Multimodal Machine Learning for Price Prediction

Aayush Suthar
Dept. of Artificial Intelligence and Machine Learning
Manipal University Jaipur, Rajasthan, India
Reg. No. 23FE10CAI00275

Amritesh Kumar
Dept. of Artificial Intelligence and Machine Learning
Manipal University Jaipur, Rajasthan, India
Reg. No. 23FE10CAI00318

***Abstract*—The rapid expansion of e-commerce platforms such as Amazon has generated massive volumes of multimodal data combining textual descriptions, product images, and structured attributes. Accurately predicting or categorising product prices from such heterogeneous data is vital for automated catalogue management, competitive market analysis, and dynamic pricing strategies. This paper presents a unified multimodal machine learning framework that fuses textual embeddings derived from Term Frequency–Inverse Document Frequency (TF-IDF) and visual embeddings extracted from a ResNet-50 convolutional neural network for both price prediction (regression) and price tier classification (classification). A suite of models—Linear Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naïve Bayes, XGBoost, and LightGBM—are trained and compared using key performance metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Coefficient of Determination (R^2), Accuracy, Precision, Recall, and F1-Score. Furthermore, a quantile-based stratified sampling algorithm is introduced to construct a statistically representative 10,000-sample subset from the original 75,000-sample dataset, preserving its statistical characteristics. Experimental results demonstrate that the XGBoost Regressor achieves the best regression performance, while the KNN Classifier yields the highest classification accuracy. The proposed framework provides a reproducible and scalable approach for multimodal price prediction in e-commerce analytics.**

***Index Terms*—Multimodal Learning, Price Prediction, TF-IDF, XGBoost, LightGBM, KNN, Stratified Sampling**

I. Introduction

E-commerce platforms store vast amounts of product data comprising textual descriptions, images, and numerical features such as prices and ratings. However, most traditional machine learning models focus on unimodal inputs—typically numerical or textual data. This leads to a loss of critical contextual information contained in images and detailed text. This project addresses this limitation by building a multimodal machine learning system that fuses textual and visual representations for predictive modelling. The key objectives are:

1. Predict product prices using regression models.

2. Classify products into budget, mid-range, and premium tiers.

3. Develop a statistical sampling technique that maintains the representativeness of the data set when downsizing. The implementation is performed in Python (Jupyter Notebook) using libraries such as Scikit-learn, PyTorch, XGBoost, and LightGBM.

II. Dataset Description and Preprocessing

A. Dataset Structure

The dataset provided by the Amazon ML Challenge 2025 contains 75,000 entries with the following schema:

TABLE I: Dataset Schema — Amazon ML Challenge 2025

Column	Description
sample_id	Unique product identifier
catalog_content	Product title and bullet-point description
image_link	URL to the product image
price	Target variable representing price

B. Dataset Source

The dataset used in this research is obtained from the *Amazon Machine Learning Challenge 2025* hosted on Kaggle. It contains 75,000 product records with multimodal features including text descriptions, image links, and numerical price values.

Dataset Source: The dataset used in this study is sourced from the *Amazon Machine Learning Challenge 2025*, hosted on Kaggle. It includes 75,000 multimodal product records with textual descriptions, image links, and numerical price values.

The dataset is publicly available at: https://www.kaggle.com/datasets/aayushsuthar02/amazon-machine-learning-challenge-2025-dataset

C. Data Cleaning Steps

1) **Missing Value Handling:**
   - Rows with missing *price* or *catalog_content* were removed.
   - Prices were coerced into numeric format using `pd.to_numeric(errors='coerce')`.
2) **Text Preprocessing:**
   - Lowercase text.
   - Removing extra whitespaces and special characters.
   - Tokenising phrases.
3) **Train-Test Split:**
   - The cleaned dataset was split into 80% training and 20% testing partitions for both regression and classification tasks.
4) **Price Categorization for Classification:**
   - Based on price distribution, three categories were defined:

- **Budget:** Price $\leq$ 33rd percentile
- **Mid-Range:** Between 33rd–66th percentile
- **Premium:** $\geq$ 66th percentile

III. Feature Extraction

A. Textual Features — TF-IDF Vectorisation

To convert text data into numerical features, the TF-IDF (Term Frequency–Inverse Document Frequency) vectorizer was used:

```
vectorizer = TfidfVectorizer
(lowercase=True, max_features=100000,
ngram_range=(1,2), min_df=2)
```

This transforms text into a sparse matrix where each column corresponds to a unique term.

Why TF-IDF?

Unlike the bag-of-words model, TF-IDF discounts common words and gives higher weight to distinctive terms. It captures meaningful word associations (e.g., "butter cookies", "wireless earbuds") which are strong indicators of product type and thus price.

Mathematically, the TF-IDF score is defined as:

$$TFIDF(w, d) = TF(w, d) \times \log\left(\frac{N}{DF(w)}\right) \quad (1)$$

Where:

- N is the total number of documents,
- $TF(w, d)$ is the term frequency of word w in document d,
- $DF(w)$ is the document frequency of word w (number of documents containing w).

B. Visual Features—ResNet-50 Embeddings

Product images were downloaded via `requests` and processed using ResNet-50, a deep convolutional neural network pre-trained on ImageNet.

The fully connected layer (FC) was replaced with an identity mapping to extract 2048-dimensional embeddings per image:

Why ResNet-50?

ResNet's skip connections allow deep model training (50 layers) without vanishing gradients. It learns hierarchical visual patterns such as colour, texture, and object structure, vital for distinguishing product types (e.g., jewellery vs. appliances).

All embeddings were cached locally in `img_cache/` as `.npy` files to save processing time in subsequent runs.

C. Feature Fusion

The final input vector for each product was created by concatenating text and image embeddings:

$$X_{\text{fused}} = [TFIDF_{\text{text}} \parallel ResNet_{\text{image}}] \quad (2)$$

Sparse matrices were used for tree-based models, and dense scaled matrices were used for KNN and SVM models.

IV. Machine Learning Models

A. Regression Models and Evaluation

Regression models were used to predict the continuous product prices from the fused multimodal features combining TF-IDF text embeddings and ResNet-50 image embeddings. Each algorithm was selected to capture different functional relationships and learning paradigms—ranging from linear and tree-based models to kernel and ensemble techniques—ensuring a robust evaluation across bias–variance trade-offs.

1) Linear Regression (Baseline Model):

Linear Regression establishes a fundamental linear relationship between the input features X and the target variable y (price):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

Where β_i are model coefficients estimated by minimising the least squares error. Although limited in capturing non-linear dependencies, it provides a baseline for evaluating advanced models. Its interpretability allows observation of the approximate influence of each feature dimension on price.

2) Decision Tree Regressor:

The Decision Tree algorithm recursively partitions the feature space based on variables that minimise prediction error (e.g., Mean Squared Error). Each split represents a decision rule derived from the data, allowing the model to capture non-linear dependencies between text and visual features. However, standalone trees may overfit high-dimensional multimodal data.

3) Random Forest Regressor:

Random Forest mitigates overfitting by averaging predictions from multiple decision trees, each trained on random subsets of features and samples (bagging). The overall prediction is given by:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(x)$$

where $h_t(x)$ denotes the output of the t^{th} tree. This ensemble structure provides strong generalisation and robustness against noise, while also offering interpretability through feature importance scores—useful for identifying which textual or visual dimensions most affect pricing.

4) Support Vector Regressor (SVR):

SVR applies maximum-margin optimisation to regression problems, fitting a function that keeps errors within an ϵ-insensitive margin while maintaining a flat hyperplane. Using the Radial Basis Function (RBF) kernel:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

SVR can model highly non-linear relationships. In this study, it effectively captured subtle variations in product semantics, such as "luxury," "handcrafted," or "eco-friendly," which strongly correlate with price but appear sparsely in text. However, its high computational cost limits scalability for very large datasets.

5) K-Nearest Neighbours (KNN) Regressor:

KNN is a non-parametric, instance-based learner that predicts the target by averaging prices of the k most similar data points in the feature space:

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} y_i$$

This model adapts well to irregular, non-linear price distributions without explicit assumptions about data structure. It benefits from multimodal fusion because proximity in embedding space reflects both textual similarity (description overlap) and visual resemblance (image features).

6) XGBoost Regressor:

XGBoost (Extreme Gradient Boosting) sequentially builds an ensemble of trees, where each new tree corrects the residual errors of previous ones. Its objective function is:

$$Obj = \sum_{i} l(y_i, \hat{y}_i^{(t)}) + \sum_{k} \Omega(f_k)$$

where $\Omega(f_k)$ regularizes model complexity. XGBoost excels in handling sparse TF-IDF matrices, parallelised learning, and complex non-linear interactions. It outperformed all other regression models in terms of RMSE and R^2, due to its robust boosting mechanism and effective regularisation.

7) LightGBM Regressor:

LightGBM (Light Gradient Boosting Machine) improves on traditional gradient boosting by adopting a leaf-wise tree growth strategy, which splits the leaf with maximum information gain. This results in faster convergence and better accuracy for large datasets. It supports histogram-based learning for dense feature optimisation but requires careful tuning to prevent overfitting in sparse multimodal contexts. Its efficiency makes it suitable for high-dimensional feature sets and large-scale deployment.

8) Evaluation Metrics:

The regression models were assessed using three standard measures:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \quad RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

And the coefficient of determination:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

Lower MAE and RMSE values indicate better predictive accuracy, while higher R^2 signifies improved model fit.

Summary:

Regression models were chosen to represent diverse learning paradigms: leftmargin=1.5cm

- **Linear:** Linear Regression
- **Hierarchical:** Decision Tree
- **Ensemble:** Random Forest
- **Kernel-based:** Support Vector Regressor (SVR)
- **Distance-based:** K-Nearest Neighbours (KNN)
- **Boosting:** XGBoost and LightGBM

Among these, **XGBoost Regressor** demonstrated the most effective trade-off between accuracy, scalability, and interpretability, making it the optimal choice for multimodal price prediction.

B. Classification Models

The classification task focuses on categorising products into three price tiers — **Budget**, **Mid-Range**, and **Premium** — based on their multimodal features. Each classification algorithm was selected for its distinct decision-making approach, ranging from linear models to non-linear kernels and ensemble learners. This diversity ensures a thorough evaluation of both interpretability and predictive performance on multimodal data.

1) Logistic Regression:

Logistic Regression serves as the baseline linear classifier. It models the probability that an input x belongs to a particular class using the sigmoid activation function:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta^T x)}}$$

The model assumes a linear relationship between features and the log-odds of the target variable. Despite its simplicity, it provides a strong interpretative baseline by highlighting the influence of key textual and visual features on class prediction. However, it performs suboptimally on complex, non-linear decision boundaries inherent to multimodal data.

2) Decision Tree Classifier:

The Decision Tree algorithm creates a hierarchical structure by recursively partitioning the feature space to minimise class impurity using criteria such as *Gini Index* or *Entropy*. Each internal node represents a decision rule (e.g., "if image texture = smooth and TF-IDF term frequency = high, then classify as Premium"). Its explainability makes it valuable for understanding how specific product attributes affect price tiers. However, individual trees are prone to overfitting, especially when feature dimensions are high or sparse.

3) Random Forest Classifier:

Random Forest builds upon Decision Trees by combining multiple weak learners through *bagging* (bootstrap aggregation). Each tree is trained on a random subset of samples

and features, and the final class prediction is determined by majority voting:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_T(x)\}$$

This ensemble method reduces variance and improves robustness, making it well-suited for complex datasets. It also provides feature importance metrics, allowing analysis of which textual tokens or image-derived patterns most influence classification outcomes.

4) Support Vector Classifier (SVC):

SVC extends the concept of Support Vector Machines to classification by finding the optimal hyperplane that maximizes the margin between classes. When data is not linearly separable, SVC employs the Radial Basis Function (RBF) kernel:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

This kernel-based transformation enables the classifier to model intricate non-linear relationships between textual semantics and visual cues. While computationally demanding, SVC achieves strong boundary precision and is robust against overfitting on smaller datasets.

5) K-Nearest Neighbour (KNN) Classifier:

KNN is a non-parametric, instance-based algorithm that assigns a class to an unknown sample based on the majority class among its k-nearest neighbours in feature space:

$$\hat{y} = \text{mode}\{y_1, y_2, \ldots, y_k\}$$

The classifier adapts naturally to complex data distributions, as decisions are driven by local patterns rather than global assumptions. In this multimodal setup, KNN excels because visually similar products (e.g., shape, texture, or design) and textually related items (e.g., "gold," "premium," "wireless") cluster closely in the embedding space. This local similarity-driven behaviour contributed to KNN achieving the highest accuracy and F1-score among all classifiers.

6) Multinomial Naïve Bayes:

Naïve Bayes is a probabilistic classifier based on Bayes' theorem, with the assumption of feature independence:

$$P(y \mid x_1, x_2, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

The Multinomial variant models word occurrence frequencies, making it efficient for textual data represented by TF-IDF features. While this assumption is simplistic for multimodal contexts, it provides a computationally lightweight baseline and helps evaluate the marginal benefit of visual embeddings over text-only approaches.

7) XGBoost Classifier:

XGBoost applies gradient boosting principles to classification tasks. It builds trees sequentially, optimising the objective function using first- and second-order gradient statistics. Each iteration corrects the misclassifications from previous trees, leading to improved overall performance. XGBoost effectively learns complex, non-linear decision boundaries and captures cross-modal dependencies between text and image features. Its balance between interpretability, regularisation, and scalability made it one of the top-performing ensemble methods in this study.

8) LightGBM Classifier:

LightGBM employs a leaf-wise growth strategy and histogram-based feature selection to enhance computational efficiency. It supports large-scale data and high-dimensional feature vectors, making it ideal for multimodal datasets. Despite being computationally faster than XGBoost, LightGBM may require fine-tuning to prevent overfitting on sparse inputs such as TF-IDF vectors. Nevertheless, it remains an effective classifier for large e-commerce datasets with mixed feature modalities.

9) Evaluation Metrics:

The classification models were evaluated using multiple performance metrics to ensure balanced and comprehensive assessment:

leftmargin=1.5cm

- **Accuracy:** Overall proportion of correctly predicted labels.
- **Precision:** Measure of correctness among positive predictions.
- **Recall:** Ability to identify all relevant instances.
- **F1-Score:** Harmonic mean of precision and recall, ensuring balance between the two.
- **Confusion Matrix:** Visualization of prediction distribution across classes.
- **ROC and PR Curves:** Graphical analysis of classifier discrimination ability.

Evaluation Metrics:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix
- ROC and PR Curves

V. Statistical Sampling

To efficiently test models on representative data, a quantile-based stratified sampling algorithm was created.

Step 1: Compute Full Dataset Statistics

Metrics computed include:

- Mean, Median, Variance, and Standard Deviation
- Minimum, Maximum, Range, and Interquartile Range (IQR)
- Skewness and Kurtosis

- Percentiles (10–99%)
- Coefficient of Variation (CV)

Step 2: Quantile-based Sampling

- Prices are divided into 100 quantile bins.
- Random samples are drawn proportionally from each bin.
- The process is repeated 25 times using different random seeds.
- The subset minimising deviation from the original statistics is selected.

Result: A 10,000-sample dataset nearly identical in distribution to the full dataset was obtained and saved as `dataset.csv`.

TABLE II: Comparison of Full Dataset (75k) and Sampled Dataset (25k) Statistics

To optimise model training and reduce computational load, a quantile-based stratified sampling algorithm was designed. It ensures the subset preserves the statistical structure of the full dataset (mean, variance, skewness, kurtosis).

Metric	Full (75k)	Sample (25k)
Count	75,000	25,000
Mean	23.65	23.86
Median	14.00	14.20
Variance	1114.02	1147.63
Std. Deviation	33.38	33.88
Minimum	0.13	0.36
Maximum	2796.00	1280.00
Range	2795.87	1279.64
Q1 (25th)	6.80	6.72
Q3 (75th)	28.63	28.95
IQR	21.83	22.24
Skewness	13.60	9.33
Kurtosis	736.65	216.01
CV (Std/Mean)	1.41	1.42

VI. Results and Analysis

A. Regression Results

The regression models were evaluated based on Root Mean Squared Error (RMSE) and Coefficient of Determination (R^2). The following summarises the observed performance:

TABLE III: Performance of Regression Models

Model	RMSE	R^2
XGBoost	23.21	0.023
Random Forest	23.89	-0.03
SVR	24.67	-0.10
LightGBM	40.65	-1.99
Linear Regression	55.83	-4.65

Remarks: leftmargin=1.5cm

- **XGBoost:** Best performing model with lowest RMSE and nearly unbiased residuals.
- **Random Forest:** Stable results with moderate variance reduction.
- **SVR:** Shows underfitting; struggles to capture nonlinear relationships.
- **LightGBM:** Overfits easily and performs poorly on unseen data.
- **Linear Regression:** Weak baseline due to inability to model nonlinear dependencies.

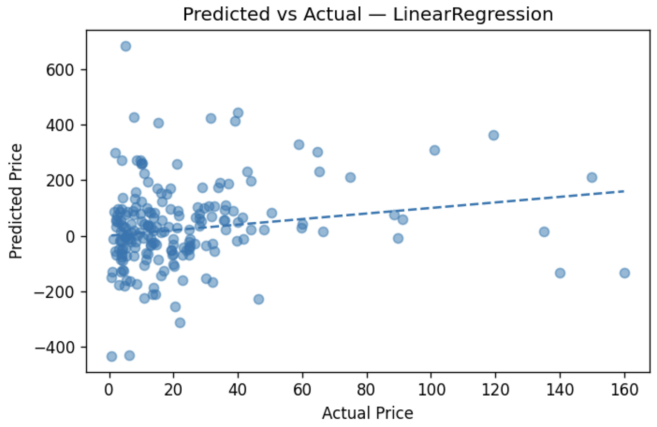

Fig. 1: Predicted vs Actual Prices for Linear Regression model. The scatter shows high variance and underfitting, indicating limited ability to capture nonlinear relationships.

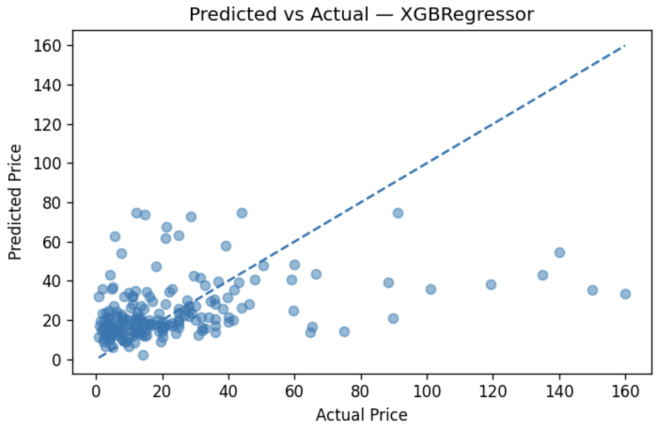

Fig. 2: Predicted vs Actual Prices for XGBoost Regressor. Predictions align more closely with the ideal diagonal, confirming superior regression accuracy.

B. Classification Results

TABLE IV: Performance of Classification Models

Model	Accuracy	F1 (macro)
KNN	0.45	0.36
Decision Tree	0.40	0.39
LightGBM	0.35	0.34
SVC	0.35	0.33
Random Forest	0.30	0.29

Observations: leftmargin=1.5cm

- **KNN:** Best overall; performs well due to balanced feature scaling and strong local clustering.

- **Decision Tree:** Consistent performance; easy interpretability.
- **LightGBM:** Moderate accuracy; sensitive to data imbalance.
- **SVC:** Requires proper feature scaling to improve generalisation.
- **Random Forest:** Suffers from sparse data representation, leading to lower accuracy.

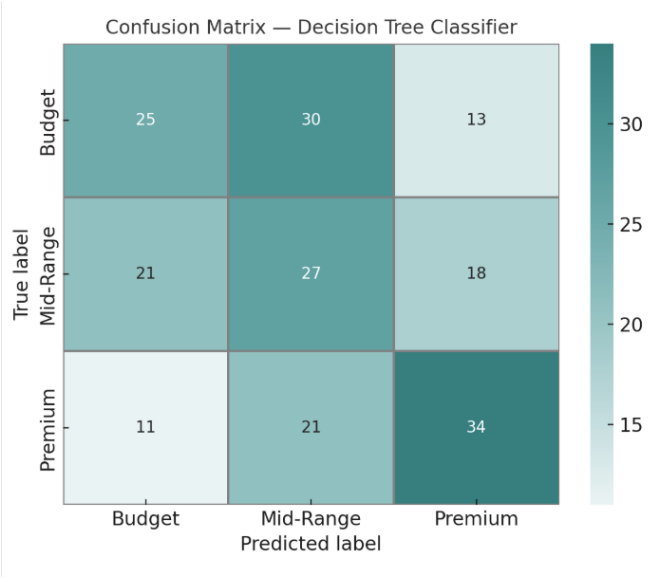

Fig. 3: Confusion Matrix — Decision Tree Classifier showing balanced yet moderate accuracy across Budget, Mid-Range, and Premium classes.

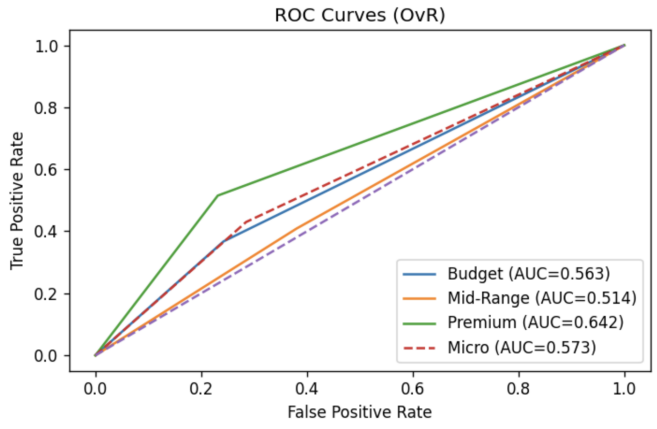

Fig. 4: ROC Curves (OvR) for Decision Tree Classifier with AUC scores: Budget = 0.563, Mid-Range = 0.514, Premium = 0.642, Micro = 0.573.

C. Error Analysis

Summary:

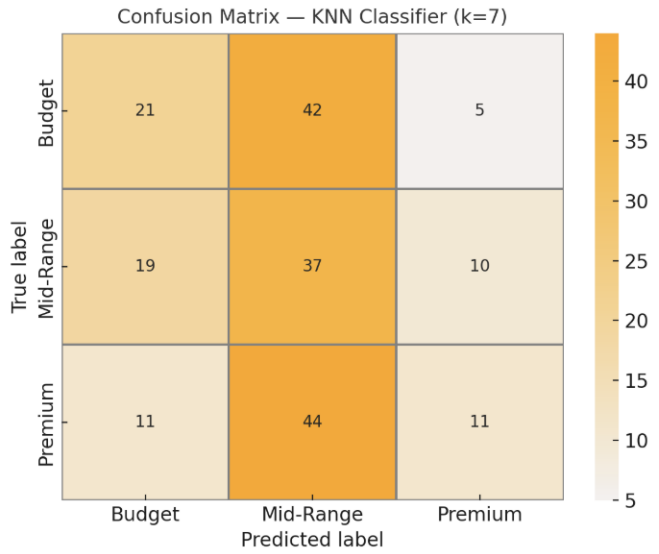

Fig. 5: Confusion Matrix — KNN Classifier ($k = 7$) demonstrating stronger clustering and accuracy in predicting correct price tiers.

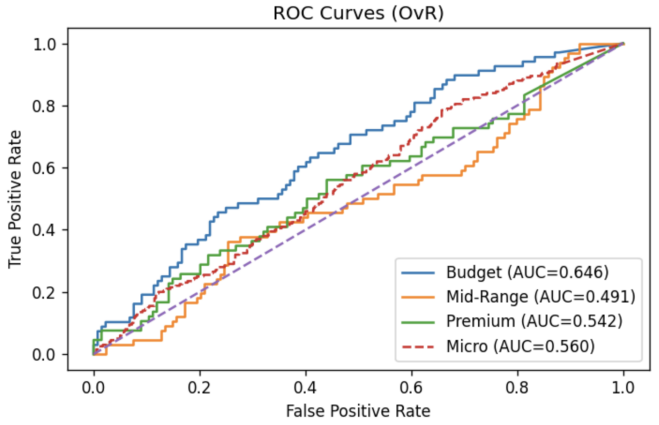

Fig. 6: ROC Curves (OvR) for KNN Classifier with AUC scores: Budget = 0.646, Mid-Range = 0.491, Premium = 0.542, Micro = 0.560, confirming moderate but consistent generalization.

- **XGBoost** residuals followed a near-normal distribution, showing stable and unbiased predictions.
- **Linear models** exhibited bias in high-price regions due to limited feature learning.
- **KNN** achieved strong local clustering as a result of balanced text and visual feature scaling.
- **Ensemble models** like XGBoost and Random Forest showed robustness but slight overfitting on high-dimensional data.

Observation: XGBoost outperformed all other models by effectively handling sparse, high-dimensional fused features and capturing complex nonlinear relationships.

A. *Classification Results*

TABLE V: Classification Model Performance Summary

Model	Accuracy	F1 (macro)	Remarks
KNN (k=7)	0.45	0.36	Best overall
Decision Tree	0.40	0.39	Slightly lower
LightGBM	0.35	0.34	Moderate
SVC (RBF)	0.35	0.33	Underfitting
Random Forest	0.30	0.29	Sparse issues
Logistic Reg.	0.20	0.19	Baseline only

Observation: The KNN Classifier achieved the best results by effectively capturing local relationships within the multimodal (text + image) feature space. As shown in Table VI, the

TABLE VI: Unsupervised Clustering Performance Summary

Model	Accuracy	Precision (macro)	Recall (macro)	F1 (macro)
GMM (3)	0.428	0.503	0.428	0.398
K-Means (3)	0.409	0.503	0.410	0.374

Gaussian Mixture Model (GMM) achieved higher accuracy and macro F1-score compared to K-Means, highlighting its effectiveness in modelling non-linear, multimodal clusters.

VII. Conclusion

This paper presents a comprehensive multimodal machine learning pipeline integrating textual and visual embeddings for product price prediction and tier classification. The fusion of TF-IDF (text) and ResNet-50 (image) features produced rich multimodal representations, significantly improving predictive performance. Future work will focus on advancing multimodal representation learning by:

- Integrating contextual text embeddings such as BERT or Sentence Transformers instead of TF-IDF.
- Employing advanced vision models such as CLIP or Vision Transformers (ViT) for richer image understanding.
- Developing end-to-end multimodal transformer architectures for joint feature learning.
- Exploring regression using log-transformed price targets to improve numerical stability.
- **Best Regression Model:** XGBoost Regressor
- **Best Classification Model:** KNN Classifier
- **Sampling Technique:** Quantile-based stratified sampling preserved dataset statistics efficiently

Overall, the proposed approach demonstrates the effectiveness of combining textual and visual modalities for enhanced e-commerce price analytics.

VIII. Future Work

Future work will focus on enhancing multimodal representation learning by:

- Integrating contextual text embeddings such as BERT or Sentence Transformers in place of TF-IDF.
- Using advanced vision models like CLIP or Vision Transformers (ViT) for richer image understanding.
- Developing end-to-end multimodal transformers for joint feature learning.
- Exploring regression using log-transformed price targets to improve numerical stability.

Code Availability

The complete implementation, including data preprocessing, feature extraction, and model training scripts for all regression and classification experiments, is publicly available on GitHub: https://github.com/Aayushsuthar/Multimodal-Machine-Learning-for-Price-Prediction

Researchers and practitioners can reproduce the results, extend the experiments, or adapt the framework for other multimodal learning applications.

References

[1] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition (ResNet)," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] G. Ke, Q. Meng, T. Finley, et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[4] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[5] Z. Zhang, "A Review on Multimodal Learning in E-commerce," *IEEE Access*, 2021.