Amazon ML Challenge Finale

SPAM_LLMs

Manav Jain
IIT (ISM) Dhanbad

Prasanth Naidu Karaka
IIT (ISM) Dhanbad

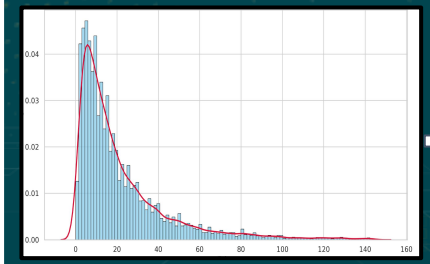
Alok Raj
IIT (ISM) Dhanbad

Rudraksh Sachin Joshi
IIT (ISM) Dhanbad

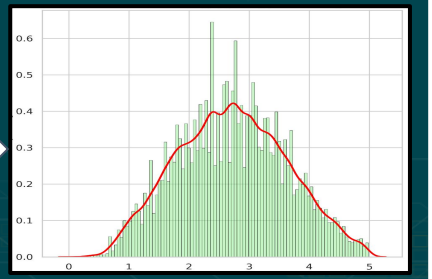
Qualitative & Quantitative Data Analysis

Distribution

Frequency Distribution of Raw Price (0-99th Percentile)



Frequency Distribution of Log Transformed Price (0-99th Percentile)



Original Price Data (Up to the 99th Percentile)
- Number of Values: 74,250
- Mean: 21.63
- Variance: 511.07

Original Price Data (Beyond the 99th Percentile)
- Number of Values: 750
- Mean: 223.04
- Variance: 20,672.09

Log-Transformed Price Data (Up to the 99th Percentile)
- Number of Values: 74,250
- Mean (log1p(Price)): 2.71
- Variance: 0.83

Log-Transformed Price Data (Beyond the 99th Percentile)
- Number of Values: 750
- Mean (log1p(Price)): 5.33
- Variance: 0.12

Data Variance

Visual Duplication Variance:

Price 1 : 39.41$
Price 2 : 29.07$
+
Varied catalogs

Textual Duplication Variance:

PAPYRUS Everyday Card

Repeated Instances (Train) : 21
Min price : 9.97$
Max price : 20.495$
Mean price : 16$

Severe positive skew → Applied log1p transform + normalization

Pre-Processing

- Catalog Content Strip: Product Description, Bullet Points, Unit, Value
- Textual Preprocessing: Emoji removal, numbers to text conversion.
- Unit Mapping Standardization: gram, gm, grams -> gm; case removal, grouping
- Unknown Units: Mapped to NA (Others)
- Prioritization: Product Description or top 5 bullet points
- Training Data Handling: Empty value processing, log normalization of price, label encoding of units

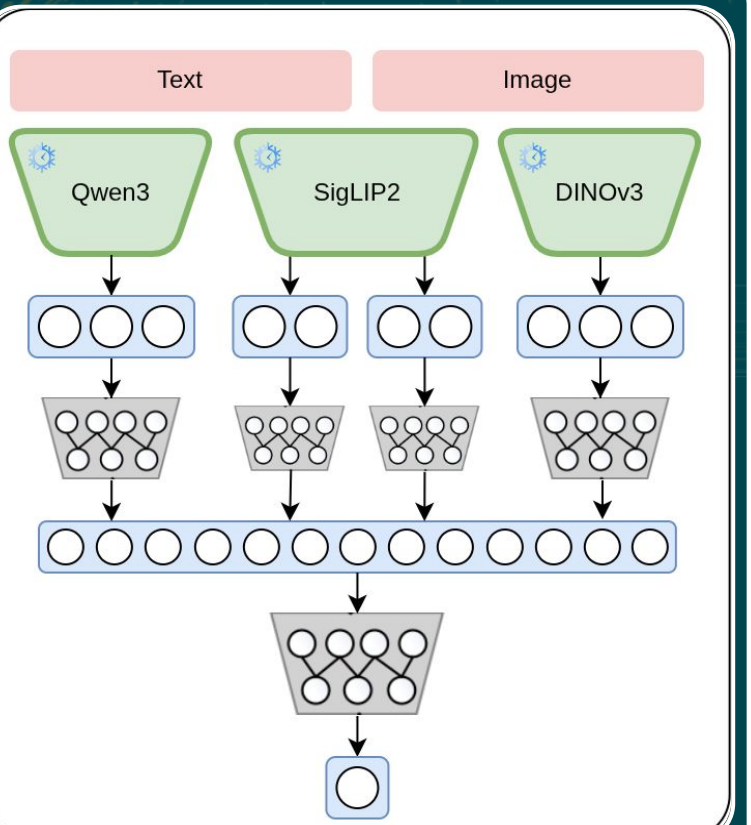
Ideation

Use Image Features, Text Features, Fused Image-Text Features, and Processed Tabular Data for training & indexing (MLPs, KNNs, LGBMs, Transformers)

Amazon ML Challenge Finale

Our Model : Architecture & Loss

Architecture



Loss Function

$$L_{\text{MSE-Log}} = \frac{1}{N}\sum_{i=1}^{N}(\log(y_i+1) - \hat{y}_i)^2$$

Latency

Embedding Generation:- 84 ms/sample
MLP head:- 30 µs/sample

*Benchmarked on P100 GPU

Concept

Input Towers: Each of the 4 embedding types (Image, Text, Text+Image) is processed by its own feed-forward tower (Linear → ReLU → BatchNorm → Dropout).
Purpose: Towers project high-dimensional embeddings into a shared 512-dimensional space, learning relevant features from each modality.
Fusion: Outputs of all 4 towers are concatenated into a 2048-dimensional unified feature vector.
Regression Head: The fused vector is passed through an MLP to predict the final price.
Advantage: Allows learning of specialized compact representations per modality before combining them for the final prediction.

Parameters

Text – 4B, Image – 0.8B, Text+Image – 2B, MLP – 8.9M x 5 (5-Fold)

Previous Approaches

Tree-Based Models: LGBM and Beyond

Parsed catalog content to extract meaningful features.
Incorporated previously generated embeddings alongside the new features.
Applied regression models combining features and embeddings.
Approach with combined features and embeddings was unsuccessful.

KNN-Based Feature Extraction

Used FAISS-based KNN for finding nearest neighbors of embeddings.
Applied to enhance feature representation with similar embeddings.
Due to small-scale dataset, performance did not meet expectations.

Cross-Attention Integration

Planned to replace separate networks with cross-attention between modalities.
Aimed to improve performance by integrating information across modalities.
The approach did not yield the expected improvements in performance. Needs further experimentation.

MinMax + Log Loss

Applied MinMax scaling to targets (0,1) and used BCE Loss to improve SMAPE metric

$$\tilde{y}_i = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}}$$

$$\mathcal{L}_{\text{BCE-minmax}} = -\frac{1}{N}\sum_{i=1}^{N}\left[\tilde{y}_i \log(\hat{y}_i) + (1-\tilde{y}_i)\log(1-\hat{y}_i)\right]$$

Expansion & Future Work

Embeddings Expansion into KNN

Post-model correction using nearest neighbors in embedding space.
Each item uses its embedding and predicted price; the model refines prediction based on local neighbor residuals.
Mechanism:
- Find k nearest neighbors within a price band x.
- Compute a weighted residual Δ from those neighbors.
- Adjust the predicted price by α·Δ.

$$\hat{y}_q^* = \hat{y}_q + \alpha \frac{\sum_{j \in N_k(z_q)} e^{-\lambda d(z_q, z_j)} (y_j)}{\sum_{j \in N_k(z_q)} e^{-\lambda d(z_q, z_j)}}$$

Controls: k, x, λ, α → locality / tolerance / decay strength.
Smooths inconsistencies, improves calibration in sparse regions.

Cross Attention & BCE-Loss

Refine cross-attention integration to improve information sharing across modalities.
Enhance MinMax + BCE loss formulation for better SMAPE stability.
Explore hybrid loss functions to improve convergence.
Conduct further experiments to evaluate performance across diverse modalities.

Hyperparam. tuning & Arch. Search

Richer embedding models, different configs of MLP head to be explored, like adapter depth, adapter dim., etc.

THANK YOU !!

- Team SPAM_LLMs