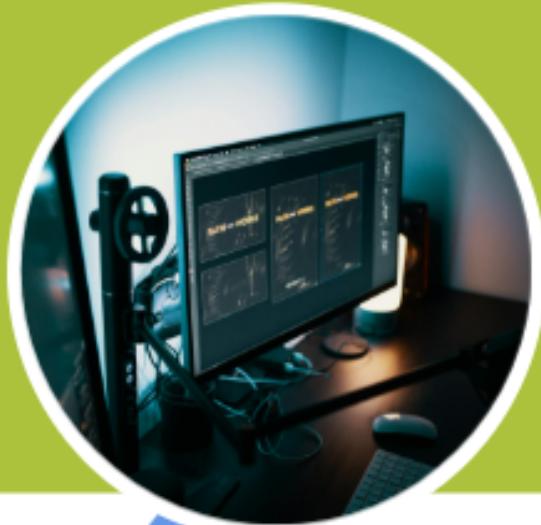




DATA Team

2030



Retail & E-Commerce analysis

in the Age of business analysis



Overview

In light of the task we take over
(Exploratory Data EDA) & (Prescriptive Analytics)
through the last four years of some retail
corporations to analyze the business situation and
highlight the weak points in addition to the strong
points, and figure out business solutions for the
upcoming plan **2026 - 2030**



Analytical Framework



We can see obviously the financial situation of our **sales (2.3)** billion by **profit (286)** million by **discount (15.62%)** on sales during the target was planned to be the **double**. (P1.Dashboard)

The overall situation is stable, and we are achieving profits; however, we have a plan in place to move forward. This plan focused on the business **dimensions** (**Seasons - regions - states - categories**) in addition to summarizing and distinguishing to **CX**



```
def calculate_discounts(discounts):
    # This function takes a list of discounts and calculates the total discount
    # applied to the total sales.
    # It returns a dictionary where the keys are the discount types and the values
    # are the total discounts applied to each type.
    # Example:
    # calculate_discounts([{"type": "Customer Loyalty", "amount": 10}, {"type": "Seasonal", "amount": 5}, {"type": "Promotional", "amount": 20}])
    # would return {"Customer Loyalty": 10, "Seasonal": 5, "Promotional": 20}
    pass

def calculate_sales(sales):
    # This function takes a list of sales and calculates the total sales.
    # It returns a dictionary where the keys are the sales categories and the values
    # are the total sales for each category.
    # Example:
    # calculate_sales([{"category": "Electronics", "amount": 100}, {"category": "Clothing", "amount": 50}, {"category": "Food", "amount": 30}, {"category": "Home Goods", "amount": 20}])
    # would return {"Electronics": 100, "Clothing": 50, "Food": 30, "Home Goods": 20}
    pass

def calculate_profit(profits):
    # This function takes a list of profits and calculates the total profit.
    # It returns a dictionary where the keys are the profit categories and the values
    # are the total profit for each category.
    # Example:
    # calculate_profit([{"category": "Electronics", "amount": 100}, {"category": "Clothing", "amount": 50}, {"category": "Food", "amount": 30}, {"category": "Home Goods", "amount": 20}])
    # would return {"Electronics": 100, "Clothing": 50, "Food": 30, "Home Goods": 20}
    pass

def calculate_discounts(discounts):
    # This function takes a list of discounts and calculates the total discount
    # applied to the total sales.
    # It returns a dictionary where the keys are the discount types and the values
    # are the total discounts applied to each type.
    # Example:
    # calculate_discounts([{"type": "Customer Loyalty", "amount": 10}, {"type": "Seasonal", "amount": 5}, {"type": "Promotional", "amount": 20}])
    # would return {"Customer Loyalty": 10, "Seasonal": 5, "Promotional": 20}
    pass

def calculate_sales(sales):
    # This function takes a list of sales and calculates the total sales.
    # It returns a dictionary where the keys are the sales categories and the values
    # are the total sales for each category.
    # Example:
    # calculate_sales([{"category": "Electronics", "amount": 100}, {"category": "Clothing", "amount": 50}, {"category": "Food", "amount": 30}, {"category": "Home Goods", "amount": 20}])
    # would return {"Electronics": 100, "Clothing": 50, "Food": 30, "Home Goods": 20}
    pass

def calculate_profit(profits):
    # This function takes a list of profits and calculates the total profit.
    # It returns a dictionary where the keys are the profit categories and the values
    # are the total profit for each category.
    # Example:
    # calculate_profit([{"category": "Electronics", "amount": 100}, {"category": "Clothing", "amount": 50}, {"category": "Food", "amount": 30}, {"category": "Home Goods", "amount": 20}])
    # would return {"Electronics": 100, "Clothing": 50, "Food": 30, "Home Goods": 20}
    pass
```

Analytical Framework



The data say that our bestseller **seasons** are the second two quarters (**Qtr3, Qtr4**) of each year, which may be influenced by occasions or weather in the USA environment. (P2 seasons)
The bestseller **regions** are (**east, west**). It may rely on the marketing plan or the types of offered categories
(P3 regions)

By extension **states** (**texas-florida-illion-michigin-virginia**)
(P5 Reigions-States)

About categories, each one has its own share, but the portions have another opinion, which we can massively search in
(P5 categories)

Analytical Framework



However, if we track the **sales & profit** in its own BI page (P6)

We can find that there are some **issues** or randomness in the **pricing** or the offers, which causes **skewness** in profit share, as per the **diagnostic** analysis we perform. So by **causal** analysis, we search all **dimensions** we determined to build the research on.

We found the **(table-bookcases-suppliers)** are **weak-selled** And the data refer to the **sub-categories** only, so our changes will focus on them and the regions we highlight as lazy purchasing.

Analytical Framework



Executive Summary:

The overall **business solution** we provide is based on the obvious analysis and data directions, and sales activity.

- Launch a **scalable marketing & advertising plan**, especially in the regions and states that have a **sales slowdown**.
- Refer to **product engineer** to design and fit the products better based on the different areas and **cultures**.
- We found items were **sold at a loss**, so we can inspect the pricing, too.
- After conducting the above point, we can **exclude** the sub-categories we highlighted if they are still weak, as they don't have their own **audience**.



Analytical Framework



Executive Summary:

We hope you will provide us

- **ratings** and evaluations from CX
- **data** gathered according to the marketing plan
- **web scraping** in your own retail software

all these to deepen the analysis to extract more precious
solutions



Thank you

Powered by data team