Computer Science and Information Systems 22(4):1707–1756 https://doi.org/10.2298/CSIS241130068L

Implementing Persona in the Business Sector by A Universal Explainable AI Framework Based on Byte-Pair Encoding

Zhenyao Liu[1,⋆], Yu-Lun Liu[2], Wei-Chang Yeh[2], and Chia-Ling Huang[3]

[1] School of Economics and Management, Taizhou University
Taizhou 225300, Jiangsu Province, China
zyliu@tzu.edu.cn
[2] Integration and Collaboration Laboratory, Department of Industrial Engineering and Engineering Management, National Tsing Hua University
Hsinchu 300044, Taiwan
yulun.lawrence@gmail.com
yeh@ieee.org
[3] Department of International Logistics and Transportation Management, Kainan University
Taoyuan 33857, Taiwan
clhuang@mail.knu.edu.tw

Abstract. In the commercial realm, particularly for businesses targeting consumers (B2C), the challenge of acquiring and retaining valuable potential customers is paramount. As chip technology continues to advance at breakneck speed, in line with Moore's Law, various innovative AI technologies have emerged, yet this also highlights the infamous "black-box" issue. Naturally, this has paved the way for the rise of Explainable AI (XAI) and machine learning. In response, this study proposes a universal explainability framework to tackle both the black-box conundrum and the limitation of customer list sizes. The framework leverages the fundamental Byte-Pair Encoding (BPE) algorithm from large language models to tokenize natural language data, integrating the results into customer data as feature columns, thereby constructing comprehensive Persona. Crucially, domain experts are involved in the model-building process, selecting and recommending features. These experts utilize depth-first search to identify additional, similar feature columns, which are then used as target categories for machine learning models. The final step involves classification tasks and prediction evaluations. The proposed framework demonstrates its effectiveness and generalizability through validation on public datasets, increasing the number of potential customers by 7.5 times compared to traditional modeling approaches. In case studies, the framework outperforms customer lists generated by experts based on past experience, yielding 2.4 times more customers, 3.8 times higher response rates, and 9 times more total respondents. More importantly, both the model-building process and predictive outcomes are interpretable through domain knowledge, enabling businesses to transfer experience and expertise, thus laying a solid foundation for large language models within the industry.

Keywords: Natural Language Processing, Byte-Pair Encoding, Persona, Explainable Machine Learning, Business Sector.

⋆ Corresponding author

1. Introduction

In the aftermath of the COVID-19 era, technological advancements have accelerated various industries. However, the adoption of technologies such as machine learning has progressed slowly in certain sectors [1, 2] . Many scholars argue that despite the improvements in tools and computational capabilities over the years, the impact of these technologies has not reached the expected levels and continues to shape our daily lives [3, 4]. Although artificial intelligence has become an indispensable part of modern life, the use of opaque "black-box" models in highly sensitive domains—such as healthcare, biomedicine, public policy, human life, judicial rights, finance, consumer products, and any area related to personal privacy and life—remains particularly problematic [5].

Consequently, literature on Explainable AI (XAI) began to experience exponential growth from 2020 onward. Research in this area spans various fields, including digital health, law, public transportation, finance, and defense, reflecting the increasing recognition of the importance of transparency and interpretability [5]. On the other hand, major technology companies released numerous large language models (LLMs) in 2023, which gained considerable popularity. These include OpenAI's ChatGPT [6] and Meta AI's Llama [7, 8], with ChatGPT alone amassing over 180 million users [6].

In fact, large language models (LLMs) have not only introduced technological transformations in domains directly related to natural language, such as customer support [9], search engines [10, 11], and text translation [12, 13], but have also been broadly applied across other interdisciplinary areas, including medicine, code assistance, education, and finance [14–17]. This signifies that LLMs possess adaptability and potential for language-related tasks across various industries and environments.

With the pulse of technological advancement, both the economy and society have undergone significant changes, profoundly altering consumer shopping and retail capabilities [18]. For businesses targeting the consumer (B2C) market, competition has become increasingly fierce, making the retention of valuable and indispensable potential customers crucial [18]. In customer relationship management, whether in automotive, aviation, retail, or e-commerce sectors, unique methods of customer segmentation are employed. Various techniques are utilized to predict future demand peaks and adjust pricing and marketing strategies, all with the aim of gaining a deeper understanding of consumer behavior and habits [19].

Beyond e-commerce and retail, the financial industry is also a prominent example of B2C commerce. Marketing campaigns are one of the primary methods for achieving corporate objectives and are crucial for banks in attracting and retaining customers. Moreover, if a company's marketing activities or strategies are not executed effectively, it can face significant challenges in meeting annual targets [20], which in turn can impact overall business performance [21] and corporate profitability [22]. In any industry focused on sales strategies, it is essential to gain a deeper understanding of each consumer, including their purchasing habits and preferences [23], and to develop appropriate marketing strategies based on their buying patterns and attributes. Marketing strategies aim to deliver greater value to both customers and the company at a lower cost. In the business realm, failing to carefully consider the process through which potential consumers purchase or receive products can lead to wasted resources [24]. Consequently, calculating the return on investment (ROI) of marketing expenditures across activities and strategies such

as physical advertising, promotional campaigns, and digital advertising is a complex yet crucial issue for decision-makers [18].

Given the aforementioned context, decision-makers or domain experts responsible for marketing strategies often reject the use of potential customer lists generated by models, primarily due to concerns about cost and career development. They tend to distrust "black-box" models and are apprehensive about the possibility of these technologies replacing their roles [25]. Furthermore, as many companies lack additional funds for validating the lists produced by these models, skepticism regarding the validity of model-generated lists persists. Consequently, experts prefer to rely on their own experience to plan marketing activities and identify potential customers [26], choosing to preserve their job security while potentially causing the company to lag behind technological advancements.

This study aims to address the issue of domain experts' reluctance to accept marketing lists generated by models by proposing an explainability framework. In addition to leveraging the fundamental principles of large language models (LLMs) to provide interpretability to data through natural language, previous literature has also employed RFM models to enhance users' understanding of data [27]. Furthermore, techniques such as LIME and SHAP can be used to supplement the interpretability of model results [28], and the use of graph-based co-occurrence descriptions can elucidate the weights and relationships between features [29], thereby improving the efficiency of information retrieval.

Moreover, involving domain experts in the construction of models to help them understand the significance of the predictive results can not only increase the number of potential customers but also enable decision-makers and experts to connect marketing activities with corporate value. This fosters trust in the model-generated results and alleviates the tension between machine learning and domain expertise [25], while also preventing manipulation of marketing variables [18]. Meeting these conditions will facilitate the integration of decision-makers' and experts' domain knowledge and experience into the models [30, 31], thereby enhancing the company's value and position in the era of large language models.

Therefore, this study develops a universal explainability framework to address issues related to domain experts' inability to accept model-generated lists and the limitations on the number of items in these lists. The framework will incorporate the following functionalities and conditions:

- The framework proposed in this study is designed to be applicable to any B2C business within the commercial sector. It will enable companies to obtain potential customer lists that are both understandable and interpretable, and that exceed the number of potential customers typically identified through conventional experience and models.
- This framework must possess both reproducibility and generalizability, allowing any industry dealing with natural language data to apply it in order to provide additional interpretability to their data and models.
- This approach leverages the fundamental principles of large language models (LLMs), specifically, tokenization algorithms, to provide additional and effective feature columns to natural language data. This enhancement makes customer data more descriptive, thereby improving readability for users and facilitating a clearer understanding of customer purchasing behavior and habits.

- In the steps of the explainability framework, involving domain experts in understanding the operation of the framework and the model-building process not only facilitates the transfer of their knowledge and experience into the model but also reduces the tension between their professional status and technological advancements.
- The predictive results of this method should surpass those generated based on past experience in terms of list size, response rate, and overall number of respondents. Additionally, the method should be applied in practical cases to achieve more effective, diverse, and precise customer relationship management and marketing strategies.

2. Preliminaries

2.1. XAI and XML

As the applications of AI and ML become increasingly widespread, the methods have also grown in complexity. Consequently, business stakeholders have become more concerned about the potential drawbacks of these models, including data-specific biases [32]. To address these concerns, Lundberg et al. introduced SHapley Additive exPlanations (SHAP) as an industry standard for interpreting machine learning models [33]. However, such interpretability often falls short of satisfying most users, leading to the consideration of post-hoc explanation methods, such as textual explanations, visual explanations, and example-based explanations [34, 35].

Due to the "black box" issue inherent in artificial intelligence and machine learning, three key elements have emerged to define XAI and XML.

- **Transparency**: A ML method is considered transparent if the model itself is easy to understand and the extraction process is transparent. This encompasses model transparency, design transparency, and algorithmic transparency [36].
- **Interpretability**: Users must be able to understand the basis on which algorithmic decisions are made within the model. They should also be capable of explaining the algorithmic criteria and hyperparameter variables within the model in comprehensible terms [36].
- **Explainability**: The definition of this element varies [36], but it is commonly understood as the user's ability to explain why the model made a specific decision, understand the rationale behind a particular prediction, and even integrate domain knowledge with the prediction to provide contextual explanations. This deeper understanding is essential for achieving true explainability [36].

With the rapid advancement of XAI and XML, their applications have become increasingly widespread across various fields [37].
In the medical field: Soares et al. utilized computed tomography (CT) scans to identify COVID-19 [38]. Morais et al., in collaboration with domain experts, examined the performance of XAI in cancer diagnosis from the perspective of experts, offering explanations that extend beyond the experts' viewpoint [39].
In the field of public policy and the judicial system: Dressel and Farid highlighted the widespread use of criminal risk assessment systems. They emphasized the necessity of providing explanations in key decisions within these systems to maintain fairness and avoid racial bias [40].

Applications based on natural language processing are also being explored in the research domain. Several authors have improved user trust in applications through the use of XAI techniques for anomaly and fraud detection [41]. Additionally, Mathews proposed an interpretable tweet classification method based on LIME, which enhances the explainability of application results, thereby increasing user engagement and trust [42].

A significant portion of applications is found in autonomous driving systems. In a fully automated system, the driving system is expected to operate in any unknown environment [43], which impacts trust and transparency compared to black-box systems. Therefore, from the perspectives of public perception and trust, as well as regulatory and legal considerations, XAI is critically important. Transparency, interpretability, and explainability are essential for developing more reliable, safe, and regulation-compliant autonomous driving systems [43].

In other domains, Murindanyi et al. utilized four tree-based machine learning methods and four standard machine learning methods to predict customer churn at Czech banks. By incorporating post-hoc explanation techniques such as LIME and SHAP, they achieved satisfactory predictive results [44]. Clement et al. proposed the XAIR process for the development of XAI, which mirrors the five steps of software development: requirements analysis, design, development, evaluation, and deployment. This process is presented as a comprehensive framework for other scholars to reference [45].

From the literature review presented in this section, it is evident that both XAI and XML share many similar elements and principles. XAI or XML fundamentally relies on three key elements: transparency, interpretability, and explainability. The application of XAI or XML in commercial domains is relatively limited, as these fields place greater emphasis on domain experts' experience. While techniques such as LIME and SHAP have demonstrated effective explanatory capabilities in literature, they may still be deemed insufficient by experts lacking data-related knowledge, leading to a lack of persuasion and practical application in industry. Additionally, due to the cost sensitivity in commercial sectors, extensive experimental costs and expenditures for model validation are often unacceptable. However, if interpretability frameworks can improve the reliance on experience in commercial fields, they are likely to contribute significantly and offer future advancements in these domains.

The issue of marketing lists being rejected by domain experts, as examined in this study, will be addressed through an interpretability framework that meets the three key elements: transparency, interpretability, and explainability. This framework will be designed to extract natural language data from customers, transforming the extracted natural language results into customer personas. Additionally, it will incorporate recommendations from domain experts to form a comprehensive solution.

2.2. Persona

Many studies have indicated that constructing persona aids in better understanding user needs, thereby facilitating personalized and precise information services [46].
Given that the early development of persona was driven by the needs of designers, scholar Travis, who specializes in user experience research, provided the following definition of the persona extraction process [47]:

- **Primary Research**: Whether the persona is determined based on real customer data or contextual interviews.

- **Empathy**: Whether the persona evokes user understanding and empathy by incorporating elements such as names, photographs, or product-related descriptions.
- **Realism**: Whether the persona appears authentic to experts in the field or frontline personnel who interact directly with customers.
- **Singularity**: Whether each persona is unique in its composition and distinct from other characters.personnel who interact directly with customers.
- **Objectives**: Whether the persona includes product-related goals and provides key descriptions that articulate these objectives.
- **Quantity**: Whether the number of personas meets the team's requirements, is sufficient for the team to remember their characteristics, and designates one persona as the primary character.
- **Applicability**: Whether the team can practically apply the persona in decision-making processes.

The seven elements outlined above play a crucial role in the effective implementation of persona techniques. They offer sufficient flexibility and adaptability, enabling practitioners to creatively explore and develop various applications of personas in practice [48].

To fulfill various objectives, personas are widely applied in software design [49], advertising [50], and technology products [46]. However, the ultimate aim remains that these personas should effectively inform and guide planning and decision-making processes [51].

Although the definition of personas is relatively broad, a review of the literature reveals that most scholars agree that personas are inherently goal-oriented. Practitioners must have a clear understanding of the purpose behind persona extraction and whether the resulting personas fulfill the initially defined objectives. Moreover, the process of developing personas should ensure that the seven essential elements are met, thereby ensuring that the personas align with expectations.

In practical applications, beyond obtaining personas through pre-classified data combined with statistical and regression methods, many scholars also employ clustering and supervised learning techniques for persona extraction. Some even derive personas from predictive outcomes. However, these approaches often fall short of achieving the initially set objectives [49, 50].

The explainability framework proposed in this study differs from traditional methods of persona extraction in the literature. It utilizes BPE to extract customers' natural language data, directly generating personas that enable experts and decision-makers to describe their behaviors and characteristics. These personas will meet the criteria of the seven key elements, ensuring not only realism and uniqueness but also alignment with the intended objectives. Additionally, BPE enhances the model's transparency, interpretability, and explainability.

2.3. The BPE

When discussing Byte Pair Encoding (BPE), it is common to reference the increasingly popular large language models (LLMs) in recent years. These models, characterized by an extensive number of parameters, are designed to understand and process natural language by modeling the semantics and probabilities of text sequences within vast datasets.

Through pre-training tasks, such as Masked Language Modeling or autoregressive prediction, large language models learn to comprehend and generate natural language effectively [52].

A well-designed pretrained transformer language model requires the implementation of various subword tokenization methods [53], among which the most renowned is BPE [54].BPE, proposed in 1994, is a straightforward data compression technique that employs a single unused byte to iteratively replace the most frequent pair of bytes in a sequence [54].

The following are the steps involved in BPE:
Step 1: we initialize the symbol vocabulary using the character vocabulary and represent each word as a sequence of characters, appending a special end-of-word symbol to the end of each word. This symbol aids in restoring the original state after translation.
Step 2: we begin iteratively calculating the frequency of all symbol pairs, where a symbol pair is a combination of each character in the vocabulary. The most frequent symbol pair is then replaced with a new symbol. For instance, if the most frequent pair is A and B, it will be replaced by a new symbol AB. In the subsequent iteration, A and B are ignored, and the frequency is calculated using AB in combination with other characters.
Step 3: each occurrence of a new symbol represents a merging operation. In other words, each merging operation generates a new symbol, which also signifies an n-gram of characters.

From the above steps and explanation, it is evident that an increase in the number of merging operations results in a larger symbol vocabulary and a corresponding increase in the granularity of the characters [55].
The BPE method merges the most frequent pairs of symbols in the entire text. Although it may appear as though BPE is performing a form of word concatenation, this is actually due to the high frequency of certain pairs. These high-frequency pairs persist and thus appear as concatenated sequences. Consequently, the most frequent pairs in the text will become prevalent in the final vocabulary. This characteristic of BPE is also why it can be effectively applied to various languages.

After understanding the operation and fundamental principles of BPE, one might consider why, for English text tokenization, spaces are not used for segmentation. From a human perspective, using spaces for tokenization seems to be the most intuitive approach. However, employing spaces or punctuation marks for segmentation results in an excessively large vocabulary. Any variation of a word would be included in the vocabulary, and if a word has multiple forms, the vocabulary size can grow exponentially. Such a large vocabulary necessitates an enormous matrix for input and output layers, increasing both memory and computational complexity [56].
Consequently, various tokenization algorithms avoid using spaces or punctuation for segmentation. This is why the BPE algorithm includes an end-of-word symbol in its implementation, a practice that also contributed to GPT-2 achieving optimal performance during its initial training [57].

In summary, BPE is a tokenization method, also referred to as a segmentation algorithm, and serves as a preprocessing technique for natural language data. It can also be applied to address the Out-Of-Vocabulary (OOV) problem [55] in natural language processing. Tokenization involves the mechanism of segmenting or dividing sentences and words into their smallest possible units [58].

The application domains of BPE include various fields. In the realm of language translation, BPE is characterized by its adaptability to different languages [59]. Additionally, numerous practical use cases of BPE have been documented: in the field of network diagnosis and detection [55], the medical and healthcare sector [60], experiments involving symbolic music for music generation and composer classification [61], and addressing the linguistic complexity on social media platforms [62].
This study proposes an interpretability framework for the business domain. The framework employs an improved version of BPE to enhance data feature dimensions, transforming tokenization results into feature fields to form Personas. By incorporating domain experts' recommendations on target fields, the framework utilizes Depth-First Search (DFS) to expand feature fields.

2.4. The DFS

DFS is a technique that has been extensively applied as a solution method for problems in combinatorial theory and AI [63]. The search process of DFS is closely related to graph theory, necessitating the introduction of certain graph-related definitions. These definitions are derived from Harary's research [64].

Let G be a graph, such that $G = (V, E)$, where V is the set of nodes and E is the set of edges. The set E consists of unordered pairs of nodes, each representing an edge. When manually drawing a graph, nodes are typically represented by circles, and the connections between these circles correspond to the edges.

Suppose we aim to search through the graph G. Initially, none of the nodes in G have been explored. We begin at an arbitrary vertex and select an edge to follow, traversing it to reach a new node, and continue this process. At each step, we choose an unexplored edge leading from the current node and traverse it. Once an edge has been traversed, it will not be explored again. This process continues until all edges in G have been traversed exactly once. This procedure constitutes the search [65].

The detailed steps of the DFS algorithm are presented below. Please refer to Figure 1 for illustration.

In summary, DFS is a graph traversal method that begins at an arbitrary vertex and explores as far as possible along each path before backtracking to visit any unvisited vertices when no further progression is possible [66].

From the perspective of text classification and Information Retrieval (IR), the concept of weighting is also applied. Blanco and Lioma proposed a graph-theoretic approach applied within the IR field, where text is modeled as a graph with edges representing the relationships between words. These relationships are then assigned corresponding weights. This method has been shown to perform on par with standard techniques in IR [29]. To address Word Sense Disambiguation (WSD), Rahmani et al. developed an unsupervised co-occurrence graph based on a corpus, which does not rely on the inherent structure and properties of the language. In other words, ambiguous words are assigned additional weights, altering the contextual structure [67].

DFS is also applied in practical cases. Du et al. proposed an algorithm that combines deep convolutional neural networks with DFS to address the problem of identifying power outage locations. In their approach, convolutional networks are used as a safety assessment tool, followed by DFS to find suitable interruption path locations. This method not only improves accuracy but also performs thousands of times faster than traditional

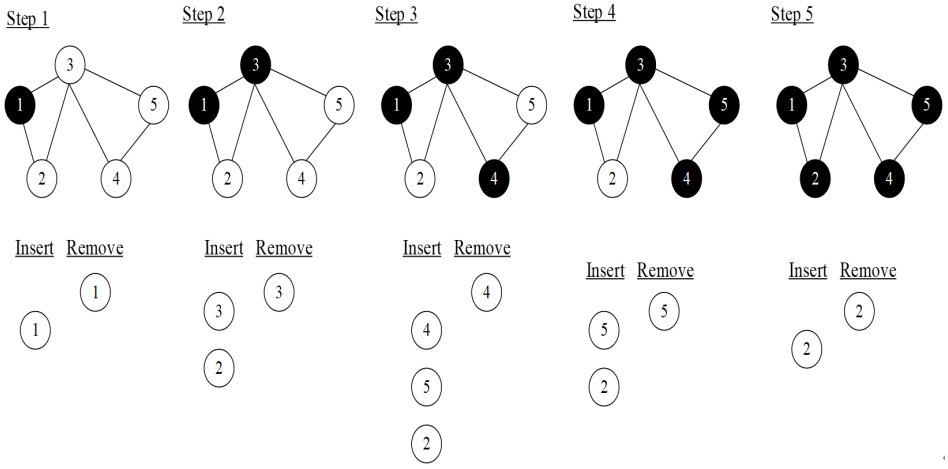

Fig. 1. The steps of the DFS algorithm

methods [68]. Mei and Gül introduced a new approach for detecting crack patterns in foundational design. After enhancing the CNN model, DFS is used for post-processing to remove isolated pixels and improve accuracy [69].

In summary, DFS is a graph-based search method that involves exploring data from one vertex to another, delving deeper into subsequent vertices, and backtracking to unvisited vertices to traverse the entire graph. Therefore, in the field of IR, many scholars use graphs to represent statements by altering edge weights. This approach not only provides models with enhanced features but also improves context and addresses issues with specialized vocabulary. In various practical cases, the characteristics of DFS combined with models enable the identification of diverse root causes and assist expert systems in effectively proposing solutions.

This study proposes a generalizable interpretability framework for application in the commercial domain. The framework enhances feature dimensions using BPE and, in conjunction with features provided by domain experts, employs DFS to identify similar feature fields. Features are represented in a graph, with edge weights based on co-occurrence to illustrate the relationships between features. The results of the DFS, combined with expert recommendations, are treated as categorical answers for a Light Gradient Boosting Machine (LightGBM). Finally, the framework performs a classification task to predict a list of potential customers highly similar to the expert recommendations.

2.5. Light Gradient Boosting Machine, LightGBM

LightGBM is an algorithmic framework based on Gradient Boosting Decision Tree (GBDT) [70]. This algorithm employs a leaf-wise tree growth strategy, designed for maximum efficiency, offering faster training speeds and minimal memory usage when handling large datasets [71].

During the model training process, decision trees are employed to generate base classifiers, and the weight parameters for each classifier are calculated iteratively. The final

model is then constructed by integrating all the base classifiers and their corresponding weights. This can be expressed by the following equation, where $f_m(X)$ represents the base classifiers and ∂_m denotes the weight parameters for each classifier, as shown in Equation (1) [72].

$$f_m(X) = \partial_1 f_1(X) + \partial_2 f_2(X) + \cdots + \partial_m f_m(X) \tag{1}$$

LightGBM offers superior predictive performance and memory efficiency compared to other classification algorithms [73]. LightGBM is highly effective in handling class imbalance issues and demonstrates strong performance in such scenarios [74]. It significantly enhances the predictive accuracy of Intrusion Detection Systems (IDS) and is notably efficient in flow classification tasks [75]. Moreover, when addressing class imbalance issues, methods such as Synthetic Minority Oversampling Technique (SMOTE) can be employed to adjust the sample distribution, yielding excellent results [76]. In the business domain, numerous studies have demonstrated that LightGBM outperforms other algorithms in terms of precision and F1 scores [77]. Additionally, applications of LightGBM often incorporate the RFM model to include customer purchasing behavior as additional features or utilize RFM combined with clustering algorithms to categorize customers before making predictions [23]. Regardless of the specific application, LightGBM consistently delivers outstanding classification performance. Although the study employs the SMOTE technique to address class imbalance, its application to extremely large-scale datasets—such as the case study involving 185 million transaction records—may hinder model generalization due to computational inefficiency and the potential introduction of noisy synthetic samples.

The interpretability framework proposed in this study involves several key components: enhancing feature dimensions through Byte Pair Encoding (BPE), incorporating expert recommendations, and utilizing Deep Feature Synthesis (DFS) to derive similar features. The final classification task is performed using LightGBM, with precision, recall, and F1 score serving as the evaluation metrics for users. Given that class imbalance is a common issue in business applications, the SMOTE technique is employed to adjust the sample distribution. Additionally, to enhance feature representation and interpretability in persona, the RFM model's monetary value is incorporated into the feature set, providing new interpretative dimensions. There are also some alternative approaches:

- **Stratified Sampling and Cost-Sensitive Learning**: During data preprocessing, oversampling the minority class or incorporating class weights during model training (e.g., using the `scale_pos_weight` parameter) can help mitigate imbalance more efficiently.
- **Ensemble Methods**: Combining undersampling techniques (e.g., RandomUnderSampler) with boosting algorithms (e.g., RUSBoost) can reduce redundancy in the majority class while preserving performance.
- **Application of Focal Loss**: Introducing dynamic weights into the loss function can down-weight the contribution of well-classified (majority class) samples and emphasize learning from hard (minority class) examples.

3. Research Methodology

To propose an interpretable framework to address the issues of customer lists being unacceptable and limited in number in the business domain, the framework will utilize the characteristics of natural language to extract customer labels. These labels will not only possess industry knowledge and interpretability but also serve as personas. By augmenting the data features with these labels and incorporating them into model training, we aim to obtain predictive results for classification tasks.

The method proposed in this study consists of three main steps:
The first step is to identify the objectives and obtain relevant raw data, which represents customer-related feature data.
The second step involves the framework proposed in this study, which first preprocesses the data using Byte-Pair-Encoding (BPE). This process extracts fact tags (F-tags) from the raw data and adds them as feature fields. Experts then define target tags (T-tags) based on the objectives, domain knowledge, and fact tags. Subsequently, Depth-First Search (DFS) is employed to identify tag combinations based on the target tags, and experts determine derivative tags (D-tags) from these results. Finally, the derivative tags are used as the basis for actual class labels, making them the target variables in the model.
The third step involves model prediction and value evaluation. By assessing the model's accuracy, recall, and F1 score, the next steps are determined. If the metrics do not meet the standards, model parameters, DFS parameters, or tags are adjusted, and the model is retrained. If the standards are met, special customers are excluded, and the resulting list is evaluated against the objectives identified in the first step. If the list does not meet the objectives, the process returns to the third step and repeats until the objectives are satisfied.

This section sequentially introduces the implementation details of the proposed framework, named Tag-Framework: Section 3.1 discusses the definition and acquisition of labels and experts. Section 3.2 explains the adjustments made to Byte-Pair Encoding (BPE) to find more root results and analyzes its complexity. Section 3.3 provides detailed explanations and examples of the adjustments made to Depth-First Search (DFS) to find tag combinations similar to the target tags. Section 3.4 analyzes the improvements to DFS in terms of time and space complexity. Section 3.5 describes the evaluation metrics for the model and the process of value evaluation.

3.1. Definition and Acquisition of Labels and Experts

Labels will vary depending on their source: first, apart from being cleaned and preprocessed according to the characteristics of the data, all data must undergo BPE preprocessing. Moreover, the generation of labels relies on the involvement of experts, specifically domain experts. According to the research and definition by Wong et al., domain experts typically lack training in data analysis, visualization, and statistics [78]. Such experts may include sociologists who analyze social phenomena in their work, sales professionals familiar with certain types of products or marketing strategies, or individuals who have deliberately practiced in areas like chess, music, healthcare, or education [79]. These experts possess advanced knowledge, business rules, and processes within their respective fields, serving as the primary source of information for the team [80], but they usually have limited awareness of technical aspects such as visualization or technology [78].

The domain experts in this study were selected based on a case study approach, involving three credit card marketing project managers (PMs) and three managers from the investment products department, each with over eight years of experience in the financial industry. These experts participated in the experiment, model development, and label selection.

Fact Tag (F-tag): Derived from the original data through BPE processing, unsuitable tags and function words are excluded. The results are then matched with extracted fields using regular expressions (regex). If a match is found, the corresponding subword is retained as a fact tag. The fact tag will serve as a feature field in the original data, with the field's value determined by the data characteristics or as a binary $0/1$ value.

Target Tag (T-tag): Determined by experts from the F-tags based on domain knowledge or target characteristics, the T-tag can consist of one or more fact tags. Experts select the T-tag from precise feature fields; otherwise, the selected value may not correspond to any existing feature, and the features are determined through experience from the data to meet the definition of an interpretable model.

Derivative Tag (D-tag): Using the T-tag as the root node, DFS is employed to identify tag combinations that are similar to the T-tag. DFS includes two parameters: depth ($pair_{len}$) and similarity ratio ($pair_{proportion}$). Differences in parameters and simple examples are detailed in Section 3.3. Upon discussion with experts, the tag combination can then be finalized as the derivative tag. Assuming DFS as the Approx function, the tag set and corresponding formulas are represented as shown in Equations (2) and (3).

$$T_{\text{tag}} \subseteq F_{\text{tag}} \tag{2}$$

$$D_{\text{tag}} = \text{Approx}(T_{\text{tag}}, F_{\text{tag}}) \tag{3}$$

According to the formulas, the T_{tag} is a subset of the F_{tag}, while the D_{tag} is derived from tag combinations identified by DFS that are close to the T_{tag}, with the final decision made by experts. In simple terms, the F_{tag} is generated by BPE, the T_{tag} is selected from the F_{tag} and determined by experts, and the D_{tag} is derived from various tag combinations found by DFS, with the final decision also made by experts, as illustrated in Figure 2.

3.2. Adjustment and Complexity Analysis of BPE

One of the objectives of this framework is to enable preprocessing of any natural language data to ensure its generalizability. Therefore, based on the conclusions drawn from the literature review, the BPE tokenization algorithm was selected for adjustment. Using alternative methods, such as splitting by punctuation or whitespace, would reduce generalizability, limiting applicability to other languages and potentially exceeding hardware constraints. Since BPE is not the most frequent tokenization method, even terms that appear only once in the dataset would be transformed into feature columns in subsequent steps, which would significantly impact memory usage [57]. Furthermore, when a term that occurs only once in the dataset becomes a feature column, it leads to a situation where only one record holds a value while all others are 0.

Another reason for adopting BPE in this study's framework is its application in the commercial sector. In addition to general consumer electronics and household products, many financial product names lack spaces and punctuation rules, and most are phrases consisting of single sentences, such as: "Green Power Global ESG Green Power ETF

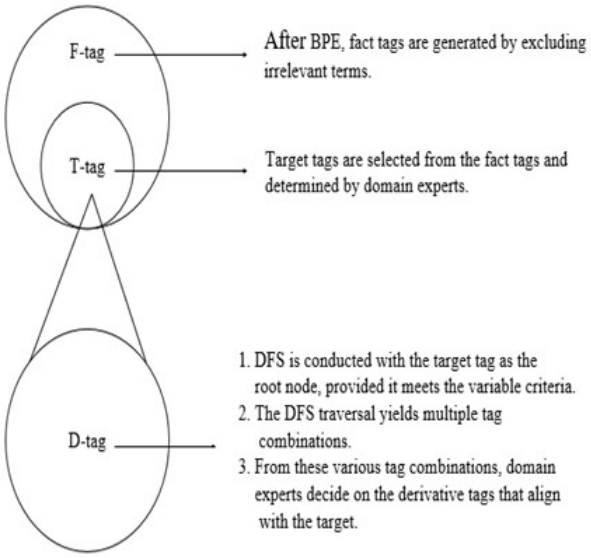

Fig. 2. Label Relationship Diagram

Fund",“US Treasury 20-Year U.S. Government Bond 20+ Year Fund",“Japan Leveraged 2x Tokyo Stock Exchange Daily Fund (Currency Hedged)",“Super Enjoy Life Variable Annuity Insurance",“Triple True Medical Hospitalization Insurance (Outward Type)",“BNP Paribas 12-Month Non-Principal-Protected Structured Note",“Franklin Templeton AI New Technology Fund N (Accumulation) (USD) (Back-End Load)", and “BlackRock Emerging Markets Bond Fund (Stable Distribution) (Monthly Distribution) (AUD Hedged)". If traditional tokenization methods were used, it would either require a customized dictionary or fail to generate reasonable morphemes, which would hinder their conversion into customer labels or feature columns. However, the characteristics of BPE can effectively resolve this issue.

The original BPE algorithm employs 2-gram characters to obtain the most frequent words. To ensure generalizability across languages such as Chinese and English, as well as specialized terminology in various industries, the byte size was modified to iterate over the data in forms of 2, 4, 6, and 8 grams. After extracting the most frequent words using 2-gram characters, the process continues with 4-gram, 6-gram, and 8-gram iterations to capture a wide range of morphemes. This approach ensures that the proposed framework can successfully extract morphemes from datasets in any natural language. The pseudocode is illustrated in Figure 3.

This code is divided into three phases, which will be explained in detail below, along with an analysis of their time and space complexities. Data Processing Phase: This phase includes converting full-width characters to half-width, removing non-essential symbols, and converting all text to lowercase to ensure data consistency. Since each character must be processed individually, the time complexity of this process is proportional to the length of the input data, denoted as $O(n)$.

```
class ExtractFtag:
    Method __init__(eng_series, eng_regex=r'[()|\[\]]'):
        Normalize eng_series to lowercase, remove specified symbols, and convert full-width characters to half-width.
        Split normalized text into words.
        Flatten list of words.

    Method strQ2B(ustring):
        Convert full-width characters in ustring to half-width.
        Return converted string.

    Method get_bpe_vocab_count(str_in_list):
        Calculate and return frequency count of sequences in str_in_list, appending '</w>'.

    Method get_stats(vocab):
        Compute and return frequency of adjacent character pairs in vocab.

    Method merge_vocab(pair, v_in):
        Merge specified character pair in v_in, update vocabulary.
        Return updated vocabulary.

    Method run_bpe(iter_num, length_thresholds=[2, 4, 6, 8]):
        For each length threshold, refine vocabulary using BPE:
            Initialize vocabulary.
            For each iteration:
                Merge highest frequency character pair.
                Record character pair if it meets current length criterion and is not previously recorded.
```

Fig. 3. Extraction of Fact Labels and BPE Pseudocode

Word Frequency Construction Phase: In this phase, the algorithm traverses the entire text to create a table mapping each unique word to its corresponding frequency. If the text contains m unique words, the time complexity of this process is $O(m)$. At the end of this phase, additional space is required to store both the processed text and the word frequency table, resulting in a space complexity of $O(m + n)$.

Iteration and Merging Phase: The core of BPE lies in repeatedly iterating and merging the most frequently occurring pairs of characters until either the iteration limit k is reached or no more character pairs can be merged. During each iteration, the algorithm calculates the frequency of all possible character pairs and selects the most frequent pair for merging. In the worst-case scenario, each iteration involves a comprehensive search through all the words, resulting in a time complexity of approximately $O(m^2)$ per iteration. Therefore, the total time complexity is $O(k \cdot m^2)$. During this phase, the frequency of each character pair is recorded. Assuming the maximum number of character pairs is p, the space complexity for this phase is $O(p)$.

In summary, the overall time complexity of the BPE algorithm designed in this study can be expressed as $O(n + m + k \cdot m^2)$, while the space complexity can be expressed as $O(n + m + p)$. However, in practical applications, adjustments and optimizations will be made based on the characteristics and structure of the data, so the actual runtime and space usage are expected to be lower than the worst-case scenario.

3.3. Adjustments to DFS and Example Explanation

One of the objectives of this framework is to address the issue of a limited number of potential customers. If targets are provided by experts and predictions are based on these targets, the number of individuals on these lists cannot be further increased. Furthermore, due to a lack of interpretability, the list may be rejected by experts or even result in a number lower than what experts would propose based on their experience. Therefore, this study employs DFS to expand the features selected by experts, allowing for the acquisition of features similar to the targets and thereby increasing the number of individuals provided by the model's predictions. DFS was originally designed to traverse an entire graph or tree until all discovered nodes are visited, as detailed in Section 2.4. However, the purpose of employing DFS in this framework is to identify label combinations that approximate T-tag in order to expand target features. Therefore, it is necessary to determine an appropriate approximation ratio through industry knowledge or expert consultations. If the parameters do not meet the specified conditions, the search will not continue further.

Therefore, this study designs two parameters for DFS: depth ($pair_{len}$) and similarity ratio ($pair_{proportion}$), to flexibly identify labels that meet the requirements. Based on Equation (3), the formula (4) representing this design is as follows:

$$D_{\text{tag}} = \underset{T_{\text{tag}}}{\text{DFS}}(F_{\text{tag}}, \text{pair}_{\text{len}}, \text{pair}_{\text{proportion}}) \tag{4}$$

The DFS function will use the initially set T-tag as the root node and search for fact labels that meet the criteria based on the variables $pair_{len}$ and $pair_{proportion}$. Here, a larger value for $pair_{len}$ indicates a greater depth extending downward from the root node, resulting in more nodes. Conversely, a larger value for $pair_{proportion}$ signifies a higher degree of association between the root node and subsequent nodes, leading to a higher proportion of co-occurrence. If $pair_{len}$ is set to 3 and $pair_{proportion}$ is set to 0.7, then the weight of the edge between the root node and the next node must be greater than 0.7, and the number of nodes below the root node must be at least 3 for it to be considered a candidate for further search. Finally, all candidate nodes are traversed, and only after this traversal are they added to the label combination list. This process continues until there are no more candidate nodes. A schematic representation is shown in Figure 4.

As shown in Figure 4, we first assess whether the weight score between the root node and the next node exceeds $pair_{proportion}$. Subsequently, we evaluate whether the total number of nodes in the graph, including the root node, is at least $pair_{len}$. Only if the root node meets both conditions will it be included in the list of candidate nodes. Therefore, the edge weight is 0.2, which does not satisfy condition 1, so this node is ignored in scenario 1; the edge weight is 0.8, satisfying condition 1, and the total number of nodes is 3, which meets condition 2, so this node is added to the list of candidate nodes in scenario 2; the edge weight is 0.8, satisfying condition 1, and the total number of nodes is 4, which meets condition 2, so this node is added to the list of candidate nodes in scenario 3.
Furthermore, if the value of $pair_{proportion}$ is set closer to 1, the association between the root node and the next node will be higher. Similarly, a larger value for $pair_{len}$ indicates a greater number of nodes. Therefore, when both DFS parameters are set to larger values, the conditions for satisfying nodes become more stringent, resulting in fewer label combinations.
Conversely, if both parameters are set to smaller values, the number of label combinations

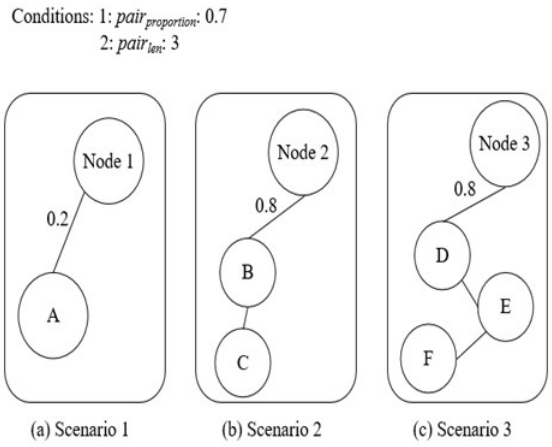

Fig. 4. DFS Search Schematic

will be significantly higher. The actual settings should be determined based on the desired objectives and discussions with experts regarding these variables.

The association between nodes can be confirmed through the weight scores (edge scores) between nodes and the set $pair_{proportion}$. This weight score is calculated based on the values of fact labels and co-occurrence (the frequency with which two features appear together). The calculation is explained as follows.

This score is derived from summing the value of a label with the values of other columns where it appears simultaneously. This approach reflects the correlation between the occurrence of labels together [71]. In other words, if one P-tag frequently co-occurs with another P-tag across multiple records, the score on the edge between these two P-tags will be higher.

After all scores are calculated, normalization will be performed. Since we are primarily interested in the relationship between the highest score and other scores to confirm the association between labels, each label's score is divided by the highest score. This process ensures that all edge weight scores fall within the range of 0 to 1. A score closer to 1 indicates a higher degree of correlation between the two labels.

In summary, after performing a descending order sort, the edge weights can reveal the co-occurrence between each pair of labels. Additionally, the top 10 most common P-tags and labels with 0 co-occurrence, as well as records with a root node score of 0, will be excluded. An example is provided in Table 1.

Using Table 1 as an example, we calculate the co-occurrence weight between the fact label A (node A, root node) and other labels (nodes). We then proceed to traverse the data.

- In the first row, the score for A is 0.5. The corresponding values for B, C, and D are 0.2, 0, and 0.1, respectively. Since the label C has a corresponding value of 0, its co-occurrence with A is also 0, and thus it does not contribute to the weight. The scores for labels B and D will be updated accordingly, with B's score increasing to 0.7 (0.5 + 0.2) and D's score increasing to 0.6 (0.5 + 0.1).

Table 1. Example Table for Weight Calculation

	F-tag			
No.	**A**	**B**	**C**	**D**
1	0.5	0.2	0.0	0.1
2	0.0	0.3	0.4	0.0
3	0.6	0.1	0.2	0.3

- In the second row, the score for A is 0 and the root node is 0, so this record can be ignored.
- In the third row, the score for A is 0.6. The corresponding values for B, C, and D are 0.1, 0.2, and 0.3, respectively. Therefore, the scores for labels B, C, and D will all increase: B's score will be 0.7 (0.6 + 0.1), C's score will be 0.8 (0.6 + 0.2), and D's score will be 0.9 (0.6 + 0.3).
- Following the calculation of the final scores, the sum of the scores obtained by all labels results in the final scores for label A with respect to the other labels: B: 1.4, C: 0.8, and D: 1.5.
- After normalization and descending order sorting, the sequence of labels D, B, and C is determined, with the highest score being 1.5. By dividing all label scores by 1.5, the final co-occurrence scores of labels D, B, and C with label A are obtained as 1, 0.93, and 0.53, respectively. The results indicate that label D has the highest co-occurrence with A, followed by B, while label C has the lowest co-occurrence.
- Therefore, we can create a graph with A as the root node, connected to nodes D, B, and C. The corresponding edge scores are 1, 0.93, and 0.53, respectively, as illustrated in Figure 5.

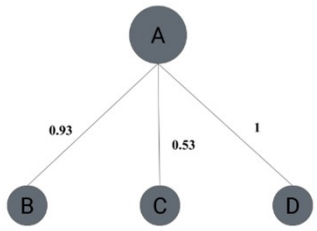

Fig. 5. Graph Generated with A as the Root Node

3.4. Complexity Analysis of DFS

The D-tag code is divided into two phases. The following sections will explain the code and analyze its time and space complexities. Please refer to Figure 6 for details.

```
Class Dtag
    Method __init__(dataFrame)
        self.dataFrame = dataFrame
        self.columns = list of dataFrame's columns
        self.linkNodeReport = empty dictionary

    Method findMajorityColumns(number=10)
        count non-zero values for each column in dataFrame
        sort columns by count in descending order, take top number
        return set of top column names

    Method linkNode(rootName, majoritySet=empty set)
        get scores for rootName
        create matrix excluding rootName column
        initialize scores array with zeros, length = number of columns in matrix

        for each row in matrix
            if root score is non-zero
                find indices of non-zero values in row
                update scores array with non-zero scores + root score

        sort columns in matrix by scores in descending order
        update scores array to match sorted order
        record in linkNodeReport: sorted columns and scores, excluding majoritySet and zeros
```

Fig. 6. Pseudocode for Deriving Labels

Phase One: Column Identification
In this phase, the entire dataset is traversed, examining each row and column to compute the count of non-zero values and perform sorting for each column. If n represents the number of rows (samples) and m denotes the number of columns (features), the time complexity for this method is $O(n * m)$. The time complexity for the sorting operation is $O(n*log_m)$. Since $m*n$ is significantly larger than $m*log_m$, the overall time complexity for the first phase will be dominated by $O(n * m)$. Phase Two: Node Connection

In this phase, calculating the weight scores for each node requires traversing the entire dataset, resulting in a time complexity of $O(n * m)$. Additionally, sorting the nodes has a time complexity of $O(m * log_m)$. Therefore, the total time complexity for the second phase is $O(n * m + m * log_m)$.

In both phases described above, the space complexity is determined by the number of columns in the data and the scores and sorted nodes for each column, resulting in a space complexity of $O(m)$. Overall, the time complexity for processing data with the D-tag class is primarily determined by the traversal of the data. As the number of samples and features in the dataset increases, the computational load will also increase, leading to a time complexity of $O(n * m)$.

Refer to Figure 7 for the analysis of the time and space complexities of label searching and DFS pseudocode. The time complexity of DFS is typically expressed as $O(V + E)$, where V represents the number of nodes and E represents the number of edges. In this study, the DFS code is adapted based on the D-tag, so in the worst-case scenario, if each node is connected to every other node, the number of edges approaches V^2. Consequently, the time complexity is close to $O(V^2)$. Regarding space complexity, the primary considerations are the storage of the node set during traversal and temporary nodes. Therefore, in the worst-case scenario, where all nodes' visit states and paths need to be stored, the space complexity is $O(V)$.

```
Class DtagSearch inherits Dtag
    Method __init__(dataFrame, majoritySet, ptagRoot)
        super().__init__(dataFrame)
        self.majoritySet = majoritySet
        self.ptagRoot = ptagRoot
        initialize adjacencyDict and adjacencyDictRaw as empty dictionaries
        for each root in ptagRoot, link node with root and majoritySet
        prepare adjacencyDict and adjacencyDictRaw for DFS

    Method runDFS(pairLength, pairProportion)
        Define inner method DFS(node, adjacencyDict, visited, tempPair)
            if node not in adjacencyDict
                if tempPair length >= pairLength, record tempPair
                return
            for each connectedNode in adjacencyDict[node]
                if connection strength > pairProportion
                    if connectedNode not in tempPair, check length and uniqueness, then record
                    if connectedNode not visited, mark as visited and recurse with DFS

        for each root in ptagRoot
            initialize pairFeature as empty list
            call DFS with root, adjacencyDict, set with root, and list with root
            record DFS results for root in dfsDtagResult
```

Fig. 7. Pseudocode for Deriving Labels and DFS Search

3.5. Model Metrics and Value Evaluation

When evaluating a model, various metrics are used to compare performance, and specific evaluation criteria are applied to datasets with class imbalance issues, as relying solely on accuracy can be misleading [81]. Typically, a confusion matrix is employed to provide statistical data on true and false results [82], as illustrated by the relationships in the following table.

Table 2. Confusion Matrix for Binary Classification

Predicted Class	Actual Class	
	True (1)	False (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Saito et al. have designed various model evaluation metrics [82]. Commonly used metrics include:

- **Precision**: Focuses on evaluating the predicted positive results.

$$\frac{TP}{TP + FP} \tag{5}$$

- **Recall, TP Rate, Sensitivity**: Concerns the results when the actual class is 1.

$$\frac{TP}{TP + FN} \tag{6}$$

- **F1 Score**: Considers all aspects of the confusion matrix simultaneously.

$$\frac{2 \times TP}{2 \times TP + FP + FN} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

The results of the three metrics are considered better the closer they are to 1 and worse the closer they are to 0. Vujović's research indicates that scores greater than 0.93 are classified as excellent, scores above 0.8 as good, and scores above 0.6 as satisfactory [81]. Given that this study places a higher emphasis on the overall performance of the model, the F1 score is used as the primary evaluation metric. Only models with an F1 score greater than 0.8 are considered for value assessment.

In the value assessment, to simulate the customer lists generated by domain experts based on past experience, the target labels are used to create a benchmark model. The predictions from this model are treated as the customer lists produced by experts, and are compared with those generated by the explainable framework to evaluate the differences.

4. Experimental Results and Analysis

This study aims to propose an explainable framework to address the marketing list issue in the business domain.

4.1. Identify Objectives and Acquire Relevant Data

To ensure that the test data closely reflects the marketing list issue in case studies, the selected dataset must meet the following criteria: the dataset should include fields with natural language, with enough detail to allow domain experts to interpret the characteristics of the data. For example, customer purchase records should contain descriptive statements such as product descriptions, product names, or transaction-related information, and the data should be relevant to the customers. Under these conditions, the dataset aligns with the definition of model explainability [83–86]. Consequently, the Amazon Sales public dataset is chosen for the experiment, which consists of 16 feature fields per record.

Table 3. Description of the Public Dataset(1)

Dataset Name	Feature Count	Number of Records	Source
Amazon Sales	16	1465	[87]

– `product_id`: **String** - Product ID, each product has a unique identifier
– `product_name`: **String** - Product Name, including detailed content
– `category`: **String** - Product Category
– `discounted_price`: **Numeric** - Price after Discount
– `actual_price`: **Numeric** - Actual Price
– `discount_percentage`: **Numeric** - Discount Percentage
– `rating`: **Numeric** - Product Rating Score

- `rating_count`: **Numeric** - Number of User Ratings
- `about_product`: **String** - Product Description
- `user_id`: **String** - User ID of the Reviewer
- `user_name`: **String** - Name of the Reviewer
- `review_id`: **String** - User Review ID
- `review_title`: **String** - Review Title
- `review_content`: **String** - Review Content
- `img_link`: **String** - Product Image URL
- `product_link`: **String** - Product Official Website URL

4.2. Explainable Framework

F-tag The public dataset serves as the raw data for this experiment. After data cleaning and conversion between full-width and half-width characters, the `product_name` and `about_product` fields were selected as the target columns for BPE. This process resulted in the extraction of 723 and 293 subwords, respectively. Due to space constraints, only a portion of the subwords is displayed; see Tables 4 and 5 for detailed information.

Table 4. Subwords for the `product_name` Field

Column Name	Number of Subwords	Sample Subwords	Method of Generation
`product_name`	723	accessor accessories adapter apple/dell apple/dell/lenovo black-heart black/char black/chartre black2v9 blackxcd- capicity cappuccin carecase certified …	BPE

After obtaining the subwords and conducting data cleaning to exclude unnecessary and duplicate subwords, the comparison between the `product_name` and `about_product` fields using regular expressions yielded a total of 177 F-tags, as shown in Table 6. The fact labels do not necessarily represent complete words, as the final labels are determined by the frequency of characters in the vocabulary.

The F-tags are treated as feature columns and added to each data entry, resulting in the original dataset having 177 additional columns, making a total of 193 columns after including the fact labels. The F-tag values are expressed as 0/1, where 1 indicates that the record contains the feature associated with the tag, and 0 indicates its absence. This means

Table 5. Subwords for the `about_product` Field

Column Name	Number of Subwords	Sample Subwords	Method of Generation
about_product	293	anti-rust anti-wrink anti-wrinkle- appropri assistant attachment authenti availability backlight bluetooth bn59-013 borosilic breaking cameramem centimet …	BPE

that a record with a value of 1 indicates that the customer's purchase includes a product description or name that possesses the corresponding tag characteristics. See Figure 8 for illustration.

Table 6. Table for Public Dataset of F-tab

Label Category	Number of Labels	Partial Results of F-tag	Generation Method
F-tag	177	['technolog', 'experienc', 'manufacture', 'temperat', 'addition', 'connectiv', 'material', 'features', 'transmis', 'recharge', … , 'notebook', 'straight', 'thorough', 'attachment', 'guidelin', 'instruction', 'upholster', 'sandwich', 'resistance', 'component']	Based on the roots generated from Tables 4 and 5, data cleaning and regular expression matching were conducted to obtain.

T-tag and label combinations In experiments conducted on public datasets, this study incorporated the recommendations of domain experts—credit card marketing product managers (PM). Three experts collaboratively discussed and selected a target label set from 177 factual labels, resulting in the following labels: sensitivity, lightweight, and durability. The experts expressed the desire to identify label combinations from the public datasets that are similar to sensitivity, lightweight, and durability.

Based on the design of the Depth-First Search (DFS) algorithm, we treat the T-tag as the root node to locate the corresponding D-tags. Through trial and error and expert discus-

Original Columns: total 16 columns				**Additional F-tag:** total 177 columns			
Product_id	…	…	Product_link	notebook	straight	…	component
B07JW9H4JI	…	…	…	0	1	…	0
…	…	…	…	…	…	…	…

Fig. 8. The public dataset with the addition of F-tag columns

sions, the parameters for the depth-first search, namely $pair_{len}$ and $pair_{proportion}$, were set to 3 and 0.4, respectively. The DFS results identified multiple label combinations for the three target labels, specifically (74 sets, 74 sets, and 35 sets). For detailed results, refer to Table 7.
The label combinations represent factual labels that co-occur with the target labels, allowing experts to identify which labels are related to their experience-based target labels. Through these label combinations, experts can better understand the relationships between the target labels and other relevant labels.

D-tag Based on the DFS results from the previous step, multiple label combinations were generated. These results need to be discussed with experts, who will determine two derived label sets based on their experience, the characteristics of the data, and the scope of interpretability. The two sets identified are: (convenient, warranty) and (function, protection).
The selection of derived labels must effectively convey the meaning of the target labels. Therefore, the derived labels (convenient, warranty) and (function, protection) were chosen to correspond to the target labels (sensitivity, lightweight, durability).
The aforementioned D-tags are treated as the actual categories for the model and are assigned to each data point. If the data contains the specified combinations, it is labeled as 1; otherwise, it is labeled as 0, as shown in Figure 9. After labeling the data, it is possible to determine which customers purchased products featuring (convenient, warranty) or (function, protection), or whether the product descriptions include items with (convenient, warranty) or (function, protection).
After labeling, separate models were developed for each derived label combination. As a result, with the two derived label sets, two models were generated: one to predict potential customers for (convenient, warranty) and another for (function, protection).

Total 193 columns							
Product_id	…	Function	Protection	…	convenient	warranty	label
B07JW9H4JI	…	1	1	…	0	0	1
…	…	…	…	…	0	0	0
B072NCN9M4	…	…	…	…	1	1	1

Fig. 9. The public dataset with the addition of F-tag columns

Table 7. Partial results table of DFS label combinations

T-tag	Number of combinations	Partial results of label combinations	Generation method
sensitivity	74	[['sensitiv', 'experienc', 'experience'], ['sensitiv', 'features', 'warranty'], . . . , ['sensitiv', 'features', 'warranty', 'devices.', 'charging', 'compatibl', 'experience'],. . . , ['sensitiv', 'function', 'devices.'], ['sensitiv', 'function', 'protection'], ['sensitiv', 'function', 'features'], ['sensitiv', 'function', 'convenient'], ['sensitiv', 'function', 'compatibl'], ['sensitiv', 'function', 'capacity']]	DFS
lightweight	74	[['lightweight', 'compatibl', 'devices.'], ['lightweight', 'compatibl', 'devices.', 'charging'],['lightweight', 'compatibl', 'devices.', 'charging', 'transfer'], ['lightweight', 'compatibl', 'devices.', 'charging', 'warranty'], ['lightweight', 'compatibl', 'devices.', 'charging', 'warranty', 'manufactur'],['lightweight', 'compatibl', 'devices.', 'charging', 'warranty', 'manufacture'], . . . , ['lightweight', 'function', 'devices.'], ['lightweight', 'function', 'protection'], ['lightweight', 'function', 'features'], ['lightweight', 'function', 'convenient'], ['lightweight', 'function', 'compatibl'], ['lightweight', 'function', 'capacity']]	DFS
durability	35	[['durability', 'charging', 'devices.'], ['durability', 'charging', 'devices.', 'compatibl'], ['durability', 'charging', 'devices.', 'compatibl', 'smartphon'], ['durability', 'charging', 'devices.', 'compatibl', 'warranty'],. . . , ['durability', 'charging', 'devices.', 'transmission'], ['durability', 'charging', 'devices.', 'transmis'] ,['durability', 'charging', 'warranty'], ['durability', 'charging', 'connector'],['durability', 'charging', 'smartphon']]	DFS

4.3. Model Prediction and Value Assessment

The experiments conducted on the public dataset were implemented using Python 3.10.5 in Visual Studio Code, on a MacBook Pro 2023 with an M2 chip and 32GB of RAM. Missing values in the dataset were handled by imputing the mean. For BPE processing, the fields (`product_name`, `about_product`) were converted to half-width characters and lowercase English letters, and full-width spaces were replaced with half-width spaces. Following Sections 4.1 and 4.2, the objective of this section is to predict potential customers for products that possess two sets of derived labels. The machine learning model utilized is LightGBM, as described by Aditya et al. in the literature review [20]. This model not only offers excellent data adaptability but also provides accurate and standardized hyperparameter settings. Hyperparameter adjustments were made based on the parameters and data characteristics discussed by Gupta et al. [88]. Table 8 details the hyperparameter settings for the model.

Table 8. Model Hyperparameter Settings Table

Model Type	Parameter Category	Setting Value
LightGBM	`num_leaves`	60
	`max_depth`	8
	`extra_trees`	True
	`random_state`	42
	`sampling_strategy`	0.8
	`train, val`	0.75, 0.25

Based on the settings in the above table, the model's prediction results and scores are shown in Table 9. For the first set of derived labels (convenient, warranty), the precision is greater than 0.8; for the second set (function, protection), the precision reaches 0.99. Both sets of D-tags meet the model standards outlined in Step 3 (precision, recall, and F1 score all exceeding 0.8). Therefore, the list of predicted customers can be subjected to a value assessment, excluding clients who are refused or blacklisted by the company. The final step is to verify whether the list aligns with the domain experts' objectives, thereby producing a final, interpretable list of potential customers.

Table 9. Model Prediction Scores Table

D-tag	Evaluation Category	Score
(convenient, warranty)	precision	0.833
	recall	0.833
	F1 score	0.833
(function, protection)	precision	0.999
	recall	0.833
	F1 score	0.909

In the value assessment, to simulate the customer list proposed by experts based on past experience, modeling based on the target labels was used as a benchmark for comparison. Since there were no individuals meeting all three criteria (sensitivity, lightweight, durability), three combinations were used for data labeling: (sensitivity, lightweight), (sensitivity, durability), and (lightweight, durability). These combinations served as the actual categories for the model. Predictions were made using the same parameter settings, and the results are presented in the table below.

Table 10. Model Prediction Scores Based on Past Experience Table

Actual Category	Evaluation Category	Score
(sensitiv, lightweight, durability)	precision	0.999
	recall	0.666
	F1 score	0.800

Next, we compared the number of individuals on the lists, as detailed in Table 11. At an F1 score threshold greater than 0.8, the model predictions based on the explainability framework identified 17 potential customers, while the directly modeled numbers based on past experience identified only 2. This result indicates that the explainability framework predicts 8.5 times more potential buyers than the past experience model. It also highlights that the targets selected based on past experience may result in no customers meeting the criteria, which contributes to modeling challenges and low explainability, exemplifying the so-called cold start problem.

Table 11. Value Assessment Table

F1 Score Threshold > 0.8	Explainability Framework	Past Experience
Predicted Number of Buyers	11 (Number of Individuals)	2 (Number of Individuals)
Difference	5.5x	1x

In summary, the results of the small-sample experiments on the public dataset validate the effectiveness of the framework. It is evident that the parameters of DFS significantly impact the number of model predictions. Additionally, the choice of label combinations can be affected by whether the DFS parameters are set too low or too high. Furthermore, the derived labels determined by domain experts based on these label combinations can substantially influence the prediction results, potentially leading to class imbalance. Therefore, it is advisable to apply SMOTE to augment and adjust the sample as needed.

4.4. Interpretability and persona

In the interpretability section, the modeling results for derived labels are analyzed using LIME for local explanations and SHAP for global explanations. First, the explanation plot

for (convenient, warranty) is presented in Figure 10, followed by the explanation plot for (function, protection) as shown in Figure 11.

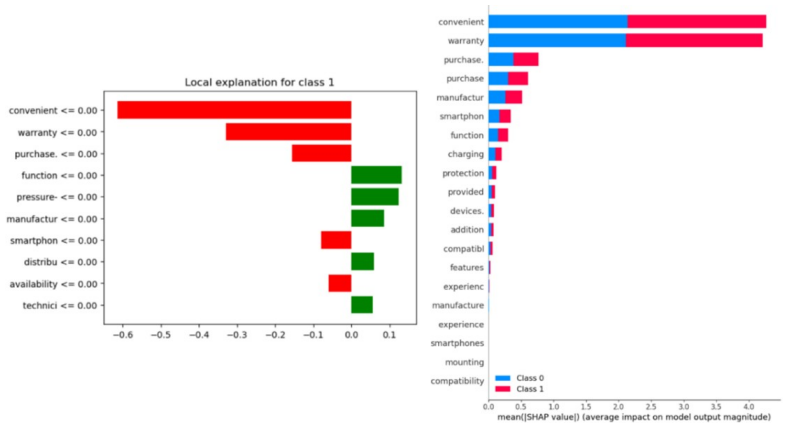

Fig. 10. LIME and SHAP Explanation(1)

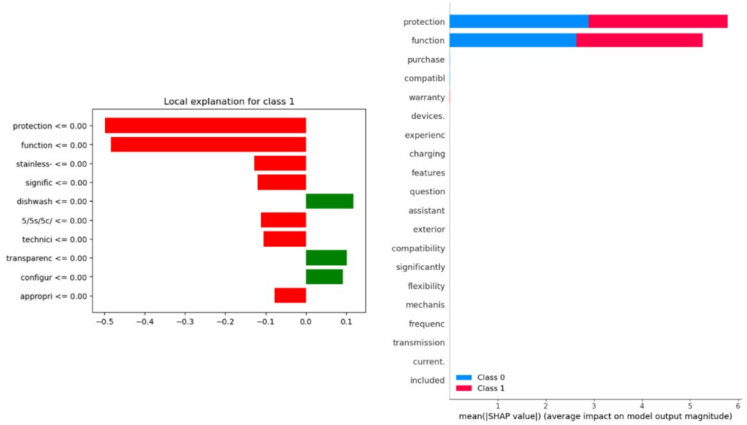

Fig. 11. LIME and SHAP Explanation(2)

Figure 10 shows the LIME explanation on the left. In the LIME plot, the Y-axis is sorted by the features with the greatest impact on the prediction, while the X-axis represents each feature's contribution to the prediction. A negative direction (in red) indicates a lower likelihood of predicting class 1, while a positive direction (in green) suggests a higher likelihood of predicting class 1. Therefore, when the values of 'convenient,' 'warranty,' and 'purchase' are less than or equal to 0, they exhibit a strong negative contribu-

tion (-0.6), influencing the model to not predict class 1. Conversely, if these values are greater than 0, they show a strong positive contribution. 'Function,' 'pressure,' and 'manufactur' have positive contributions, which slightly push the prediction toward class 1.
The right side of Figure 10 presents the SHAP explanation. In the SHAP plot, the Y-axis ranks the features by importance, with the most important ones listed at the top. Blue corresponds to class 0, and red corresponds to class 1. If a feature significantly affects both classes, the corresponding row will display both colors. Therefore, 'convenient' and 'warranty' naturally have a significant impact on the model's output, while 'purchase' also exhibits a certain level of influence. Additionally, 'manufactur' and 'smarphon' show moderate SHAP values, indicating a smaller but still notable effect on the model's predictions.
On the other hand, the label co-occurrence provided by this framework allows for the calculation of co-occurrence scores in the prediction list based on (convenient, warranty), as shown in Table 12.

Table 12. Co-occurrence Scores for D-tags(1)

D-tag	F-tag	Co-occurrence Score
(convenient, warranty)	warranty	1.000
	manufactur	0.372
	convenient	0.344
	manufacture	0.313
	provided	0.288
	features	0.242
	devices.	0.226
	charging	0.217
	compatibl	0.170
	protection	0.161
	experienc	0.155
	experience	0.155
	function	0.145
	addition	0.136
	installation	0.133

The table above lists the top 15 ground truth labels based on co-occurrence ranking. From the table, it can be observed that, in addition to the labels 'convenient' and 'warranty' themselves, the label most closely related to them is 'manufactur,' which has the highest co-occurrence score. This is followed by 'provided' and 'features,' among others. These results help reveal the relationships between labels and can be used to validate the reasonableness of the model's feature explanations in Figure 10. Similarly, in the LIME explanation on the left side of Figure 11, it can be seen that when the values of 'function' and 'protection' are less than 0, they provide a sufficiently negative contribution, leading the model to predict against class 1. In the right-side plot, 'function' and 'protection' exhibit the highest average SHAP value (5.8), meaning the model can determine whether a customer is a potential target based solely on these derived labels. Furthermore, Table

13 shows that the co-occurrence of 'function' and 'protection' is quite high, while the co-occurrence of other labels drops below 0.34, which fully explains the feature ranking results provided by SHAP.

Table 13. Co-occurrence Scores for D-tags(2)

D-tag	F-tag	Co-occurrence Score
(function, protection)	function	1.000
	protection	0.872
	warranty	0.345
	experienc	0.303
	experience	0.303
	charging	0.297
	devices.	0.250
	compatibl	0.228
	features	0.228
	technology	0.218
	technolog	0.218
	functional	0.207
	smartphon	0.202
	performanc	0.202
	convenient	0.180

In the persona analysis, this study randomly selected one instance from the model prediction results for each of the two sets of derived labels and listed the non-zero ground truth labels, as shown in the table below. These results were then analyzed by domain experts.

Table 14. Persona and F-tags(1)

D-tag	F-tag	Value
(convenient, warranty)	experienc	1
	purchase	1
	complete	1
	resistant	1
	comfortabl	1
	lightweight	1
	playback.	1
	experience	1
	convenient	1
	warranty	1

Table 14 indicates that the purchaser values experience in product descriptions and actively makes purchases. They typically prefer products with attributes such as resis-

tance, lightweight, playback capability, convenience, and warranty. Experts suggest that in the prediction list of derived labels associated with (convenient, warranty), the persona corresponds to a tech-savvy professional. Such individuals usually prioritize unique experiences and lightweight products, and they value high-performance gadgets, which aligns with a high relevance to playback and experience. Alternatively, they might be active lifestyle enthusiasts who emphasize comfort, convenience, and durability, indicating that they may engage in outdoor activities or travel and require items that are both durable and portable.

Table 15. Persona and F-tags(2)

D-tag	F-tag	Value
	connector	1
	function	1
	computer	1
	transfer	1
(function, protection)	interfer	1
	compatibl	1
	protection.	1
	convenient	1
	conductor	1

Table 15 reveals that the purchaser values connectors and functionality in product descriptions and desires features such as computers, transfer capabilities, and protection. Experts suggest that in the prediction list of derived labels associated with (function, protection), the persona corresponds to IT professionals, including engineers, remote workers, or technical specialists. Given their emphasis on computer-related labels and their concern for convenience and transfer functions, these characteristics align well with this type of persona. In summary, to ensure that the model's prediction results are sufficiently interpretable, it is necessary to adopt various expert recommendations based on the characteristics and domain knowledge of different datasets. Integrating the RFM model to assign values to ground truth labels can provide experts with additional analytical space in persona. Additionally, using DFS with optimal parameters to explore label combinations and expand the feature scope is essential for the model to identify a larger and more reasonable number of potential customers. In summary, the above steps can be categorized into three major phases: identifying objectives and obtaining relevant data, implementing the interpretability framework, and conducting model prediction and value assessment. The interpretability framework itself comprises three labeling processes: ground truth labels, target labels, and derived labels. The following discussion will explain the general validation process based on these steps.

4.5. Generalizability Validation

Due to the rarity of test sets that fit specific case scenarios, publicly available datasets such as Google Play Store Apps [89] and Amazon Products Sales Dataset 2023 [90] are

not suitable. These datasets either lack user IDs or have product names or descriptions that have already become class labels, making it impossible to enhance interpretability through BPE. However, the effectiveness and feasibility of the framework have been validated through small sample experiments in the preceding sections. To assess the framework's generalizability and reproducibility, this study has selected other similar datasets for experimentation. **Define Objectives and Acquire Relevant Data** In summary, for the generalizability validation, the UCI Online Retail dataset [91], hereafter referred to as Public Dataset 2, was used. Each record in this dataset contains 6 feature columns, with a total of 541,909 online transaction records, as listed below:

Table 16. Description of the Public Dataset(2)

Dataset Name	Feature Count	Number of Records	Source
UCI Online Retail	6	541909	[91]

- `Description`: **String** - Product Name
- `Quantity`: **Value** - quantity purchased for the record
- `InvoiceDate`: **String** - purchase date for the record
- `unit_price`: **Value** - unit price of the product
- `CustomerID`: **Value** - user ID
- `Country`: **String** -user's country of residence

Interpretability Framework

F-tag The selected dataset utilizes the Description column for BPE processing, resulting in 442 word stems. After excluding duplicates and non-applicable stems, a comparison between the stems and the values in the Description column using regular expressions produced 194 ground truth labels. Due to space constraints, only a subset of these stems is displayed in the table below.

After obtaining the ground truth labels, as listed in Table 18, they were incorporated into the dataset as feature columns. Consequently, 194 new columns were added to the dataset, resulting in a total of 200 feature columns.

To represent the purchasing characteristics of customers, the M (Monetary Value) component from the RFM model was used as the value for the ground truth labels. The calculation for the monetary value is as follows: (Total purchase amount by the customer for the products corresponding to the label) / (Number of purchases of that product) / 2. This represents the average spending per product by the customer over a two-year period. Subsequently, the data was merged based on CustomerID, retaining only the CustomerID and ground truth label columns, resulting in a total of 195 columns.

In summary, for consumer A, if there is only one record in the data matching the label "popcor," the total amount for that record is divided by 2 to determine the value of "popcor" for customer A. If consumer B has two purchase records in the data that match the label, the amounts for these two records are summed, divided by the number of records,

Table 17. Partial Word Stems from the Description Column

Field Name	Number of Stems	Partial Stem Results	Generation Method
		artific	
		butterfi	
		butterfli	
		butterfly/	
		campho	
Description	442	...	BPE
		windmil	
		wirele	
		yellow	
		yellow/	
		yuleti	

Table 18. Public Dataset of F-tag Table 2: Label

Lable Categories	Number of Labels	Partial F-tag Results	Generation Method
F-tag	194	['childre', 'strawb', 'butter', 'scandina', 'babush', 'victori', 'dinosa', 'garden', 'sketch', 'popcor', ... , 'revolu', 'toilet', 'square', 'artific', 'glass', 'cabinet', 'candle']	Based on the word stems generated from Table 12, cleaning and regular expression matching were performed to obtain

and then divided by 2 to determine the value of the label "glass" for customer B. The consolidated results will form a new dataset.
Based on the combination of the RFM model and the ground truth labels, we can determine the average spending amount of each customer on products corresponding to the label over a two-year period. This information helps to understand the average purchasing power of each customer in online shopping.

T-tags and combinations of labels The T-tags were determined by domain experts—a credit card marketing project manager—based on the objectives of the task, using the ground truth labels. After discussion among three experts, the target labels were decided to be ('childre', 'candle', 'decorati', 'chocolate'). The experts expressed the aim of identifying label combinations in the public dataset that are similar to those related to children, candles, decorations, and chocolate.
Next, the parameters $pair_{len}$ and $pair_{proportion}$ for the DFS were set. To verify the reproducibility of the framework, these parameters were set to 3 and 0.4, respectively. This configuration means that the label combinations should include at least three nodes and that the co-occurrence between the root node and the subsequent node should be above 0.4. The DFS search results for the target labels were (147 combinations, 147 combinations, 136 combinations, 147 combinations), as detailed in Table 19.

D-tag Based on the experts' experience, data characteristics, and the scope of interpretability, two sets of derived labels were selected from the label combinations: (christ, colour) and (garden, flower). The selection of D-tags must be able to convey the meaning of the target labels. Therefore, the derived labels (Christmas, colorful) and (garden, flowers) were chosen to correspond to the target labels (children, candles, chocolate, decorations).

The D-tags mentioned above will be treated as the actual categories for the model and assigned to each record in the dataset. If a record contains one of the specified combinations, it will be marked as 1; otherwise, it will be marked as 0. After labeling the data, it will be possible to identify which customers purchased products related to either (Christmas, colorful) or (garden, flowers).

Model Prediction and Value Assessment The experimental environment for Public Dataset 2 was the same as that used for Public Dataset 1. Missing values in the dataset were imputed with the mean values. The BPE processing for the column (Description) was standardized to lowercase and half-width characters, with full-width spaces converted to half-width spaces and extra spaces removed.

After completing steps one and two, the task objective is to predict potential customers for products that possess either of the two sets of derived labels. To ensure the stability of the experiment, the machine learning model used is LightGBM, with hyperparameters configured as outlined in Table 8. Using the settings from Table 8, the model prediction results and scores are obtained as shown in Table 20. For the first set of derived labels (Christmas, colorful), the precision of the predictions reaches 0.98; for the second set of derived labels (garden, flowers), the precision is 0.97. Both sets of derived tags meet the model standards from step three (with precision, recall, and F1 score all exceeding 0.8). Consequently, the customer lists predicted by the model can be subjected to value assessment.

Table 19. DFS Label Combination Results(2)

T-tag	Number of Combinations	Partial Label Combination Results	Generation Method
childre	147	[. . . , [’childre’, ’colour’, ’glass.1’, ’christ’, ’decorati’, ’candle.1’, ’candle’, ’black’], [’childre’, ’colour’, ’glass.1’, ’christ’, ’decorati’, ’candle.1’, ’candle’, ’black’, ’cakestand’], . . . , [’childre’, ’colour’, ’glass.1’, ’christ’, ’decorati’, ’drawer’], [’childre’, ’colour’, ’glass.1’, ’garden’]. . . ,]	DFS
candle	147	[. . . , [’candle’, ’candle.1’, ’colour’, ’glass.1’, ’butter’], [’candle’, ’candle.1’, ’colour’, ’glass.1’, ’drawer’], [’candle’, ’candle.1’, ’colour’, ’christ’],. . . .]]	DFS
decorati	136	[[’decorati’, ’christ’, ’colour’], [’decorati’, ’christ’, ’colour’, ’glass.1’], [’decorati’, ’christ’, ’colour’, ’glass.1’, ’silver’], . . . , [’decorati’, ’christ’, ’drawer’], [’decorati’, ’christ’, ’butterf’], [’decorati’, ’christ’, ’lanter’], [’decorati’, ’christ’, ’butterfly’]]	DFS
chocolate	147	[[’chocolate’, ’colour’, ’glass.1’, ’christ’, ’decorati’, ’candle.1’, ’candle’, ’black’, ’garden’], [’chocolate’, ’colour’, ’glass.1’, ’christ’, ’decorati’, ’candle.1’, ’candle’, ’black’, ’garden’, ’cakestand’], . . . , [’chocolate’, ’colour’, ’glass.1’, ’christ’, ’drawer’], [’chocolate’, ’colour’, ’glass.1’, ’christ’, ’butterf’],. . .]	DFS

Table 20. Model Prediction Scores Table 2

D-tag	Evaluation Categories	Score
(christ, colour)	precision	0.983
	recall	0.8093
	F1 score	0.887
(garden, flower)	precision	0.978
	recall	0.866
	F1 score	0.919

In the value assessment, the T-tags (‘childre’, ‘candle’, ‘decorati’, ‘chocolate’) are used as a benchmark based on the experts’ past experience for comparison. The T-tags are treated as the actual categories, and the same parameter settings are employed for modeling. The prediction scores are listed in the table below.

Table 21. Prediction Scores Table 2 Based on Historical Experience

Actual categories	Evaluation Categories	Score
(‘childre’, ‘candle’, ‘decorati’‘, ‘chocolate’)	precision	0.955
	recall	0.773
	F1 score	0.85

At an F1 score level greater than 0.8, a comparison of the number of potential customers was conducted. The interpretability framework identified 512 potential buyers, while the number predicted based on historical experience modeling was 68. The results indicate that the number of potential customers predicted by the interpretability framework is 7.5 times that of the historical experience modeling, as detailed in the table below.

Table 22. Value Assessment(2)

F1 Score >0.8	Interpretability Framework	Historical Experience
Predicted Number of Buyers	512 (Number of Individuals)	68 (Number of Individuals)
Difference	7.5x	1x

Interpretability and Persona In the interpretability section, the modeling results for the derived labels were analyzed using LIME for local explanations and SHAP for global explanations. First, the explanation diagram for (Christmas, colorful) is shown in Figure 12. Following that, the explanation diagram for (garden, flowers) is presented in Figure 13.

In Figure 12, the left side shows the LIME explanation. In the LIME diagram, when the value of ”colour” is less than or equal to 0, it has a strong negative contribution (-0.4), influencing the model not to predict class 1. Conversely, if the value is greater than 0, it has a strong positive contribution. Additionally, when ”christ” exceeds 7.07 and ”scandina” exceeds 0.49, they provide positive contributions, pushing the prediction towards class 1. Even ”silver” values between 0 and 4.1 have a positive contribution towards class 1. On the right side of Figure 4-6, the SHAP explanation is presented. The SHAP diagram indicates that ”colour” and ”christ” have significant impacts on the model’s output, while ”scandina,” ”silver,” and even ”garden” and ”decorati” also exhibit certain degrees of influence. On the other hand, the co-occurrence scores in the prediction list were calculated based on the label pairs (Christmas, colorful) as provided by this framework, as

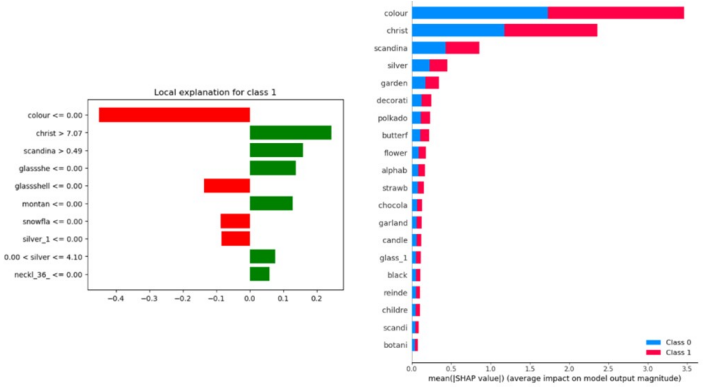

Fig. 12. Public Dataset 2: LIME and SHAP Explanation Diagrams 1

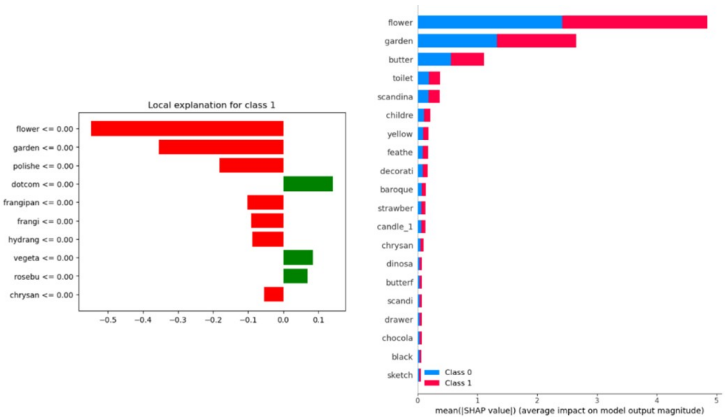

Fig. 13. Public Dataset 2: LIME and SHAP Explanation Diagrams 2

shown in Table 23. The table indicates that, aside from the labels themselves, the label most strongly associated with (Christmas, colorful) is ”flower,” followed by ”cakestand,” which also ranks high in co-occurrence. Conversely, ”garden” and ”silver” exhibit lower levels of co-occurrence, which slightly diverges from the explanation results in Figure 12.

In Figure 13, the LIME explanation on the left indicates that when the values of ”flower,” ”garden,” and ”polishe” are less than or equal to 0, they have a sufficient negative contribution, causing the model to not predict class 1. Conversely, when ”dotcom” is less than or equal to 0, it has a positive likelihood of predicting class 1. On the right side of the figure, the SHAP explanation shows that ”flower” and ”garden” have the highest average SHAP values (5), while ”butter” and ”toilet” also have a notable impact on the model in this classification. Additionally, Table 24 shows that, aside from ”flower” itself, ”christ” and the derived labels appear together quite frequently, followed by the ”garden” label.

Table 23. Based on D-tags, Co-occurrence Value(3)

D-tag	F-tag	Co-occurrence Scores
(christ, colour)	christ	1.000
	colour	0.672
	flower	0.374
	cakestand	0.274
	black	0.221
	glass_1	0.179
	garden	0.164
	drawer	0.158
	silver	0.123
	cabinet	0.112
	candle	0.1096
	candle_1	0.1096
	botani	0.1092
	decorati	0.1035
	alphab	0.1035

The co-occurrence scores for ”black,” ”colour,” ”darwer,” and ”cakestand” with the derived labels are all greater than 0.4, indicating a high level of co-occurrence. This finding slightly diverges from the feature explanation results in Figure 13, but it still provides experts with greater interpretability and comprehensibility of the model's prediction results.

In the persona section, this study randomly selected one instance from the model prediction results for each of the two sets of derived labels and listed the factual labels with RFM values greater than 0, as shown in the table below. These were then analyzed by domain experts.

From Table 25, it can be seen that the purchaser places significant importance on the descriptive features of the product, such as ”decorati” and ”silver,” and even prefers descriptions that include ”feathe.” The first three features have high consumption amounts and frequencies over the past two years, with scores exceeding 5.9. Additionally, the purchaser values specific themes (”charlot”) and stripe elements. Experts suggest that, for the derived labels corresponding to (Christmas, colorful), the persona is likely that of a home decoration enthusiast. This individual enjoys beautifying and decorating items and may be a homemaker, interior designer, or a working professional who values ritualistic elements, potentially even an admirer of Scandinavian style (scandina, scandinavi, scandi).

From Table 26, it is evident that the purchaser has a strong preference for products described with ”doughn” in the product descriptions. The consumption frequency and amount over the past two years for this label are significantly higher, reaching a ratio of 12.5, far surpassing the second most frequent label, ”flower.” Other related labels include ”breakf” (breakfast), ”citron” (lemongrass), and ”thermo” (temperature control) products. Experts suggest that, for the derived labels corresponding to (flower, garden), the persona is likely that of a home baking and breakfast enthusiast. The labels indicate that the customer enjoys baking and preparing breakfast, and could also be a baker selling homemade

Table 24. Based on D-tags, Co-occurrence Value(4)

D-tag	F-tag	Co-occurrence Scores
(flower, garden)	flower	1.000
	christ	0.840
	garden	0.791
	black	0.587
	colour	0.579
	drawer	0.461
	cakestand	0.445
	botani	0.355
	cabinet	0.329
	glass_1	0.273
	candle	0.251
	candle_1	0.251
	butter	0.246
	butterf	0.239
	stripe	0.223

Table 25. Persona and F-tags(3)

D-tag	F-tag	RFM value
(christ, colour)	decorati	7.5
	silver	6.36
	feathe	5.9
	charlot	4.25
	stripe	4.12
	colour	3.75
	christ	3.26
	scandina	1.25
	scandinavi	1.25
	scandi	1.25

Table 26. Persona and F-tags(4)

D-tag	F-tag	RFM value
(flower, garden)	doughn	12.50
	flower	4.20
	garden	4.19
	breakf	4.08
	citron	3.97
	citronel	3.97
	thermo	2.97
	candle	2.86
	candle_1	2.86
	black	1.77

goods. Additionally, the customer might be a nature lover or a retiree, having ample time for floral arrangements and baking activities.

In summary, the experimental results with the UCI Online Retail dataset (Public Dataset 2) demonstrate the generalizability and reproducibility of the explainability framework. It is also observed that when the content of the target labels is more relevant, the label combinations become highly correlated, resulting in numerous similar label combinations within the nodes.

When the target labels are ('childre', 'candle', 'decorati', 'chocolate'), domain experts interpret the relevance from a natural language perspective, noting that candles and decorations are related, chocolate and children are related, and chocolate color is also related to decorations. This aligns with the setting of target labels being interrelated. Consequently, the derived labels that were determined—(Christmas, colorful) or (garden, flowers)—are also relevant. Decorations are associated with candles and Christmas, as well as being colorful and even with flowers. Children are related to both Christmas and gardens. Many garden decorations fall within the category of decorations, indicating that despite different label combinations, they remain highly related. This is evident from Tables 22 and 23, where the co-occurrence rankings of the two sets of derived labels both reflect the presence of the other set of derived labels.

Additionally, the experimental results indicate that directly modeling using the expert-provided target labels can lead to issues such as a small number of customers and a lack of explanatory power, potentially even causing cold start problems. However, by employing the described method, which utilizes DFS to identify co-occurrence and reasonably extend the most relevant features, it is possible to broaden the scope of similar targets and increase the number of potential customers in the model. Moreover, assigning values to factual labels through the RFM model and computing the co-occurrence of labels based on derived labels can effectively enhance the interpretability of the model's potential customer list for experts. This approach increases trust in the model's results and facilitates the analysis of personas.

In summary, by integrating domain knowledge with natural language understanding, experts can provide additional explanations for the labels in the prediction results and reassess whether their target settings are aligned with the outcomes. After verifying the generalizability and reproducibility of the framework, Section 4.6 applies the same steps for case validation.

4.6. Case validation

After validating the effectiveness and generalizability of the interpretability framework using publicly available datasets, this study further applies the interpretability framework to cases in the financial industry where marketing lists have been rejected.

Identify objectives and obtain relevant data For this study's case, marketing lists, transaction data, and customer datasets provided by one of the top three financial holding companies in Taiwan were used, as detailed in Table 27. After data integration and consultation with domain experts, a total of 185,199,048 transaction records were obtained, with each record containing four feature columns, as listed below.

- `customer_id`: **String** - Customer Code for Each Transaction
- `monetary`: **Numeric** - Transaction Amount for Each Entry

Table 27. Dataset Description Table

Dataset Name	Number of Features	Number of Records	Data Period	Purpose
Transaction Data	4	185,199,048	2021/8-2023/8	Obtain F-tags
Customer Information	4	7,314,643	2023/1-2023/8	Training and Prediction
Marketing Lists	N/A	N/A	N/A	Define Target Labels

– `date`: **String** - Year and Month of Each Transaction
– `remark`: **String** - Transaction Notes

Interpretability Framework

F-tag After processing the transaction notes field using Byte Pair Encoding (BPE), a total of 1,465 word stems were generated. Following the removal of duplicates, non-descriptive, and irrelevant stems, regular expression matching was conducted, resulting in 319 ground truth labels. Due to restrictions imposed by financial regulations and personal data protection laws [92, 93], only a portion of the ground truth labels can be provided, as detailed in Table 28.

Table 28. Partial F-tags Table for the Case Dataset

Label Category	Number of Labels	Partial Results of F-tags	Generation Method
F-tag	319	['Stocks', 'Allowance', 'Taipei City', 'Steak', 'Streaming Media Platform', 'Dividend', 'Holiday Cash Flow', ... , 'Policy Loans', 'Funds', 'Credit Card', ...]	Based on the word stems generated by BPE, data cleaning and regular expression matching were performed to obtain

In the customer information dataset, after data integration, a total of 7,314,643 records were obtained. Thus, as of the end of August 2023, there were 7,314,643 customers. This dataset contains four feature columns, consistent with those in the transaction data. The next step involves incorporating the 319 ground truth labels generated by Byte Pair Encoding (BPE) as additional feature columns into the customer dataset, resulting in a total of 323 feature columns.

To represent each customer's spending level, the F-tag values in the case study are expressed using the RFM model's expenditure amount, calculated as (Total Transaction Amount) / (Total Number of Transactions). The total number of transactions is calculated as the sum of the occurrences of transactions corresponding to the label within the customer's two-year transaction history, indicating the total number of transactions involving that label over the two years. The total transaction amount is computed using the 'monetary' field from the transaction data, reflecting the total amount of transactions associated with the label over the two-year period, expressed in ten thousand units.

According to the RFM description provided above. When calculating the fund label field for Customer A, the relevant transaction records associated with the fund label in the transaction data are used. The total transaction amount, expressed in ten thousand units, is then divided by the total number of transactions, and the resulting value is entered into Customer A's fund label field. Similarly, when calculating the credit card label field for Customer B, the relevant transaction records associated with the credit card label in the transaction data are used. The total transaction amount, expressed in ten thousand units, is divided by the total number of transactions, and the resulting value is entered into Customer B's credit card label field.
Through the application of the RFM model as described above, the F-tag values provide insights into the relationship between the total transaction amount and the number of transactions associated with each label for the customer up to the end of August 2023.

T-tag and combinations of labels The marketing list dataset was pre-screened by domain experts, specifically the investment products department manager, to define the target T-tags. Based on the F-tags derived from the transaction data, and following discussions among the three experts, a total of eight T-tags were determined, including but not limited to: (Dividend, Year-End Bonus, Holiday, Retirement Pension, etc.). The remaining four target labels are related to the descriptions of investment products and internal product sensitivity, and therefore, not all T-tags can be disclosed.
With the target labels established, this case study sets the DFS parameters $pair_{len}$ and $pair_{proportion}$ to 3 and 0.6, respectively, to identify similar label combinations with the target label as the root node. The aim was to find combinations where there are at least 3 nodes and the co-occurrence between labels exceeds 0.6. The DFS search results, sorted by the number of combinations, are as follows: (119 combinations, 16 combinations, 108 combinations, 3 combinations, 133 combinations, 59 combinations, 60 combinations, 43 combinations). To avoid violations of financial regulations and personal data protection laws [93], and to prevent the illegal misuse of personal information, the specific label combinations cannot be disclosed.

D-tag Following a discussion among three experts, including the investment products department manager, it was decided that there are 11 derived label combinations. Since the D-tags involve aspects such as the characteristics of investment products, customer response rates, consumer habits, subscription outcomes, and company sensitivity, disclosing these could potentially lead to the inadvertent exposure of specific customer characteristics and legal issues. To prevent misunderstandings related to the use of personal data and any illegal intentions [93], the D-tags cannot be disclosed.

Model Prediction and Value Assessment The experiments for this case study were implemented using Python 3.6 and PySpark 1.4 in the Cloudera Data Science Workbench. The hardware used consisted of an Intel Xeon 64-bit processor with 16 cores and 128GB of RAM.
As described in Sections 4.3 and 4.5, the LightGBM model was used with D-tags treated as the actual class for training. The prediction task involved identifying investment product purchasers between September 1, 2023, and October 31, 2023. The results are detailed in the table below, with an average precision of 0.941, a recall of 0.899, and an F1 score of 0.938 for the 11 D-tags.

Table 29. Model Prediction Scores

D-tag	Evaluation categories	Score
11 D-tag sets	Average precision	0.941
	Average recall	0.899
	Average F1 score	0.938

Since all three evaluation metrics exceed the threshold of 0.8 set within the interpretability framework, the process moves on to the next step of value assessment. This study uses A/B testing to validate actual effectiveness. A list of potential customers for the same investment products identified by domain experts based on past experience is used as the benchmark (Group A), consisting of 91,018 individuals. The proposed framework predicts 226,998 potential buyers (Group B), which is 2.493 times higher than the expert-provided list.

Digital advertisements for investment products were targeted to the customer lists provided by each group (A and B) through an advertising deployment system. Under identical advertising copy, this study tracked and compiled data only for customers within one month after the advertisement was deployed. Due to the challenges in objectively defining transaction tracking and effectiveness of investment products, data on customer repurchase rates or cost-effectiveness could not be obtained for comparison. Therefore, only customer response rates and the number of respondents were compared.

The test results indicate that the customer response rate is 3.8 times higher than that of the list provided by industry experts, and the total number of responses is 9 times greater, as detailed in Table 30.

Table 30. Value Assessment Table-3

Under an F1 Score Greater Than 0.8	Interpretability Framework	Previous Experience
Predicted Number of Purchasers	226,998 (Number of individuals)	91,018 (Number of individuals)
Difference	2.493x	1x
Customer Response Rate Within One Month of Advertisement Deployment	3.8x	1x
Number of Responses Within One Month of Advertisement Deployment	9x	1x

Interpretability and Persona Due to regulatory constraints, it is not possible to display LIME and SHAP model visualizations. However, by utilizing an explainability framework and anonymizing the data, some co-occurrence labels can still be presented. The order of the labels does not reflect their actual ranking, and identifiable labels have been omitted, as shown in the table below.

Table 31. Table-5 of Co-occurrence Scores for D-tags

D-tag	F-tag	Co-occurrence Score
(ETF A, ETF B, ETF C)	technology industry	0.719
	regular installment plans	0.644
	group dining	0.355
	Shin Kong Mitsukoshi	0.181
	afternoon tea	0.140

The table presents a randomly selected de-identified label from 11 derived label sets for co-occurrence analysis. It reveals that among the labels associated with the three ETFs, aside from the strong correlation with their own labels, the ”technology industry” fact label exhibits a particularly high level of co-occurrence. Additionally, terms such as ”regular installment plans” frequently appear in the descriptions of these products, attracting a diverse range of customers. In the persona analysis, Table 32 indicates that the purchaser

Table 32. Persona and F-tags (5)

D-tag	F-tag	RFM value
(ETF A, ETF B, ETF C)	rent	9.50
	afternoon tea	8.27
	group dining	6.83
	technology industry	4.29
	steak	2.21

frequently has the label ”rent” noted in their account records, and they tend to spend a significant amount on afternoon tea and group dining. This individual is also likely employed in the technology industry. Experts suggest that the profile aligns closely with that of an engineer working in a science park. Alternatively, it could describe a financially savvy individual, possibly a landlord, who specializes in managing rental properties. This person predominantly invests in technology-related portfolios, which may explain the high frequency and expenditure on afternoon tea and group dining.

The validation results not only significantly increased customer response rates, response volume, and the number of targeted individuals but also enabled domain experts to interpret the results through personas derived from the data. This enhances data transparency and model interpretability, allowing experts to better understand the predictive outcomes and reducing skepticism toward the model. These findings further demonstrate the practical feasibility of the explainability framework and its potential for increased profitability.

5. Conclusions and Discussion

This study addresses cases where the customer lists generated by the model were deemed unacceptable. Specifically, it focuses on issues such as the inability to explain the lists, the features of the lists failing to persuade domain experts and decision-makers, and the limitations on the number of individuals in the lists based on past experience. To address these challenges, this research proposes an explainability framework as a solution.

The framework integrates BPE and a three-tier labeling system to enhance the interpretability of the model's results. Fact labels expand the feature dimensions of customer data, enabling users to perceive the functional dimensions of the data through natural language [94]. Industry experts can select target labels from the fact labels and determine derived labels for the model's actual categories based on label combinations identified by DFS. This allows domain experts to fully understand that the customer lists predicted by the model are generated from natural language features, and further explanations can be provided through personas. This approach fully satisfies the transparency, comprehensibility, and interpretability requirements of XML [36]. Moreover, the relationship between derived and target labels can be further elaborated to provide contextual explanations, aligning with the definition of XAI [37].

This study develops a general explainability framework to address challenges in the business domain, where industry experts or decision-makers may reject model-generated potential customer lists, and where the number of marketing list recipients is limited by past experience. The experimental results show that:

1. By labeling data with natural language, this framework enhances data interpretability for any user and produces comprehensible potential customer lists. It effectively increases response rates and the number of recipients on the lists, offering a higher chance of generating greater corporate profits.
2. Although designed for the business domain, the framework is repeatable and generalizable, applicable to any dataset involving natural language. It can be adopted to enhance both the feature dimensions and readability of data, helping users better understand its behavior and characteristics.
3. Grounded in the experience of domain experts and decision-makers, this framework successfully transfers prior knowledge and domain expertise into the model. In the future, experts can confidently leverage technological advancements, and managers can more easily monitor changes in customer consumption patterns and habits.

The proposed framework can be further extended to improve its adaptability across industries and alignment with cutting-edge technologies through the following directions:

- Multimodal Data Fusion: In retail scenarios, integrating product images (e.g., clothing design sketches) with textual reviews via vision-language models such as CLIP can generate cross-modal tags, thereby enriching user profiling.
- Federated Learning: In privacy-sensitive domains such as finance and healthcare, distributed model training enables collaborative modeling (e.g., credit risk assessment across banks) while preserving user data privacy by avoiding raw data exchange.
- Replacing DFS with Graph Neural Networks (GNNs): Instead of heuristic DFS-based search, the label co-occurrence structure can be directly modeled using GNNs. Graph Attention Networks (GAT), in particular, can capture complex inter-label relationships (as discussed in Section 4.6.4), offering a more expressive alternative.

- Improving Interpretability with Large Language Models (LLMs): Prompt engineering techniques can leverage models like GPT-4 to automatically generate semantic explanations for tags (e.g., defining the business meaning of "durability"), thereby reducing reliance on domain experts.

While the proposed framework demonstrates strong performance, several limitations should be acknowledged:

- Dependence on Data Quality: The framework's effectiveness relies heavily on the completeness and accuracy of natural language fields (e.g., product descriptions). High levels of noise—such as spelling errors or ambiguous expressions—may lead to suboptimal tag generation by BPE. For example, as shown in Table 4, subwords like "cappuccin" require manual correction to align with intended semantics.
- Expert Involvement Overhead: The DFS-generated tag combinations require manual filtering by domain experts. As illustrated in Section 4.2.3, only 2 out of 74 candidate D-tag combinations were selected for downstream use, which limits the framework's automation in knowledge-scarce scenarios.
- Computational Bottlenecks: Both BPE and DFS may incur substantial memory and time costs when applied to large-scale datasets, such as the 185 million transaction records used in the case study. Distributed computing frameworks (e.g., Apache Spark) or approximate algorithms may be necessary to improve scalability.
- Limited Adaptability to Dynamic Data: The current framework does not account for data distribution shifts over time (e.g., evolving consumer preferences). Future work should explore online learning mechanisms to periodically update the tag taxonomy and maintain robustness under dynamic conditions.

Acknowledgments. This research was supported in part by the National Science and Technology Council, R.O.C. under grant MOST 110-2221-E-007-107-MY3, NSTC 112-2221-E-007-086 and NSTC 113-2221-E-007-117-MY3.

References

1. A. Rakipi, O. Shurdi, and J. Imami, "Utilization of data mining and machine learning in digital and electronic payments in banks," *Corporate and Business Strategy Review*, vol. 4, no. 4, pp. 243–251, 2023.
2. W. Yeh, M. Chuang, and W. Lee, "Uniform parallel machine scheduling with resource consumption constraint," *Applied Mathematical Modelling*, vol. 39, no. 8, pp. 2131–2138, 2015.
3. W. Yeh and S. Wei, "Economic-based resource allocation for reliable grid-computing service based on grid bank," *Future Generation Computer Systems*, vol. 28, no. 7, pp. 989–1002, 2012.
4. K. Pousttchi and M. Dehnert, "Exploring the digitalization impact on consumer decision-making in retail banking," *Electronic Markets*, vol. 28, no. 3, pp. 265–286, 2018.
5. P. Angelov, E. Soares, R. Jiang, N. Arnold, and P. Atkinson, "Explainable artificial intelligence: an analytical review," *WIREs Data Mining and Knowledge Discovery*, vol. 11, no. 5, p. 2021, 2021.
6. J. Achiam and et al., "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.
7. H. Touvron and et al., "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

8. "Introducing llama: A foundational, 65-billion-parameter language model." https://ai.meta.com/blog/large-language-model-llama-meta-ai/. Accessed: 2024/4/2.
9. S. Spatharioti, D. Rothschild, D. Goldstein, and J. Hofman, "Comparing traditional and llm-based search for consumer choice: A randomized experiment," *arXiv preprint arXiv:2307.03744*, 2023.
10. H. Corley, J. Rosenberger, W. Yeh, and T. Sung, "The cosine simplex algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 27, pp. 1047–1050, 2006.
11. B. Arcila, "Is it a platform? is it a search engine? it's chatgpt! the european liability regime for large language models," *Journal of Free Speech Law*, vol. 3, p. 455, 2023.
12. W. Yeh, "Novel binary-addition tree algorithm (bat) for binary-state network reliability problem," *Reliability Engineering and System Safety*, vol. 208, p. 107448, 2021.
13. M. Karpinska and M. Iyyer, "Large language models effectively leverage document-level context for literary translation, but critical errors persist," in *Proceedings of the Eighth Conference on Machine Translation*, 2023.
14. W. Yeh, "A new branch-and-bound approach for the n/2/flowshop/f+ cmax flowshop scheduling problem," *Computers & Operations Research*, vol. 26, no. 13, pp. 1293–1310, 1999.
15. A. Thirunavukarasu, D. Ting, K. Elangovan, L. Gutierrez, T. Tan, and D. Ting, "Large language models in medicine," *Nature Medicine*, vol. 29, no. 8, pp. 1930–1940, 2023.
16. C. Luo, B. Sun, K. Yang, T. Lu, and W. Yeh, "Thermal infrared and visible sequences fusion tracking based on a hybrid tracking framework with adaptive weighting scheme," *Infrared Physics & Technology*, vol. 99, pp. 265–276, 2019.
17. A. Mbakwe, I. Lourentzou, L. Celi, O. Mechanic, and A. Dagan, "Chatgpt passing usmle shines a spotlight on the flaws of medical education," *PLOS Digital Health*, vol. 2, no. 2, p. e0000205, 2023.
18. N. Chiliya, G. Herbst, and M. Roberts-Lombard, "The impact of marketing strategies on profitability of small grocery shops in south african townships," *African Journal of Business Management*, vol. 3, no. 3, p. 70, 2009.
19. T. Damrongsakmethee and V.-E. Neagoe, "Data mining and machine learning for financial analysis," *Indian Journal of Science and Technology*, vol. 10, no. 39, pp. 1–7, 2017.
20. R. Aditya and D. Satria, "Optimizing bank marketing strategies through analysis using lightgbm," *CoreID Journal*, vol. 1, no. 2, pp. 58–65, 2023.
21. S. Shim, M. Eastlick, and S. Lotz, "Search-purchase (s-p) strategies of multi-channel consumers," *Journal of Marketing Channels*, vol. 11, no. 2-3, pp. 33–54, 2004.
22. A. Faria and W. Wellington, "Validating business gaming: Business game conformity with pims findings," *Simulation & Gaming*, vol. 36, no. 2, pp. 259–273, 2005.
23. P. Chate, *Behavioral Modelling of Customer Marketing Patterns and Review Prediction Using Machine Learning Techniques*. PhD thesis, National College of Ireland, Dublin, 2022.
24. M. Muslim, Y. Dasril, A. Alamsyah, and T. Mustaqim, "Bank predictions for prospective long-term deposit investors using machine learning lightgbm and smote," *Journal of Physics: Conference Series*, vol. 1918, no. 4, p. 042143, 2021.
25. E. Broek, A. Sergeeva, and M. Huysman, "When the machine meets the expert: An ethnography of developing ai for hiring," *MIS Quarterly*, vol. 45, no. 3, 2021.
26. T. Jovanov and M. Stojanovski, "Marketing knowledge and strategy for smes: Can they live without it?," in *Thematic Collection of papers of international significance: Reengineering and entrepreneurship under the contemporary conditions of enterprise business*, pp. 131–143, 2012.
27. Y. Huang, M. Zhang, and Y. He, "Research on improved rfm customer segmentation model based on k-means algorithm," in *2020 5th International Conference on Computational Intelligence and Applications (ICCIA)*, 2020.

28. E. Soares, P. Angelov, B. Costa, and M. Castro, "Actively semi-supervised deep rule-based classifier applied to adverse driving scenarios," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019.
29. R. Blanco and C. Lioma, "Graph-based term weighting for information retrieval," *Information Retrieval*, vol. 15, no. 1, pp. 54–92, 2011.
30. X. Xie, J. Niu, X. Liu, Z. Chen, S. Tang, and S. Yu, "A survey on incorporating domain knowledge into deep learning for medical image analysis," *Medical Image Analysis*, vol. 69, p. 101985, 2021.
31. C. Deng, X. Ji, C. Rainey, J. Zhang, and W. Lu, "Integrating machine learning with human knowledge," *iScience*, vol. 23, no. 11, p. 101656, 2020.
32. V. Belle and I. Papantonis, "Principles and practice of explainable machine learning," *Frontiers in Big Data*, vol. 4, p. 688969, 2021.
33. S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
34. A. Arrieta and et al., "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.
35. Z. Liu, W. Yeh, K. Lin, C. Lin, and C. Chang, "Machine learning based approach for exploring online shopping behavior and preferences with eye tracking," *Computer Science and Information Systems*, vol. 21, no. 2, pp. 593–623, 2024.
36. R. Roscher, B. Bohn, M. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," *IEEE Access*, vol. 8, pp. 42200–42216, 2020.
37. H. Chia, "The emergence and need for explainable ai," *Advances in Engineering Innovation*, vol. 3, no. 1, pp. 1–4, 2023.
38. E. Soares, P. Angelov, S. Biaso, M. Froes, and D. Abe, "Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification," *MedRxiv*, 2020.
39. F. Morais, A. Garcia, P. Santos, and L. Ribeiro, "Do explainable ai techniques effectively explain their rationale? a case study from the domain expert's perspective," in *2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2023.
40. J. Dressel and H. Farid, "The accuracy, fairness, and limits of predicting recidivism," *Science Advances*, vol. 4, no. 1, p. eaao5580, 2018.
41. A. Smith-Renner, R. Rua, and M. Colony, "Towards an explainable threat detection tool," in *IUI Workshops*, 2019.
42. S. Mathews, "Explainable artificial intelligence applications in nlp, biomedical, and malware classification: A literature review," in *Advances in Intelligent Systems and Computing*, pp. 1269–1292, Springer International Publishing, 2019.
43. A. Das and P. Rad, "Opportunities and challenges in explainable artificial intelligence (xai): A survey," *arXiv preprint arXiv:2006.11371*, 2020.
44. S. Murindanyi, B. Mugalu, J. Nakatumba-Nabende, and G. Marvin, "Interpretable machine learning for predicting customer churn in retail banking," in *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2023.
45. T. Clement, N. Kemmerzell, M. Abdelaal, and M. Amberg, "Xair: A systematic metareview of explainable ai (xai) aligned to the software development process," *Machine Learning and Knowledge Extraction*, vol. 5, no. 1, pp. 78–108, 2023.
46. Y. Han, "Research on precise service of academic journals based on user profile," *Acta Editologica*, vol. 2, pp. 142–146, 2021.
47. D. Travis, "How to create personas your design team will believe in." `https://www.userfocus.co.uk/articles/personas.html`. Accessed: 2024/4/2.
48. Y. Chang, Y. Lim, and E. Stolterman, "Personas: from theory to practices," in *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, pp. 439–442, 2008.

49. L. W., O. K., L. C.G., and C. H.J., "User profile extraction from twitter for personalized news recommendation," in *16th International conference on advanced communication technology*, pp. 779–783, IEEE, 2014.
50. M. Raghuram, K. Akshay, and K. Chandrasekaran, "Efficient user profiling in twitter social network using traditional classifiers," in *Advances in Intelligent Systems and Computing*, pp. 399–411, Springer International Publishing, 2015.
51. R. Bonnie, "The power of the persona." `https://www.pragmaticinstitute.com/resources/articles/product/the-power-of-the-persona/`. Accessed: 2024/4/2.
52. Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly," *High-Confidence Computing*, vol. 4, no. 2, p. 100211, 2024.
53. K. Bostrom and G. Durrett, "Byte pair encoding is suboptimal for language model pretraining," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.
54. P. Gage, "A new algorithm for data compression," *The C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.
55. J. Zhan and et al., "An effective feature representation of web log data by leveraging byte pair encoding and tf-idf," in *Proceedings of the ACM Turing Celebration Conference-China*, pp. 1–6, 2019.
56. "Summary of the tokenizers." `https://huggingface.co/docs/transformers/tokenizer_summary#summary-of-the-tokenizers`. Accessed: 2024/4/2.
57. Thomwolf, "Bpe tokenizers and spaces before words." `https://discuss.huggingface.co/t/bpe-tokenizers-and-spaces-before-words/475`. Accessed: 2024/4/10.
58. R. A. and S. Borah, "Study of various methods for tokenization," in *Applications of Internet of Things*, pp. 193–200, Springer Singapore, 2020.
59. X. Gutierrez-Vasques, C. Bentz, and T. Samardžić, "Languages through the looking glass of bpe compression," *Computational Linguistics*, vol. 49, no. 4, pp. 943–1001, 2023.
60. N. Tavabi and K. Lerman, "Pattern discovery in physiological data with byte pair encoding," in *Multimodal AI in Healthcare*, pp. 227–243, Springer International Publishing, 2022.
61. N. Fradet, N. Gutowski, F. Chhel, and J. Briot, "Byte pair encoding for symbolic music," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.
62. H. Liu, "Byte-pair and n-gram convolutional methods of analysing automatically disseminated content on social platforms," *MDPI AG*, 2020.
63. N. Nilsson, *Principles of Artificial Intelligence*. Springer Berlin Heidelberg, 1982.
64. F. Harary, "The explosive growth of graph theory," *Annals of the New York Academy of Sciences*, vol. 328, no. 1, pp. 5–11, 1979.
65. R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.
66. C. Photphanloet and R. Lipikorn, "Pm10 concentration forecast using modified depth-first search and supervised learning neural network," *Science of The Total Environment*, vol. 727, p. 138507, 2020.
67. S. Rahmani, S. Fakhrahmad, and M. Sadreddini, "Co-occurrence graph-based context adaptation: a new unsupervised approach to word sense disambiguation," *Digital Scholarship in the Humanities*, vol. 36, no. 2, pp. 449–471, 2020.
68. Y. Du, F. Li, T. Zheng, and J. Li, "Fast cascading outage screening based on deep convolutional neural network and depth-first search," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2704–2715, 2020.
69. Q. Mei and M. Gül, "Multi-level feature fusion in densely connected deep-learning architecture and depth-first search for crack segmentation on images collected with smartphones," *Structural Health Monitoring*, vol. 19, no. 6, pp. 1726–1744, 2020.

70. A. Syah, F. Helmiah, N. Irawati, and N. Hasibuan, "Depth first search algorithm in the expert system for diagnosis of palm oil growth obstacles," in *4TH INTERNATIONAL CONFERENCE ON CURRENT TRENDS IN MATERIALS SCIENCE AND ENGINEERING 2022*, 2024.
71. G. Logeswari, S. Bose, and T. Anitha, "An intrusion detection system for sdn using machine learning," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, pp. 867–880, 2023.
72. W. Cai, R. Wei, L. Xu, and X. Ding, "A method for modelling greenhouse temperature using gradient boost decision tree," *Information Processing in Agriculture*, vol. 9, no. 3, pp. 343–354, 2022.
73. G. Ke and et al., "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, vol. 30, 2017.
74. B. Wardani, S. Sa'adah, and D. Nurjanah, "Measuring and mitigating bias in bank customers data with xgboost, lightgbm, and random forest algorithm," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 1, pp. 142–155, 2023.
75. Y. Hua, "An efficient traffic classification scheme using embedded feature selection and lightgbm," in *2020 Information Communication Technologies Conference (ICTC)*, 2020.
76. N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
77. J. Ponsam, S. Gracia, G. Geetha, S. Karpaselvi, and K. Nimala, "Credit risk analysis using lightgbm and a comparative study of popular algorithms," in *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, 2021.
78. Y. Wong, K. Madhavan, and N. Elmqvist, "Towards characterizing domain experts as a user group," in *2018 IEEE Evaluation and Beyond-Methodological Approaches for Visualization (BELIV)*, pp. 1–10, 2018.
79. P. Fadde and P. Sullivan, "Developing expertise and expert performance," in *Handbook of Research in Educational Communications and Technology: Learning Design*, pp. 53–72, 2020.
80. K. Chandrasekaran, *Domain-Driven Design with Java - A Practitioner's Guide: Create simple, elegant, and valuable software solutions for complex business problems*. Packt Publishing, 2021. https://ddd-practitioners.com/home/glossary/domain-expert/.
81. Vujović, "Classification model evaluation metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021.
82. T. Saito and M. Rehmsmeier, "Basic evaluation measures from the confusion matrix." https://classeval.wordpress.com/introduction/basic-evaluation-measures/, 2017.
83. P. Le, M. Nauta, V. Nguyen, S. Pathak, J. Schlötterer, and C. Seifert, "Benchmarking explainable ai - a survey on available toolkits and open challenges," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023.
84. A. Holzinger, "From machine learning to explainable ai," in *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, 2018.
85. Z. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.
86. C. Molnar, *Interpretable machine learning*. Lulu.com, 2020.
87. J. Karkavelraja, "Amazon sales dataset." https://www.kaggle.com/datasets/karkavelrajaj/amazon-sales-dataset. Accessed: 2024/4/2.
88. A. Gupta, A. Raghav, and S. Srivastava, "Comparative study of machine learning algorithms for portuguese bank data," in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021.
89. Lavanya, "Google play store apps." https://www.kaggle.com/datasets/lava18/google-play-store-apps. Accessed: 2024/4/10.
90. P. Lokesh, "Amazon products sales dataset 2023." https://www.kaggle.com/datasets/lokeshparab/amazon-products-dataset. Accessed: 2024/4/2.
91. D. Chen, "Online retail." UCI Machine Learning Repository, 2015.
92. "Personal data protection act." https://law.moj.gov.tw/LawClass/LawAll.aspx?PCODE=G0380233. Accessed: 2024/4/2.

93. "Banking act." https://law.fsc.gov.tw/LawContent.aspx?id=GL000624. Accessed: 2024/4/2.
94. A. Caramazza and J. Shelton, "Domain-specific knowledge systems in the brain: The animate-inanimate distinction," *Journal of Cognitive Neuroscience*, vol. 10, no. 1, pp. 1–34, 1998.

Zhenyao Liu is currently an Assistant Professor of the School of Economics and Management, Taizhou University in Jiangsu Province, China. He received Ph.D. degree from the Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan. His research areas are soft computing and machine learning.

Yu-Lun Liu received M.S. degree from the Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan.

Wei-Chang Yeh received the M.S. and Ph.D. degrees from the Department of Industrial Engineering, University of Texas at Arlington. He is currently a Chair Professor of the Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan. Most of his research is focused on algorithms, including exact solution methods and soft computing. He has published more than 300 research articles in highly ranked journals and conference papers.

Chia-Ling Huang is currently a Professor of the Department of International Logistics and Transportation Management, Kainan University, Taiwan.

Received: November 30, 2024; Accepted: June 25, 2025.