ML Challenge 2025: Smart Product Pricing Solution Report

Team Name: Ctrl + Alt + Learn
Team Members: BHAGYAWANTH(Leader), JONAN PURO, S DHIRAJ, TANYA GARG
Submission Date: October 2025

---

1. Executive Summary

We designed a multimodal pricing predictor that learns from product metadata, textual descriptions, and image features using a **Supervised Autoencoder + Stacking Ensemble**. The system extracts deep image representations, aligns them with structured and textual data, and fuses them through ensemble learning for accurate price estimation.

2. Methodology Overview

2.1. Pipeline Flow

Our end-to-end pipeline follows five key stages:

1. **Data Preprocessing:** Clean metadata, generate text embeddings (e.g., Sentence-BERT), and extract 2048-D CNN image features.
2. **Supervised Autoencoder Training:** Image embeddings are compressed into a 512-D bottleneck representation that learns both visual reconstruction and price alignment.
3. **Feature Fusion:** Structured, textual, and autoencoded image vectors are concatenated into a single standardized feature matrix.
4. **Model Training:** LightGBM, XGBoost, and MLP regressors are trained in parallel using 5-fold cross-validation, each capturing different feature dynamics.
5. **Stacking Ensemble:** Out-of-fold predictions from the base models form meta-features, which train a LightGBM meta-model to produce final prices.

Conceptually: The pipeline first converts all modalities into numeric representations, learns high-level joint features via autoencoding, and then aggregates model opinions through stacking to minimize prediction bias.

3. Model Architecture

3.1. Supervised Autoencoder Module

The autoencoder converts raw 2048-D image embeddings into meaningful compressed vectors. **Architecture:**

- Encoder: Linear layers (2048→1024→512, ReLU).
- Decoder: Reconstructs features (512→1024→2048).
- Regressor Head: Linear(512→1) predicting log(price).
- Joint Loss: $L = 0.5L_{recon} + 0.5L_{reg}$ (MSE).

This ensures the 512-D bottleneck captures both visual content and pricing cues, creating supervised embeddings ready for fusion.

3.2. Feature Fusion and Standardization

After training, embeddings are merged:

$$X_{full} = [X_{structured}|X_{text}|X_{image(bottleneck)}]$$

Each block is normalized separately using `StandardScaler`. This unified feature space ensures balanced influence of all modalities.

3.3. Stacking Ensemble Learning

Base Models:

- LightGBM (lr=0.05, 31 leaves)
- XGBoost (max depth=5, lr=0.05)
- MLP (2 hidden layers: 128–64, ReLU)

Their predictions are used as new features for the meta-model (LightGBM with 300 estimators, lr=0.03), which learns how to optimally combine them. This two-level architecture stabilizes errors and boosts prediction robustness.

4. Model Performance

Validation Metrics: RMSE = 0.2758, R^2 = 0.9143, SMAPE = 21.74%. **Per Model RMSE:** LightGBM 0.3165, XGBoost 0.3361, MLP 0.3499. **Per-Fold Results:**

Fold	LGBM	XGB	MLP
1	0.3169	0.3680	0.3552
2	0.3124	0.3590	0.3559
3	0.3168	0.3621	0.3520
4	0.3163	0.3606	0.3516
5	0.3165	0.3661	0.3540
Mean	**0.3158**	**0.3632**	**0.3537**

The ensemble achieved the lowest RMSE, demonstrating effective feature fusion and model diversity.

5. Conclusion

The pipeline sequentially transforms multimodal inputs into a unified representation, learns supervised visual embeddings, and blends predictions through stacking. The design leverages deep learning for feature compression and ensemble learning for robustness. Future work will extend to transformer-based multimodal encoders and uncertainty-aware regression.

Appendix

A. Code: Google Drive Repo

B. Models: Autoencoder weights, scalers, base and meta models.

C. Supplementary: Loss curves, feature importances, correlations.

Note: Compact one-page explanation of the Smart Product Pricing pipeline.