Sales Probability Modeling: Dahlia Inventory Depletion

# 1. Environment Setup & Dependencies

Objective: Install and import libraries needed for modeling inventory depletion and prediction.

• Installed packages: catboost, feature\_engine, plot-metric, scipy

• Imported: pandas, numpy, matplotlib, plotly, scikit-learn, CatBoostClassifier, BigQuery libraries

# 2. Data Ingestion from Google BigQuery

Objective: Fetch multiple inventory and order-related tables from BigQuery.

• Queried tables: product\_listings, products, product\_categories, offers, orders\_level\_1/2, etc.

• Used credentials and mounted Google Drive for seamless access.

# 3. Initial Data Cleaning & Structuring

Goal: Combine datasets, unify schema, and resolve duplicates.

• Merged products with categories & subcategories.

• Renamed key columns and dropped irrelevant/duplicate data.

# 4. Underscore & String Cleaning

Goal: Normalize text features for consistency.

• Cleaned leading/trailing underscores, lowercased and standardized text.

# 5. Inventory Classification (CORE vs NON\_CORE)

Goal: Separate and process data streams based on inventory class.

• Created df\_inv (CORE) and non\_core\_inv, standardized dates, and removed duplicates.

# 6. Merging Offers & Non-Core Inventory

Goal: Match inventory records with offer details.

• Merged offers based on SKU, seller, and date; computed offer\_day\_diff\_updated\_inv.

# 7. Temporal Shifting for Inventory Movement

Goal: Track depletion by comparing current and previous inventory records.

• Used shift(1) and calculated depletion and time between updates.

# 8. Depletion Logic by Product Category

Goal: Apply category-sensitive thresholds to adjust depletion logic.

• Categories segmented into low, mid, high depletion types with custom rules.

# 9. Dummy Generation for Negative Classes

Goal: Introduce synthetic examples for class balancing.

• Created dummies for low shelf life, low time, and high depletion.

# 10. Listing Type Assignment

Goal: Classify inventory into types: fresh, obsolete, excess, etc.

• Used shelf life and recovery rate to determine listing type.

# 11. Liquidation Channel Simulation

Goal: Model inventory movement through liquidation paths.

• Channels: friends/family, retailer, liquidator, etc. | Weighted by SKU priority.

# 12. Sensitivity Correction for Time & Shelf Life

Goal: Restore correlation post-explosion.

• Grouped data and scaled 'time' and 'shelf\_life\_remaining\_days\_prev'.

# 13. Final Cleansing & Enrichment

Goal: Final prep before modeling.

• Imputed missing values, created depletion\_percent, filtered bad records.

# 14. Model Training Dataset Prep

Train columns used:

• Categorical: sku\_number, brand, product\_category, product\_subcategory, seller\_name, listing\_condition, liquidation\_channel

• Numerical: time, depletion\_percent | Label: possible (1 or 0)

# 15. Model Training (CatBoostClassifier)

Goal: Predict whether an inventory listing is realistically sellable.

• Used CatBoostClassifier with tuned parameters. Trained on ~200k rows.

# ✅ Summary

The pipeline builds a robust sales probability model by consolidating datasets, engineering features, simulating dummy data, and predicting liquidation success likelihood. Enables smarter decisions around listing, inventory clearance, and strategy.

# 16. Handling Nulls & Anomalies in Shelf Life

Goal: Ensure model robustness by filling missing shelf life values.

• Calculated median shelf life per priority segment (P1, P2, P3).

• Applied these medians to impute missing shelf\_life\_remaining\_days.

• Ensured all rows used in training had valid time and shelf life values.

# 17. Handling Negative Time Values

Goal: Fix data quality issues where time difference was negative.

• Identified ~700 rows with negative `time`.

• Applied absolute value transformation to maintain temporal consistency.

# 18. Feature Transformation for Depletion Percent

Goal: Derive a normalized metric for depletion.

• Computed: depletion\_percent = (depletion / total\_units\_prev) \* 100

• Applied to both actual and dummy rows, enabling a common learning signal.

# 19. Dataset Balancing and Filtering

Goal: Ensure balanced training data with both possible and impossible scenarios.

• Tagged real-world examples with `possible=1`.

• Added synthetic negatives with `possible=0` (low shelf life, high depletion, etc.)

• Merged all, dropped duplicates, and retained ~430,000 examples.

# 20. Evaluation Strategy (Suggested)

Goal: Ensure real-world usefulness and accuracy of the probability model.

• Evaluate using metrics like AUC, Precision@K, and Uplift on real liquidation events.

• Future scope includes segment-wise modeling or multi-task learning for each channel.

# 21. Use Cases and Business Value

🎯 Use Cases:

• Flag SKUs unlikely to sell – reduce listing costs and avoid customer frustration.

• Identify best liquidation channel for a product based on sell-through probability.

• Enable priority-based recommendation engine for faster stock clearance.

💡 Business Impact:

• Higher inventory turnover.

• Improved recovery from unsold stock.

• Reduced dependency on disposal methods.