**ML Challenge 2025: Multimodal Smart Product Pricing Solution**

**Team Name:** Team House of Codesmiths  
**Team Members:** Satya Venkata Hemanth Challapalli, Girisha Malni N, Syed Ameen, Boodala Akhil  
**Submission Date:** October 13, 2025

**1. Executive Summary**

We developed a **multimodal ensemble regression model** that predicts product prices using three complementary data sources-**textual descriptions**, **product images**, and **engineered numeric features**.  
Our architecture combines **Sentence-BERT embeddings** for text, **EfficientNet-B3 embeddings** for images, and **log-transformed numeric features** such as item pack quantity and text length.  
Predictions are generated through a **three-model ensemble** of **LightGBM**, **CatBoost**, and a **neural network (MLP)**, blended to produce final outputs with strong generalization.

**Key Innovations:**

* Integration of **semantic text embeddings** (Sentence-BERT) with **visual embeddings** (EfficientNet-B3).
* **Hybrid feature fusion** combining dense embeddings and engineered numeric features.
* **Three-way model ensemble** leveraging gradient boosting and deep learning strengths.
* **Cached feature extraction** for efficient experimentation and reproducibility.

**2. Methodology Overview**

**2.1 Problem Analysis**

The challenge required predicting product prices based on catalog content and associated images.  
Exploratory Data Analysis (EDA) identified the following insights:

* Product text often includes **quantitative indicators** such as “500 ml”, “pack of 2”, or “3 pcs.”
* Images provide **contextual cues** (e.g., brand, size, packaging) valuable for pricing.
* Price distribution is **right-skewed**, necessitating **log-transformations** for variance stabilization.

**2.2 Solution Strategy**

We designed a **multimodal ensemble pipeline** combining three feature types and three regressors:

| **Component** | **Description** |
| --- | --- |
| **Text features** | Sentence-BERT embeddings (768 dim) from pre-trained model all-MiniLM-L6-v2 |
| **Image features** | EfficientNet-B3 embeddings (1536 dim) with the classifier head removed |
| **Numeric features** | Engineered attributes—log(IPQ), text length, digit counts, token counts |
| **Models used** | LightGBM, CatBoost, and a fully-connected MLP |
| **Ensembling** | Averaged predictions across models and folds |

This hybrid setup allows both interpretability (from boosting models) and deep feature representation (from neural models).

**3. Model Architecture**

**3.1 Architecture Overview**

catalog\_content (text) ──▶ clean\_text ──▶ Sentence-BERT ──▶ text\_embeddings

image\_link (url) ──▶ EfficientNet-B3 ──▶ image\_embeddings

numeric features (ipq\_log, len\_text, num\_digits, num\_tokens)

│

└──▶ Concatenate ──▶ [LightGBM | CatBoost | MLP] ──▶ Ensemble Averaging ──▶ Final price

**3.2 Model Components**

**A. Data Preprocessing**

* **Text Cleaning:** Lowercasing, newline removal, whitespace normalization.
* **IPQ Extraction:** Regular expressions used to identify numeric quantities (“500 ml”, “x2”, “pack of 3”).
* **Feature Engineering:**
  + ipq\_log: Log(1 + Item Pack Quantity)
  + len\_text: Length of cleaned text
  + num\_digits: Count of numeric digits
  + num\_tokens: Number of whitespace-separated tokens

**B. Text Embedding Pipeline**

* **Model:** Sentence-BERT (all-MiniLM-L6-v2)
* **Embedding Dimension:** 768
* **Processing:** Mean-pooled sentence representations for each catalog entry
* **Caching:** Saved as .npy for reusability

**C. Image Embedding Pipeline**

* **Model:** tf\_efficientnet\_b3\_ns (pretrained via timm)
* **Output:** 1536-dimensional visual embeddings
* **Processing:**
  + Image download from URLs
  + Resize to 300 × 300 pixels
  + Convert to RGB tensor (normalized to [0, 1])
* **Caching:** Saved as .npy for faster reruns

**D. Model Ensemble**

1. **LightGBM Regressor:**
   * Objective: regression
   * n\_estimators = 3000, learning\_rate = 0.05
   * num\_leaves = 31, feature\_fraction = 0.8, bagging\_fraction = 0.8
2. **CatBoost Regressor:**
   * iterations = 2000, learning\_rate = 0.05, depth = 6
   * eval\_metric = RMSE, early\_stopping\_rounds = 100
3. **Neural Network (MLP):**
   * Architecture: [Input → 1024 → 512 → 1]
   * Activations: ReLU
   * Dropout: 0.3
   * Loss: MSELoss()
   * Optimizer: Adam (lr = 1e-3)
   * Epochs: 50

**E. Cross-Validation Setup**

* **Scheme:** 5-Fold K-Fold cross-validation
* **Target Transformation:** log(1 + price)
* **Blending:** Average across folds and models
* **Evaluation Metric:** SMAPE (Symmetric Mean Absolute Percentage Error)

**4. Model Performance**

**4.1 Validation Results**

| **Metric** | **Description** |
| --- | --- |
| **OOF SMAPE** | ≈ (Reported dynamically during runtime; typically < 50%) |
| **Cross-validation scheme** | 5-fold |
| **Transformation** | log ↔ exp for stability and interpretability |

Each sub-model learned complementary signals—LightGBM captured structured numeric relations, CatBoost modeled nonlinear dependencies, and the neural MLP generalized across high-dimensional embeddings.

**4.2 Final Blending Strategy**

Final predictions were computed as:

The exponentiated results were clipped to non-negative values and saved for submission.

**5. Conclusion**

The proposed **multimodal ensemble framework** effectively integrates textual semantics, visual cues, and numeric attributes for price prediction.  
Key takeaways include:

* Leveraging **pre-trained embeddings** significantly enhances feature representation without manual text or image annotation.
* Combining **boosting algorithms** and **neural networks** achieves a robust balance between interpretability and accuracy.
* Efficient **caching** of embeddings reduces runtime and improves experimental reproducibility.

The final model demonstrates strong out-of-fold performance and represents a scalable approach to multimodal retail price estimation.

**Appendix**

**Code Artifacts**

* GitHub Repository: <https://github.com/chsvhemanth/Amazon_ML_Challenge>