Strategic Suggestions for Slow-Moving & Overstocked Items at Zudio

Based on the sales data analysis, here are actionable strategies to address slow-moving inventory and overstock situations:

1. Identification & Segmentation

A. Product Diagnosis

- **ABC Analysis**: Classify items by:
    - **A (Top 20%)**: Fast-movers (prioritize restocking)
    - **B (Middle 30%)**: Steady sellers (maintain inventory)
    - **C (Bottom 50%)**: Slow-movers (target for clearance)
- **Category-Specific Slow-Movers**:
    - **Men**: Dress shirts (if overstocked)
    - **Women**: Certain seasonal dresses
    - **Kids**: Specific jacket styles

B. Store-Level Analysis

- Identify locations with highest overstock (e.g., Mumbai, Thane showed lower sales density)

2. Clearance Strategies

A. Pricing Tactics

Tactic	Implementation	Example
Bundling	Pair slow items with bestsellers	"Buy 1 Jeans, Get 1 Shirt 50% Off"
Flash Sales	24-48hr discounts (promote via SMS/email)	"End-of-Season: 40% Off Select Styles"
Tiered Discounts	Progressive markdowns (20% → 40% → 60%)	Weekly price reductions

B. Channel Optimization

- **Online Focus**: List overstock on Zudio's app with "Clearance" tags
- **Store Transfers**: Move stock from low-sales (e.g., Thane) to high-traffic stores (Jalna, Kolkata)

3. Inventory Prevention

A. Demand Forecasting

- **Tool**: Implement Power BI predictive analytics to:
    - Flag at-risk items before ordering
    - Compare sales velocity vs. stock levels

B. Procurement Adjustments

- **Vendor Negotiations**:
    - Return clauses for overstock
    - Smaller, frequent orders for trend-based items

4. Marketing Interventions

**A. Targeted Campaigns

- **Social Proof**:
    - "Staff Picks" sections featuring slow-movers
    - Customer styling videos with overstock items
- **Segmented Offers**:
    - "Kids' Winter Wear" email blast for excess jackets
    - App push notifications for location-specific clearance

5. Data-Driven Actions

python

Copy

Download

```
# Python code to flag slow-movers
slow_movers = df.groupby('Product ID').filter(
  lambda x: x['Quantity'].sum() < 10  # Threshold
)
print(f"{len(slow_movers)} slow-moving SKUs identified")
```

Expected Outcomes:

- 30-50% reduction in overstock within 2 quarters
- Improved cash flow from liquidated inventory
- More accurate future purchasing