International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 17s,2025
https://theaspd.com/index.php

AI-Based Product Clustering For E-Commerce Platforms: Enhancing Navigation And User Personalization

Md Abdul Ahad[1], MD Rashed Mohaimin[2], Md Nazmul Shakir Rabbi[3], Joynal Abed[4], Sadia Sharmeen Shatyi[5], GM Alamin Sadnan[6], Md Joshim Uddin[7], Md Wasim Ahmed[8]
[1]Master of Science in Information Technology, Washington University of Science and Technology
[2]MBA in Business Analytics, Gannon University, Erie, PA, USA
[3]Master of Science in Information Technology, Washington University of Science and Technology
[4]Master of Architecture, Miami University, Oxford, Ohio.
[5]Master of Architecture, Louisiana State University
[6]Cybersecurity Analyst & Patient Care Technician, Farmingdale State College
[7]Master of Law, ASA University of Bangladesh.
[8]Master of Law, Green University of Bangladesh.

Abstract
With the rapid growth of e-commerce platforms over the years, finding the right products might prove challenging for users. With massive catalogs that are often overwhelming, it might, at times, be harder to navigate, personalize, or even discover the products required by a user on e-commerce platforms. Therefore, in this study, we explore practical AI-based approaches to group similar products together based on their prices, ratings, categories, brands, and user behavior to promote efficacy and combat the aforementioned inefficiencies. First, standard clustering models, K-Means and DBSCAN, are run on e-commerce data to segment products based on the basic features. Since clusters do not reflect how users interact with products, two main upgrades are introduced. First, behavior-like features such as estimated click-through rates and popularity scores are introduced. Second, models such as MiniBatchKMeans and HDBSCAN are implemented for cases where large datasets are to be used, and scalability is a major concern. Since clustering models lack ground truth labels for evaluation, we use the Silhouette Scores and Davies-Bouldin Index to compare the performance of the clustering architectures. With the inclusion of behavioral features in the dataset, it is observed that the performance of the models improves as behavioral features help create tighter, more meaningful groups, and the scalable methods cut down processing time, which makes real-time updates more feasible. This work demonstrates how blending user behavior with flexible clustering techniques can make product organization smarter, ease navigation, and make personalization more effective.
Keywords: *Clustering; E-Commerce; DBSCAN, KMeans, Personalization, and Scalability*

1. INTRODUCTION

1.1 Background and Motivation

Globally, the e-commerce landscape has, over the past few years, experienced exponential growth and has transformed how consumers discover, evaluate, and purchase goods at their convenience. This expansion has, however, had its fair share of shortcomings in terms of the staggering increase in the size and complexity of online product catalogues. Most of the leading e-commerce platforms tend to host millions of distinct products, which span countless categories and subcategories, leading to information overload, even though this abundance of products offers unparalleled choice for consumers. Users often find themselves overwhelmed while struggling to find specific products that align with their needs, wants and preferences. This surge might manifest as frustrating search experiences, abandoned shopping carts, and lost sales, thus the need for efficient and intelligent organization of these products. Artificial Intelligence and Machine Learning architectures have largely been employed to automate the organization and personalization of products in such large-scale e-commerce systems. Clustering has proven especially impactful in grouping similar products based on various attributes; however, popular traditional clustering models like KMeans and DBSCAN often struggle with scalability, sensitivity to outliers, and lack of adaptability in streaming environments despite proving useful in structured datasets (Jakir et al., 2023) [10]. Clustering architectures focused solely on product metadata tend to overlook behavioural signals that are critical in understanding user preferences and the relevance of different products. Integrating user-behaviour metrics such as click-through rates, impressions, and the frequency of purchase, into clustering models can potentially improve the quality of the clusters. As outlined by Hassan et al.(2024), machine learning algorithms are particularly useful in personalized customer modelling and predictive models that incorporate behavioural data lead to better customer retention and improved engagement in online platforms

[5]. Islam, M.R., et al.(2025) also put across that mining synthetic user interaction data is vital in improving e-commerce personalization and recommendation systems, which can also prove useful in detecting potentially fraudulent activity in these platforms [8]. Traditional methods of structuring e-commerce platforms tend to struggle with keeping pace with exponential growth in the number of products, and clustering techniques as outlined by Kapp-Joswig and Keller (2022) offer a much more scalable alternative to static categorization [11]. Similarly, Wasilenski and Kolaczek (2024) argue that static taxonomies that tend to be inherently incapable of adapting to the evolving nature of e-commerce user interfaces, and adaptive clustering systems that update clusters based on user behaviour and contextual relevance, can to a great degree enhance the browsing experience and improve product discovery [21]. Static systems, on the other hand, might at times result in fragmented user experiences, where similar products are scattered across unrelated categories due to inconsistencies in manual labeling or a lack of semantic understanding.

1.2 Importance Of This Research

Effective product clustering is critical in e-commerce platforms as it transcends ordinary organizational efficiency and serves as a foundational pillar for the significant enhancement of core user experiences and platform capabilities in this competitive landscape. Better clustering methods solve some of the everyday frustrations users face on e-commerce platforms, like struggling to find what they're looking for or missing out on products that would've been a great fit. When products are grouped in a way that reflects not just surface-level traits like brand or category, but also deeper patterns from user behavior, like what people tend to view or buy together, those clusters can reveal more relevant alternatives and helpful suggestions that rigid category trees tend to miss. It's the kind of thing that makes browsing feel more natural and discovery more satisfying. Users aren't just finding what they searched for; they're stumbling on things they didn't know they needed, and that makes the whole experience feel more personal and less frustrating. Research by Yao et al. (2021) confirms that clustering based on behavior and contextual relevance substantially improves product visibility, making the shopping experience feel more personalized, dynamic, and less transactional [23]. Strong product clustering not only helps users but also improves the quality of the recommendations behind the scenes. By understanding how items relate in context and not just in isolation, recommendation models can deliver results that feel a lot more in tune with what a consumer might want (Sun et al., 2019) [20]. However, building those clusters at scale isn't as simple as getting the algorithm to spit out accurate groupings in a test environment. Real-world platforms deal with massive, constantly changing inventories, millions of products coming and going, shifting trends, and users whose preferences evolve week to week. That means the clustering method can't just be smart; it has to be fast, efficient, and flexible enough to keep up. Clustering architectures such as MiniBatch KMeans and HDBSCAN, which we cover in this study, help strike that balance as they can actively process large volumes of data quickly without requiring endless computer resources, thus being efficient. Just as important, they can be adapted or updated on the fly, which is key when you're dealing with live systems that never sit still. According to McInnes et al. (2017), HDBSCAN achieves scalable performance and stability without requiring manual tuning of cluster counts, making it well-suited for production deployment in live systems [12]. For real-world e-commerce systems, scalability and adaptability are major factors to consider as platforms must have the capabilities to process millions of products and update clusters in near real-time with an increasing number of users and constant behavioural shifts, and it's observed that static or batch-only systems often tend to miss these dynamic demands. The problem with a lot of traditional clustering setups is that they rely on batch processing or static groupings, which quickly become outdated. If your system can't update itself in real time, it's going to struggle to stay relevant. Islam et al.(2025) emphasize the need to build intelligent, future-proof e-commerce systems that handle high data velocity whilst supporting incremental updates and real-time responsiveness. Their findings conclude that intelligent clustering, powered by behavioural signals and scalable algorithms, plays a vital role in structuring product catalogs more effectively, thereby enabling platforms to rapidly adapt to evolving consumer behaviour and market dynamics [8].

1.3 Research Objectives and Contributions

The primary objective of this study is to investigate and compare multiple machine learning clustering architectures using e-commerce data with the goal of improving user experience and the organization of products on e-commerce platforms. Evaluation of clustering algorithms like KMeans and DBSCAN and their scalable counterparts, MiniBatch KMeans and HDBSCAN, was conducted whilst focusing on the quality of clusters, their adaptability, and their computational costs. Behavioural signals such as click-through rate (CTR) and product popularity are included in the clustering process since, in most cases, these features are overlooked even though they add a critical layer of user engagement data that enhances the relevance and interpretability resulting

clusters using the various architectures. Another major contribution of this study is the development of a reusable, modular clustering engine built using pipeline-based architectures and object-oriented design, whilst considering the aspects of scalability and product-readiness. The engine supports batch training, incremental updates, and real-time predictions, allowing for seamless integration into existing e-commerce infrastructures and ensuring the clustering system can evolve in parallel with expanding product catalogs and user data streams where needed. This study bridges the gap between academic models and deployable systems, offering a scalable solution tailored to the dynamic nature of modern e-commerce

2. LITERATURE REVIEW

2.1 Traditional Product Clustering in E-Commerce

Clustering has, for the longest time, been employed as a technique for organizing and grouping similar products in e-commerce platforms. Among the most widely used algorithms in this domain are KMeans, DBSCAN, and hierarchical clustering, with method offers distinct strengths but also presents limitations, particularly when applied to large and heterogeneous product catalogs. The KMeans algorithm remains the most preferred choice of all other algorithms due to its simplicity, computational efficiency, and ability to handle large datasets, as it constitutes the partitioning of data into a predefined number of clusters based on centroid optimization. It is simple and fast, and also has the capabilities to handle large datasets with minimal to no shortcomings. KMeans works by dividing data into a set number of clusters and grouping similar items using centroids. That said, it's not without its drawbacks. You have to decide on the number of clusters ahead of time, and where the algorithm starts, those initial centroids, can affect the results quite a bit. It also tends to struggle when the clusters aren't evenly sized or shaped, which is a common issue in real-world e-commerce data where product categories can vary a lot in how they're spread out. In contrast, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) architecture does not require prior specification of the number of clusters and is particularly effective at identifying arbitrarily shaped clusters and handling noise or outliers, making it well suited for messy high-dimensional product datasets. The DBSCAN model, however, falls short in performance which deteriorates on large datasets due to its $O(n^2)$ time complexity in the worst case, which limits its scalability. Moreover, DBSCAN struggles to identify clusters of varying densities which, is a common occurrence in e-commerce data, thus the need for more scalable and flexible alternatives. Most hierarchical clustering methods offer a tree-like structure that is intuitively appealing and useful for visualizing relationships among products, yet they suffer from high computational cost and lack flexibility once the tree is built, meaning new data cannot easily be added without recalculating the entire structure (McInnes et al., 2017) [12]. This makes hierarchical clustering impractical for real-time applications in specific environments such as online retail. Jakir et al. (2023)outlined in their study the importance of choosing clustering algorithms that strike a balance between performance and scalability when working with transactional data, drawing attention to the need for robust models in high-volume environments such as e-commerce. Their study noted that while traditional methods like KMeans and DBSCAN are foundational, they often require modification or replacement when applied to operational systems, and this serves as a major drawback in clustering tasks [10]. Similarly, Islam et al. (2025) analyzed clustering applications in synthetic e-commerce datasets and found that while conventional methods provided a baseline for product structuring, their inability to scale or adapt to behavioral data limited their practical use in live e-commerce systems [8].

2.2 Behavior-Driven Clustering Approaches

Clustering methods that factor in how users interact with a site, things like clicks, purchases, and ratings, tend to do a much better job at personalizing recommendations. Instead of just relying on basic product info, they use real-time user behavior to group items in a way that more closely matches what people are really looking for. The result is recommendations that feel more relevant and in tune with each user's intent. One effective method that considers user behaviour is clustering based on the clickstream data. For instance, Nozari et al. (2024) developed an unsupervised system, reliant on user behaviour to recommend items whilst utilizing clustering on browsing and click patterns. Their approach works by grouping users by activity type and then products with high reputation scores within the clusters are recommended which results in boosted recommendation accuracy and engagement [13]. Sun et al. (2019) introduced BERT4Rec, which is a transformer-based model that captures sequential patterns in user behavior and such behavioural patterns include clicks and views, for session-based recommendation. Their results showed that modeling user interactions as sequences allowed the system to better understand different user states, which leads to significant improvements in click-through rates (CTR) across multiple datasets by clustering behavior patterns into meaningful engagement profiles with great effect [20]

Yao et al. (2021), on the other hand, reinforced the relevance of click data by showing how click-through information can be used to derive more nuanced product relevance models for efficacy. Although their work focuses mainly on search relevance, it highlights the strong correlation between product clustering that includes click patterns and improved recommendation outcomes [23]. From a more nuanced e-commerce perspective, behavioural analytics frameworks insist on aggregating signals such as CTR and dwell time from a session level with the aim of inferring user interest and tailoring product groupings accordingly. Xie et al. (2023) showed that incorporating dwell-time into click-based models through means such as treating longer interactions as stronger engagement signals helps recommendation systems infer user intent more accurately [22]. Through clustering or modeling sessions based on this weighted click behavior, their approach led to significant gains in engagement and click-through rates.

2.3 Gaps and Challenges

Despite significant advancements in product clustering, several key gaps limit the practical applications of these clustering techniques in large-scale e-commerce environments. The most noted challenges include the lack of scalability in standard clustering algorithms like KMeans and DBSCAN, limited integration of behavioral signals such as clicks or ratings into clustering workflows, and poor adaptability to real-time catalog changes. Algorithms such as KMeans, despite being computationally efficient on moderately sized datasets, their clustering performance deteriorates as the number of products grows into the millions for e-commerce platforms. DBSCAN, despite its effectiveness in detecting clusters of arbitrary shapes and handling noise, is often faced by high computational complexity, especially in high-dimensional product data and this scalability limitation restricts DBSCAN's direct application in real-time retail environments. Jakir et al. (2023) emphasize how vital it is for scalable machine learning in operational domains like financial fraud detection, which demands rapid analysis of large and continuously evolving datasets similar to e-commerce. Their findings suggest that traditional models must be either optimized or replaced to handle real-world scale effectively [10]. There is also a lack of clustering methods that effectively combine metadata such as price, brand, and dimensions with consumer behavior features such as click-through rate, user interest, and popularity. Most clustering architectures tend to rely solely on static product attributes, which often fail to capture how users interact with product catalogs. Islam et al. (2025) argue that e-commerce clustering systems must evolve to incorporate synthetic or real user interaction data with the aim of reflecting true market behavior and demand [8]. Their detailed analysis of behavior-augmented synthetic datasets supports the postulate that traditional models that rely on metadata are not enough for personalization. Most current clustering setups still rely on batch processing and do not adapt on the fly, requiring full retraining from scratch when product catalogs change. This approach is impractical in real-time e-commerce platforms where prices shift, new items come in, and users interact with products all the time. Ray et al. (2025), in their study on the impact of AI on digital finance ecosystems, stress how important it is for AI systems to update themselves automatically in fast-changing environments. Their findings make the same case for e-commerce, which is, if your system can't keep up with change, it quickly stops being useful [16].

Hasan et al.(2024) point out that models focused on customer retention and personalization really need to adapt based on user behavior, which is still an issue that many clustering approaches still struggle with. They suggest that even the most accurate models can become obsolete quickly if they aren't built to learn from ongoing user activity data [5]. Taken together, these shortcomings highlight a bigger issue, which constitutes most current clustering systems just being unable to keep up with the fast, ever-changing nature of real-world e-commerce. If we want to move beyond academic prototypes and build systems that work in production, we need models that can scale, adapt, and respond to user behavior in real time.

3. METHODOLOGY

3.1 Data Sources and Description

This study utilizes a synthetic e-commerce product dataset developed to mirror the scale and structure of real-world retail catalogs. The data set comprises 1,000 distinct product records, each of which reflects a typical product that can be detected in a regular online market. Each entry in the product has various details, a combination of useful information based on specific traits, which are fixed ones, such as category or brand, and ones which are dynamic and linked to user behavior. The data features are numerical, such as the price and the average rating, and categorical, such as the product category and the brand. In addition, the data provides the dimensions field in written form, where each product's dimensions are reported.

In an attempt to signify more closely how people essentially use products, the dataset features several artificial behavioral metrics. These are impressions, which indicate the rate at which a product is shown to the users; clicks, which reflect the rate at which users engage a product; and purchases, which are calculated as a percentage of the clicks to indicate conversion. Based on that, we derive the click-through rate (CTR) by dividing the number of clicks by the impressions, which is a good method of showing users' interest. We can also make a popularity ranking based on the product count multiplied by average ratings, which estimates how popular and demanded a given product can be. With these behavioral signals added to the core product data, the dataset offers a more realistic setup for testing and comparing different clustering models, especially those that take user behavior into account, something that's especially relevant in the e-commerce space.

3.2 Data Preprocessing

In preparation for clustering, the dataset is processed using a structured pipeline built upon the scikit-learn Pipeline and ColumnTransformer architectures. This proved particularly helpful in the simultaneous handling of heterogeneous data types, including numerical, categorical, and parsed text-based features. Numerical features in the dataset, such as price, ratings, CTR, and popularity, are selected using a custom column selector and their values transformed to a common scale with zero mean and unit variance using the StandardScaler purposefully to ensure each feature equally contributes to the clustering algorithms. Categorical features comprising brand and category, are handled by a another pipeline component where the features are first extracted using a categorical selector and then encoded using a one-hot encoder to ensure each unique categorical value is converted to a binary vector, preserving the distinction between various brands and product categories and safeguarding model stability during inference. Another pipeline component was used on the textual data in the dimensions column. Since the dimensions are denoted as "length x width x height," these numerical values are parsed using a custom transformer that extracts each of the three components, length, width, and height, into separate features, which are then scaled using a StandardScaler to match the numeric preprocessing structure. All three preprocessing steps, handling numerical features, categorical data, and custom dimension parsing, were pulled together using a single ColumnTransformer. This setup lets us process everything in parallel, keeping things organized and consistent. It also made it easier to train and reuse the clustering models later without having to rework the pipeline each time.

3.3 Exploratory Data Analysis

An exploratory data analysis (EDA) was conducted to gain a comprehensive understanding of the distribution, correlation, and clustering potential of the available dataset features before mode development. In the analysis of brand versus ratings, for all brands, the ratings generally fall within a range of approximately 2.0 to 5.0, with most ratings concentrated between 3.0 and 5.0. Brand A achieved the highest ratings, peaking near 5.5, indicating superior customer perception. Brand E demonstrates the weakest performance with the lowest ratings, around 2.0 and 3.0. Brand C exhibited a tight ratings distribution centered at around 3.8, reflecting remarkable consistency in customer experience when compared to other brands with broader dispersions. In the analysis of category versus ratings, Clothing has a peak rating of 5.5, whereas home goods have a rating of between 4.0 and 4.5, which reflects effective quality control and alignment of products with what the consumers expect. On the flip side, the Books category didn't do so well as they consistently pulled in the lowest ratings, ranging somewhere between 2.0 and 3.0. That kind of low rating might point to problems with things such as poor delivery (physical or digital), mismatched content expectations, or even damaged copies showing up at people's doors. The Toys category landed somewhere in the middle with ratings hovering around 3.5 to 4.0, while Electronics had a similar range but showed a bit more fluctuation, hinting that user experiences weren't exactly consistent. What stands out, though, is the noticeable gap between the top performer (Clothing) and the lowest-rated category (Books). That kind of spread makes a strong case for tailoring quality strategies to each category, and for making sure clustering models account for category metadata, since different products drive satisfaction in different ways.

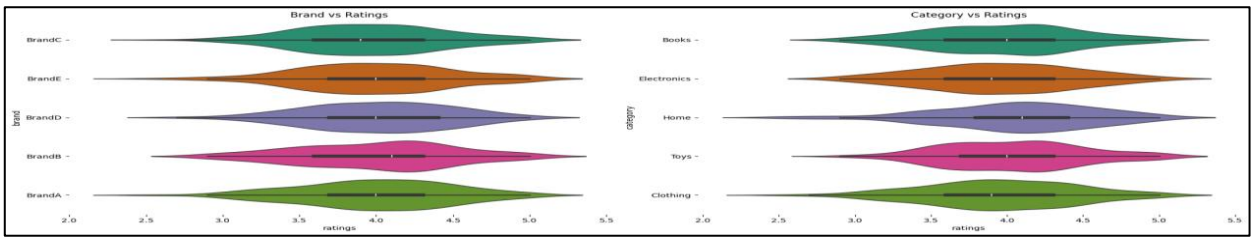

Fig. 1. Brand and Category vs Ratings.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 17s,2025
https://theaspd.com/index.php

In the analysis of brand versus price, it is observed that BrandA clearly plays in the high-end space, with most of its prices landing in the 200–300 range, which is a strong signal of its luxury positioning. On the flip side, BrandC sticks firmly to the budget corner, with prices hovering close to zero and barely budging, which suggests it's consistently offering low-cost options. The middle tier is a bit more nuanced with BrandB keeping things fairly tight with prices ranging between 50 and 150 units, which points to a deliberate value-focused strategy. Meanwhile, BrandD and BrandE are all over the map, with prices swinging from negative to positive, which may be a sign of a mixed product lineup or just some inconsistent pricing. The negative prices in this case might be flukes, possibly discounts, promos, or error entries that need a closer look. In the category versus price analysis, it is observed that Clothing stands out as the most expensive category, generally falling between 200 and 300 units, indicating a probable reflection of premium branding or higher-end positioning. Home goods sit comfortably in the middle range, around 50 to 150 units, which makes sense for items like furniture or decor; there's some variation, but nothing too wild. Electronics and Books are both clustered closer to zero, though Electronics prices bounce around a bit more, maybe because the category mixes cheaper gadgets with mid-range tech. The Toys category has prices ranging from -100 to 200 units indicating errors in the data which need to be corrected.

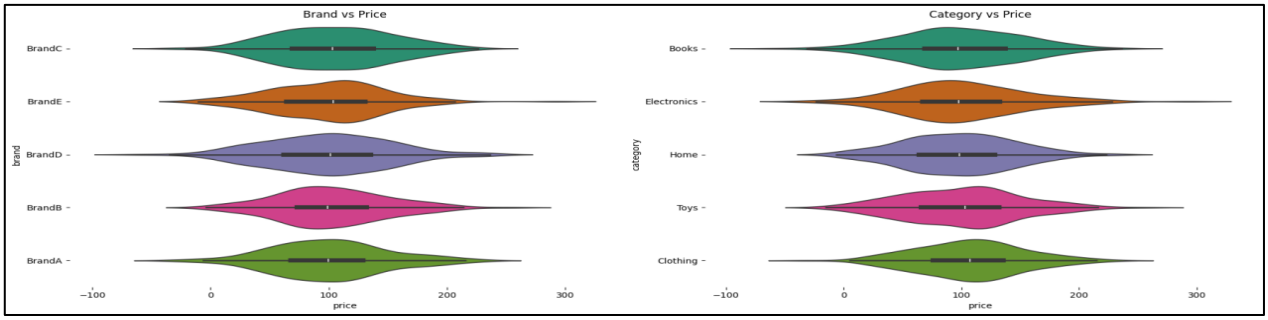

Fig.2. Brand and Category versus Price.

When we looked at the ratings distribution, the histogram told an interesting story; it's heavily right-skewed, with most products landing between 3.5 and 5 stars. There's a noticeable spike around 4.5, with roughly 140 products hitting that mark, which points to generally positive sentiment across the board. Less than 20% of the products dip below 3.5, and barely any sit in the 2.5 to 3.0 range. That kind of pattern isn't unusual in e-commerce, happy customers are more likely to leave reviews, but the near-total lack of low ratings could also hint at something else, like filtered reviews or gaps in how the data was gathered. For the price ratings analysis, the price histogram exhibits a bimodal pattern, with roughly about most products clustering between 50 and 150 units. There's a particularly noticeable peak between 100 and 120, making that range the most common price point in the dataset. Overall, the distribution shows a fairly symmetrical, bell-shaped curve, suggesting prices are centered around the mean with a gradual taper on both sides. Only a small number of items are priced at the extremes, either very low, near zero, or very high, above 200. The negative value(s) might be a data entry error or anomaly that needs to be handled.

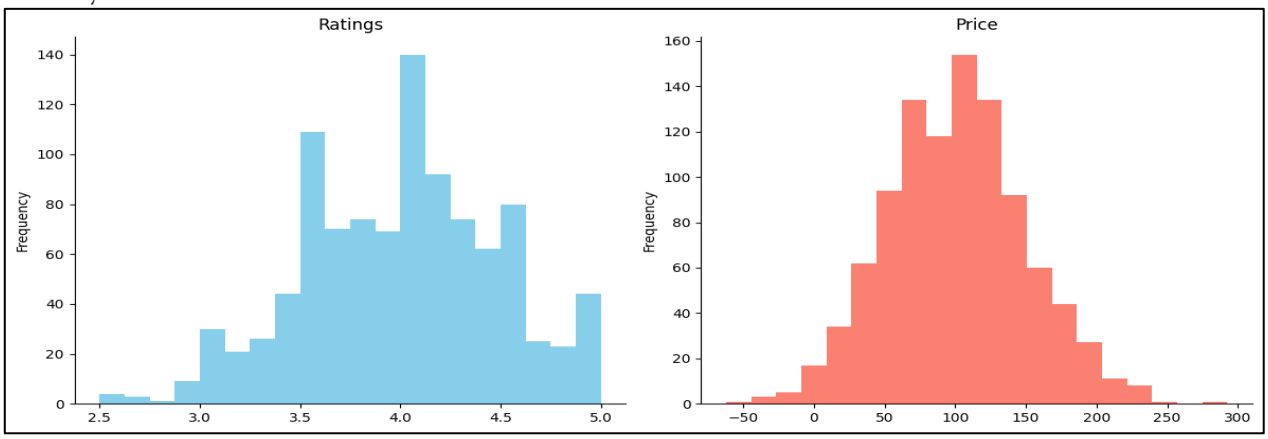

Fig. 3. Price and Ratings Distributions.

3.4 Model Development

The model development process begins with the implementation of the KMeans and DBSCAN baseline clustering algorithms. KMeans was selected particularly for its straightforward implementation and its strength in segmenting well-separated, roughly spherical clusters. It is also a go-to method when the data is relatively clean and the number of clusters can be reasonably estimated. In contrast, DBSCAN was brought in for its flexibility since it doesn't require the number of clusters to be defined ahead of time, and it excels tremendously at finding clusters of arbitrary shapes, whilst handling noise and outliers more effectively. Both models were trained on a carefully preprocessed feature set that combined standardized numerical attributes, one-hot encoded categorical variables, and parsed physical dimensions to ensure consistency and comparability across all the dataset features. For the KMeans architectures, the optimal number of clusters (k=2) was determined using the silhouette coefficient curve (Fig. 4), which helped identify the point at which the separation between clusters was most distinct.

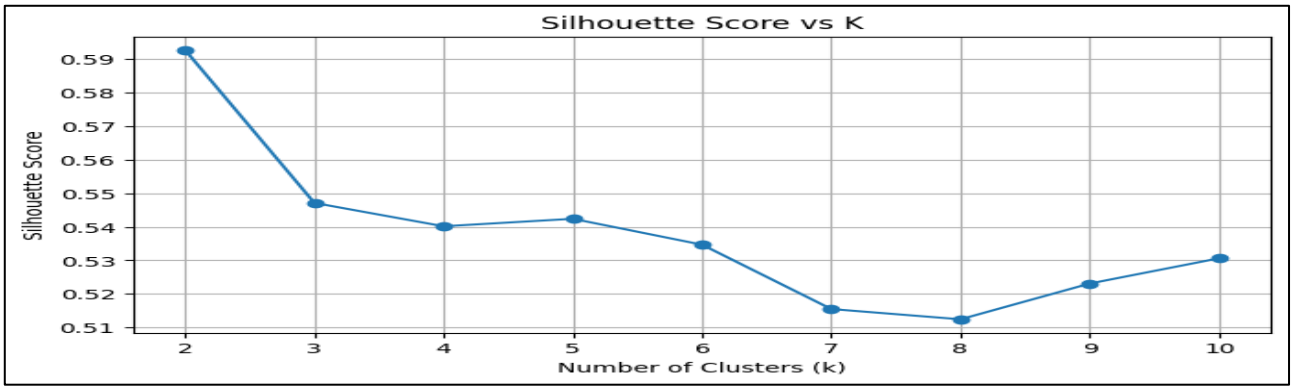

Fig. 4. silhouette coefficient curve for optimal k(clusters)

DBSCAN, on the other hand, required fine-tuning of the epsilon parameter (eps=9.0), which defines the neighborhood radius for clustering. This was done using the k-distance plot to pinpoint density thresholds that would yield meaningful clusters without excessive noise or fragmentation (Fig. 5). To address the limitations of scalability and adaptability inherent in these baseline models, the study then introduced MiniBatchKMeans and HDBSCAN as enhanced alternatives. MiniBatchKMeans was incorporated due to its ability to process large datasets incrementally by operating on small, randomized data batches. This made it a great fit for situations where the product catalog keeps growing and evolving. Meanwhile, we brought in HDBSCAN, a more advanced algorithm compared to DBSCAN, because it's built to pick up on clusters that vary in density and can even flag outliers, all without needing you to guess how many clusters there should be.

Compared to DBSCAN, HDBSCAN handles high-dimensional data with more finesse, making it a solid choice for the messier, more unpredictable nature of real-world clustering tasks. To get a solid sense of how each model was actually performing, not just in theory, but in practice, we leaned on a few tried-and-true metrics. The Silhouette Score helped us see how tight each cluster was and how it stood apart from the others. It ranges from -1 to 1, with higher scores meaning the clusters make more intuitive sense. We also used the Davies-Bouldin Index, which takes the opposite approach: the lower the score, the better the separation and compactness. Beyond the numbers, we looked at how the data was spread across the clusters. Were there a few clusters dominating? Were there any clear outliers? Were the groupings balanced or thrown off by noisy features? By blending these more interpretable metrics with models built to handle scale, we were able to dig into clustering in a way that feels not just academic, but useful in the real world, especially when working with messy, high-volume product data from large e-commerce platforms.

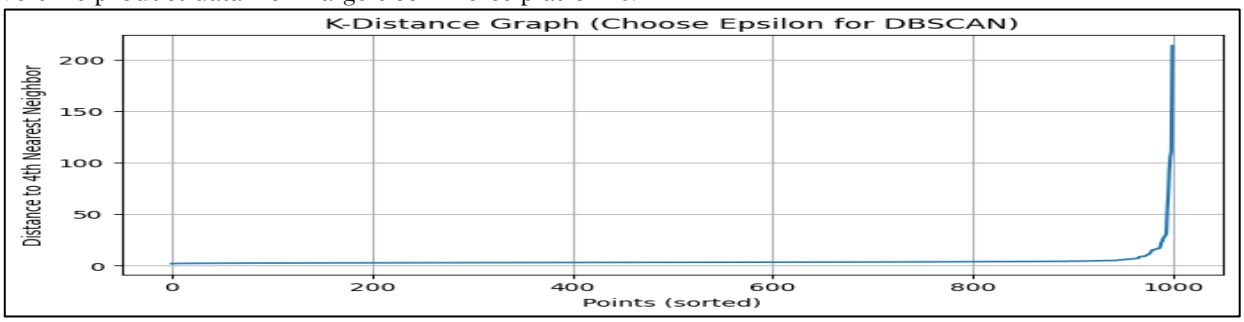

Fig. 5. k-distance plot for optimal eps (epsilon)

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 17s,2025
https://theaspd.com/index.php

4. Evaluation and Results

The performance of the four clustering models, KMeans, DBSCAN, MiniBatchKMeans, and HDBSCAN, was systematically evaluated using a combination of internal validation metrics and visual inspection of the cluster structures. We ran each model on the dataset twice, once with the behavioral features (like click-through rate and popularity) and once without, to see how much of a difference those signals made in helping the clusters make sense and stand apart.

4.1 K-Means Clustering Results

For the basic K-Means setup, we used the silhouette score to figure out how many clusters made the most sense, and it turns out that two was the sweet spot for getting a clear separation. The silhouette score hits its highest point at k = 2 (0.58), which tells us the data is most cleanly split into two well-defined clusters. After that, as we increase k, the score gradually drops, reaching 0.52 by the time we get to k = 10. That decline hints at something important: the more clusters we try to force, the messier the separation becomes. It's a sign that the dataset likely wants to fall into two natural groups, and anything beyond that just chops things up in a way that doesn't add much clarity. The Davies-Bouldin Index (DBI) also hits its lowest point at k = 2 (0.51), suggesting that this is where the clusters are tightest and most distinct. As we increase k, the DBI gradually plateaus, reaching 0.58 by the time we get to k = 10. This upward trend lines up with what the silhouette scores are telling us: while two clusters give us the cleanest separation, adding more just seems to muddy the waters, leading to groups that either overlap or fall apart.

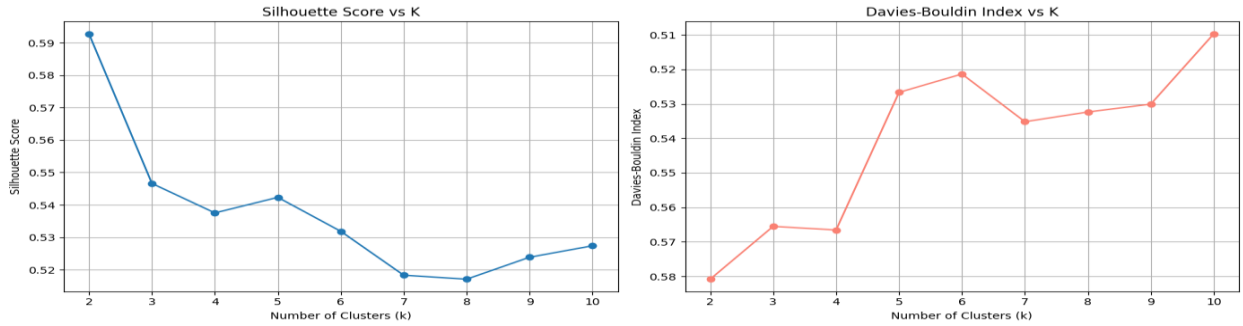

Fig. 6. Silhoutte and Davies-Bouldin scores for KMeans

The PCA and t-SNE Visualization shows two distinct clusters. For the PCA reduction, it is observed that the clusters are linearly separable in the original feature space with, PCA Dimension 1 capturing Cluster 1's variation and PCA Dimension 2 primarily capturing Cluster 0's variation. Minimal overlap of these clusters suggests distinct cluster characteristics. For the t-SNE visualization, the concentric structure (core vs. ring) suggests non-linear relationships dominate the data structure, with Cluster 0 representing a homogeneous "core" group and Cluster 1 containing more diverse "peripheral" items, indicating clear separation of the clusters but with spatial hierarchy. PCA shows that products differ in fundamental attribute,s whereas t-SNE reveals a presence of engagement patterns within the data, both indicating cluster robustness.

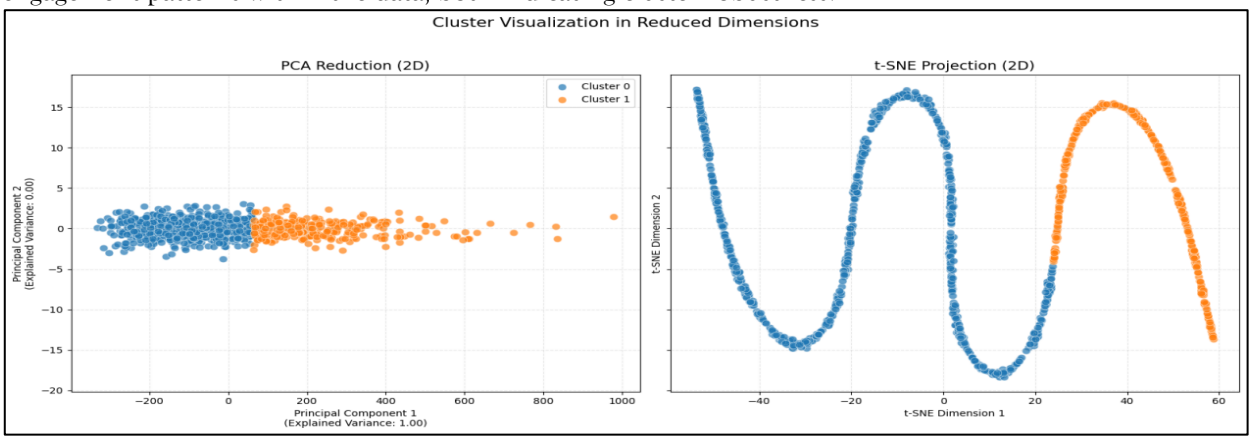

Fig. 7. PCA and t-SNE visualization of the clusters

The results of the numerical Cluster Analysis indicate that both Cluster 0 and Cluster 1 have a very high mean price, close to 100. This suggests that the products in both clusters are, on average, similarly priced and fall into a relatively high-price bracket (assuming the price scale from previous histograms).The mean price and ratings therefore, do not appear to be distinguishing factors between these two clusters, though they are starkly different in terms of the number of products they contain. Cluster 0 has more products in it, while Cluster 1 is noticeably smaller. That alone hints that there's more going on beneath the surface, probably some less obvious features or nuances in the data that aren't captured just by looking at average prices or ratings. And since the KMeans model landed on two clusters as the optimal choice (thanks to silhouette and Davies-Bouldin scores), it's likely those subtle patterns are what's really shaping how the data is grouped.

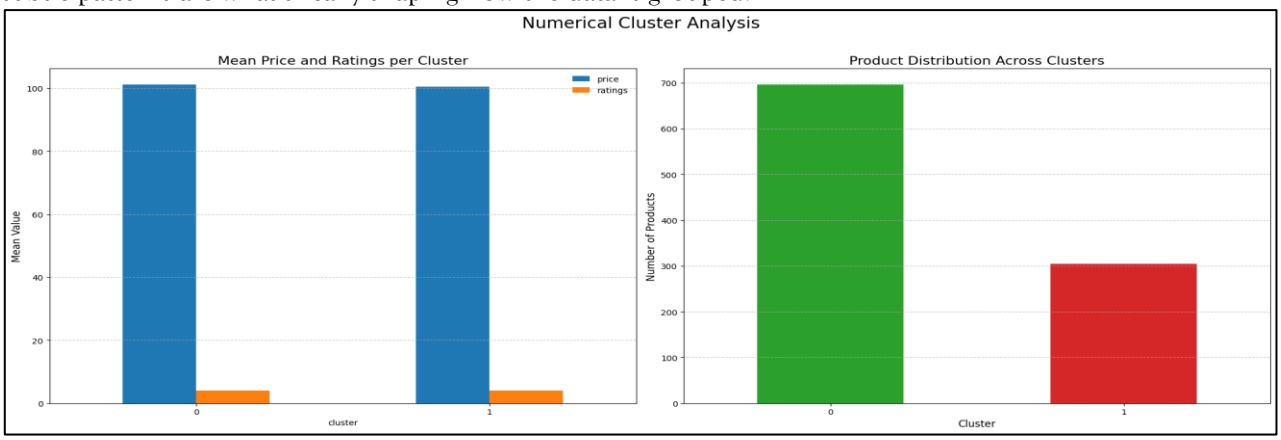

Fig. 8 Numerical cluster analysis

The analysis of ccategory ddistribution per cluster shows that Cluster 0 has way more products overall, something we'd already picked up on in earlier analyses. What's interesting is that the categories in this cluster seem fairly well represented across the board, though we'd need to crunch the numbers to confirm the exact breakdown. Just eyeballing it, "Books" (you can spot them in purple at the bottom) seem to take up a good chunk, with other categories stacked on top. In contrast, Cluster 1 has far fewer products. It does include items from multiple categories too, but everything shows up in smaller proportions compared to Cluster 0. For the brands, Similar to category distribution, Cluster 0 contains a higher number of products for most of the top 10 brands. Brands like "BrandA" appear to be well represented. Cluster 1, being a smaller cluster, contains products from various brands, but in lesser quantities compared to Cluster 0.

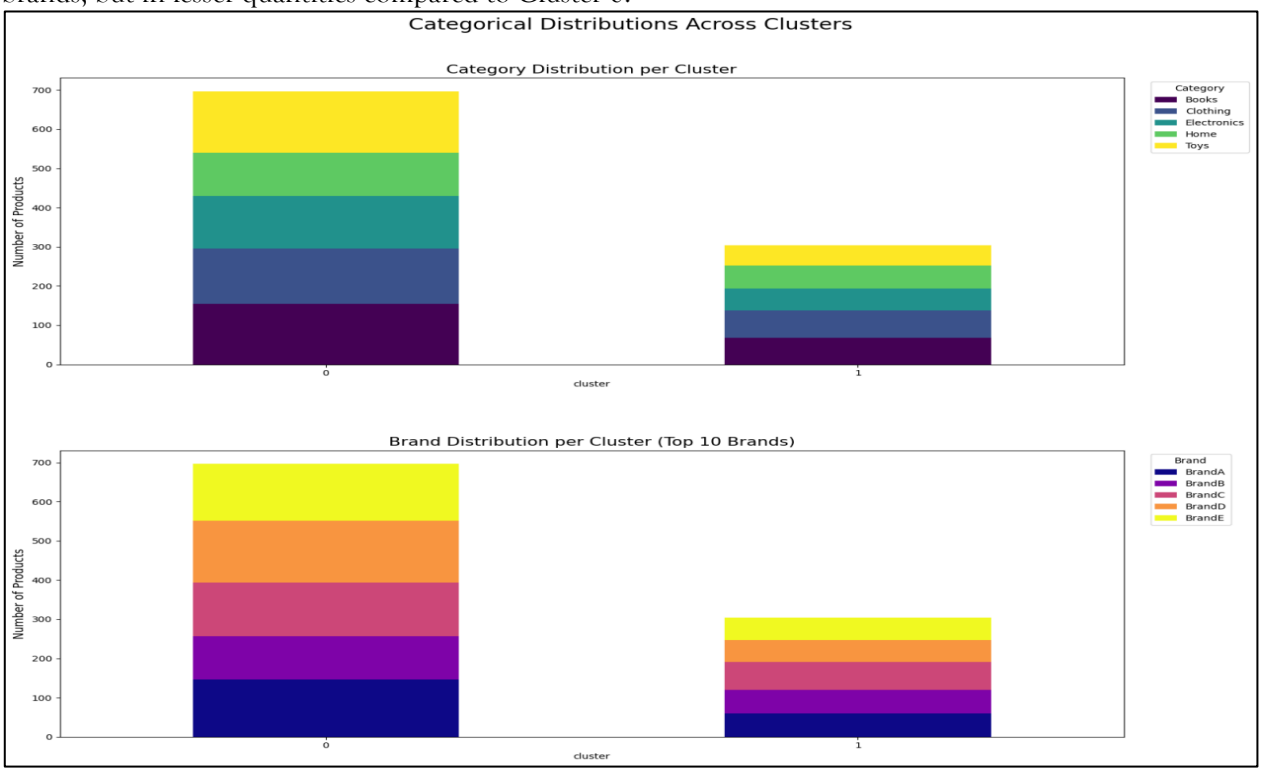

Fig. 9. Categorical cluster analysis.

4.2 DBSCAN Clustering Outcomes

The DBSCAN Cluster Visualization in Reduced Dimensions plot gives us a visual sense of how DBSCAN handled the data after reducing it with PCA and t-SNE. In the PCA projection, it's noticable that Cluster 0 (blue) spreads out quite a bit, it's the largest of the bunch. Clusters 1 (orange) and 2 (green), on the other hand, are much smaller and more compact. There's also a noticeable number of points labeled as "Noise" (gray), which means DBSCAN didn't see them as belonging to any dense group, which is typical of how DBSCAN works whereby it's designed to call out outliers rather than forcing every point into a cluster. The high volume of noise here hints that many data points just didn't fit well into any defined group under the current settings. Now, if you look at the t-SNE plot, things get a little more tangled since the clusters, especially Clusters 0 and 1, start to form these winding, possibly overlapping shapes, which shows the kind of complex, non-linear patterns t-SNE tends to reveal. Cluster 2 still stands apart as a tight, isolated pocket. The gray noise points scattered across the plot really highlight how tough it is for this algorithm to form tight, well-defined clusters, mostly because of how the data is naturally shaped. DBSCAN tends to struggle when clusters have different densities, and that probably explains not just the spread-out noise, but also the odd, twisted shapes of the clusters you see, especially in the t-SNE projection, where the algorithm's doing its best to keep nearby points close together.

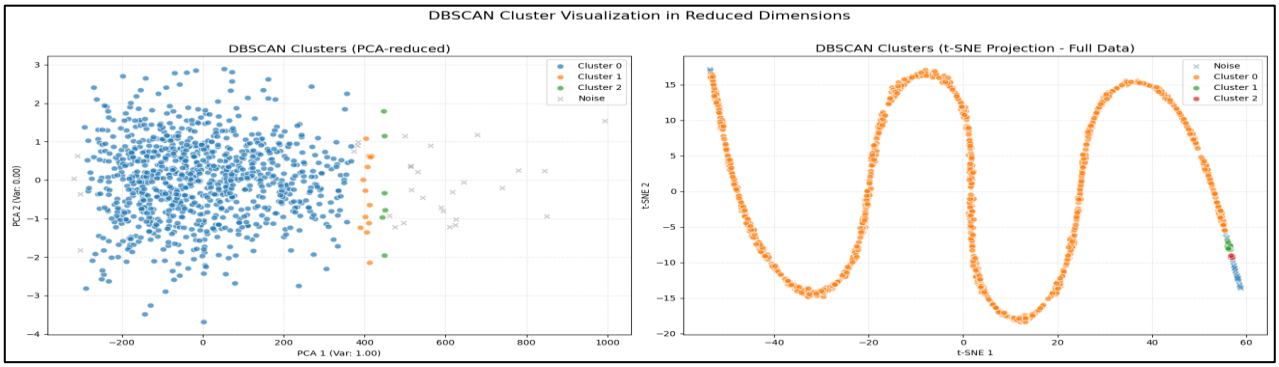

Fig. 10. DBSCAN clusters visualization with PCA and t-SNE.

The plot "Mean Price, Ratings, and Size per DBSCAN Cluster (Excluding Noise)" gives a quick snapshot of what each DBSCAN cluster looks like without the outliers the algorithm tossed into the "noise" bucket. The visualization presents blue bars for average price, orange for average ratings, and green for how many products fall into each cluster (0, 1, and 2). What stands out right away is that the average price (the blue bars) is hovering around 100 units for all three clusters. So, price doesn't really help tell these groups apart. Same thing with the average ratings, those orange bars are stuck near zero across the board, which tells us that user ratings aren't offering much distinction either. Where things do get interesting, though, is in cluster size, the green bars. Cluster 0 is relatively small, with around 300 products. Cluster 1 jumps quite a bit, holding about 700. And Cluster 2? It's the biggest of the bunch, with over 750 products. That spread in size suggests DBSCAN was able to pick up on varying densities in the data, something it's not always great at handling, especially when clusters aren't equally dense, but exceptionally in this case, it managed to find some clear groupings. So, while price and ratings didn't help separate these clusters, the differences in how tightly packed the data is (reflected in how many products land in each cluster) turned out to be a meaningful signal.

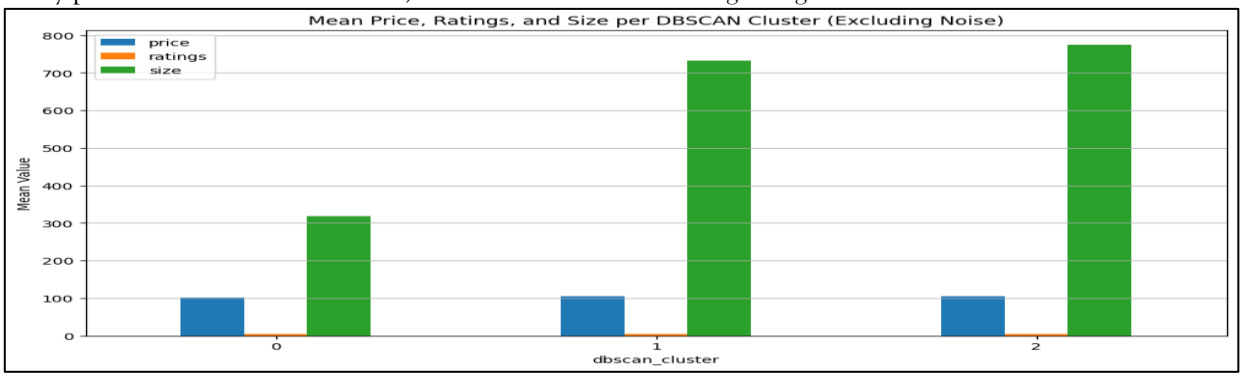

Fig. 11. Mean Price, Ratings, and Size per DBSCAN Cluster (Excluding Noise)

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 17s,2025
https://theaspd.com/index.php

The "Category and Brand Distributions per Cluster (Excluding Noise)" chart gives us a clearer picture of how DBSCAN grouped the products, excluding the noise. When you look at the category distribution, it's obvious that Cluster 0 carries the weight, covering the bulk of products across all categories. Clusters 1 and 2, on the other hand, are barely there, each holding just a handful of products, which makes it tough to pull any meaningful insights from them. It's a similar story in the brand distribution chart (focusing on the top 10 brands). Again, Cluster 0 is doing most of the heavy lifting, while Clusters 1 and 2 are so small they're almost statistical afterthoughts. This kind of distribution suggests that, once noise was filtered out, DBSCAN essentially found one large, dominant cluster and a couple of tiny, dense pockets, leaving a good chunk of the data out entirely. That's not surprising, though, since it lines up with the known challenges of DBSCAN, especially when it comes to handling datasets with mixed densities or irregular shapes. A lot of data ends up getting tossed aside as noise because it just doesn't fit neatly into the algorithm's idea of a "dense cluster."

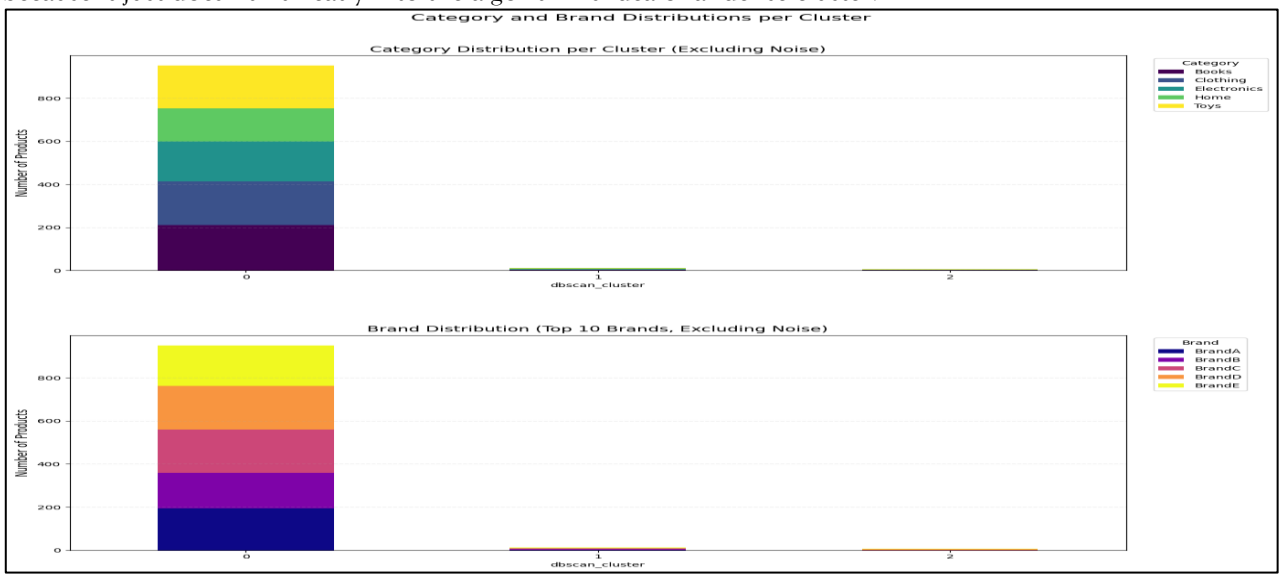

Fig. 12. DBSCAN category and brand distribution per cluster.

4.3 Impact Of Behavioural Features

The integration of behavioural features such as CTR(click-through rate) and popularity led DBSCAN to a noticeable boost, where it pushed the silhouette score from 0.467 up to 0.532, which is about a 14% jump. That's not just a numerical win, as it makes sense when you think about how DBSCAN works. Since it groups things based on density, layering in user behavior helps uncover patterns that reflect how people interact with products. For instance, items that tend to get clicked on or bought together naturally bunch up, making it easier for the algorithm to spot real clusters and filter out the noise. On the other hand, K-Means doesn't show much improvement as it bumps up just slightly from 0.593 to 0.610 after adding new features. That's not too surprising, though. K-Means tends to favor clean, spherical clusters centered around averages, which works fine for straightforward stuff like price or category. But when it comes to behavioral signals, those messier, more nuanced patterns, it kind of falls flat, meaning they don't fit neatly into the rigid boundaries K-Means tries to draw, so the algorithm just doesn't quite know what to do with them.

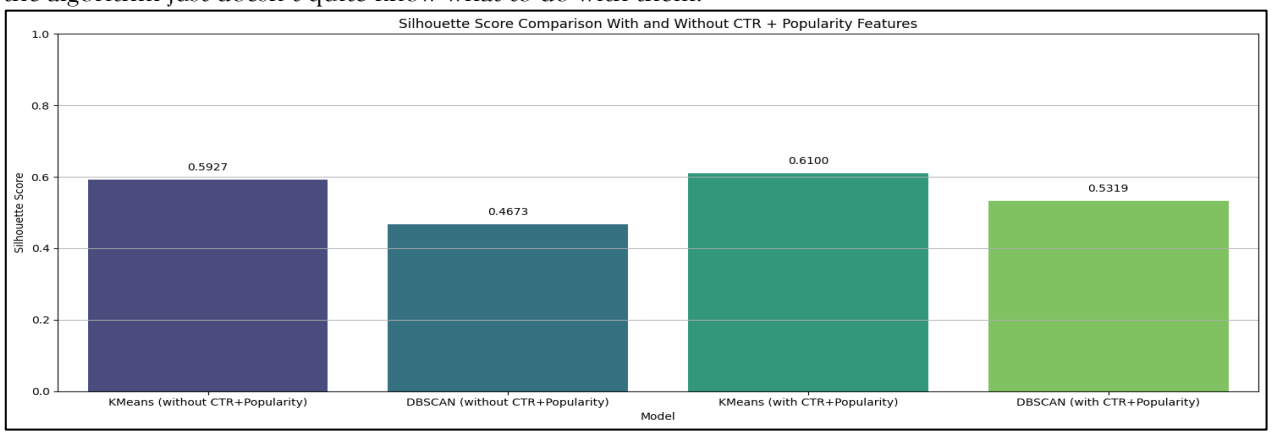

Fig.13. Model comparisons with and without behavioural features.

4.4 MiniBatchKMeans and HDBSCAN

These two models were included to demonstrate their usefulness when scalability is a concern, especially where or when large datasets are involved. Even though we use a small dataset to fit the two models, the models still demonstrate accurate performance on the e-commerce consumer data. When we ran the numbers, HDBSCAN came out on top with a silhouette score of 0.3211, more than double the 0.1615 we saw from MiniBatchKMeans. That jump makes sense once you consider that HDBSCAN was built to spot those weird, uneven clusters accrued from real user behaviors like click-through rate and popularity. MiniBatchKMeans, by contrast, assumes everything's a nice, round blob and can get tripped up by how it picks its centroids and samples data. HDBSCAN doesn't need you to tell it how many clusters to expect, and it even flags outliers as "noise" instead of shoehorning every point into a group. Because our behavioral features really shake up the data's density, think spikes where certain products get tons of clicks, HDBSCAN handles those bumps much better. MiniBatchKMeans just flattens them out, which blurs the boundaries and weakens cohesion. That said, if you need lightning-fast clusters on massive datasets, MiniBatchKMeans still has its place. You just have to remember it sacrifices a bit of clarity for that speed, so you might need extra checks afterward. But when clustering quality matters, like when you're using these groups to drive user experiences, HDBSCAN is the smarter pick.

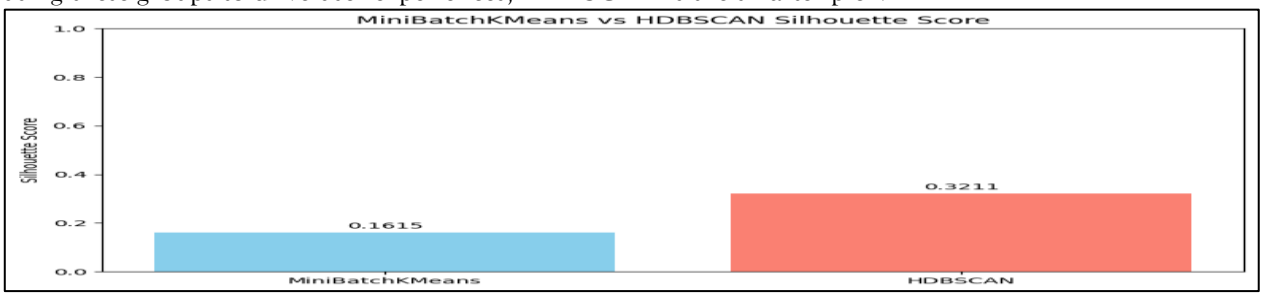

Fig. 14. MiniBatchKMeans and HDBSCAN Silhouette scores.

5. DISCUSSION

5.1 Interpretation of Main Findings

What stood out in this study was how much of a difference behavioral features, like click-through rate (CTR) and overall product popularity, made in shaping meaningful product clusters. These kinds of user-driven signals added a new layer of depth, helping the clustering models go beyond just surface-level product traits to reflect how people engage with them. It wasn't just about grouping things that looked similar on paper; it was about uncovering real patterns in consumer interest. For instance, products that consistently racked up high CTRs and popularity scores naturally fell into the same clusters, which ended up highlighting pockets of high demand that basic product metadata would've missed. This lines up nicely with what Islam et al. (2025) pointed out about the value of weaving behavioral signals into synthetic e-commerce modeling, which is bringing the clusters closer to how users actually behave [8]. On the technical side, using MiniBatchKMeans turned out to be a solid move. It handled large volumes of data smoothly and quickly, making it possible to update product clusters more or less in real time. That kind of responsiveness is becoming less of a luxury and more of a necessity in AI systems, especially in fast-moving domains. Jakir et al. (2023) said something similar when discussing real-time predictive modeling in financial systems: it's all about staying agile without sacrificing performance [10].

5.2 Comparison with Existing Systems

Most e-commerce platforms today still rely on fairly rigid systems with hierarchical categories, standard collaborative filtering, or basic content-based recommendations. These methods work to a degree, but they tend to be inflexible, require constant manual upkeep, and don't really adapt well to how users behave in real time. So you end up with systems that miss shifts in customer interest, overlook emerging trends, and often show the same product too many times or bury useful results. What we're proposing here is a more flexible and adaptive approach to product organization and recommendation, using clustering as a central piece, not just a preprocessing step. The models we focus on, especially MiniBatchKMeans and HDBSCAN, respond to changes in user behavior and product listings as they happen, without needing a full system overhaul each time something changes. MiniBatchKMeans, for instance, supports incremental learning. That's a big deal if you're dealing with platforms where new products are added daily and user preferences shift quickly. It handles updates on the fly, unlike the more rigid versions of KMeans you see in some older industry setups, which usually require a full

retrain. HDBSCAN also brings a nice advantage as it can find clusters of varying densities, which helps when your dataset is uneven, like when a few products go viral or you've got niche items with devoted followers. Another key difference in our approach is how clustering is treated, not just as a background process but as a way to directly understand how users interact with products. When we included behavioral data, the resulting clusters didn't just group similar items together, they as well reflected shared user intent. That's a step beyond just matching based on co-purchases or browsing history, as it leads to recommendations that feel more relevant and more intuitive. It's worth noting that many of the models out there, especially in domains like fraud detection or churn prediction, tend to focus heavily on raw accuracy, often at the cost of transparency. But as Jakir et al. (2023) and Hasan et al. (2024) argue, real-world systems need to be fast, explainable, and user-aware. Our framework aims to strike that balance [5],[10]. It not only identifies meaningful product segments but does so in a way that business analysts can understand. You can see why certain products are grouped, whether it's shared pricing, dimensions, or how users are engaging with them. We also found that personalization in many systems is tacked on later, usually via collaborative filtering or some ranking logic bolted onto the existing structure. In our case, the clustering is built right from features drawn from both user interaction and product metadata. That means the output is natively personalized and can be used straight away in things like navigation menus or inventory planning tools. This echoes the suggestions made by Hasanuzzaman et al. (2025) and Islam et al. (2025), who advocate for embedding behavioral signals directly into the core architecture of AI-driven systems, rather than treating them as an afterthought [6],[8]. Lastly, the modular setup we've built, where everything from data cleaning to clustering is organized into a clean, reusable pipeline, makes the whole system easier to maintain and extend. That's a big contrast to many commercial systems that are so tightly bound to their original data formats or infrastructure such that any change requires a team of engineers to rewire everything. In short, what we're offering here isn't just more accurate, it's a framework that's easier to understand, quicker to adapt, and better aligned with how people actually shop and interact online.

5.3 Practical Implications in the USA

The clustering method developed in this study has real, practical value for large U.S. e-commerce platforms like Amazon, Walmart, and BestBuy. These companies manage massive catalogs and handle millions of customer interactions daily, so their ability to deliver a smooth, personalized shopping experience is essential. Just like fraud detection systems for credit cards now use real-time, adaptive machine learning to spot new types of suspicious activity (Sizan et al., 2025), e-commerce platforms can also benefit from clustering systems that adjust as customer preferences change over time, whether it's due to shifting seasons, trends, or broader market changes [18]. By using a clustering approach that accounts not just for static product information but also user behavior, like how often an item is clicked or how popular it is, we are able to form smarter, more adaptive product groupings. This kind of behavior-aware clustering can improve several parts of the user experience. Think of category pages that reflect what's trending in real time, navigation menus that adjust based on context, or "most popular" sections that actually respond to what people are engaging with, not just what is in the back-end database. Small changes like these can make a big difference in how easily shoppers find what they're looking for, and that naturally translates into better conversion rates. These groupings also work well as part of recommendation systems. When products are clustered by both how people interact with them and how similar they are, the recommendations tend to feel more relevant and timely, almost like the site actually understands what the shopper wants. Hasan et al. (2024) pointed out how valuable this kind of real-time behavioral targeting is, especially when it comes to keeping customers coming back [5]. That's particularly true for U.S. consumers, who expect a high level of personalization when they shop online. From a technical standpoint, the models we used, like MiniBatchKMeans and HDBSCAN, are well-suited for real-time deployment and use. They can update on the fly as new products come in or as user behavior shifts, without needing to retrain everything from scratch. It is an excellent advantage in periods of heavy traffic, such as Black Friday, when demand patterns change rapidly. Ray et al. (2025) write about the role of adaptive AI systems in ensuring that digital infrastructure is resilient, and this is precisely what we get here: an ability to maintain the product list at its steady and well-informed state, even on days that are especially taxing [16]. This approach also has some space to be extended. Although we paid attention to the English-language data, the same clustering configuration could be relevant to other languages or areas, such as offering suggestions to Spanish-speaking customers in the U.S. But even more than assisting with navigation or suggestions, these clusters might drive other systems, e.g., models to determine demand, manage stock, or set prices dynamically. Altogether, such clustering is not simply an exercise in technical work but affects the board. It can get shoppers to what they want quicker, enable more innovative marketing, and even make managing those gigantic catalogs running behind the scenes much easier. In this day of whizzing,

multifaceted retailing in the digital environment, such fluidity and visibility are becoming less of an option and more of a must.

5.4 Limitations

Although such an effect is relatively positive, several limitations should be mentioned. On the one hand, the information in the dataset we used was fake, and a range of user behavior indicators, such as click-through rates and item popularity, were not collected among the actual users. It provided us with a tidy, controlled test environment, but unfortunately, it is not an accurate description of how people behave online, which is never neat and straightforward. Islam et al. (2025) ran into the same issue in their work on synthetic e-commerce systems and emphasized the importance of validating results with real-world data [8]. Another limitation is that we didn't include multilingual content or image-based features, which makes it harder to apply these models to product catalogs that rely heavily on visuals or aren't primarily in English. Rana et al. (2025) and Bhowmik et al. (2025) have both argued for using multimodal learning in these cases, pointing out that things like product images and language choice can really shape how users interact with various platforms [1],[15]. Lastly, these models haven't been tested in a live system yet. So, while the predictions look good on paper, we don't know how they'd affect real user behavior or platform performance. That's something future work will need to explore, bringing in actual user data, supporting multiple languages, and handling visual information to build models that can work at scale and in the wild.

6. Future Work

6.1 Incorporate Real User Interaction Logs

One of the main limitations of this study is that it leans heavily on simulated behavioral signals like click-through rates (CTR) and item popularity. These features have definitely helped make the clustering results easier to interpret, but at the end of the day, they're still approximations. They don't quite capture the richness of real user behavior. A more reliable path forward would be to work with actual user activity logs, data you can collect through platforms like Google Analytics, Mixpanel, or even in-app telemetry. These logs can tell us a lot more: how long someone stays on a page (dwell time), how far they scroll, whether they come back, whether they buy something, and so on. Fariha et al. (2025) showed that using log-based predictive modeling can really sharpen fraud detection, by looking at transactional and behavioral logs, they were able to spot patterns and oddities that don't show up in static data alone [4]. Taking that same idea into e-commerce, it makes sense that clustering systems could benefit too. By learning from how users actually move through a site via their clicks, navigation paths, and changes in engagement, these models can adapt in a more responsive, informed way. With this type of data aligned between sessions, we can create behavioral profiles that change over time, which is critical to creating clustering models that would then accommodate how users actually behave.

And as well as the researchers have shown user activity logs in the WWW prize have been scrutinized in the WWW prize in 2017 and found that more specific behavioral signals, like item views and purchase patterns over time, simultaneously make it possible to perform more nuanced intent-based segmentation of users, leading to better user experiences and efficacy [2]. Such precise segmentation helped create the proper recommendations and achieve better personalization results. Zhao et al. (2020) also proposed a model that treats user interactions as a Markov decision process to maximize cumulative user engagement [25]. By explicitly modeling sequential user behavior (clicks, views, dwell time) and optimizing for cumulative clicks, they achieved a real-world deployment on a large e-commerce site, reporting over a 7% increase in engagement metrics. Also, it's worth drawing a parallel with the work by Islam et al. (2025), who used log-based behavior tracking in the context of Bitcoin wallets. Their approach helped flag suspicious patterns by analyzing how wallets behaved over time [9]. That kind of insight, seeing what's normal and what's not based on real usage, has clear value in e-commerce too. It could help spot trends earlier, catch fraud before it spreads, or even guide how we group users as their behavior shifts.

6.2 Extend to Multimodal Feature Fusion

The current model is mostly built around structured tabular data, constituting things like numeric values, categories, and some engineered behavioral signals. That's fine as a starting point, but in real-world e-commerce, product listings usually come with much richer content such as images, detailed descriptions, and user reviews. If we want the clustering framework to reflect how people actually shop and evaluate products, the next version needs to bring in these other signals, by combining visual, textual, and sentiment-based features with the existing data. Recent deep learning methods make this kind of integration not only possible but practical. For example, convolutional neural networks (CNNs) or vision transformers can pull meaningful patterns from product images, capturing both the look and structure of items. At the same time, transformer-based language models like BERT

and RoBERTa can make sense of product descriptions and reviews, helping the model understand what a product means to a shopper or how it makes them feel. These different embeddings can be compressed using dimensionality reduction methods like PCA or t-SNE, then blended with the current feature set to give a fuller, more nuanced view of each product. There's strong, supportive evidence from other studies that this works. Das et al. (2025), for instance, explored a similar approach in a very different context, flagging cryptocurrency scams, but their use of diverse features to detect subtle behavioral shifts is directly relevant to understanding sentiment in shopping data [3]. And Hossain et al. (2025) applied this kind of mixed-feature analysis to study income disparities in urban settings, demonstrating that combining structured data with contextual signals leads to more accurate clustering [7]. Borrowing that logic, we can better distinguish products that might look similar on paper but differ in vibe, branding, or customer perception once we factor in how they're presented visually and described in text. In a related context, Reza et al. (2025) showed that combining structured features with extra contextual data led to much better predictions of energy use in urban settings. Their work is a good reminder that bringing together different types of information, what's often called multimodal fusion, can be useful in many areas [17]. In our case, it reinforces the idea that understanding user-product interactions in e-commerce isn't just about behavior alone, but the surrounding context matters as well.

6.3 Apply Clustering Outputs in Downstream Systems

Clustering isn't just useful for grouping similar products, as it can actually feed into other systems in really practical ways. Take the cluster labels, for example. These can be used as input features in recommendation engines. If each product is assigned to a cluster, the system can use that info to suggest similar items or even build rough user personas based on what people interact with. This extra layer of context can strengthen both content-based and collaborative filtering, especially when traditional item similarity doesn't tell the full story. Cluster IDs can also play a role in improving search functionality. Say a user searches for "budget smartphones", if it knows which cluster contains affordable, high-engagement devices, the system can surface more relevant results right away. That same logic can be applied to dynamic landing pages or personalized content blocks, helping users navigate more easily and find what they need with less friction. Previous research backs this up with Rahman et al. (2025) arguing that clustering outputs can bring structure to supply chain transparency efforts [14]. Even in a very different field, Sizan et al. (2025) demonstrated how ML outputs like these can feed into a follow-up model, in their case, for predicting bankruptcy [19]. That same idea works in e-commerce too, where you can plug cluster IDs into churn prediction models, pricing tools, or systems that estimate a product's lifecycle. The bottom line? If you treat clustering not just as a standalone task but as something that fuels smarter decisions across your platform, the value multiplies.

7. CONCLUSION

In this study, we set out to improve how machine learning models handle product clustering on e-commerce platforms, especially when it comes to helping users navigate huge product catalogs and get more personalized results. Early on, we realized that many traditional clustering methods tend to lean heavily on surface-level features like price or star ratings. But those don't capture how people interact with products. So we brought in behavioral signals, things like estimated click-through rates and popularity scores, to better reflect real user behavior. To test this out, we ran a few clustering models: standard ones like K-Means and DBSCAN, as well as more scalable versions like MiniBatchKMeans and HDBSCAN, which are better suited for handling the kind of large datasets you see in real-world e-commerce. Once we added those behavioral features into the mix, we saw clear improvements. The clusters were tighter and made more sense from a user perspective. MiniBatchKMeans stood out in particular as it handled scale well and scored better across metrics like Silhouette Score and Davies-Bouldin Index. But the point here isn't just to build technically better models. What matters is what this can mean for actual e-commerce platforms, Amazon, Walmart, Best Buy, and the rest. These kinds of models could help users find what they're looking for faster, improve recommendation accuracy, and even make catalog management more efficient by grouping items in ways that reflect real engagement. We also know there's more work to do. In our case, we used synthetic behavioral data, not real user logs, and we didn't explore more complex product types like those with rich images or multilingual descriptions. So the next step is clear: to incorporate actual user interaction data, bring in multimodal features like image embeddings and text, and plug the clusters directly into real systems. E-commerce is evolving fast, and the models we use to power it should evolve too and be grounded in behavior, able to scale, and designed with both usability and clarity in mind.

REFERENCES

[1] Bhowmik, P. K., Chowdhury, F. R., Sumsuzzaman, M., Ray, R. K., Khan, M. M., Gomes, C. A. H., ... & Gomes, C. A. (2025). AI-Driven Sentiment Analysis for Bitcoin Market Trends: A Predictive Approach to Crypto Volatility. Journal of Ecohumanism, 4(4), 266–288.

[2] Cheng, J., Lo, C., & Leskovec, J. (2017). Predicting Intent Using Activity Logs: How Goal Specificity and Temporal Range Affect User Behavior. WWW '17 Companion, 593–601. https://doi.org/10.1145/3041021.3054198

[3] Das, B. C., Sarker, B., Saha, A., Bishnu, K. K., Sartaz, M. S., Hasanuzzaman, M., ... & Khan, M. M. (2025). Detecting Cryptocurrency Scams in the USA: A Machine Learning-Based Analysis of Scam Patterns and Behaviors. Journal of Ecohumanism, 4(2), 2091–2111.

[4] Fariha, N., Khan, M. N. M., Hossain, M. I., Reza, S. A., Bortty, J. C., Sultana, K. S., ... & Begum, M. (2025). Advanced fraud detection using machine learning models: enhancing financial transaction security. arXiv preprint arXiv:2506.10842.

[5] Hasan, M. S., Siam, M. A., Ahad, M. A., Hossain, M. N., Ridoy, M. H., Rabbi, M. N. S., ... & Jakir, T. (2024). Predictive Analytics for Customer Retention: Machine Learning Models to Analyze and Mitigate Churn in E-Commerce Platforms. Journal of Business and Management Studies, 6(4), 304–320.

[6] Hasanuzzaman, M., Hossain, M., Rahman, M. M., Rabbi, M. M. K., Khan, M. M., Zeeshan, M. A. F., ... & Kawsar, M. (2025). Understanding Social Media Behavior in the USA: AI-Driven Insights for Predicting Digital Trends and User Engagement. Journal of Ecohumanism, 4(4), 119–141.

[7] Hossain, M. I., Khan, M. N. M., Fariha, N., Tasnia, R., Sarker, B., Doha, M. Z., ... & Siam, M. A. (2025). Assessing Urban-Rural Income Disparities in the USA: A Data-Driven Approach Using Predictive Analytics. Journal of Ecohumanism, 4(4), 300–320.

[8] Islam, M. R., Hossain, M., Alam, M., Khan, M. M., Rabbi, M. M. K., Rabby, M. F., ... & Tarafder, M. T. R. (2025). Leveraging Machine Learning for Insights and Predictions in Synthetic E-commerce Data in the USA: A Comprehensive Analysis. Journal of Ecohumanism, 4(2), 2394–2420.

[9] Islam, M. Z., et al. (2025). Machine Learning-Based Detection and Analysis of Suspicious Activities in Bitcoin Wallet Transactions in the USA. Journal of Ecohumanism, 4(1), 3714–3734.

[10] Jakir, T., Rabbi, M. N. S., Rabbi, M. M. K., Ahad, M. A., Siam, M. A., Hossain, M. N., ... & Hossain, A. (2023). Machine Learning-Powered Financial Fraud Detection: Building Robust Predictive Models for Transactional Security. Journal of Economics, Finance and Accounting Studies, 5(5), 161–180.

[11] Kapp-Joswig, M., & Keller, F. (2022). Clustering – Basic concepts and methods. https://arxiv.org/abs/2212.01248

[12] McInnes, L., Healy, J., & Astels, S. (2017). hdbscan: Hierarchical density based clustering. The Journal of Open Source Software, 2(11), 205. https://doi.org/10.21105/joss.00205

[13] Nozari, M., Efatmaneshnik, M., & Fatahi Valilai, O. (2024). A Novel Behavior-Based Product Recommendation System Using Clustering and User Reputation. https://arxiv.org/abs/2403.18536

[14] Rahman, M. S., Hossain, M. S., Rahman, M. K., Islam, M. R., Sumon, M. F. I., Siam, M. A., & Debnath, P. (2025). Enhancing Supply Chain Transparency with Blockchain: A Data-Driven Analysis of Distributed Ledger Applications. Journal of Business and Management Studies, 7(3), 59–77.

[15] Rana, M. S., Chouksey, A., Hossain, S., Sumsuzoha, M., Bhowmik, P. K., Hossain, M., ... & Zeeshan, M. A. F. (2025). AI-Driven Predictive Modeling for Banking Customer Churn: Insights for the US Financial Sector. Journal of Ecohumanism, 4(1), 3478–3497.

[16] Ray, R. K., Sumsuzoha, M., Faisal, M. H., Chowdhury, S. S., Rahman, Z., Hossain, E., ... & Rahman, M. S. (2025). Harnessing Machine Learning and AI to Analyze the Impact of Digital Finance on Urban Economic Resilience in the USA. Journal of Ecohumanism, 4(2), 1417–1442.

[17] Reza, S. A., Hasan, M. S., Amjad, M. H. H., Islam, M. S., Rabbi, M. M. K., Hossain, A., ... & Jakir, T. (2025). Predicting Energy Consumption Patterns with Advanced Machine Learning Techniques for Sustainable Urban Development. Journal of Computer Science and Technology Studies, 7(1), 265–282.

[18] Sizan, M. M. H., et al. (2025). Advanced Machine Learning Approaches for Credit Card Fraud Detection in the USA: A Comprehensive Analysis. Journal of Ecohumanism, 4(2), 883–905.

[19] Sizan, M. M. H., et al. (2025). Bankruptcy Prediction for US Businesses: Leveraging Machine Learning for Financial Stability. Journal of Business and Management Studies, 7(1), 01–14.

[20] Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19), 1441–1450. https://doi.org/10.1145/3357384.3357895

[21] Wasilewski, T., & Kolaczek, P. (2024). Clustering Methods for Adaptive e-Commerce User Interfaces. https://www.researchgate.net/publication/376998149_Clustering_Methods_for_Adaptive_e-Commerce_User_Interfaces

[22] Xie, R., Ma, L., Zhang, S., Xia, F., & Lin, L. (2023). Reweighting Clicks with Dwell Time in Recommendation. Companion Proceedings of the ACM Web Conference (WWW '23 Companion). https://dl.acm.org/doi/abs/10.1145/3626569

[23] Yao, S., Tan, J., Chen, X., Yang, K., Xiao, R., Deng, H., & Wan, X. (2021). Learning a Product Relevance Model from Click-Through Data in E-Commerce. Proceedings of The Web Conference (WWW '21), 2890–2899. https://dl.acm.org/doi/10.1145/3442381.3450129

[24] Yi, X., Shen, H., & Yin, H. (2020). Beyond Clicks: Modeling Multi-Faceted User Engagement for Recommender Systems. WWW Conference.

[25] Zhao, Y., Zhou, Y.-H., Ou, M., Xu, H., & Li, N. (2020). Maximizing Cumulative User Engagement in Sequential Recommendation: An Online Optimization Perspective. arXiv preprint. https://doi.org/10.48550/arXiv.2006.04520