ML Challenge 2025: Smart Product Pricing Solution

Team Name: Test Data
Team Members: Kanahia, Anubhav Chandak, Animesh Tripathy, Kaustubh Kumar
Submission Date: 13th October 2025

1 Introduction

This study focuses on predicting product prices using both textual and visual information. After performing extensive data cleaning and preprocessing, four different modeling approaches were evaluated. The final submission leverages an ensemble of these models to achieve improved predictive performance.

2 Methodology Overview

2.1 Exploratory Data Analysis & Data Cleaning

Critical observations from Exploratory Data Analysis (EDA) and data cleaning informed the modeling strategy:

- **Price Skewness:** The 'price' distribution was highly skewed (Skewness: 13.768, Max: 2796.00). This necessitated a **log(1 + price) transformation** for model training to normalize the target and align with the SMAPE metric.
- **Duplicate Handling Strategy:** Exact duplicates (same content and image) had their prices **averaged** and one representative record was kept. Other forms of duplicates (e.g., same content, different image) were retained, as they might represent genuine product variations or richer context.
- **Outlier Handling:** Price outliers above the **99.6$^{\text{th}}$** percentile (199.99) were removed, resulting in the removal of **0.39%** of the data. This clipping stabilized training by significantly reducing skewness (**-80.4%**) and standard deviation (**-24.7%**) (See Table).

Table 1: Price Distribution Before and After Clipping (99.6$^{\text{th}}$ Percentile)

Statistic	**Original (Post Duplicate Handling)**	**After Clipping**	**Change (%)**
Count	74,944	74,648	-0.39
Max Price	2796.00	199.99	-92.8
Skewness	13.768	2.703	-80.4

2.2 Solution Strategy

- **XGBoost Approach (M1):** First, we established a strong baseline by analyzing the relationship between price and catalog content. We used an XGBoost model with text embeddings and engineered features to gauge the predictive power of textual data alone.
- **EfficientNet-BERT Approach (M2):** To incorporate visual information, we developed a dual-encoder model using lightweight, yet powerful, pre-trained backbones: **EfficientNet-B4** (∼19M params) for image embeddings and **BERT-base-uncased** (∼110M params) for text modeling. This allowed us to assess the value of combining both modalities.
- **OpenAI CLIP ViT-L/14 Approach (M3):** To benchmark the performance of another large-scale Vision-Language Model (VLM), we fine-tuned **"openai/clip-vit-large-patch14"** on the same dataset setup. This model achieved a validation SMAPE of **41.97**%.

- **LAION-CLIP ViT-H-14 Approach (M3):** Finally, we leveraged a large, pre-trained multi-modal model, **"laion/CLIP-ViT-H-14-laion2B-s32B-b79K"**. Its powerful zero-shot and fine-tuning capabilities on joint image-text representations provided a significant performance uplift, leading to its selection as our final model.

3 Model Architecture

M2. EfficientNet-BERT Dual-Encoder

Model M2 uses `bert-base-uncased` for text (up to 256 tokens) and `efficientnet_b4` for images (380×380). Image and text features are combined using a Multihead Attention block, followed by a multi-layer regression head to make predictions.

M3. OpenAI CLIP ViT-L/14 (VLM)

Model M3 uses `openai/clip-vit-large-patch14` to get embeddings for text and images. The vision-language encoder is fine-tuned, and a multilayer regression head predicts the output from the combined embeddings.

M4. LAION-CLIP ViT-H-14 (VLM)

Model M4 uses the `laion/CLIP-ViT-H-14` encoder to create 1024-dimensional embeddings for both text and images. The embeddings are joined and passed through a deep regression head to produce the final prediction.

4 Model Performance

All models were trained using **np.log1p(price)** as the target with **SMAPELoss**. The validation split was **15**%.

Table 2: Cross-Validation Performance Summary

Model Architecture	Best SMAPE	Validation MAE	Validation RMSE
XGBoost (M1)	**53.78**%	N/A	N/A
EfficientNet-BERT (M2)	**43.17**%	9.67	21.04
OpenAI CLIP ViT-L/14 (M3)	**41.97**%	9.56	21.94
LAION-CLIP ViT-H-14 (M4)	**40.55**%	9.48	20.46

5 Conclusion

Following a comprehensive evaluation of four distinct approaches, we adopted an ensemble strategy using GridSearch, combining models M2, M3 and M4 to construct the final predictive model.

6 Appendix

6.1 Code Artefacts

- **Code Repository:** Google Drive Folder (Code and Resources)

6.2 Hugging Face Model References

The following pre-trained models from the Hugging Face Hub were utilized in our experiments:

- **Sentence Transformer (M1):** `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`
- **BERT Base Uncased(M2):** `https://huggingface.co/bert-base-uncased`
- **Efficientnet B4 (M2):** `https://huggingface.co/google/efficientnet-b4`
- **OpenAI CLIP ViT-L/14 (M3):** `https://huggingface.co/openai/clip-vit-large-patch14`
- **LAION-CLIP (M4):** `https://huggingface.co/laion/CLIP-ViT-H-14-laion2B-s32B-b79K`