Data-Driven Discount Strategy for Eniac

Are Discounts Beneficial?

Group 2: Asli Özdemir Strollo, Azumi Sifau Muhammed, Carlos Victor Montefusco Pereira, Olaf Bulas

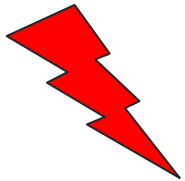
The Discount Debate

Marketing Team Lead

"Discounts drive customer acquisition, satisfaction, retention, and growth."

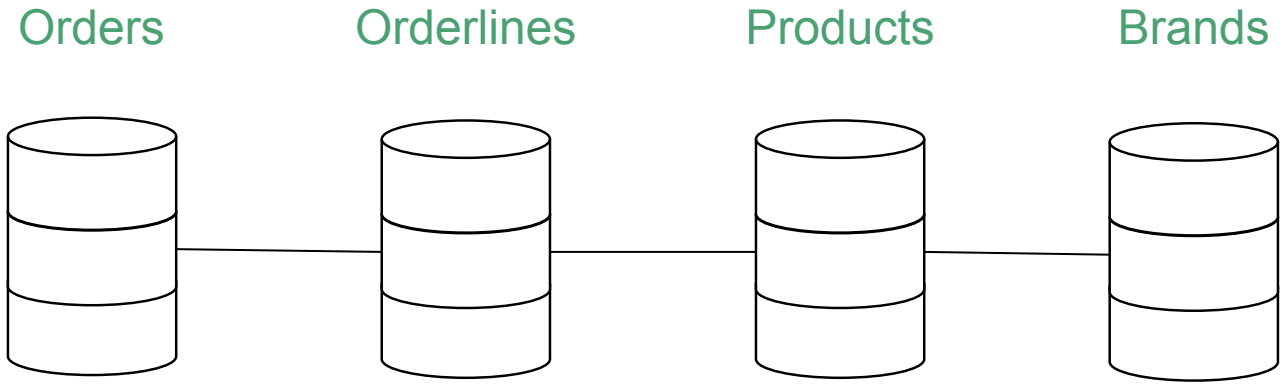
Board of Investors

"Increased orders, decreased revenue. Focus on quality over low prices."



Is there a clear winner in this debate? Let's delve into the facts.

Eniac Dataset



Data Collection Improvements

Problems with the data tables:

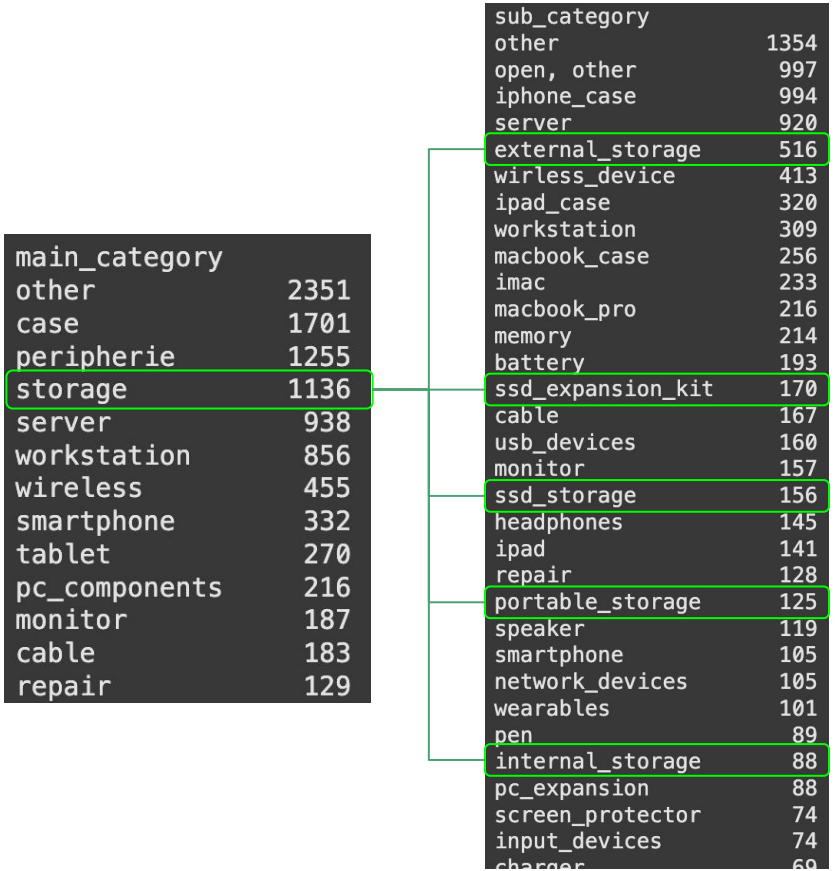
- Wrong **data types** for prices and dates
- Big amount of product **duplicates**
- **Price Data Issues** with decimal places and points
- **Data Redundancy** in all tables
- Outdated and **unnecessary data**

For more details ask our team and look into the slides in the appendix

Product Categorization

Easier understanding, simpler analysis & reporting.

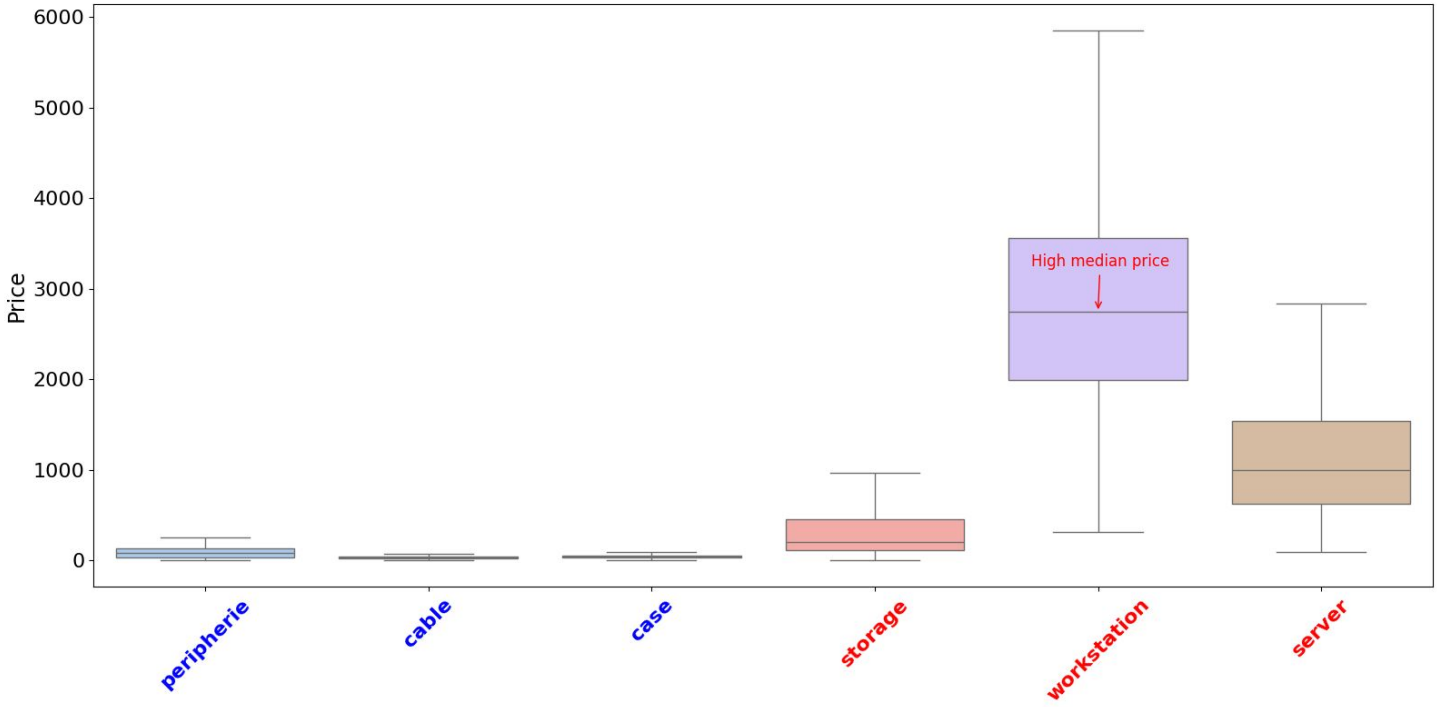
Over **76%** of products already **categorized**!



main_category	
other	2351
case	1701
peripherie	1255
storage	1136
server	938
workstation	856
wireless	455
smartphone	332
tablet	270
pc_components	216
monitor	187
cable	183
repair	129

sub_category	
other	1354
open, other	997
iphone_case	994
server	920
external_storage	516
wirless_device	413
ipad_case	320
workstation	309
macbook_case	256
imac	233
macbook_pro	216
memory	214
battery	193
ssd_expansion_kit	170
cable	167
usb_devices	160
monitor	157
ssd_storage	156
headphones	145
ipad	141
repair	128
portable_storage	125
speaker	119
smartphone	105
network_devices	105
wearables	101
pen	89
internal_storage	88
pc_expansion	88
screen_protector	74
input_devices	74
charger	69

Price Distribution Across Categories



"Categories" - >

- High price range
- Product tiers: Strategic discount

"Categories" ->

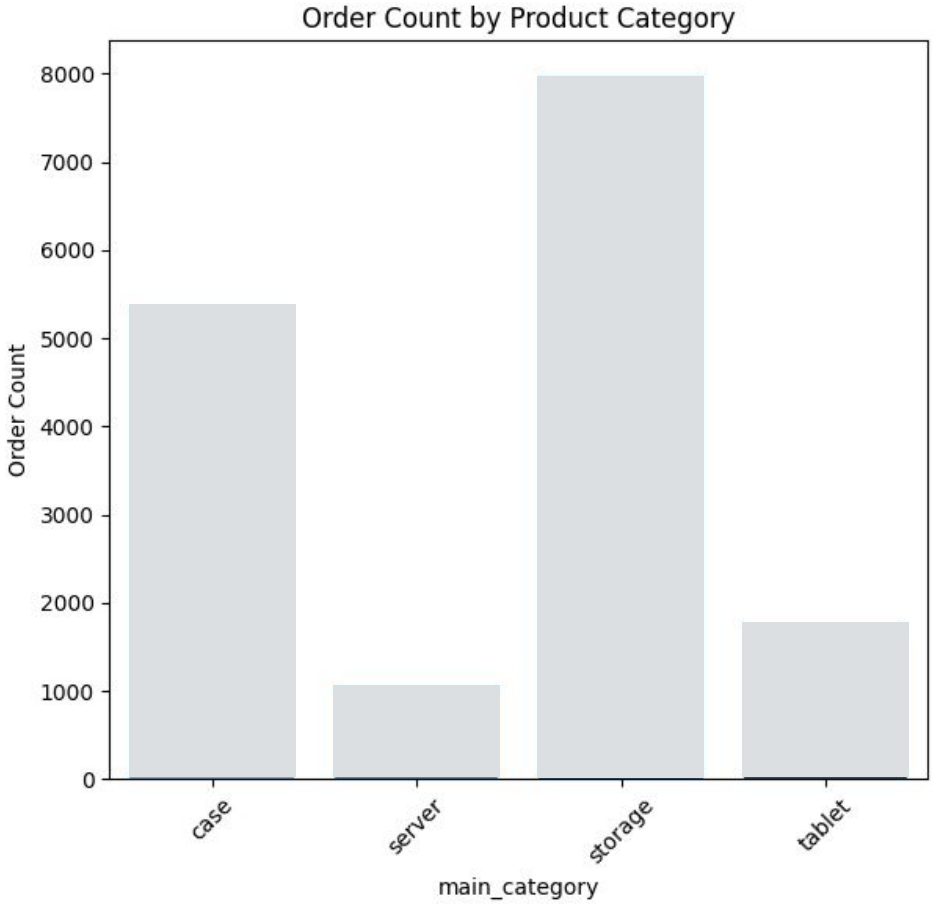
- Low price range
- Aggressive discounts can eat up all profit

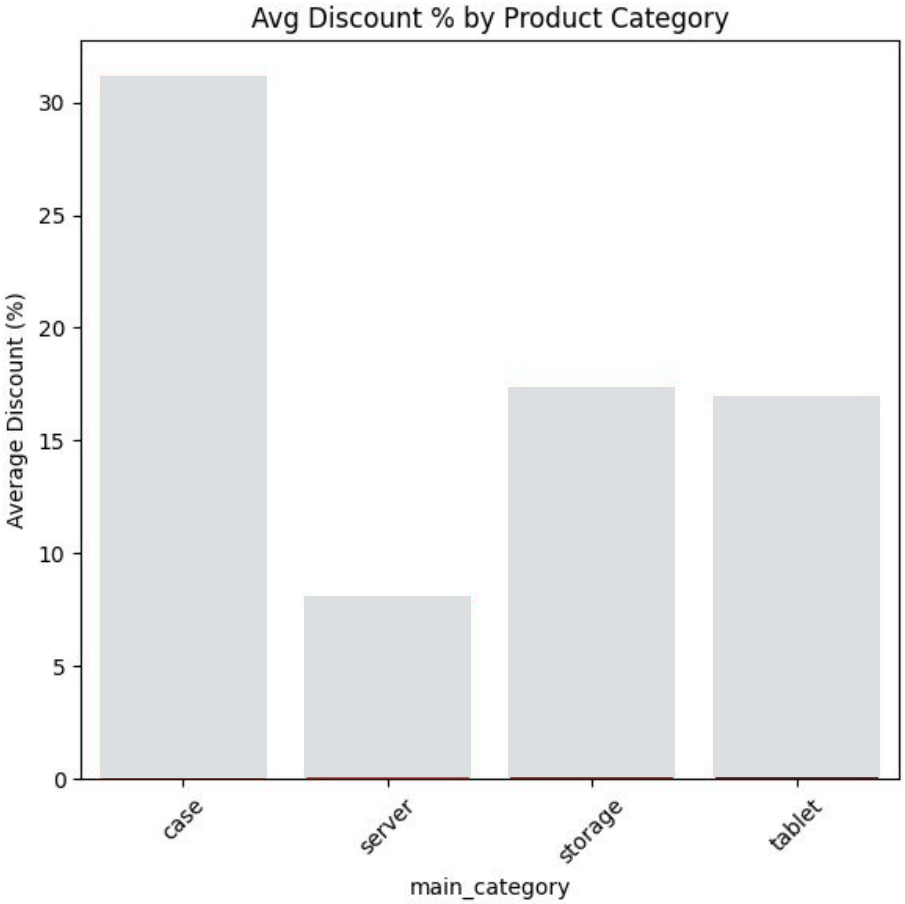
95%

of all products sold were **discounted**.

Discounting was the norm, not the exception

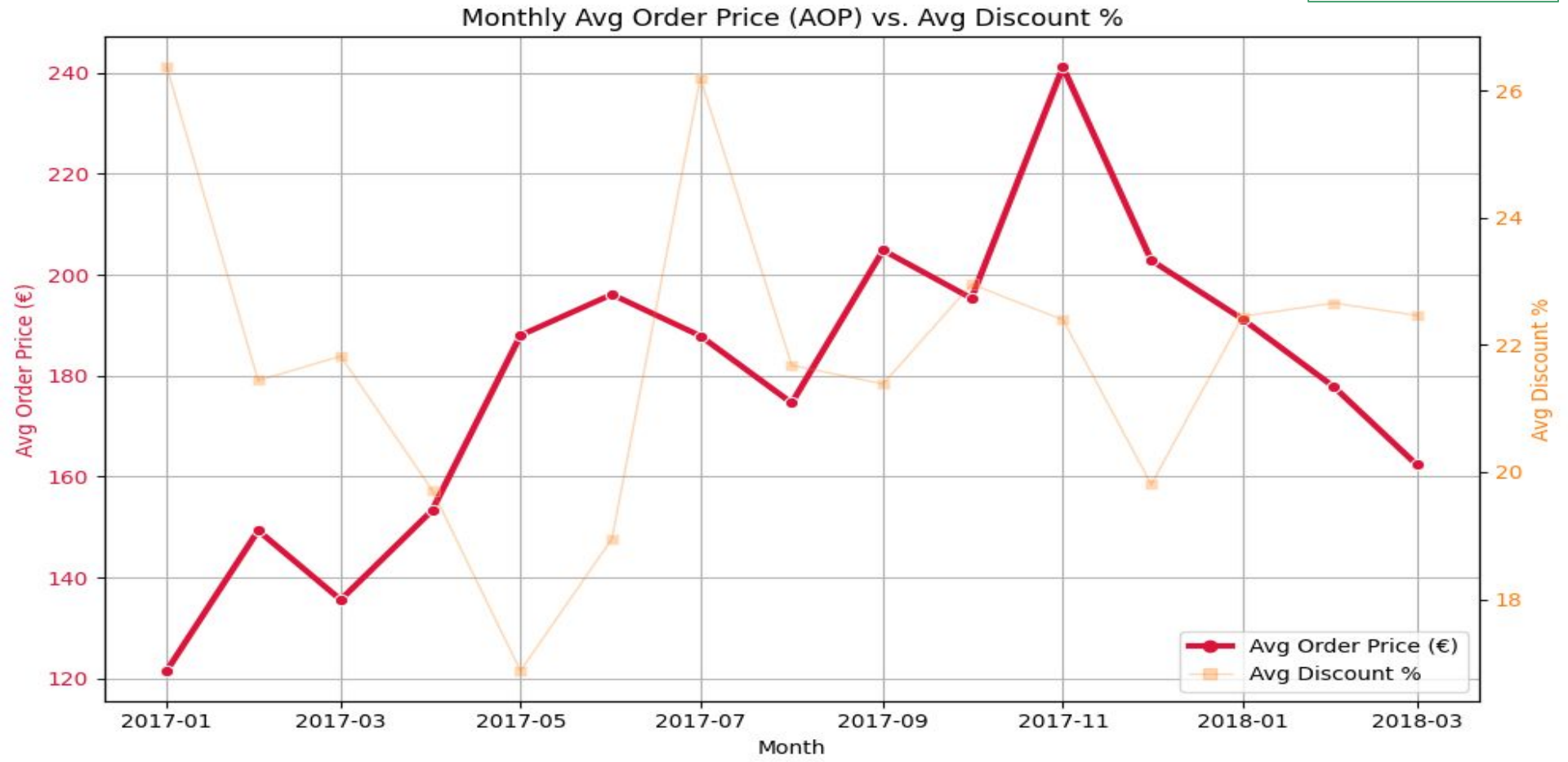
Discounts Don't Drive Order Volume



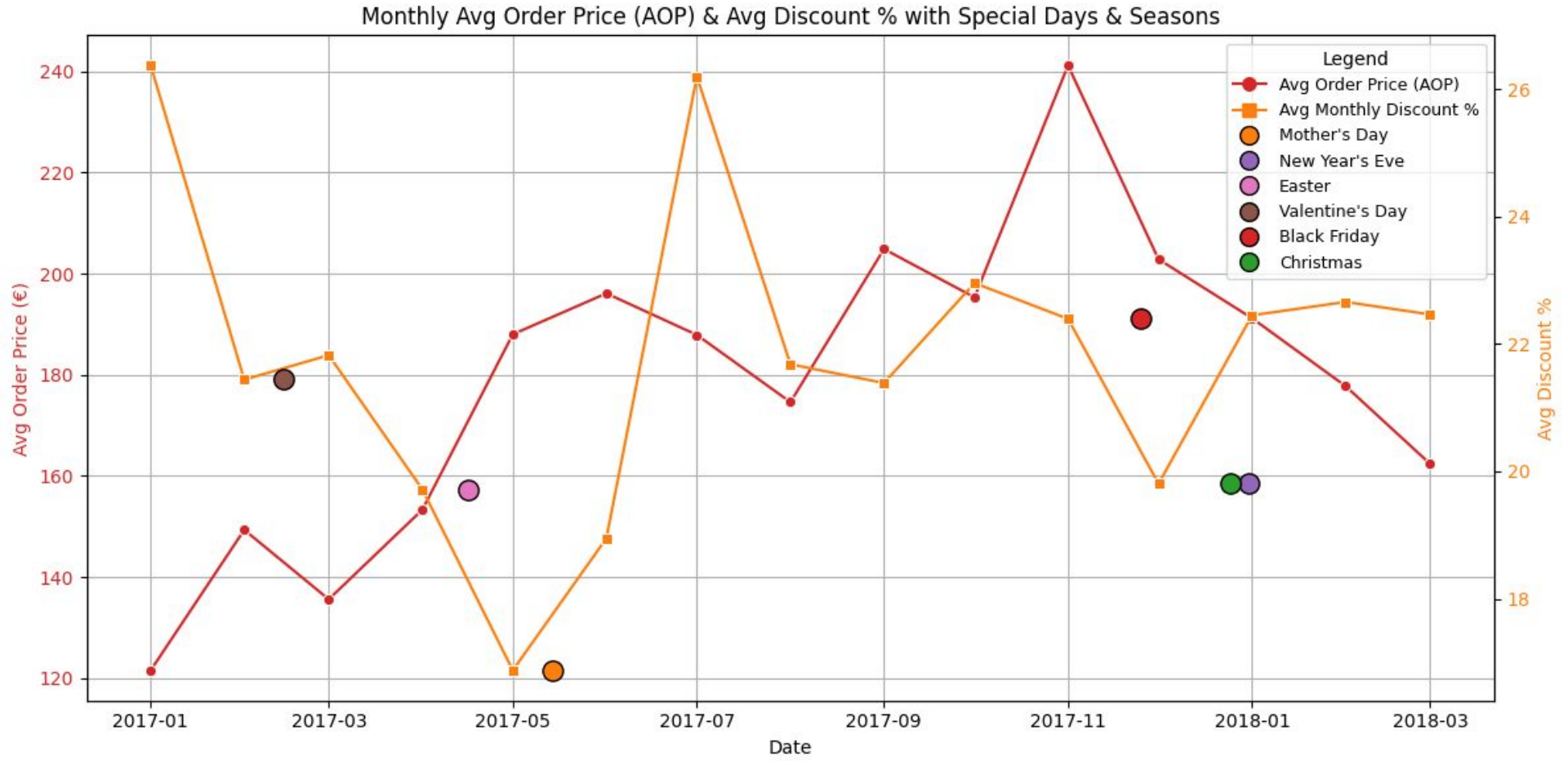


Do Discounts And Special Days Drive AOP Peaks/Drops?

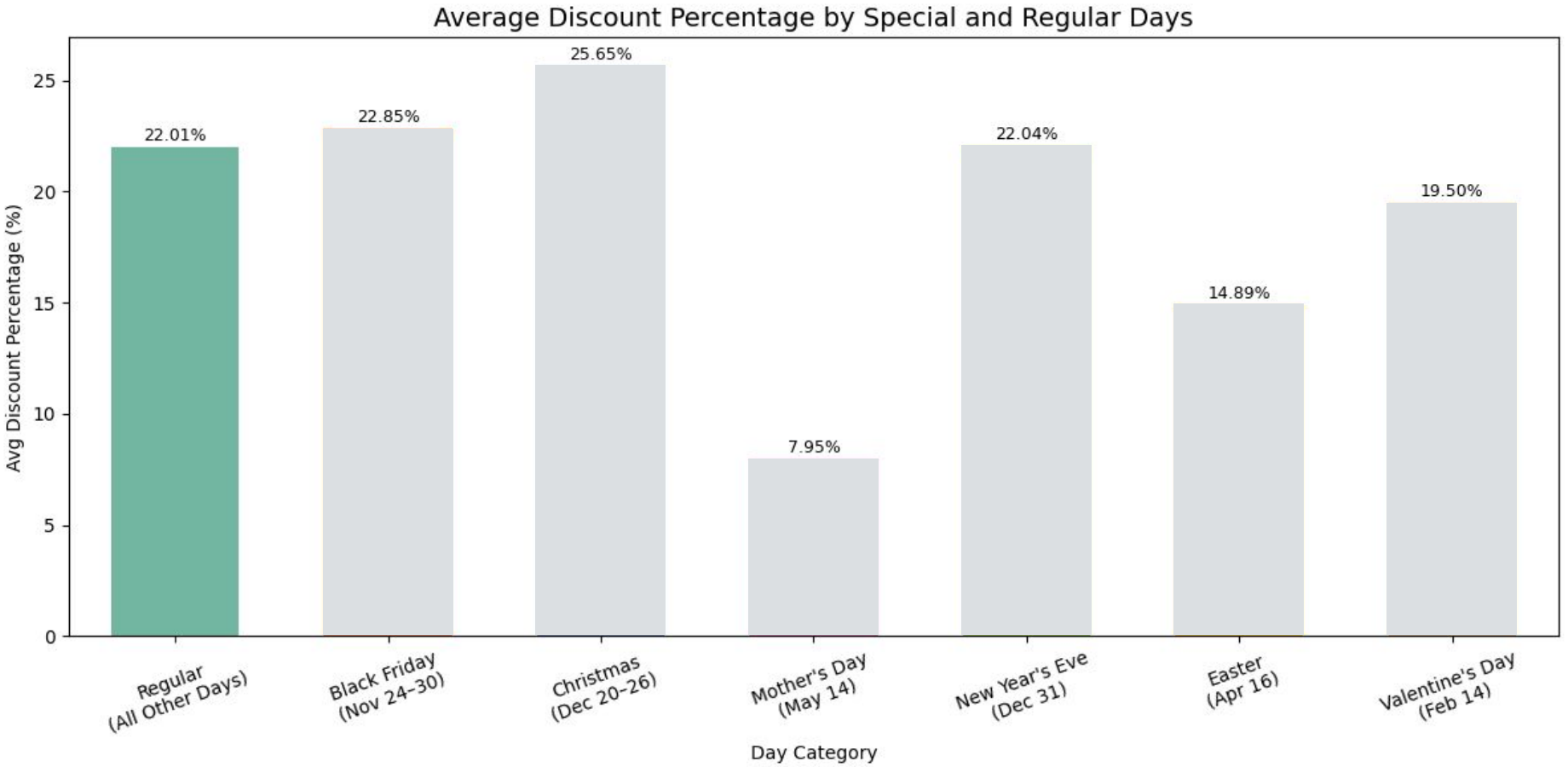
Avg Order Price = Revenue / Number of Orders



Discounts Have Little AOP Impact



Discount Behavior on Special and Regular Days

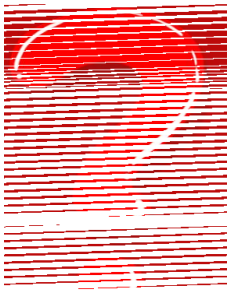


The most sold product over time: Samsung SSD

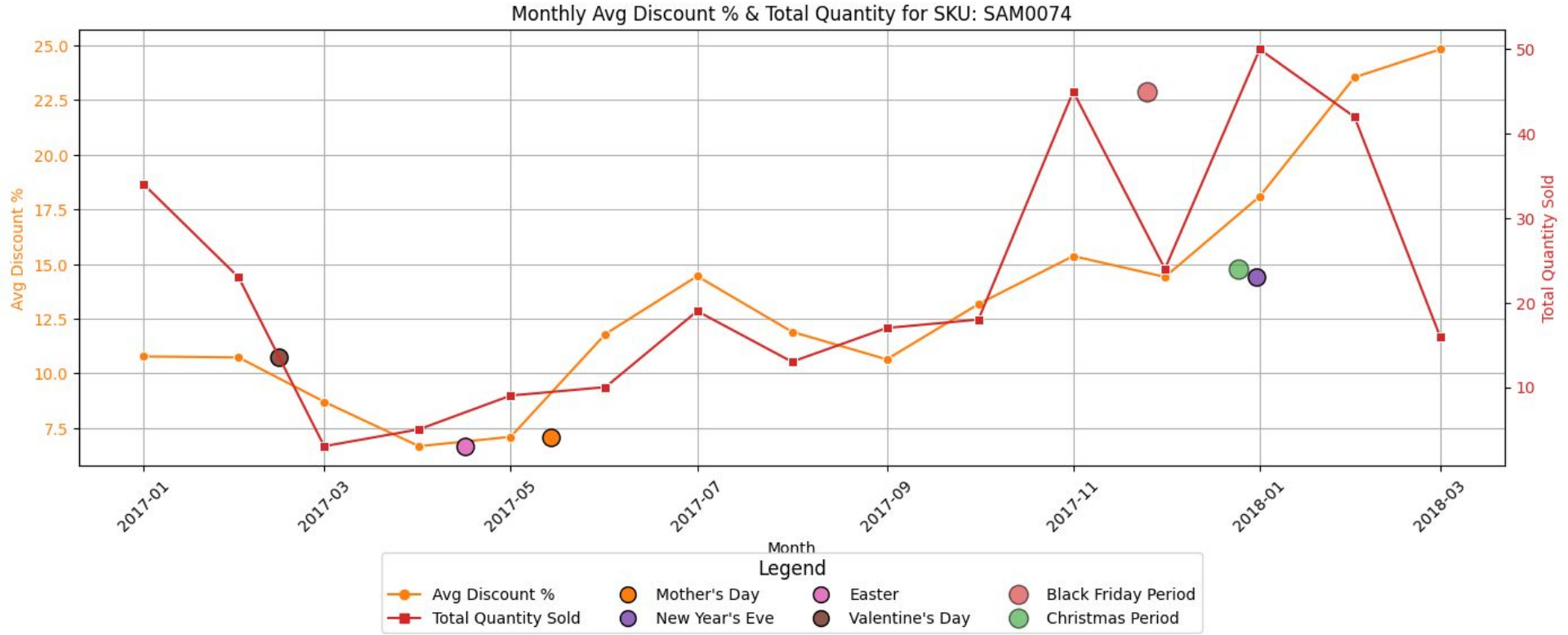
It was sold 328 times with 40 different prices over time

How effective were the discounts on the sales of this product?

	name	main_category
79	Samsung 850 EVO SSD Disk 500GB	storage



Some special days drive sales more than discounts.



Recommendation

We support the board of investors' decision to avoid frequent, aggressive discounts. Our data indicates that such discounts rarely boost sales and often lead to lower revenue. Instead, our company's strength lies in direct customer contact, which builds higher retention and growth through quality, not low prices.

Sources & Appendix

Detailed slides on data collection enhancements are up next.

Sources:

Slide 4 - Image Source
Slide 6 - Image Source

Data Collection Improvements - Data Types

- Prices = numeric
- Dates = datetime

Changes made to data types in 3/4 of the tables

Using strings for prices and dates to analyse the data accordingly

```
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   id                216250 non-null  int64
 1   id_order          216250 non-null  int64
 2   product_id        216250 non-null  int64
 3   product_quantity  216250 non-null  int64
 4   sku               216250 non-null  object
 5   unit_price        216250 non-null  float64
 6   date              216250 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(1), int64(4), object(1)
```

Brands

DCI - Brands Table

Solution:

Problems:

1. Remove short column
2. Rename long column
3. Remove duplicates from the brand_name column
4. Adding a dedicated primary key.
5. Add brand_id as a foreign key to the products table.

- Short key is reliant on the SKU structure
- Data redundancy with brands appearing multiple times in long column.

	brand_id	brand_name
0	0	8Mobility
1	1	Acme
2	2	Adonit
3	3	Aiino
4	4	Akitio
5	5	Allocacoc
6	6	Apple
7	7	Band&Strap
8	8	Beats
9	9	Belkin

name	price	in_stock	brand_id
one 6s 32GB Silver	529.00	0	7
0GB RAM I 4TB (...	771.33	0	120
r PCIe 2.5 "Sata ...	78.99	0	119
X Transparent S...	19.95	1	99
y 8 Plus / 7 Plus ...	39.99	1	118
0TB (8x10TB) Se... Na	5836.89	0	120
Monitor UP2716D "	683.99	0	29
.5GHz Retina 5K...	2659.00	0	7
38 "IPS Marco Slim	269.99	1	29
VGA Adapter Black	37.99	1	10

DCI - Products Duplicates

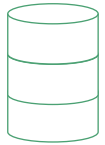
We had 8746 duplicates in the products table. **45%** of all products!

Add a input validation check to save storage and get a better overview

```
RangeIndex: 19326 entries, 0 to 19325
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   sku          19326 non-null  object
 1   name         19326 non-null  object
 2   desc         19319 non-null  object
 3   price        19280 non-null  object
 4   promo_price  19326 non-null  object
 5   in_stock     19326 non-null  int64
 6   type         19276 non-null  object
dtypes: int64(1), object(6)
memory usage: 1.0+ MB
```

```
Index: 10580 entries, 0 to 19325
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   sku          10580 non-null  object
 1   name         10580 non-null  object
 2   desc         10573 non-null  object
 3   price        10534 non-null  object
 4   promo_price  10580 non-null  object
 5   in_stock     10580 non-null  int64
 6   type         10530 non-null  object
dtypes: int64(1), object(6)
memory usage: 661.2+ KB
```

Products



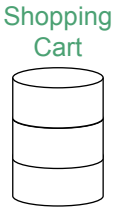
DCI - Price Data Issues & Solution

92% of promo prices used incorrect formats like multiple decimal points and more than two decimal places

- Implement a **'float' data type for prices** to prevent multiple decimal points.
- Add an **input validation check** to reject prices with more than two decimal places and trigger an error.

price	promo_price
107	814.659
1.568.206	1.568.206
566.35	5.659.896
29.99	237.925
24.99	22.99
79.99	56.99
199.99	1.441.174

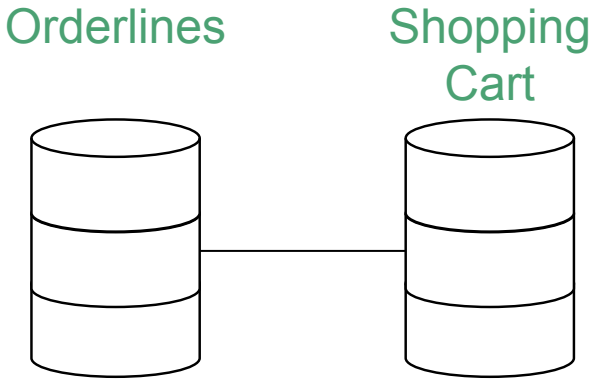
DCI - Shopping Cart



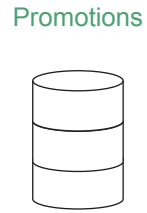
Problem: Current system uses Orderlines table for temporary "Shopping Basket" items, leading to old data like carts from 2017

Solution: Create a **Shopping Cart table**

Purpose: Stores temporary items (add, remove, adjust quantity) before an order is finalized.



DCI - Promotions

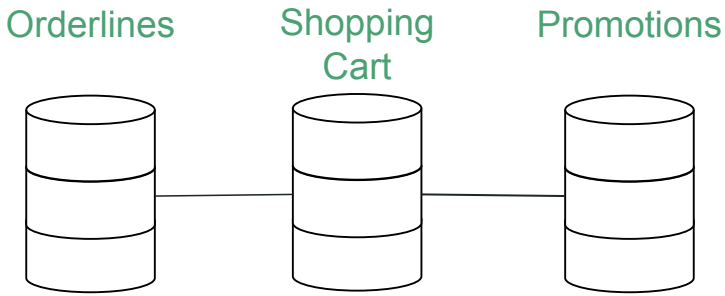


Remove promotion_price from Products table and create a **Promotions table**

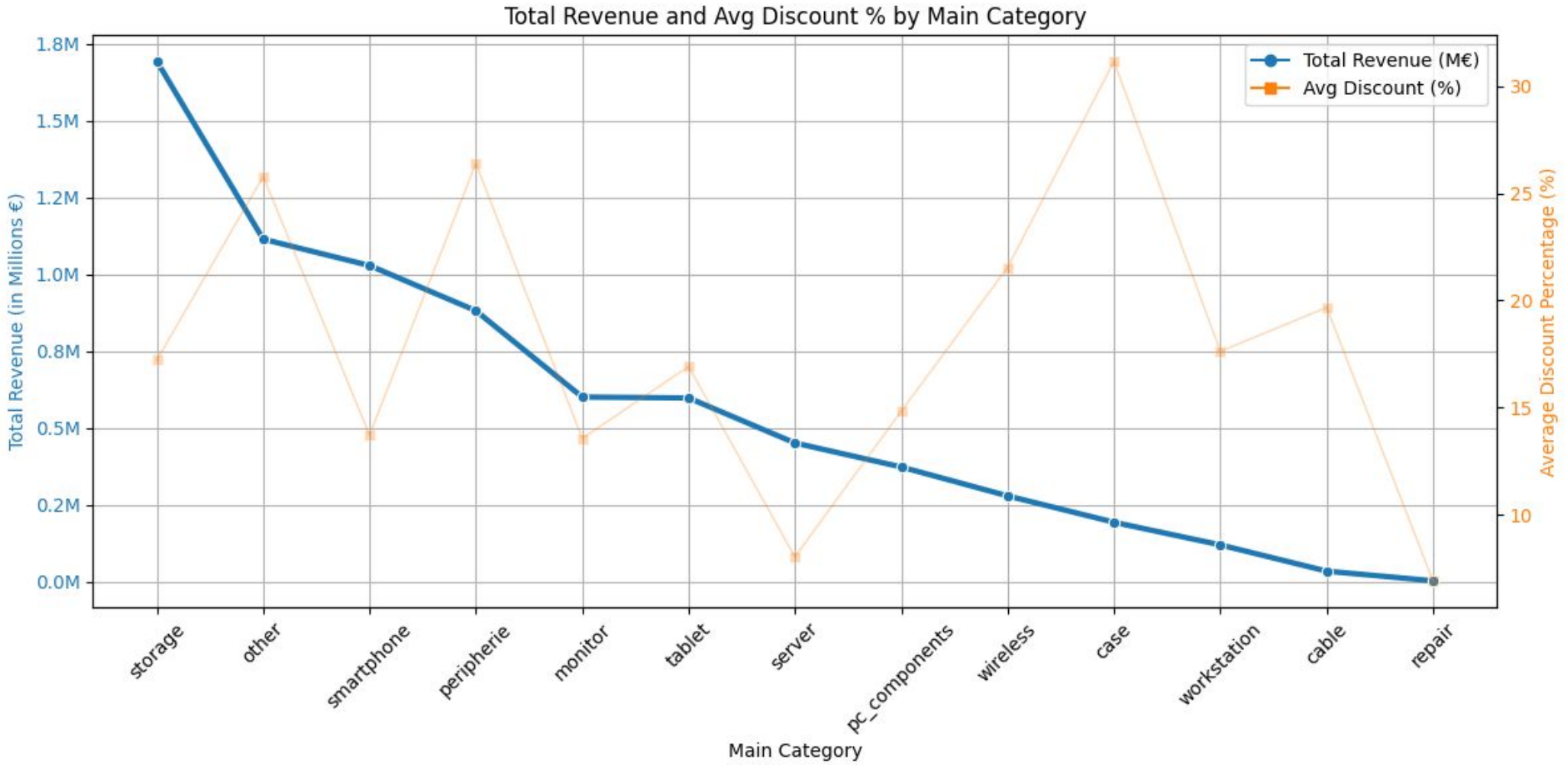
Avoids Redundancy: No need to update every product when promotions changes.

Flexibility: Allows dynamic, time-sensitive, and complex promotion rules.

Single Source of Truth: Promotions table centrally manages all promotion logic.



Discounts are ineffective due to consistently low prices



Total revenue doesn’t seem to depend much on discounts, regardless of price level.