📘 Dahlia Depletion Modeling – Detailed Documentation

🎯 Objective: Predict the depletion percentage of an SKU using regression. This model estimates what portion of stock will be liquidated within a lifecycle window based on inventory metadata, listing condition, and liquidation channels. It complements probability and time-based models.

# 1. Data Extraction and Joining

• Queried data from BigQuery tables: product\_listings, products, categories, subcategories, offers, orders.

• Joined orders and offers via tracking number and SKU to compute 'ordered units' and 'offered units'.

• Renamed and merged fields, e.g. recovery rate, offer price, total\_units\_ordered.

# 2. Core vs Non-Core Inventory Processing

• CORE: Cleaned underscores, normalized strings, converted dates, dropped missing brand/category/subcategory.

• NON\_CORE: Deduplicated on SKU+expiry+update, converted date formats, and merged with closest valid offer based on time lag.

# 3. Tracking Inventory Changes Over Time

• Created shifted versions of inventory rows to compare time intervals and quantity drops.

• Calculated:

– `depletion = total\_units\_prev - total\_units`

– `time = updated\_on - updated\_on\_prev`

# 4. Depletion Adjustment Using Category Thresholds

• Computed mean depletion per category and split into low, mid, and high depletion categories.

• Applied category-wise time cutoffs (20, 45, 60 days) to adjust 'true' depletion values to total\_units\_prev where needed.

# 5. Outlier Handling & Priority Corrections

• Applied correction: If pollen units ordered > depletion, cap depletion at pollen units.

• For P1 SKUs: 80% chance of randomly reducing depletion between 5% to 20%.

# 6. Synthetic Data Creation

• Generated 4 types of synthetic rows:

– High Time: Increased time, scaled depletion upward

– Low Time: Reduced time, scaled depletion downward

– High Shelf Life: Shelf life increased, depletion scaled

– Low Shelf Life: Shelf life decreased, depletion scaled

• Ensured total\_units\_prev always ≥ depletion in all synthetic cases.

# 7. Listing Type Annotation

• Classified records as `fresh`, `obsolete`, or `excess` using shelf life and recovery thresholds.

• Added randomness: 5% records labeled as `damaged` and `made\_to\_order`.

# 8. Multi-Channel Liquidation Modeling

• Exploded each record into 6 liquidation channels (e.g. pollen, retailer, liquidator).

• Computed `depletion\_other\_than\_pollen = depletion - pollen\_liquidated\_units`.

• Distributed depletion using custom weights for each priority and channel.

• Used `apply(get\_depleted\_units\_via\_liquidation\_channel)` logic per channel.

# 9. Temporal Sensitivity Correction

• Created a `channel\_group` identifier to represent rows from the same original record.

• Adjusted `time` and `shelf\_life` using group-wise max depletion to maintain feature-label correlation.

# 10. Final Feature Engineering

• Merged seller personas, imputed missing priority/seller\_type with empty string.

• Removed rows with 0 `total\_units\_prev` or null depletion%.

• Calculated `depletion\_percent = (depletion / total\_units\_prev) \* 100`.

• Filled null `shelf\_life\_remaining\_days` using priority-wise medians (P1: 67, P2: 179, P3: 439 days).

• Joined Google Sheet shelf-life data for future analysis/validation.

# 11. Model Training – CatBoost Regressor

• Used CatBoost with RMSE loss, learning\_rate=0.6, depth=5, early stopping at 25 rounds.

• Target = log-transformed depletion\_percent (log1p).

• Input features:

– Categorical: SKU, brand, category, subcategory, seller\_name, listing\_condition

– Numerical: total\_units, time, shelf\_life, recovery\_rate, etc.

• Data split into training and validation using `train\_test\_split`.

# 12. ✅ Use Cases & Business Value

• Estimate how much of listed inventory will realistically sell via different liquidation routes.

• Feed into planning dashboards, markdown engines, and seller scorecards.

• Enables sell-through forecasting beyond pollen – better multi-channel planning.

• Enables targeted decisioning: when to push to liquidators, mark as loss, or discount further.