ML Challenge 2025: Smart Product Pricing Solution

Team Name: Deep Thinkers
Team Members: Bhargav P Y, Chethan T P, Aiman Shariff, B R Abhijith
Submission Date: 13-10-2025

---

1. Executive Summary

This solution addresses the pricing challenge using a Multimodal Learning approach centered on an optimized LightGBM Regressor. Our method strategically fuses features derived from three modalities: text (product content), image (product photo), and tabular (numerical/categorical data). Key innovations include a Text Embedding Ensemble (MiniLM-L6 and MPNet-base) for richer semantics and rigorous validation using the SMAPE metric and a 5-fold cross-validation strategy.

---

2. Methodology Overview

2.1 Problem Analysis

The core of the ML Challenge was a regression task requiring the prediction of product price based on multimodal features. The target variable was identified during EDA as highly right-skewed (heavy tail of high-priced items), necessitating a log1p transformation to normalize the distribution and stabilize model training.

Key Observations:
Target Skewness: The original price distribution was severely right-skewed, with the median being significantly lower than the mean, and a few extreme outliers driving the maximum value (indicating extreme outliers).

Multimodality: Product price is clearly influenced by structured numerical data (base_value, item_pack_quantity), descriptive text (catalog_content), and visual image data, mandating a feature fusion approach.

Feature Skewness & Scaling: Numerical features (item_pack_quantity, base_value) also exhibited heavy skewness. The decision to use Quantile Transformation (after log1p) was based on achieving a near-perfect Gaussian distribution, which often improves the performance of tree-based models.

2.2 Solution Strategy

Approach Type: Hybrid Multimodal Learning with a Stacked Model (Feature Fusion)
Core Innovation: Ensembled Text Feature Extraction (Concatenation of two distinct Sentence Transformer model outputs) combined with a robust Gradient Boosting Machine (LightGBM) for final prediction.

---

3. Model Architecture

3.1 Architecture Overview

The solution follows a multi-stage feature extraction and fusion pipeline:

Parallel Feature Extraction: Text and Image features are generated independently.
Tabular Processing: Numerical and categorical features are scaled, encoded, and aligned.
Feature Fusion: All feature components (X_simple, E_text, E_image) are loaded as individual arrays. For training, these arrays are dynamically sliced and concatenated for each fold to avoid memory fragmentation.
Prediction: A single LightGBM Regressor is trained per fold, and the final result is an ensemble average of the 3 models trained on the log(price).

3.2 Model Components

Final Predictor: LightGBM Regressor

Text Processing Pipeline:

- Preprocessing steps: Text cleaning (lowercase, remove noise/non-ASCII, normalize whitespace).
- Model type: Ensemble of two Sentence Transformer models.
- Key parameters: [Model 1: all-MiniLM-L6-v2 (384 dimensions), Model 2: all-mpnet-base-v2 (768 dimensions), Embeddings concatenated to form a 1152-dimensional text vector]

Image Processing Pipeline:

- Preprocessing steps: Standard ImageNet Compose transform (Resize (256), CenterCrop (224), ToTensor, Normalize).
- Model type: Pretrained ResNet50 (Convolutional Neural Network)
- Key parameters: Input size: 224×224, Backbone: ResNet50 pretrained on ImageNet, Output layer: Global Average Pooling, Embedding size: 2048, Classification head removed. Crucially, missing images were imputed with a 2048-dimensional zero vector to maintain matrix size, with the image_missing flag carrying the predictive signal.

Tabular Feature Engineering:

- Scaling: item_pack_quantity and base_value features were log1p-transformed, then fitted and transformed with a Quantile Transformer (output_distribution='normal') to normalize to a Gaussian distribution.

- Encoding: Target Encoding (TE): Applied to item_name. To eliminate data leakage, the TE feature was not pre-calculated. Instead, it was calculated on-the-fly within each K-Fold iteration, using the train_index target values to encode the corresponding val_index samples. Missing/new items were imputed with the train_log_price_mean. One-Hot Encoding (OHE) was applied to the binned base_unit, with columns rigorously aligned to the training set.

---

4. Model Performance

4.1 Validation Results

- **SMAPE Score:** 52.44%

5. Conclusion

Our approach successfully integrates text, image, and tabular features into a unified matrix for prediction by a powerful LightGBM model. We learned the necessity of rigorous, step-wise execution, requiring thoroughness at every stage. This involved constantly optimizing the pipeline through continuous refinement and incorporating best practices, while maintaining calmness and patience when troubleshooting errors.

Appendix

A. Code artefacts

https://drive.google.com/file/d/1K5KeR-Y_aj6nJb9ke_ps0XlC93jnOJuf/view?usp=sharing

B. Additional Results