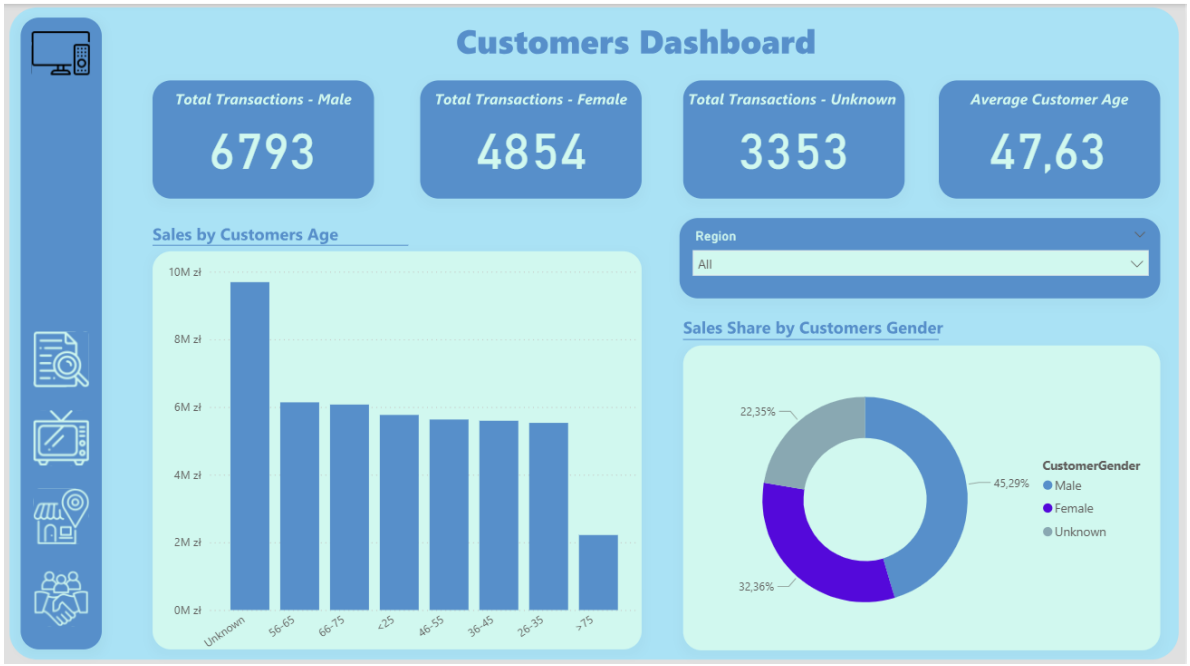
Stores Dashboard

KPIs for Top-performing Store and Dealer, total sales by store, and dealer sales summary.



Customers Dashboard

KPIs for gender-split transactions and average customer age, plus visuals for sales by customer age and sales share by gender.



Insights & Recommendations

- TV is the strongest category; ensure breadth in high-demand models and accessories.
- PlayStation 5 leads unit demand; maintain stock buffers around promotions and holidays.
- LG OLED C2 55" contributes the highest revenue; price-elasticity checks and attach-rate bundles (soundbars, extended warranties) can lift margin.
- BLIK dominates payments; feature BLIK in checkout communications and targeted ads.
- July peaks in sales; align staffing, inventory, and marketing campaigns to seasonal spikes.

Limitations & Next Steps

Limitations

- Synthetic dataset; real-world biases and seasonality may differ.
- No cost or margin data, so profitability analysis is out of scope.

Planned improvements

- Add margin/costs for profitability views and cohort LTV.
- Introduce AI-assisted forecasting and anomaly detection.
- Enrich customer attributes for better segmentation and propensity modeling.

Appendix: DAX Measures (selected)

- Total Transactions

```
Total Transactions = COUNTROWS ( Sales_Transactions )
```

- Total Sales

```
Total Sales = SUM ( Sales_Transactions[TotalAmount] )
```

- Total Quantity

```
Total Quantity = SUM ( Sales_Transactions[Quantity] )
```

- Top Selling Product

```
Top Selling Product =
CALCULATE(
    MAX(Dim_Products[ProductName]),
    TOPN(
        1,
        ADDCOLUMNS(
            VALUES(Dim_Products[ProductName]),
            "Product Sales Count", CALCULATE(COUNTROWS(Sales_Transactions))
        ),
        [Product Sales Count], DESC
    )
)
```

- Top Performing Store

```
Top Performing Store =
CALCULATE(
    MAX(Dim_Stores[StoreCity_Canonical]),
    TOPN(
        1,
        ADDCOLUMNS(
            VALUES(Dim_Stores[StoreCity_Canonical]),
            "Store Sales Count",
CALCULATE(SUM(Sales_Transactions[TotalAmount]))
        ),
        [Store Sales Count], DESC
    )
)
```

- Top Performing Dealer

```
Top Performing Dealer =
CALCULATE(
```

```
    MAX(Dim_Salesreps[Dealer]),
    TOPN(
        1,
        ADDCOLUMNS(
            VALUES(Dim_Salesreps[Dealer]),
            "Dealer Sales Count",
CALCULATE(SUM(Sales_Transactions[TotalAmount]))
        ),
        [Dealer Sales Count], DESC
    )
)
```

- Top Performing Category

```
Top Performing Category =
CALCULATE(
    MAX(Dim_Products[ProductCategory]),
    TOPN(
        1,
        ADDCOLUMNS(
            VALUES(Dim_Products[ProductCategory]),
            "Category Sales Count", CALCULATE(COUNTROWS(Sales_Transactions))
        ),
        [Category Sales Count], DESC
    )
)
```

- Sales YTD

```
Sales YTD = TOTALYTD ( [Total Sales], Dim_Dates[Date] )
```

- Sales Rep Rank

```
Sales Rep Rank = RANKX ( ALL ( Dim_Salesreps[Dealer] ), [Total Sales], , DESC
)
```

- Product Share

```
Product Share = DIVIDE ( [Total Sales], CALCULATE ( [Total Sales], ALL (
Dim_Products ) ), 0 )
```

- AVG Unit Value

```
AVG Unit Value = DIVIDE([Total Sales],[Total Quantity])
```

- AVG Order Value

```
AVG Order Value = DIVIDE(SUM(Sales_Transactions[TotalAmount]),
COUNT(Sales_Transactions[TransactionID]))
```

- AVG Customer Age

```
AVG Customer Age = AVERAGE(Sales_Transactions[CustomerAge])
```

- Average Price

```
Average Price = DIVIDE ( [Total Sales], [Total Quantity], 0 )
```