Annie's Magic Numbers - Strategic Sales & Inventory Analysis Documentation

Author: Alonso Castro
Platform: Databricks (PySpark)
Purpose: Advanced sales and inventory analysis with strategic insights

Executive Summary

This analysis transforms raw sales, inventory, and purchasing data into actionable strategic insights using Apache Spark. The system processes over millions records to identify profit opportunities, optimize product portfolios, and provide data-driven recommendations for business growth.

Key Results Overview

Financial Performance
- Total Portfolio Analysis: Complete profit analysis across all products
- Average Margin: 32.4% across the product portfolio
- Strategic Categorization: Products classified into 6 performance tiers

Strategic Insights Discovered
1. Star Products: 31,268 high-profit, high-margin items generating $35.4 in profit
2. Strategic Opportunities: 1,110 products with potential for improvement rather than discontinuation
3. Inventory Optimization: Identified overstocked items worth $244 requiring promotional strategies

Technical Architecture

1. Data Pipeline (Lines 21-67)
python
```
def load_and_validate_tables():
    """Load tables with error handling and basic validation"""
```

- Input Sources: 6 different data tables (purchases, prices, inventory, invoices, sales)
- Validation: Automated record counting and null value detection
- Error Handling: Comprehensive try-catch blocks for data integrity

2. Data Preparation & Cleaning (Lines 68-132)
The system implements three specialized cleaning functions:

Sales Data Preparation
- Converts data types (Integer for quantities, Double for currency)
- Filters out invalid transactions (zero quantities/amounts)
- Ensures data quality with null value removal

Purchase Data Preparation
- Standardizes vendor information

- Calculates weighted average costs per product
- Handles price variations across different purchase dates

Inventory Management
- Reconciles beginning and ending inventory levels
- Calculates inventory turnover rates
- Identifies slow-moving vs fast-moving products

3. Advanced Cost Calculation (Lines 134-148)
python
def calculate_weighted_avg_cost(purchases_df):
    """Calculate weighted average cost per product"""

Innovation: Uses weighted averages instead of simple averages for more accurate cost calculations, accounting for:
- Multiple purchase prices over time
- Varying quantities purchased
- Seasonal price fluctuations

4. Comprehensive Profit Analysis (Lines 150-194)
The system calculates multiple performance metrics:

Metric	Formula	Business Value
Total Cost	Unit Cost × Sales Quantity	Accurate COGS calculation
Profit	Sales Dollars - Total Cost	Direct profitability measure
Margin %	(Profit ÷ Sales Dollars) × 100	Pricing efficiency
ROI %	(Profit ÷ Total Cost) × 100	Investment return
Inventory Turnover	Sales Quantity ÷ Beginning Inventory	Stock efficiency

5. Strategic Analysis Framework (Lines 230-290)

Performance Categorization System
The code implements a sophisticated 6-tier classification:

1. Star Products (Profit > $1,000, Margin > 20%)
   - High profit, high margin items
   - Focus for growth and premium positioning

2. Strong Performers (Profit > $500, Margin > 10%)
   - Consistent contributors to bottom line
   - Candidates for market expansion

3. Solid Contributors (Profit > $0, Margin > 5%)
   - Steady performers requiring maintenance
   - Potential for optimization

4. Volume Players (Profit > $0, High Volume)

- Low margin but high turnover
- Candidates for cost reduction

5. Strategic Opportunities (Negative profit, High volume)
- KEY INNOVATION: Instead of immediate discontinuation
- Targeted for strategic intervention

6. Needs Review (All others)
- Requires detailed individual analysis

6. Strategic Opportunities Identification (Lines 196-258)

The "Nurture vs Discard" Philosophy
Core Innovation: Rather than simply identifying products to drop, the system categorizes struggling products into strategic intervention categories:

High Volume Losers (Price Adjustment Strategy)
- Criteria: Negative profit BUT high sales volume (>50 units)
- Strategy: Price adjustment rather than discontinuation
- Logic: Market demand exists; pricing is the issue

High Turnover, Low Margin (Volume Strategy)
- Criteria: 0-15% margin BUT high turnover (>2x)
- Strategy: Leverage volume for supplier negotiations
- Logic: Fast-moving inventory can support lower margins

Overstocked Products (Promotion Strategy)
- Criteria: High inventory (>100 units), Low turnover (<0.5x)
- Strategy: Promotional campaigns and bundling
- Logic: Clear inventory through strategic marketing

Products for Discontinuation
- Criteria: Significant losses (<-$1,000), Low volume (<20 units), Poor margins (<-10%)
- Strategy: Only then consider discontinuation
- Logic: No viable path to profitability

7. Advanced Visualization System (Lines 348-450)
The code generates 6 strategic visualizations:

1. Performance Matrix: Bubble chart showing profit vs margin (size = volume, color = turnover)
2. Brand Comparison: Top 10 brands by profit with detailed metrics
3. Margin Distribution: Histogram showing portfolio margin spread
4. Revenue vs Volume: Scatter plot identifying efficiency patterns
5. Strategic Matrix: Boston Consulting Group-style matrix (Stars, Cash Cows, Question Marks, Dogs)
6. Top vs Bottom: Comparative analysis of best and worst performers

Results Interpretation

Top Performers Analysis
- Best Product: Jack Daniels No.7 Black generating $27.67 profit margin
- Best Brand: Hennessy with $179.24 in total profit
- Highest Margin Products: Achieving 100% margins in specialty categories

Strategic Opportunities Identified
- Price Adjustment Candidates: 1,110 products requiring pricing strategy review
- Volume Strategy Products: High-turnover items suitable for supplier negotiations
- Promotion Opportunities: Overstocked items worth $244 requiring clearance strategies

Portfolio Health Indicators
- Margin Distribution: Well-balanced with mean of 32.4%
- Performance Categories: Healthy distribution across all tiers
- Strategic Balance: More products in "opportunity" than "discontinuation" categories

Business Impact & Recommendations

Immediate Actions (0-30 days)
1. Price Optimization: Adjust pricing on 1,110 identified products
2. Inventory Clearance: Launch promotional campaigns for overstocked items
3. Supplier Negotiations: Leverage volume data for better terms

Medium-term Strategy (30-90 days)
1. Product Repositioning: Move struggling products to different market segments
2. Bundle Creation: Combine slow movers with popular items
3. Seasonal Campaigns: Time-based promotional strategies

Long-term Portfolio Management (90+ days)
1. Performance Monitoring: Implement monthly reviews using this framework
2. Predictive Analytics: Extend analysis to forecast future performance
3. Market Expansion: Scale successful products to new markets

Technical Scalability

Performance Optimizations
- Spark SQL: Leverages distributed computing for large datasets
- Lazy Evaluation: Efficient processing through Spark's optimization engine
- Memory Management: Proper caching and partitioning strategies

Code Modularity
- Function-based Architecture: Each analysis component is independently testable
- Error Handling: Comprehensive validation at each processing stage
- Documentation: Extensive inline comments for maintainability

Conclusion

This analysis demonstrates that sophisticated business intelligence doesn't require expensive software—it requires strategic thinking implemented through robust code. The "nurture vs discard" philosophy transforms traditional product analysis from a purely elimination exercise into a strategic optimization opportunity.

Key Innovation: By focusing on strategic interventions rather than automatic discontinuation, businesses can recover potentially millions in revenue from products that appear to be underperforming but actually have strategic value.

The system provides a replicable framework for continuous business optimization, turning data analysis into actionable strategic advantage.