E-commerce Product Categorization using NLP and Machine Learning

Mummidi Devi Siva Rama Saran, Akshayaa B K, R Sai Raghavendra, Poornima N, and Sachin Kumar S

Amrita School of Artificial Intelligence, Coimbatore, Amrita Vishwa Vidyapeetham, India
{ramasaranmummidi, akshayaabk1908, sairaghavendra179, poornima.n2425}@gmail.com, s_sachinkumar@cb.amrita.edu

Abstract. E-commerce, or electronic commerce, has revolutionized the way businesses operate, and consumers engage in transactions, also having a profound impact on global economies. From the early days of online retail to the current era of seamless digital transactions, the evolution of e-commerce has been marked by innovations in payment systems, security protocols, and user experience. In this work, the key problem addressed is product categorization based on the text description of a product in a particular category. Natural language processing techniques are used to eliminate unwanted and irrelevant information from the text description of the products. The normalized text description is then vectorized using the Term Frequency-Inverse Document Frequency, Bag of words, word2Vec, fastText, and BERT methods. This vectorized data is input into the machine learning models to classify into different product categories. Support vector machines resulted in the highest accuracy in classifying the product categories, which is 95.19%. By accurately categorizing products, it becomes easier for users to find the relevant products, improving the overall user experience. This can lead to increased customer satisfaction and potentially higher sales.

Keywords: E-commerce · Natural language processing · Machine learning · Text analysis · Product categories · Term Frequency-Inverse Document Frequency. · Bag of words · word2Vec · fastText · BERT.

1 Introduction

Electronic commerce (e-commerce) emerged in the late 20th century with the advent of the internet, gradually transforming the way people buy and sell goods. Initially, it gained momentum through online marketplaces and electronic data interchange. The convenience of shopping from the comfort of one's home and the ability to browse a vast array of products propelled its popularity [1]. E-commerce's success is rooted in its ability to simplify the shopping experience, offering a diverse range of products at the click of a button. Its appeal lies in the convenience, time-saving, and accessibility it provides, making it a preferred choice for consumers seeking simplicity in meeting their shopping needs [2].

E-commerce has revolutionized global trade, fostering economic growth by enabling businesses to reach a broader market and consumers to access a diverse range of products and services, thus contributing significantly to the expansion of the global economy [3]. Filters and sort options in e-commerce streamline the shopping experience, helping users swiftly find products tailored to their preferences and priorities. Navigating through categories such as electronics, household items, books, clothing, and accessories refines the shopping experience on e-commerce sites, enabling users to pinpoint their desired products with ease and precision [4].

Classifying products into categories based on text descriptions is helpful for several reasons [5]. Firstly, it aids in organizing and structuring large datasets, making it easier for both sellers and consumers to navigate and find relevant products. This classification also facilitates advanced search and recommendation systems, improving the overall user experience. However, it is a challenging task due to the diversity and variability of product descriptions. Different sellers may use varying language and terminology to describe similar products. This creates ambiguity and requires sophisticated algorithms to accurately categorize items [6].

In this work, various natural language processing (NLP) techniques are employed to normalize the text, reducing noise, and removing irrelevant information. Additionally, these methods also diminish the curse of dimensionality by removing irrelevant details from the text. Term Frequency-Inverse Document Frequency (TF-IDF) vectorization is utilized to capture the essence of representing a document in numerical form, reflecting the importance of words in the document within a larger corpus. Additionally, Word2Vec is employed to generate continuous vector representations of words, capturing semantic relationships through context-based learning. FastText enhances this by considering subword information, improving representation for rare words. Furthermore, BERT (Bidirectional Encoder Representations from Transformers) is used to provide deep contextual understanding by capturing word meaning in both directions within the text, offering superior performance in many natural language processing tasks. Subsequently, various machine learning (ML) algorithms are employed to classify the text into categories based on the normalized and vectorized descriptions. The hyperparameters of the ML classifiers are fine-tuned to further enhance the metrics. A comprehensive comparative analysis is conducted to identify the most optimal model for this task.

The remaining sections of this work are organized as follows: Section 2 presents a summary of the literature survey, while Section 3 outlines the process of data preparation. Section 4 provides detailed information on the proposed methodology, and Section 5 offers an overview of the results obtained in this study. This is followed by the conclusion and future scope in Section 6.

2 Related Work

Machine learning has been widely used for various e-commerce applications in recent years. [7] introduced MEP-3M, a substantial and extensive e-commerce dataset. This dataset serves as a valuable source for studying vision-language and e-commerce, given its size, diverse types of data, structured organization, and detailed categorization. Fused ML approach is utilized for product classification in [8]. An advanced e-commerce recommendation system with an intelligent meta-search engine, which outperforms traditional keyword and category searches, showcasing its potential for efficient cross-supplier product searches, is proposed in [9]. The problem of product category matching is addressed using ML in [10]. A novel distributional semantics representation and a two-level ensemble approach for efficient product classification is proposed in [11]. In [12], the utilization of deep neural networks for the classification of product categories is presented. Another deep learning approach is utilized for search intent analysis in [13]. A semi-supervised approach integrated with NLP is used for text classification [14]. Support vector machines are utilized to categorize products in [15]. Convolutional sequence to sequence learning is also employed for identifying the category of products [16].

Multi-granularity matching attention network proposed in [17] enhances the representation learning of queries and categories, effectively mitigating the expression gap between informal queries and categories. Hierarchical structure based classification method is utilized for query classification in [18]. In [19], a user-centric categorization framework is proposed, revealing inconsistent and uneven personalization in commercial websites. A blend of autoencoders and deep belief nets is employed to classify products into 28,338 categories, achieving an 81% accuracy [20]. NLP techniques are utilized to analyze product reviews, which are then classified into five ratings ranging from 1 to 5, signifying lowest to highest, respectively, using supervised ML algorithms [21]. [22] employed sentiment analysis to examine customer reviews on an e-commerce platform, demonstrating the effectiveness of machine learning algorithms with 85% accuracy in classifying reviews as highly trustworthy or not. Building upon recent advancements in e-commerce research, the proposed methodology introduces a novel fusion of NLP techniques, TF-IDF text vectorization, and fine-tuned supervised ML algorithms using optuna method for the effective classification of product categories.

3 Data Preparation

This section provides details about the dataset, an overview of the different normalization procedures used, and also the data pre-processing.

3.1 Dataset

The dataset used [23] in this research is an e-commerce dataset that has two features: one is the categories of products, and the other is the description of

various products in these categories. The number of instances for each category of products and the corresponding category names are presented in Table 1.

Table 1. Dataset Summary

Category	Number of Instances
Electronics	5,308
Household	10,564
Books	6,256
Clothing & Accessories	5,674

3.2 Data Normalization

Natural language processing (NLP) is employed for the task of text normalization. The following are the NLP techniques utilized to normalize the text description of the products:

- **Case Folding:** Converting text to lowercase in NLP is often done to ensure uniformity and improve the efficiency of text processing. It helps in treating words with different cases as the same, reducing the dimensionality of the data.
- **Text Cleaning:** White spaces, punctuation symbols, stop words, and unicode characters like emoji, HTML tags, etc., are removed to focus on the essential linguistic elements, stripping away noise and irrelevant details to enhance the accuracy and efficiency of classifier models.
- **Text Abbreviation Handling:** Substitution of acronyms and contractions in NLP improves text normalization and consistency for more accurate language understanding and processing.
- **Lexical Processing:** Stemming and lemmatization in NLP help reduce words to their root forms, improving text analysis by simplifying variations of words to a common base. Stemming involves chopping off prefixes or suffixes, while lemmatization considers the word's base or dictionary form for a more accurate transformation.
- **Removing Non-alphabetic Words:** Non-alphabetic words are removed to enhance text analysis by focusing on the meaningful content and reducing noise. A regular expression is used to tokenize the input text and filter out the words that contain non-alphabetic characters, leaving only those composed entirely of alphabetic characters. Finally, the filtered words are joined back into a string, effectively removing any non-alphabetic words from the original text.
- **Part-of-speech Tagging:** Part-of-speech (POS) tagging helps identify the grammatical category of each word in a sentence, aiding in syntactic and semantic analysis.

Fig. 1 depicts the text normalization process of a sample text example for a clear understanding of the proposed text normalization procedure.

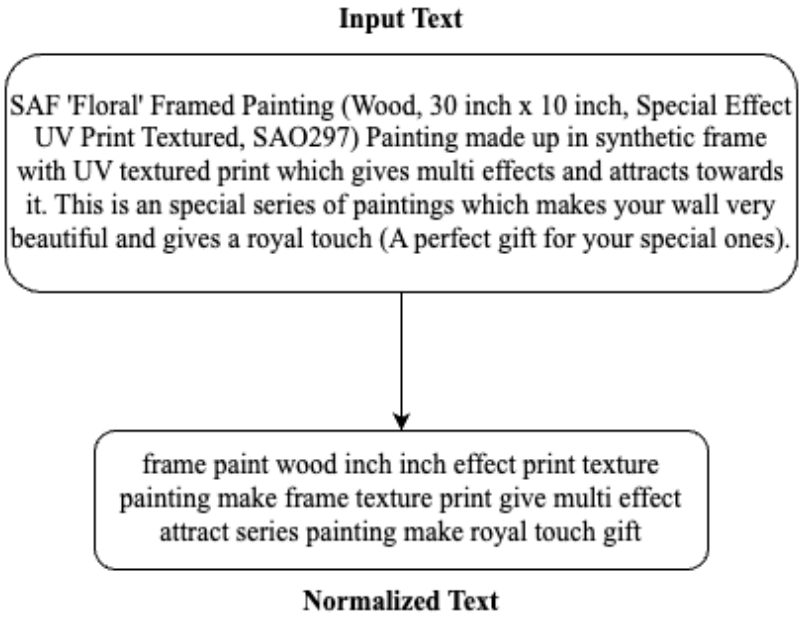

Fig. 1. Illustration of the normalization procedure using sample input text.

3.3 Data Pre-processing

The instances with null values are dropped from the dataset. Removing missing or null values in machine learning is crucial for several reasons. Firstly, these values can disrupt the training process and lead to inaccurate models, as algorithms may struggle to handle undefined or incomplete data. Additionally, many machine learning algorithms require complete datasets for effective training. Label encoding is done to convert categorical data into numerical format for algorithmic processing.

4 Methodology

This section provides a detailed overview of the proposed approach. The text data in the dataset undergoes standard normalization processes, including stemming, lemmatization, and the removal of stop words, for the TF-IDF and Bag of Words (BoW) models. Subsequently, the dataset undergoes preprocessing to eliminate null values, and label encoding is applied to the target feature in the dataset, i.e., category. The normalized text is then input into the TF-IDF model for vectorization, ensuring consistency and meaningful input for accurate model learning. Similarly, for Word2Vec, FastText, and BERT embeddings, partial normalization is applied, considering only selected text normalization processes. This approach is chosen to avoid the loss of valuable information inherent in

standard normalization procedures when utilizing pre-trained embeddings. Before feeding the tokenized words to the pre-trained models to obtain embeddings, only a few selected text normalization processes are applied. These representations are subsequently utilized as input features for the machine learning classifier to categorize the product classes. Fig. 3 illustrates the proposed methodology for the classification task.

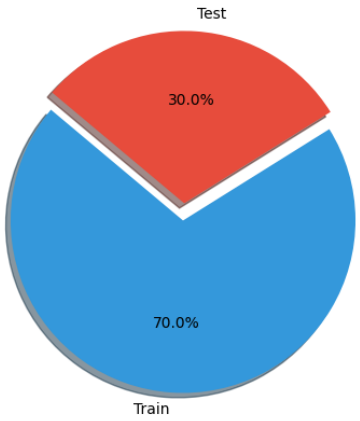

Fig. 2. Illustration of the data split into training and test segments.

4.1 Text Vectorization

Term Frequency-Inverse Document Frequency (TF-IDF) is a powerful technique in text vectorization that evaluates the importance of a word within a document relative to its frequency across a collection of documents. It consists of two components: term frequency (TF), which measures how often a term appears in a document, and inverse document frequency (IDF), determining the rarity of a term across the entire document set. By multiplying these values, TF-IDF assigns higher weights to terms that are frequent in a document but rare in the overall corpus, capturing the essence of the document's content while lessening the impact of common words.

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D) \tag{1}$$

where:

- t represents a term (word).
- d represents a document.
- D represents the entire document corpus.
- TF(t,d) is the term frequency of term t in document d, measuring how often t appears in d.

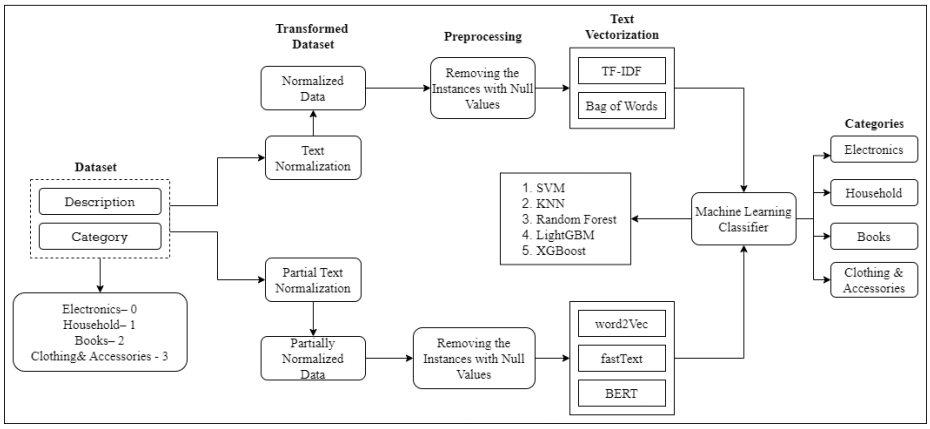

Fig. 3. A schematic block diagram of the proposed approach.

– IDF(t,D) is the inverse document frequency of term t in the corpus D, representing the rarity of t across the entire set of documents.

The TF-IDF product combines these two components to give a weighted value that highlights the importance of a term in a specific document while considering its rarity in the overall corpus.

Bag of Words (BoW) is a fundamental technique in natural language processing that represents text data by counting the frequency of words in a document. It disregards the order and context of words, focusing solely on their occurrence. BoW builds a vocabulary from the entire corpus of documents and generates a vector for each document, where each element represents the frequency of a word in the vocabulary within that document.

$$\text{BoW}(w, d) = \text{Count}(w, d) \tag{2}$$

where:

– w represents a word.
– d represents a document.
– Count(w, d) is the count of word w in document d.

Word2Vec is a popular technique for word embedding, which represents words in a continuous vector space. It captures semantic relationships between words by learning distributed representations of words based on their context in large corpora. Word2Vec employs either the Continuous Bag of Words (CBOW) or Skip-gram model to learn word embeddings, enabling the representation of words with similar meanings as vectors close to each other in the vector space.

$$\text{Word2Vec}(w) = \text{WordEmbedding}(w) \tag{3}$$

where:

- w represents a word.
- WordEmbedding(w) is the vector representation of word w learned by the model.

FastText is an extension of Word2Vec that considers subword information by breaking words into character n-grams. By including subword units, FastText captures morphological similarities and improves the representation of rare and out-of-vocabulary words. This approach enhances the performance of word embeddings, particularly for languages with complex morphology and spelling variations.

$$\text{FastText}(w) = \text{SubwordEmbedding}(w) \tag{4}$$

where:

- w represents a word.
- SubwordEmbedding(w) is the vector representation of word w incorporating subword information.

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art model for natural language understanding. It generates deep bidirectional embeddings by pre-training a transformer-based architecture on large text corpora. BERT captures contextual information from both directions within a sentence, resulting in rich representations of words that account for their surrounding context. Fine-tuning BERT on specific downstream tasks further enhances its performance in tasks such as text classification, question answering, and named entity recognition.

$$\text{BERT}(w, C) = \text{ContextualEmbedding}(w, C) \tag{5}$$

where:

- w represents a word.
- C represents the context of word w within a sentence.
- ContextualEmbedding(w, C) is the vector representation of word w considering its context C.

4.2 Classification

In this study, machine learning classifiers are utilized to distinguish between the four product categories based on the text descriptions of the products. The ML models utilized in this study are as follows:

- Support Vector Machines (SVM)
- Light Gradient-Boosting Machine (LightGBM)
- K-Nearest Neighbors (KNN)
- Random Forest (RF)
- Extreme Gradient Boosting (XGBoost)

SVMs are powerful classifiers that find optimal hyperplanes in high-dimensional spaces. LightGBM excels in handling large datasets and is renowned for its speed and efficiency. KNN classifier relies on proximity, assigning labels based on the majority class of its k-nearest neighbors. RF is an ensemble learning method that constructs a multitude of decision trees to enhance predictive accuracy and control overfitting. XGBoost is another boosting algorithm, known for its scalability and performance. The data split for training and testing segments is depicted in Fig. 2

4.3 Evaluation Metrics

The evaluation metrics used in this work are listed in Table 2.

Table 2. Error Metrics for Performance Assessment of ML Models.

S.No	**Error Metric**	**Formula**
1	Accuracy	$\frac{\text{True Positives} + \text{True Negatives}}{\text{Total Number of Predictions}}$
2	Precision	$\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$
3	Recall	$\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$
4	F1 Score	$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

5 Results and Discussions

The evaluation of various machine learning models across different text embeddings is illustrated in Table 3. The Support Vector Machine (SVM) classifier consistently demonstrates the highest performance, achieving accuracy, precision, recall, and F1 scores around 0.9520 with TF-IDF embeddings. LightGBM also performs well, with accuracies ranging from 0.9347 to 0.9416 across different

embeddings, showing its versatility. XGBoost achieves the highest accuracy of 0.9446 with Word2Vec embeddings, while Random Forest (RF) and K-Nearest Neighbors (KNN) show competitive results but generally perform slightly lower than SVM, LightGBM, and XGBoost across most embeddings.

Table 3. Illustration of performance evaluation metrics of the ML models.

Embeddings	**Model**	**Evaluation Metrics**			
		Accuracy	**Precision**	**Recall**	**F1 score**
TF-IDF	**SVM**	**0.9520**	**0.9524**	**0.9520**	**0.9519**
	KNN	0.9074	0.9089	0.9074	0.9072
	LightGBM	0.9347	0.9350	0.9347	0.9347
	RF	0.9275	0.9284	0.9275	0.9274
	XGBoost	0.9237	0.9246	0.9237	0.9236
Bag of words	**SVM**	0.9311	0.9298	0.9304	0.9301
	KNN	0.8020	0.8374	0.8145	0.8100
	LightGBM	0.9385	0.9402	0.9370	0.9385
	RF	0.9334	0.9386	0.9300	0.9339
	XGBoost	0.9277	0.9314	0.9249	0.9279
word2Vec	**SVM**	0.9339	0.9342	0.9339	0.9339
	KNN	0.9362	0.9368	0.9362	0.9362
	LightGBM	0.9416	0.9416	0.9416	0.9415
	RF	0.9279	0.9291	0.9279	0.9279
	XGBoost	0.9446	0.9447	0.9446	0.9446
fastText	**SVM**	0.9151	0.9161	0.9151	0.9151
	KNN	0.9055	0.9085	0.9055	0.9061
	LightGBM	0.9366	0.9368	0.9366	0.9366
	RF	0.9249	0.9262	0.9249	0.9249
	XGBoost	0.9404	0.9403	0.9404	0.9403
BERT	**SVM**	0.9220	0.9221	0.9220	0.9220
	KNN	0.8947	0.8960	0.8947	0.8947
	LightGBM	0.9203	0.9206	0.9203	0.9203
	RF	0.8984	0.9023	0.8984	0.8982
	XGBoost	0.9247	0.9250	0.9247	0.9246

SVM's superior performance can be attributed to its ability to handle high-dimensional spaces and its effectiveness in finding the optimal hyperplane that maximizes the margin between classes, which is particularly beneficial in text classification tasks. With Bag of Words embeddings, LightGBM achieves the highest accuracy of 0.9385, while SVM maintains a strong performance across all embeddings. FastText embeddings show LightGBM and XGBoost performing exceptionally well, with accuracies of 0.9366 and 0.9404, respectively. BERT embeddings, which capture deep contextual information, exhibit robust performance with XGBoost achieving an accuracy of 0.9247 and SVM following closely with 0.9220. Overall, these results underscore the effectiveness of combining advanced text embeddings with robust classifiers like SVM, LightGBM, and XG-

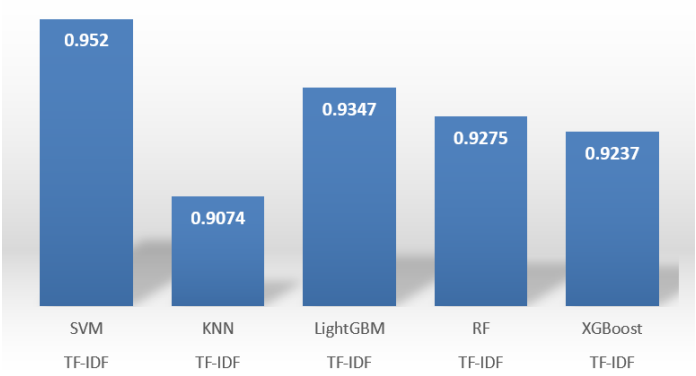

Fig. 4. Illustration of the performance comparison of the ML algorithms.

Boost for accurate and reliable text classification tasks. The comparison of these models' effectiveness based on classification accuracy is depicted in Fig. 4, highlighting the relative strengths of each approach.

Table 4. Illustration of performance evaluation metrics for each class obtained using SVM.

Class	**Evaluation Metrics**			
	Accuracy	**Precision**	**Recall**	**F1 score**
Electronics	0.9255	0.9373	0.9255	0.9314
Household	0.9666	0.9434	0.9666	0.9549
Books	0.9330	0.9579	0.9330	0.9453
Clothing & Accessories	0.9704	0.9757	0.9704	0.9731

The remarkable performance of SVM model can be observed from the confusion matrix, depicted in Fig. 5. Out of 8134 test data points, 7743 are correctly classified, indicating high precision and accuracy in the model's predictions. This impressive performance reflects the model's ability to navigate the data landscape effectively and make accurate assessments.

In the t-SNE plot illustrated in Fig. 6, the clusters are well-separated, revealing the efficiency in capturing the underlying patterns or trends within the data. The observed overlap appears normal, indicating that the selected features can easily distinguish the classes. Moreover, there is a minimal presence of outliers in the plot, signifying a sparse number of instances with a probability of being misclassified.

Fig. 7 depicts the receiver operating characteristic (ROC) curves with areas under the curves (AUC). The AUC values for all the classes are close to 1, signifying that the model is effective at distinguishing between the classes. The

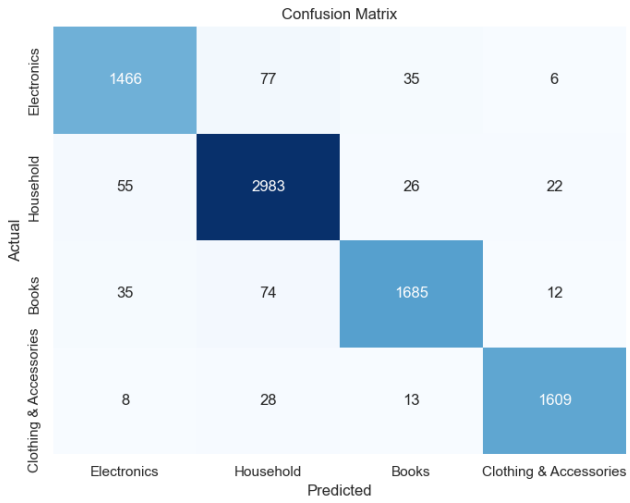

Fig. 5. Illustration of the confusion matrix derived using SVM.

curves are closer to the upper-left corner of the plot, indicating fewer false positives and false negatives. The achieved results, overall, underscore the robustness and reliability of the proposed approach in classifying instances across diverse product categories. This study thus stands as a valuable benchmark in the field of e-commerce research.

6 Conclusion

In this work, a machine learning recognition architecture based on NLP is proposed to accurately classify products into different categories using the product description as the input feature. This is achieved by leveraging the text description data of the products within each category. NLP-based text data normalization is performed before the classification task to ensure consistent and meaningful input for accurate model learning. Among the various models evaluated, the SVM classifier with TF-IDF embeddings achieved the highest accuracy of 95.20%, demonstrating its robustness and effectiveness in this task.

Future work could focus on enhancing text normalization through spell-checking and other preprocessing techniques in NLP to further improve input data quality. Exploring alternative text vectorization methods, such as Word2Vec, FastText, and BERT, in combination with advanced machine learning algorithms, can enhance classification performance. Additionally, developing product recommendation systems represents a promising avenue to extend the impact of this research, potentially offering a more personalized and satisfying user experience for customers in e-commerce. Expanding this work to include these im-

Fig. 6. Visualization of the intricate relationships in the data using a t-SNE plot.

provements could significantly boost the overall effectiveness and applicability of the classification architecture in real-world scenarios.

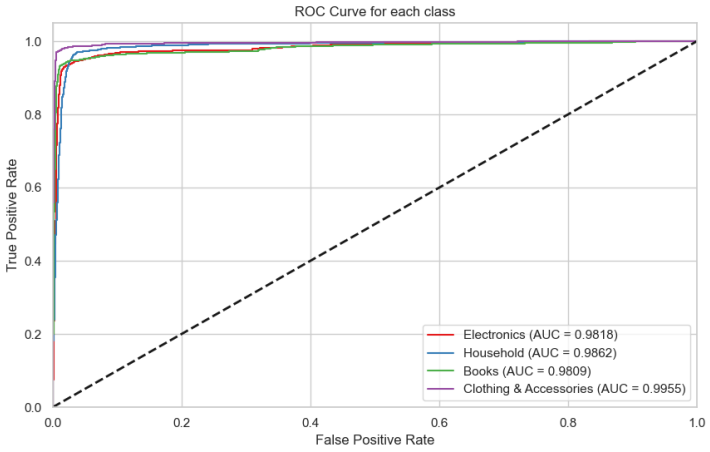

Fig. 7. Illustration of the ROC-AUC curves.

Bibliography

[1] Manal Loukili, Fayçal Messaoudi, and Mohammed El Ghazi. Machine learning based recommender system for e-commerce. *IAES International Journal of Artificial Intelligence*, 12(4):1803–1811, 2023.

[2] Francisco Munoz, Clyde W Holsapple, and Sharath Sasidharan. E-commerce. In *Springer Handbook of Automation*, pages 1411–1430. Springer, 2023.

[3] Harikumar Pallathadka, Edwin Hernan Ramirez-Asis, Telmo Pablo Loli-Poma, Karthikeyan Kaliyaperumal, Randy Joy Magno Ventayen, and Mohd Naved. Applications of artificial intelligence in business management, e-commerce and finance. *Materials Today: Proceedings*, 80:2610–2613, 2023.

[4] Huang Huang, Adeleh Asemi, and Mumtaz Begum Mustafa. Sentiment analysis in e-commerce platforms: A review of current techniques and future directions. *IEEE Access*, 2023.

[5] Sabina-Cristiana Necula. Exploring the impact of time spent reading product information on e-commerce websites: A machine learning approach to analyze consumer behavior. *Behavioral Sciences*, 13(6):439, 2023.

[6] Wenhu Yu, Zhiqiang Sun, Haifeng Liu, Zhipeng Li, and Zhitong Zheng. Multi-level deep learning based e-commerce product categorization. In *eCOM@ SIGIR*, 2018.

[7] Fan Liu, Delong Chen, Xiaoyu Du, Ruizhuo Gao, and Feng Xu. Mep-3m: A large-scale multi-modal e-commerce product dataset. *Pattern Recognition*, 140:109519, 2023.

[8] R Anitha and D Vimal Kumar. E-commerce product classification using fused machine learning. *Scandinavian Journal of Information Systems*, 35(1):795–808, 2023.

[9] Ick-Hyun Kwon, Chang Ouk Kim, Kyung Pil Kim, and Choonjong Kwak. Recommendation of e-commerce sites by matching category-based buyer query and product e-catalogs. *Computers in Industry*, 59(4):380–394, 2008.

[10] Mayank Kejriwal, Ke Shen, Chien-Chun Ni, and Nicolas Torzec. An evaluation and annotation methodology for product category matching in e-commerce. *Computers in Industry*, 131:103497, 2021.

[11] Vivek Gupta, Harish Karnick, Ashendra Bansal, and Pradhuman Jhala. Product classification in e-commerce using distributional semantics. *arXiv preprint arXiv:1606.06083*, 2016.

[12] Dehong Gao, Wenjing Yang, Huiling Zhou, Yi Wei, Yi Hu, and Hao Wang. Deep hierarchical classification for category prediction in e-commerce system. *arXiv preprint arXiv:2005.06692*, 2020.

[13] Ali Ahmadvand, Surya Kallumadi, Faizan Javed, and Eugene Agichtein. Deepcat: Deep category representation for query understanding in e-commerce search. *arXiv preprint arXiv:2104.11760*, 2021.

[14] Guy Horowitz, Stav Yanovsky Daye, Noa Avigdor-Elgrabli, and Ariel Raviv. Consistent text categorization using data augmentation in e-commerce. *arXiv preprint arXiv:2305.05402*, 2023.

[15] Dan Shen, Jean David Ruvini, Manas Somaiya, and Neel Sundaresan. Item categorization in the e-commerce domain. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1921–1924, 2011.

[16] Idan Hasson, Slava Novgorodov, Gilad Fuchs, and Yoni Acriche. Category recognition in e-commerce using sequence-to-sequence hierarchical classification. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 902–905, 2021.

[17] Chunyuan Yuan, Yiming Qiu, Mingming Li, Haiqing Hu, Songlin Wang, and Sulong Xu. A multi-granularity matching attention network for query intent classification in e-commerce retrieval. In *Companion Proceedings of the ACM Web Conference 2023*, pages 416–420, 2023.

[18] Lvxing Zhu, Kexin Zhang, Hao Chen, Chao Wei, Weiru Zhang, Haihong Tang, and Xiu Li. Hcl4qc: Incorporating hierarchical category structures into contrastive learning for e-commerce query classification. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3647–3656, 2023.

[19] Dezhi Wu, Il Im, Marilyn Tremaine, Keith Instone, and Murray Turoff. A framework for classifying personalization scheme used on e-commerce websites. In *36th Annual hawaii international conference on system sciences, 2003. Proceedings of the*, pages 12–pp. IEEE, 2003.

[20] Ali Cevahir and Koji Murakami. Large-scale multi-class and hierarchical product categorization for an e-commerce giant. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 525–535, 2016.

[21] Deepak Dharrao, Sarika Deokate, Anupkumar M Bongale, and Siddhaling Urolagin. E-commerce product review classification based on supervised machine learning techniques. In *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 1934–1939. IEEE, 2023.

[22] Hrutuja Kargirwar, Praveen Bhagavatula, Shrutika Konde, Paresh Chaudhari, Vipul Dhamde, Gopal Sakarkar, and Juan C Correa. E-commerce product's trust prediction based on customer reviews. In *Congress on Intelligent Systems*, pages 375–383. Springer, 2023.

[23] Gautam. E commerce text dataset, July 2019.