

Walmart

Here's an explanation for each measure and DAX calculation in the context of Walmart Inventory Optimization project. I'll cover the purpose, functionality, and impact on decision-making:

Age Grouping Measure

- **Purpose:** Segment customers by age to analyze purchasing behavior across different age groups. Groups include:
 - 0-18, 19-29, 30-39, 40-49, 50-59, 60+, and an "Other" category for ungrouped values.
- **Impact:** Age segmentation enables Walmart to target marketing and promotions to specific age groups, optimizing engagement and relevance.

Calculated Columns DAX

1. **Is Repeat Customer** : `IF(COUNTROWS(FILTER('WalMart','WalMart'[customer_id] EARLIER('WalMart'[customer_id]))) > 1,"Yes","No")`
`IF(COUNTROWS(FILTER('WalMart','WalMart'[customer_id] = EARLIER('WalMart'[customer_id])) > 1, "Yes","No")`
 - **Purpose:** Identifies if a customer is a repeat buyer by counting instances of the customer_id.
 - **Impact:** Highlights customer loyalty and purchasing frequency, enabling Walmart to focus retention efforts and loyalty programs on repeat customers.
2. **HourOfDay**: `FORMAT('Walmart'[transaction_date], "hh:mm")`
 - **Purpose:** Extracts the hour and minute from transaction_date for analyzing peak shopping times.
 - **Impact:** Helps identify peak selling hours, allowing Walmart to optimize staffing and inventory levels during high-traffic periods.
3. **City** : `LEFT(WalMart[store_location], SEARCH(",", WalMart[store_location]) - 1)`
 - **Purpose:** Extracts the city from the store_location field.
 - **Impact:** Facilitates regional sales analysis, enabling location-specific insights and inventory allocation.
4. **State** : `SWITCH(TRUE(), RIGHT([store_location], LEN([store_location]) - SEARCH(" ", [store_location]) - 1) = "CA", "California", RIGHT([store_location], LEN([store_location]) - SEARCH(" ", [store_location]) - 1) = "FL", "Florida", RIGHT([store_location], LEN([store_location]) - SEARCH(" ", [store_location]) - 1) = "IL", "Illinois", RIGHT([store_location], LEN([store_location]) - SEARCH(" ", [store_location]) - 1) = "NY", "New York", RIGHT([store_location], LEN([store_location]) - SEARCH(" ", [store_location]) - 1) = "TX", "Texas", -- Add other states as needed)`
 - **Purpose:** Derives the state from the store_location column for geographic analysis.
 - **Impact:** Helps Walmart analyze sales and stockouts by state, allowing for more effective regional inventory planning.

DAX Measures

1. **TotalSales:** `SUMX(WalMart, WalMart[quantity_sold] * WalMart[unit_price])`
 - **Purpose:** Calculates the total sales revenue.
 - **Impact:** Offers a top-level view of financial performance, essential for assessing demand and revenue trends.
2. **Total Sales Volume:** `SUM(WalMart[quantity_sold])`
 - **Purpose:** Sums total units sold across all transactions.
 - **Impact:** Reflects demand levels, informing supply chain and stocking strategies.
3. **AverageOrderValue:** `DIVIDE([TotalSales],DISTINCTCOUNT(WalMart[transaction_id]))`
 - **Purpose:** Calculates the average revenue per transaction.
 - **Impact:** Identifies high-value customers and effective upselling strategies, aiding in revenue-maximizing decisions.
4. **Count of Repeat Customers:** `COUNTROWS(FILTER('WalMart','WalMart'[Is Repeat Customer]= "Yes"))`
 - **Purpose:** Counts the number of repeat customers, useful for loyalty analysis.
 - **Impact:** Allows Walmart to measure customer retention and tailor loyalty incentives for frequent buyers.
5. **ForecastAccuracy:** `DIVIDE(SUM(WalMart[actual_demand]), SUM(WalMart[forecasted_demand]), 0)`
 - **Purpose:** Measures the accuracy of demand forecasting by comparing forecasted vs. actual demand.
 - **Impact:** Supports continuous improvement in forecasting, helping Walmart reduce stockouts and overstock.
6. **Holiday Sales & Non-Holiday Sales:** `CALCULATE([TotalSales],WalMart[holiday_indicator] = TRUE)`
`CALCULATE([TotalSales],WalMart[holiday_indicator] = FALSE)`
 - **Purpose:** Separates sales based on holidays and non-holidays to analyze seasonal demand.
 - **Impact:** Enables Walmart to prepare for demand spikes, ensuring adequate stock during holidays to maximize revenue.
7. **PeakSellingHour:** `MAXX(TOPN (1,ADDCOLUMNS (SUMMARIZE ('Walmart','Walmart'[HourOfDay]),"SalesCount", COUNT('Walmart'[transaction_id])),[SalesCount],DESC),[HourOfDay])`
 - **Purpose:** Identifies the peak hour with the most transactions.
 - **Impact:** Informs staffing and stock levels during peak hours, enhancing customer experience and sales efficiency.
8. **PromotionPercentage:** `DIVIDE(CALCULATE(COUNTROWS(WalMart), WalMart[promotion_applied] = TRUE),COUNTROWS(WalMart), 0)`
 - **Purpose:** Calculates the percentage of transactions involving promotions.
 - **Impact:** Helps Walmart understand the reach and frequency of promotions, refining future strategies.
9. **PromotionSalesDifference:** `[RevenueFromPromotions] - [RevenueBeforePromotions]`
 - **Purpose:** Quantifies the revenue uplift due to promotions.
 - **Impact:** Evaluates promotional effectiveness, allowing Walmart to optimize promotional investment.

10. RevenueBeforePromotions & RevenueFromPromotions: SUMX(FILTER(WalMart, WalMart[promotion_applied] = FALSE), WalMart[quantity_sold] * WalMart[unit_price])

SUMX(FILTER(WalMart, WalMart[promotion_applied] = TRUE), WalMart[quantity_sold] * WalMart[unit_price])

- **Purpose:** Separates revenue into promotional and non-promotional sales.
- **Impact:** Enables ROI calculations for promotions, helping Walmart optimize its promotional strategy.

11. Selected Context: IF(HASONEVALUE('WalMart'[product_name]), "Product: " & SELECTEDVALUE(WalMart[product_name]), IF(HASONEVALUE(WalMart[City]), "City: " & SELECTEDVALUE(WalMart[City]), "Select a Product or City"))

- **Purpose:** Displays context-specific information (e.g., product or city) based on selected filters.
- **Impact:** Enhances report interactivity and clarity, allowing stakeholders to view focused insights.

12. Stockout Rate by Store: DIVIDE(CALCULATE(COUNTROWS(WalMart), WalMart[stockout_indicator] = TRUE), COUNTROWS(WalMart))

- **Purpose:** Calculates the stockout rate per store.
- **Impact:** Helps Walmart identify high-stockout stores, guiding location-specific inventory adjustments.

N.B : (In the DAX formulas provided, 'WalMart' refers to the table name in your Power BI data model.)

These measures collectively empower Walmart with actionable insights into customer behavior, demand patterns, and inventory needs. Each measure aids in decision-making, optimizing inventory levels, promotional effectiveness, and overall operations.

