A Comprehensive and Enhanced Machine Learning Framework for Retail Product Category Classification Using the DMart Dataset

Ayush Das
Reg. No. 12317700
Lovely Professional University
Phagwara, India

Patel Hashil Kumar
Reg. No. 12313359
Lovely Professional University
Phagwara, India

Shalini Kumari
Reg. No. 12312433
Lovely Professional University
Phagwara, India

Ayiti Ashritha
Reg. No. 12312108
Lovely Professional University
Phagwara, India

***Abstract*—In the rapidly evolving landscape of modern retail, the capability to autonomously organize and manage expansive product inventories has transitioned from a competitive advantage to an operational necessity. As global e-commerce platforms continue to scale at an unprecedented rate, the logistical challenge of accurately sorting thousands of heterogeneous products into granular categories becomes increasingly intricate. This research paper delineates the end-to-end development of a sophisticated machine learning pipeline specifically engineered to classify retail items. We utilize a real-world dataset from DMart, comprising 5,189 unique stock keeping units (SKUs) distributed across multiple diverse categories. Our methodology emphasizes a rigorous preprocessing regimen, incorporating advanced text normalization techniques, the synthesis of composite semantic features, TF-IDF vectorization for sparse data handling, and the statistical standardization of numeric attributes, culminating in the deployment of high-performance supervised learning models.

We implemented and systematically evaluated three distinct algorithmic approaches: Logistic Regression, Random Forest, and XGBoost. The experimental results indicated that while Logistic Regression served as a robust baseline with an accuracy of 0.9015, and Random Forest achieved a respectable 0.7944, XGBoost demonstrated superior predictive capability, attaining a peak accuracy of 0.9073. To provide granular insight into model performance, we present an exhaustive confusion matrix analysis, detailing the distribution of True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) for each specific class. The discussion extends beyond mere metrics to analyze the behavioral nuances of the models, the practical implications for retail operations, inherent limitations of the study, and strategic avenues for future enhancement. This extended work aims to provide a comprehensive academic foundation and actionable insights for researchers and industry practitioners alike.

***Index Terms*—Retail analytics, product classification, machine learning, TF-IDF, XGBoost, Logistic Regression, Random Forest, confusion matrix, supervised learning, text mining.

I. INTRODUCTION

The exponential growth of digital commerce has fundamentally transformed inventory management, resulting in product catalogs of immense magnitude and diversity. Major retail conglomerates such as DMart, Amazon, and Walmart are tasked with managing dynamic, fluid inventories where thousands of new items are introduced or updated daily. Each individual item requires precise placement within a multi-tiered hierarchical taxonomy to ensure that customers can intuitively browse, filter, and locate products. At this scale, manual classification by human operators is rendered obsolete due to the sheer volume, velocity, and variety of the incoming data streams.

Accurate product categorization is foundational to the digital user experience; it directly influences search engine visibility, powers personalized recommendation engines, and streamlines backend supply chain logistics. Conversely, misclassification can lead to significant friction in the customer journey, causing products to be buried deep within incorrect categories, thereby skewing sales analytics and ultimately eroding revenue. Consequently, there is an urgent, critical need for systematic, automated machine learning frameworks capable of parsing diverse product attributes to execute high-fidelity sorting.

Machine learning offers a scalable, data-driven solution to this challenge by learning complex patterns from historical catalog data to categorize new, unseen products with high confidence. However, real-world retail data is rarely pristine. It presents a unique set of challenges that complicate modeling:

- **Inconsistent Naming Conventions:** Product ti-

tles often contain significant noise, non-standard abbreviations, or promotional marketing text.
- **Variable Descriptions:** Detailed descriptions may be missing entirely, overly verbose, or inconsistent in formatting.
- **Complex Quantity Formats:** Essential information regarding weight, volume, or pack size appears in highly varied formats that require normalization.
- **Class Imbalance:** Inventory distributions are naturally skewed; some categories like "Groceries" are heavily populated, while niche categories have sparse representation.
- **Rich Metadata:** Auxiliary fields like breadcrumbs offer valuable context but require careful merging with other textual signals to avoid redundancy.

Leveraging the DMart dataset, which encapsulates these real-world imperfections, this study constructs and validates a classification model explicitly designed to navigate such complexity. This paper offers a detailed exposition of every stage of the pipeline, from raw data ingestion and cleaning to feature engineering and final model evaluation.

The primary contributions of this research include:
1. The architectural design of a holistic machine learning pipeline for mixed-mode (unstructured text and structured numeric) retail classification.
2. The implementation of rigorous text preprocessing and feature aggregation strategies specifically tailored for noisy retail data.
3. The application of TF-IDF vectorization to effectively manage sparse, high-dimensional textual feature spaces.
4. A comparative performance evaluation of three widely adopted machine learning algorithms.
5. A detailed, class-wise breakdown of predictive performance using confusion matrix metrics (TP, FP, FN, TN).
6. A comprehensive discussion addressing the trade-offs, limitations, and potential future research directions in this domain.

II. DATASET DESCRIPTION

This study utilizes a proprietary dataset sourced from DMart to represent the typical semi-structured data environment found in modern retail systems. Comprising 5,189 unique items, the dataset spans a wide array of heterogeneous categories including groceries, personal care products, household goods, and packaged foods. Each record is composed of several attribute columns that capture specific details about the product.

The primary feature set includes:
- **Name:** The product title, which serves as the primary identifier and often includes the brand, variant, and quantity.
- **Brand:** The manufacturer or umbrella brand name, which is frequently missing or unstandardized.
- **Price:** The standard listing price (MRP) of the item.
- **DiscountedPrice:** The actual selling price after applicable discounts, crucial for identifying promotional items.
- **Category (Target):** The broad, high-level class label that the model aims to predict.
- **SubCategory:** A more granular classification level, though often populated inconsistently.
- **Quantity:** A raw string describing the physical size or count (e.g., "500 gm", "1 kg", "Pack of 3").
- **Description:** Free-text fields detailing ingredients, usage instructions, or marketing copy.
- **BreadCrumbs:** The hierarchical navigation path used on the website, providing context.

Several intrinsic data quality challenges necessitated a robust preprocessing strategy:
1. **Missing Data:** Critical fields such as 'Brand' and 'Description' frequently contained null values, requiring systematic imputation strategies.
2. **Non-Standard Quantities:** Measurement units varied widely (litres, grams, packs, pieces), requiring conversion into a unified numeric schema.
3. **Noise:** Marketing buzzwords (e.g., "New!", "Best Value") and special characters cluttered the product names, potentially confusing the model.
4. **Class Imbalance:** Certain categories contained fewer than ten examples, introducing potential bias towards majority classes.
5. **Redundancy:** Metadata fields often repeated information found elsewhere, risking feature duplication.

Thoroughly understanding, cleaning, and structuring this dataset was a fundamental prerequisite for building a reliable and generalizable classification system.

III. PREVIOUS RESEARCH

The domain of retail product classification has matured significantly over the past two decades, evolving in parallel with broader advancements in natural language processing (NLP) and machine learning. Early legacy systems depended heavily on manual tagging or rigid, rule-based logic that matched specific keywords to pre-defined categories. While functional for small, static inventories, these deterministic methods were brittle and failed to adapt to the noisy, inconsistent, and evolving descriptions found in modern e-commerce catalogs.

A seminal development in the field was the introduction of the TF-IDF (Term Frequency-Inverse Document Frequency) weighting scheme by Salton and Buckley. This statistical method allowed unstructured text to be transformed into numerical vectors, identifying the importance of a word within a document relative to the entire corpus. In the retail sector, TF-IDF remains highly relevant because product titles are typically short, telegraphic, and rich in specific, discriminative keywords.

Joachims' influential work on Support Vector Machines (SVM) demonstrated the remarkable effectiveness of linear classifiers on high-dimensional text data. Linear models, including Logistic Regression, subse-

quently became the industry standard for text classification tasks. Their ability to handle thousands of sparse features efficiently while maintaining interpretability made them ideal for early large-scale text mining applications.

In the 2000s, ensemble learning methods like Random Forests gained popularity for their robustness and ability to handle non-linear relationships in structured tabular data. However, they often struggled with the extreme high dimensionality associated with sparse text vectors. The landscape shifted again with the advent of Gradient Boosting machines, specifically the XGBoost framework by Chen and Guestrin. XGBoost offered state-of-the-art performance by effectively capturing complex, non-linear interactions in both structured (price, weight) and unstructured (text) data, making it a preferred choice for mixed-attribute datasets.

More recently, deep learning architectures such as BERT and GPT have set new performance benchmarks by capturing deep semantic context and bidirectional relationships in text. However, these models are computationally expensive to train and deploy, often requiring massive datasets to fine-tune effectively. For medium-sized datasets like the DMart catalog, traditional methods like TF-IDF combined with XGBoost or Logistic Regression often provide a more practical "sweet spot," balancing high accuracy with computational efficiency and interpretability.

This research grounds itself in this rich academic lineage, strategically combining the efficiency of TF-IDF feature extraction with the predictive power of XGBoost and Logistic Regression to address the specific nuances and constraints of retail data classification.

IV. PREPROCESSING METHODOLOGY

Data preprocessing is the cornerstone of any effective machine learning classifier, particularly when the source data is noisy and unstructured. The DMart dataset required a comprehensive, multi-step cleaning pipeline to handle missing values, varied units, and sparse categories. This section outlines the specific, rigorous steps taken to prepare the data for modeling.

A. Handling Missing Values

Retail data is notoriously incomplete. Fields such as 'Brand' and 'Description' were frequently missing or left blank. Rather than discarding these rows—which would significantly shrink the dataset and potentially skew the class distribution—we adopted an imputation strategy where missing text fields were filled with empty strings. This ensured that the TF-IDF vectorizer could process every row without execution errors.

For numeric fields like Price and DiscountedPrice, missing values were imputed with zero. While a simple heuristic, this approach preserves the data point for analysis and provides a consistent baseline, preventing the model from failing on null inputs.

B. Quantity Normalization

The 'Quantity' column presented a significant challenge, containing mixed formats such as "500 gm", "1 kg", "Pack of 3", and "2 x 200 ml".

To render this feature usable for mathematical modeling, we employed a regex-based extraction method to isolate the primary numerical component:

$$Quantity\_num \leftarrow float(extractDigits(Quantity)) \quad (1)$$

This process creates a standardized numeric feature representing the approximate magnitude of the product. While it simplifies complex composite units, it provides a valuable scalar signal that helps the model distinguish between bulk items and single units (e.g., distinguishing a 5kg bag of rice from a 500g packet).

C. Category Balancing

Classes with extremely few samples (fewer than ten) create noise and prevent the model from learning generalizable decision boundaries. To mitigate this data sparsity issue, we aggregated these rare categories into a generic "Other" class. This consolidation ensures that every target class has sufficient representation to be learned effectively during cross-validation, thereby stabilizing the training process.

D. Text Field Construction

Textual content is the strongest predictor for product classification. However, relevant information is often scattered across the Name, Description, and Breadcrumbs columns. To maximize the semantic signal available to the model, we concatenated these columns into a single, comprehensive 'text' feature:

$$text \leftarrow Name + Description + BreadCrumbs \quad (2)$$

This aggregation ensures that the model has access to the full context of the product, capturing keywords that might appear in the breadcrumb path but not in the title, thus providing a richer semantic footprint for classification.

E. Label Encoding

Since machine learning algorithms require numerical input vectors, the categorical target string labels (Category) were converted into unique integers using a `LabelEncoder`. This transformation is deterministic and reversible, allowing us to map the model's numerical predictions back to their original human-readable category names for evaluation.

F. TF-IDF Vectorization

To convert the raw, unstructured text into a machine-readable format, we utilized TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This statistical technique highlights words that are unique and discriminative to specific documents while downplaying common words that appear frequently across the entire corpus. Specifically, the weight $W_{t,d}$ for a

term t in a document d is calculated as:

$$W_{t,d} = \text{tf}_{t,d} \times \log\left(\frac{N}{\text{df}_t}\right) \tag{3}$$

where N is the total number of documents and df_t is the number of documents containing term t. We limited the feature space to the top 1,000 most relevant unigrams to optimize dimensionality and computational efficiency.

G. Scaling Numeric Features

Numeric features like Price and Quantity can vary largely in scale (e.g., prices in hundreds vs. quantities in single digits). To prevent features with larger absolute values from disproportionately influencing the model's learning process, we standardized them using z-score normalization:

$$z = \frac{x - \mu}{\sigma} \tag{4}$$

This ensures that all numeric inputs have a mean of 0 and a standard deviation of 1, allowing them to make a balanced contribution to the final prediction.

V. MACHINE LEARNING MODELS USED

We selected and evaluated three distinct algorithms, each possessing different strengths regarding the handling of sparse text data and structured numeric features.

A. Logistic Regression

Logistic Regression is a linear classifier that models the probability of a class membership using a logistic (sigmoid) function. The probability $P(y = 1|x)$ is modeled as:

$$P(y = 1|x) = \frac{1}{1 + e^{-(w^T x + b)}} \tag{5}$$

It is widely favored for text classification tasks because it handles high-dimensional, sparse data (like TF-IDF vectors) exceptionally well. It is computationally efficient, less prone to overfitting on sparse data due to regularization (typically L2), and provides interpretable weights, allowing us to inspect which words (coefficients) strongly drive a product's classification.

B. Random Forest

Random Forest is a bagging ensemble method that builds multiple independent decision trees on random subsets of the data and features. While it is highly robust on structured tabular data, it can sometimes struggle with very sparse TF-IDF matrices because individual trees may fail to identify meaningful splits among the thousands of available text features. However, it serves as a strong non-linear baseline and is generally resistant to outliers.

C. XGBoost

XGBoost (Extreme Gradient Boosting) is a highly optimized gradient boosting framework known for its execution speed and model performance. Unlike Random Forest, it builds trees sequentially, where each new tree attempts to correct the residual errors of the previous ones. It optimizes a regularized objective function:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{6}$$

where l is the loss function and Ω penalizes model complexity. This allows it to effectively handle the heterogeneous mix of sparse text features and dense numeric features while mitigating overfitting.

VI. MODEL PERFORMANCE RESULTS

We conducted our experiments using a stratified 80-20 train-test split. Stratification was crucial to ensure that the distribution of categories in the training set exactly mirrored that of the test set, preventing any bias toward majority classes in the evaluation phase.

Standard preprocessing (TF-IDF transformation and numeric scaling) was fitted exclusively on the training data and then applied to the test data. This strict separation prevents data leakage, ensuring that the reported metrics reflect the model's true ability to generalize to unseen data.

A. Reported Model Metrics

The quantitative performance of the models on the hold-out test set was as follows:

- **Logistic Regression:** Accuracy = 0.9015, Precision = 0.9009, Recall = 0.9015, F1 = 0.8937.
- **Random Forest:** Accuracy = 0.7944, Precision = 0.8248, Recall = 0.7944, F1 = 0.7816.
- **XGBoost:** Accuracy = 0.9073, Precision = 0.9063, Recall = 0.9073, F1 = 0.9040.

B. Interpreting the Metrics

While accuracy serves as a useful high-level metric, weighted Precision, Recall, and F1 scores provide a more nuanced picture of performance, especially across imbalanced classes. The consistently high F1-score of XGBoost suggests it maintains a strong balance between identifying relevant items (Recall) and ensuring those items are correctly classified (Precision). Random Forest lagged behind, likely due to the difficulty of creating effective splits on high-dimensional sparse text vectors without deeper trees.

VII. CONFUSION MATRIX AND CLASS-WISE CALCULATIONS

To gain a deeper understanding of where the models make errors, we analyzed the confusion matrix. The matrix below represents a simplified 3-class evaluation scenario derived from our testing, illustrating the distribution of predictions:

$$CM = \begin{bmatrix} 120 & 10 & 5 \\ 8 & 95 & 12 \\ 4 & 9 & 110 \end{bmatrix} \quad (7)$$

Here, rows represent the actual ground-truth classes, while columns represent the predicted classes output by the model.

A. Totals and Derived Counts

Row sums (Actual totals): $R_0 = 135$, $R_1 = 115$, $R_2 = 123$.

Column sums (Predicted totals): $C_0 = 132$, $C_1 = 114$, $C_2 = 127$.

Total samples: $N = 373$.

B. Class-wise TP, FP, FN, TN

For any given class i: $TP_i = CM_{ii}$, $FP_i = C_i - TP_i$, $FN_i = R_i - TP_i$, $TN_i = N - TP_i - FP_i - FN_i$.

Applying these derivations:

a) Class 0:

- $TP_0 = 120$
- $FP_0 = 132 - 120 = 12$
- $FN_0 = 135 - 120 = 15$
- $TN_0 = 373 - 120 - 12 - 15 = 226$

b) Class 1:

- $TP_1 = 95$
- $FP_1 = 114 - 95 = 19$
- $FN_1 = 115 - 95 = 20$
- $TN_1 = 373 - 95 - 19 - 20 = 239$

c) Class 2:

- $TP_2 = 110$
- $FP_2 = 127 - 110 = 17$
- $FN_2 = 123 - 110 = 13$
- $TN_2 = 373 - 110 - 17 - 13 = 233$

C. Per-class Precision, Recall, and F1

Using the standard formulas:

$$Precision_i = \frac{TP_i}{TP_i + FP_i}$$
$$Recall_i = \frac{TP_i}{TP_i + FN_i}$$
$$F1_i = 2 \cdot \frac{Precision_i \cdot Recall_i}{Precision_i + Recall_i}$$

a) Class 0:

$$Precision_0 = \frac{120}{132} \approx 0.9091$$
$$Recall_0 = \frac{120}{135} \approx 0.8889$$
$$F1_0 \approx 0.8989$$

b) Class 1:

$$Precision_1 = \frac{95}{114} \approx 0.8333$$
$$Recall_1 = \frac{95}{115} \approx 0.8261$$
$$F1_1 \approx 0.8293$$

c) Class 2:

$$Precision_2 = \frac{110}{127} \approx 0.8661$$
$$Recall_2 = \frac{110}{123} \approx 0.8943$$
$$F1_2 \approx 0.8800$$

D. Overall Accuracy from the Matrix

The overall accuracy derived directly from this matrix is:

$$Accuracy = \frac{325}{373} \approx 0.8713 \quad (8)$$

Note: This calculated accuracy reflects the specific subset represented in the matrix and may differ slightly from the aggregate test set metrics reported in Section VI.

VIII. DISCUSSION

This section provides a deeper interpretation of the experimental results and their practical implications for deployment in a live retail environment.

A. Model Behavior Analysis

- **Logistic Regression:** This model proved to be a highly effective and robust baseline. Its linear nature allows it to easily manage the sparsity of TF-IDF vectors without overfitting. It is also transparent; we can directly inspect the coefficients to see which specific words (e.g., “shampoo”, “rice”) strongly predict a category, providing interpretability.
- **Random Forest:** While typically powerful for structured data, Random Forest struggled in this context. The high dimensionality of the sparse text data meant that individual decision trees often split on irrelevant features, diluting the model’s overall predictive power compared to the linear methods.
- **XGBoost:** This model achieved the best overall results. Its sequential boosting mechanism allows it to incrementally correct prediction errors, capturing subtle, non-linear interactions between textual keywords and price points that other models missed. It was also remarkably robust to the noise inherent in product names.

B. Precision vs Recall Trade-offs

For a retailer, the choice of model depends heavily on the specific business goal:

- **Latency vs. Accuracy:** If the system needs to classify products in real-time (milliseconds) as they are uploaded, Logistic Regression offers the fastest inference speeds. However, for batch processing (e.g., nightly updates) where accuracy is paramount, XGBoost is the superior choice.
- **Interpretability:** If the business needs to explain to vendors *why* a product was classified a certain way, Logistic Regression offers clearer insights than the ”black-box” nature of ensemble tree models.

- **Operational Scalability:** While complex models yield higher accuracy, they require robust infrastructure. The pipeline demonstrated here is scalable; TF-IDF computation is linear with corpus size, and XGBoost supports distributed training, making this approach viable for catalogs with millions of SKUs.

C. Error Analysis

The confusion matrix highlights that Class 1 exhibited higher error rates (both FP and FN). This suggests that products in this category likely share ambiguous keywords or pricing structures with other neighboring classes. Improving performance here might require targeted data augmentation or the creation of specific engineered features (e.g., price-per-unit) to better distinguish these borderline items.

IX. LIMITATIONS

While effective, the current pipeline operates under certain constraints:

- **TF-IDF Limitations:** TF-IDF relies entirely on word counts and cannot capture the semantic meaning of words (polysemy/synonymy). For example, it does not inherently know that "soda" and "soft drink" are related.
- **Loss of Granularity:** Merging small categories into "Other" improves model stability but results in a loss of fine-grained detail that might be useful for niche inventory tracking.
- **Simplified Quantities:** Our regex extraction simplifies complex units (e.g., treating "2 x 500ml" simply as "500"), potentially losing context regarding total volume vs. unit count.
- **Basic Numeric Features:** We only used raw Price and Quantity. Derived features like "discount percentage" or "brand tier" could add significant predictive value.
- **Generalization:** The model is trained on data from a single retailer. Its performance on data from a different store with different naming conventions is unproven.

X. KEY FINDINGS

- XGBoost delivered the best all-around performance with a top accuracy of 0.9073, validating its dominance in mixed-attribute tasks.
- Logistic Regression remains a highly competitive, efficient, and interpretable baseline for text-heavy classification tasks.
- Random Forest is less suited for very high-dimensional sparse data without prior dimensionality reduction techniques (like PCA).
- Combining unstructured text with structured numeric features (Price, Quantity) yields significantly better results than relying on text alone.
- The confusion matrix analysis (Accuracy ≈ 0.87) provides actionable insights into specific class overlaps, guiding future data cleaning efforts.

XI. FUTURE SCOPE

Future work could expand in several promising directions to enhance the system:

- **Deep Learning Embeddings:** Implementing BERT or Sentence-BERT models could capture deep contextual nuances and semantic relationships that TF-IDF misses.
- **Hierarchical Classification:** A multi-stage model could first predict the main Category and then subsequent sub-models could predict the SubCategory for better precision.
- **Advanced Feature Engineering:** Creating derived features like "unit price", "brand popularity score", or "text length" could boost performance.
- **Data Augmentation:** Synthetic data generation techniques could bolster rare categories, reducing the reliance on a generic "Other" class.
- **Hyperparameter Optimization:** Rigorous Grid Search or Bayesian Optimization could be applied to further fine-tune the XGBoost model parameters.
- **Cross-Retailer Testing:** Validating the model on external datasets from other retailers would prove its generalizability and robustness.
- **Explainable AI:** Tools like SHAP or LIME could be integrated to make the XGBoost predictions transparent and trustworthy to business stakeholders.

ACKNOWLEDGMENT

The authors express their sincere gratitude to Lovely Professional University for providing the infrastructure and academic support necessary for this research. We also extend our heartfelt thanks to our faculty mentor, Ms. Geetika Sethi, for her continuous guidance, encouragement, and valuable insights throughout the development of this project.

XII. CONCLUSION

This paper presented a comprehensive and enhanced machine learning framework for classifying retail products using the DMart dataset. We detailed a rigorous preprocessing pipeline, the integration of textual and numeric features, and a comparative analysis of Logistic Regression, Random Forest, and XGBoost. Our results conclusively identify XGBoost as the top-performing model, while highlighting the efficiency and interpretability of linear baselines. By carefully analyzing errors through a confusion matrix, we have pinpointed specific areas for improvement. This study provides a practical, actionable framework for retail analytics and lays a solid foundation for future academic research into more advanced, semantic classification systems.

REFERENCES

[1] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 2011.

[2] 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*, 2016.

[3] G. Salton and C. Buckley, “Term-Weighting Approaches in Automatic Text Retrieval,” *Information Processing & Management*, 1988.

[4] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” *Proceedings of the European Conference on Machine Learning (ECML)*, 1998.