Journal of Economics, Finance and Accounting Studies (JEFAS)
ISSN: 2709-0809
DOI: 10.32996/jefas
Journal Homepage: www.al-kindipublisher.com/index.php/jefas

JEFAS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Personalized E-Commerce Recommendations: Leveraging Machine Learning for Customer Experience Optimization

Joynal Abed[1]✉, Kazi Nehal Hasnain[2], Kazi Sharmin Sultana[3], Maksuda Begum[4], Sadia Sharmeen Shatyi[5], Mohotasim Billah[6] and GM Alamin Sadnan[7]

[1]*Master of Architecture, Miami University, Oxford, Ohio.*
[2]*Master of Science in Information Technology (MSIT), Westcliff University, Irvine, CA*
[3]*MBA in Business Analytics, Gannon University, Erie, PA*
[4]*Master of Business Administration, Trine University.*
[5]*Master of Architecture, Louisiana State University*
[6]*Master of Science in Computer Science, Washington University of Virginia(WUV)*
[7]*Cybersecurity Analyst & Patient Care Technician, Farmingdale State College*
Corresponding Author: Joynal Abed, Email: abedj2@miamioh.edu

| ABSTRACT

E-commerce ventures are increasingly turning to personalization as a key differentiator in the competitive digital market. Business in the marketplace is becoming more personal to gain closer engagements. Machine learning has transformed the possibility of personalizing shopping practices through the analysis of vast amounts of data that can discern user preferences and anticipate future actions. With such clever algorithms embedded in their websites, online retailers will be able to provide customers with a plethora of relevant, timely, and personalized product recommendations, leading to improved user satisfaction as well as business metrics, including click-through rates, conversion rates, and average order value. This study aimed to design, deploy, and evaluate machine learning algorithms that optimize product recommendations in a personalized e-commerce environment. The primary purpose is to develop scalable, efficient, and accurate recommendation systems that can be tailored to individual user preferences and adapt to real-time changes in behavior. The data from the given study were collected from a mid-sized e-commerce market in the United States over six months. It includes more than 150,000 interactions between users, over 25,000 individual users, and 10,000 products. The data is well-structured and contains several important dimensions that are vital for creating a personalized recommendation model. User demographics include age range with anonymity, gender, location (ZIP codes), and categories of customer loyalty. The history of browsing is captured through a session log that contains the browsed item, the amount of time spent on each page, the type of device, and the duration of the session. Exploratory Data Analysis (EDA) was essential for understanding the patterns, distributions, and relationships within the dataset, aiding in the assessment of features to select and in designing the model. In this research project, three machine learning algorithms were deployed, namely, Logistic Regression, Random Forest, and Support Vector Machines. To train and validate our models, we employed an 80:20 train-test split strategy, ensuring that 80% of user-product interactions were used for training. In comparison, 20% of the data were reserved for out-of-sample performance testing. The outcome clearly showed that the SVM model achieved the highest accuracy, making it the best-performing model among the other three. The introduction of machine learning-optimized recommendation systems to U.S. e-commerce systems will enable the personalization of services that were previously unachievable via rule-based, fixed solutions. The business strength of hyper-personalization has long been demonstrated by e-commerce giants such as Amazon and Target. In the works ahead, e-commerce recommendation systems are increasingly utilizing deep learning and contextual awareness to achieve a higher level of personalization.

| KEYWORDS

Personalization, Machine Learning, E-Commerce, Customer Experience, Recommendation Systems, User Engagement, Data Analytics, Predictive Modeling

| ARTICLE INFORMATION

ACCEPTED: 01 August 2024 **PUBLISHED:** 15 August 2024 **DOI:** 10.32996/jefas.2024.6.4.10

I. Introduction

Background

Personalization is a crucial aspect of current e-commerce, as the processes of communication between businesses and their customers are transforming (Bharwaj et al., 2024). The user's attention span is short, and competition is fierce in an era where people are spoiled for choice and products; a customized experience is a distinct advantage that increases customer satisfaction and loyalty. According to the U.S. Census Bureau, in 2023, e-commerce sales in the United States totaled approximately $1.1 trillion, indicating a growing trend toward online shopping (Alqurashi et al., 2023). As millions of consumers make their purchases online every day, it has become not a luxury but a necessity to provide relevant and personalized product recommendations. Machine learning is a crucial component of this transformation, as it enables the identification of patterns from large datasets and facilitates informed predictions (Behare et al., 2025). The McKinsey & Company report suggests that a tailored experience can result in a 10-30 percent increase in sales, underscoring the importance of custom, personalized recommendations for effective customer interaction. As online consumers experience an overload of options, they have increased their desire to have personal experiences. Artificial intelligence, particularly machine learning, offers a powerful set of tools that can analyze large volumes of consumer data to identify patterns that lead to personalized recommendations. It is by taking advantage of customer behavior, their likes and dislikes, and their buying patterns that e-commerce sites can generate dynamic recommendations that are acceptable to users and ultimately improve the shopping experience, while also increasing the chance of conversion.

Neural networks, support vector machines, and ensemble methods are machine learning methods that enable conclusions based on complex behavioral data. This knowledge will be vital in recommending products based on past experiences, browsing preferences, shopping trends, and social media activity (AG et al., 2025). Retail giants such as Amazon and Netflix have proven the efficiency of personalized recommendation engines. As an example, Amazon claims that its recommendation engine (based on both collaborative and content-based filtering approaches) contributes up to 35 percent of its sales (Agoro et al., 2021). Personalization via machine learning is therefore not an optional feature, but rather a core strategy that enables user engagement and commercial success. Besides generating revenue, these machine learning-based recommendation systems significantly enhance the overall customer experience by alleviating decision fatigue and presenting relevant options in real-time. Since the number of SKUs handled by e-commerce platforms is vast and the user base is diverse, automated and scored product recommendations ensure that users can find what they are looking for more quickly, resulting in a direct impact on satisfaction and retention.

According to Bitra (2025), customer experience and data-driven personalization have become a revolution in digital commerce due to their synergetic relationship. E-commerce users now expect platforms to recognize them and present content accordingly. According to the Salesforce report, titled "State of the Connected Customer" (2023), 73 percent of consumers want firms to understand their requirements and expectations. The increasing pressure underscores the need to create real-time, context-sensitive, and hyper-personalized experiences. Thus, this type of recommendation system has both a competitive advantage and adherence to changing consumer expectations and behavior in the United States when it comes to recommending advanced machine learning algorithms (Anonna et al., 2023).

Problem Statement

Even with the potential for personalized suggestions, numerous e-commerce sites face significant technical and operational challenges when implementing machine learning systems on a large scale (Bhardwaj et al., 2024). The first is the necessity to process huge amounts of data in real-time, which continuously varies depending on user interactions. Such dynamic conditions can cause traditional algorithms to fail, resulting in latency and obsolete recommendations (Alqurashi et al., 2023). Moreover, the inability of the smaller retailers to have strong infrastructure or expertise may lead to systems that are either too basic or not responsive enough to changing user preferences (Behare et al., 2025). This disconnect can be observed when customers receive suggestions that are not relevant to them, which may lower trust and the level of interaction. Boppiniti (2022) stated that scalability and the sparsity of data are other huge problems. With the increase in product catalog and growth in user base, it becomes computationally costly to maintain accurate and personalized recommendations—the cold start problem. An e-commerce platform frequently faces the problem of cold start, where new users or products have no prior data to make successful predictions. Agoro et al. (2021) noted that such deficiencies can lead to a disjointed user experience and diminish the effectiveness of marketing approaches. A 2022 Forrester report indicated that 57 percent of digital organizations failed to meet their personalization objectives due to their inability to act on customer data in real-time.

Objective

This study aims to design, deploy, and evaluate machine learning algorithms that optimize product recommendations in a personalized e-commerce environment. The primary purpose is to develop scalable, efficient, and accurate recommendation systems that can be tailored to individual user preferences and adapt to real-time changes in behavior. The paper will examine various machine learning algorithms, including matrix factorization, deep neural networks, and hybrid strategies that integrate multiple techniques to achieve better results. The study will attempt to validate these models on the most important performance metrics of e-commerce, including precision, recall, user satisfaction, and computational efficiency, through the simulation of real-world e-commerce settings and the usage of publicly available datasets, such as those available at Kaggle and the UCI Machine Learning Repository.

Furthermore, the project will address the ethical and regulatory aspects of personalization by incorporating privacy-preserving methods, including differential privacy and federated learning, thereby ensuring the protection of individual privacy. Such technologies can provide individualized recommendations without compromising user anonymity or violating data protection regulations. This two-pronged approach to technical performance and ethical integrity will ensure that the recommendations produced are not only effective but also credible and trustworthy. Its aim is not only theoretical but also to offer practical implementation solutions that can be applied by small-to-mid-sized U.S. e-commerce companies that do not require the same level of infrastructure as Amazon. Ultimately, the study will contribute to the existing body of knowledge in the field of data-driven personalization by offering practical recommendations and technical solutions that enhance customer engagement, loyalty, and conversion rates. Enhanced customer experiences through smart recommendations will help e-commerce platforms meet consumer demands, adapt to market trends, and remain competitive in a rapidly evolving digital landscape. The results will serve as a guide for incorporating advanced machine learning into business applications, particularly for American companies facing the challenges of both technological and regulatory complexity.

Significance

Choukey et al. (2023) highlighted that the strategic adoption of machine learning-based personalized recommendation systems is of immense importance to the e-commerce business in the U.S., which aims to compete in the international market. Given a digital economy in which customer demands are rapidly growing, and the loyalty that drives either brand performance or experience is becoming a dominant factor, the competitiveness of the market is closely tied to the ability to individualize shopping processes (Bitra, 2025). Deloitte's 2023 survey found that 80 percent of respondents would prefer to shop with a brand that offers personalized experiences, and that 90 percent of consumers find personalization attractive. Such a shift in consumer behavior necessitates the development of more advanced technologies that can learn in real-time and individualize their responses. Companies located in the United States that successfully adopt and utilize technologies can stand out from their competitors, both abroad and within the European Union, as the pace of e-commerce realization accelerates (Bhardwaj et al., 2024). Personalization not only becomes an instrument of engagement but also a key resource for positioning in a global market.

In financial terms, the effects of personalized recommendations are substantial, leading to increased conversion rates and higher overall revenue. According to the Boston Consulting Group (BCG) report, the increment in revenue generated by retailers that apply sophisticated personalization techniques ranges between 6% and 10%, which is three times the size of what retailers that have not implemented these techniques are gaining (Dulloo, 2024). In the United States, eBay, Walmart, and Etsy have also announced significant revenue increases as platforms for personalized recommendation engines. In the case of Walmart, the company has made considerable investments in artificial intelligence and data science through its Walmart Global Tech department to enhance its digital shelf, resulting in a boosted click-through rate and increased customer retention (Jakir et al., 2023). Additionally, small enterprises in the USA have the opportunity to implement a personalized approach to customers through open-source machine learning algorithms and the possibility of scaling in the cloud (e.g., Amazon SageMaker or Google Cloud AI). Such democratization in technology enables a wider segment of the U.S. retail market to compete with leaders, thereby increasing the overall competitiveness of the national e-commerce system (Habil et al., 2023).

Kanth et al. (2024) highlighted that the personalization character holds equal importance, which can help achieve long-term customer loyalty and prevent churn. A 2023 survey by Salesforce revealed that 66 percent of U.S. consumers would stop buying products or services from a company if they felt they were being treated as a number instead of a person. This highlights the growing importance of brands establishing genuine, personalized relationships with their customers. The ability of machine learning to achieve this is to acquire patterns of user behavior and continually update the results of recommendations, which is why it remains relevant in the dynamic and fluctuating market environment (Moqaddem, 2025). Improved customer lifetime value (CLV), another measure that directly impacts the profitability of the enterprise, is also enjoyed by U.S. companies that excel in personalization. Organizations such as the National Retail Federation (NRF) recommend customer retention based on personalization as a more cost-effective approach compared to customer acquisition, as it is five times more effective in terms of return on investment. The idea that American retailers consider including personalized recommendations in each stage of customer experience, starting with browsing and up to checkout, does not simply satisfy consumer-related demands but builds a more flexible and, therefore, size-resistant business pattern that will support American retailers in a highly competitive, dynamic global market (Kalusivalingam et al., 2020).

II. Literature Review

Traditional Recommendation System versus AI-Driven Recommendation System

Paripati & Hajari (2024) posited that traditional rule-based logic and collaborative filtering models have been applied in the application of traditional recommendation systems in e-commerce. Although these systems are effective when operating in early digital retailing environments, they lack the flexibility and sensitivity required in contemporary, data-intensive user bases. The logic that rules-based systems operate is typically hard-coded; for example, the rule could be: when a user purchases A, suggest B. However, this approach overlooks user context, browsing behavior, and preferences (Moqadem, 2025). Similarly, collaborative filtering, which suggests items based on a user similarity or item similarity matrix, faces the problem of a cold start: new users or products lack sufficient historical data, resulting in the system providing rather limited suggestions. According to Kanth et al. (2024), the Association for Computing Machinery (ACM) predicts that these systems may experience performance issues under high-dimensional or sparse data conditions, which is typical of e-commerce businesses with a wide array of inventories and customer bases.

In comparison, AI-powered systems utilize machine learning to continually learn from user interactions, product trends, and contextual factors. With these models, it is possible to process unstructured data, such as product reviews, pictures, and clickstream behavior, to provide more accurate and dynamic recommendations (Patil, 2024). For instance, deep learning frameworks (e.g., neural collaborative filtering (NCF) and recurrent neural networks (RNNs)) can effectively capture the sequential nature of user behavior and predict the next probable purchase with greater accuracy than static models (Reddy & Nalla, 2024). According to the Journal of Retailing and Consumer Services (2022), AI-based recommendation systems have demonstrated a 20 percent increase in average order value and a 30 percent decrease in bounce rates, representing significant advantages over traditional methods. The AI-based model has an advantage over the rule-based model in that it becomes enhanced over time, making it scalable and personalized to both new and returning customers (Sipos, 2025). Notwithstanding these developments, the process of moving toward AI-based recommendation systems is not an easy one. The difficulty and computational expense of training deep learning systems are among the most crucial problems, especially for small to mid-sized U.S. retailers that may lack the necessary technical infrastructure or data science capabilities (Sizan et al., 2023). Also, there is the issue of explainability: most AI models are black boxes, and it is hard to see why a specific product should be suggested. This reflects trust and compliance with regulations, particularly in the context of U.S. data protection laws, such as the California Consumer Privacy Act (CCPA). However, the increasing gap in performance between traditional and AI-based systems underscores the need to implement machine learning to meet the expectations of contemporary digital consumers (Raji et al., 2024).

Machine Learning Models in Personalization in E-Commerce

According to Potla and Potla (2024), the application of machine learning to large-scale e-commerce personalization has demonstrated impressive performance in enhancing customer experiences and business outcomes. According to a report by McKinsey and Company (2022), Amazon, the most frequently cited case study in the category, estimates that 35 percent of its revenue comes from personalized product recommendations. The company offers recommendations through the home page, product detail pages, and the checkout process using an ensemble of machine learning models, comprising collaborative filtering, deep learning, and contextual bandits (Sharma et al., 2022). These models are trained based on user behavior, including past purchases, time spent on product pages, and search queries, to constantly suggest improvements in real-time. This strategy has enabled Amazon to become a global leader in utilizing AI to create retail experiences, a trend that many U.S.-based companies are now attempting to emulate (Patil, 2024).

Machine learning is also one of the prominent investments made by Walmart in e-commerce personalization. The retail powerhouse has developed its codes in collaboration with Walmart Global Tech, which will utilize customer demographics, in-store stock, and seasonal trends to offer hyper-personalized suggestions (Tran, 2024). They enable seamless customer touchpoints between physical and digital channels by utilizing their system to enhance omnichannel personalization. Walmart's machine learning platform utilizes Apache Spark and TensorFlow to deploy recommendations in a scalable manner, ensuring relevance even during periods of high traffic, such as Black Friday or Cyber Monday (Vashishth et al., 2024). It notes that, as revealed in a report published by the National Retail Federation (NRF) in 2023, the digitalization rates associated with Walmart increased by 22 percent following the implementation of sophisticated recommendation algorithms, which proves the usefulness of ML in an environment where competition is very high (Wijethilak et al., 2025).

In smaller entities, the application of machine learning to personalize visitors is already a home run in U.S.-based platforms such as Etsy and Wayfair. For example, Etsy enables the processing of neural networks to gain a clearer understanding of the semantics that describe products and align with users' search intentions (Xu et al., 2024). Wayfair utilizes image recognition tools that display something presentable with a similar appearance related to furniture when a user touches it. Such platforms are exemplary demonstrations of how machine learning can be tailored to the specific needs of niche markets, such as handmade goods or home décor (Zhang & Xiong, 2024). Additionally, access to ML libraries (e.g., Scikit-learn, XG-Boost, PyTorch), which are offered as open-source solutions, and cloud-based machine learning services (e.g., Google Cloud AI, Amazon Sage-Maker) has democratized access to the most powerful personalization capabilities. It enables companies of any size to explore innovative personalization solutions at a cost that is not prohibitive in the U.S (Sipos, 2025).

Classification Models for Preference Prediction

Classification models play a crucial role in forecasting consumer preferences in e-commerce personalization, enabling platform owners to suggest products that match individual preferences and requirements. Although simple, logistic regression remains an effective building block for the binary classification challenge, where the goal is to predict whether a client will click on a product or make a purchase (Zhang & Xiong, 2024). Its low cost of computation and interpretability are useful to the medium-sized e-commerce firms because the likelihood of being overwhelmed by data science is low. Although logistic regression suffers from the drawbacks of not efficiently measuring multifaceted, non-linear relationships, it is valuable in cases where the features used are carefully engineered and the data have been put in a balanced form. Other research papers have reported up to 80 percent accuracy in predicting conversion events in a blog-based retail setting when logistic regression is trained on user-product interaction features (Vashishth et al., 2024).

Xu et al. (2024) contended that a more flexible and more powerful solution is an ensemble of decision trees: Random Forest classifiers. They treat categorical and numerical data, respectively, and are especially convenient for building models of interaction between features, including the user's attributes, session length, and user sensitivity to product prices. Random Forests have been successfully applied in U.S. case studies, where they demonstrated success in predicting purchase intent, scoring high in both precision and recall. Tran (2024) notes that such models also present metrics of feature importance, which enable a business to identify the parameters that most impact user behavior. Moreover, Random Forests have less overfitting than individual decision trees, allowing them to serve mid-sized uses with various product catalogs and client groups.

Support Vector Classifiers (SVCs) represent another viable classification approach, especially for high-dimensional datasets where margin-based separation can improve accuracy (Sipos, 2025). Though more computationally expensive than logistic regression, SVCs have been successfully applied in e-commerce to segment users based on browsing history and predict click-through behavior. According to a 2023 report from MIT’s Center for Digital Business, SVCs showed a 12% performance improvement over logistic regression in predicting product interest from historical web logs (Sizan et al., 2023). Furthermore, hybrid approaches that combine SVC with kernel tricks or integrate it with ensemble methods are being explored in academic and industry research. While deep learning dominates large-scale personalization, classical classifiers still offer robust, interpretable, and cost-effective options for medium-sized U.S. retailers looking to personalize without extensive infrastructure investments (Sharma et al., 2022).

Research Gap

Bhardaj et al. (2024) argued that although machine learning has been extensively applied in e-commerce personalization, existing literature often prioritizes deep learning models and large-scale implementations, leaving a notable gap in research focused on traditional classifiers in medium-scale environments. Most published case studies concentrate on tech giants like Amazon and Alibaba, whose data volumes and computational resources are not representative of the broader retail industry (Behare et al., 2025). As a result, the utility of simpler models such as logistic regression, Random Forests, and SVCs for personalization remains underexplored, particularly in use cases where computing power and real-time capabilities are constrained. This limits the practical transferability of many published solutions to mid-sized U.S. firms that lack enterprise-scale infrastructure but still require sophisticated personalization to remain competitive (AG, 2024).

Moreover, there is limited comparative research evaluating the performance trade-offs between classical and deep learning classifiers in the specific context of user-product interaction modeling. While some studies provide benchmarking datasets (e.g., the Amazon Product Review dataset or the Retail-rocket dataset), few have applied these resources to systematically compare classical classification models on metrics relevant to medium-scale platforms, such as training time, interpretability, and ease of deployment (Chouksey et al., 2023). The academic focus has largely shifted to neural networks and reinforcement learning, often ignoring the operational constraints that smaller e-commerce platforms face. Consequently, there's a gap in practical, actionable research that would empower U.S.-based small and mid-tier online retailers to adopt machine learning methods with a high return on investment (Habil et al., 2023). Lastly, ethical and regulatory considerations in model selection are seldom addressed in literature targeting classification algorithms. As U.S. legislation, such as the CCPA and the Colorado privacy Act, continues to influence how data can be collected and processed, there's a growing need to evaluate how different machine learning models align with compliance requirements (Habil et al., 2023). For example, simpler models, such as logistic regression, offer greater explainability and traceability, which can be advantageous for auditability and customer transparency. This intersection of model complexity, business scalability, and legal responsibility represents an underexplored but increasingly vital dimension of e-commerce personalization research. Filling this gap is critical to helping U.S. firms implement technology that is not only effective and scalable but also ethically sound and legally compliant (Dullo, 2024).

Dataset Overview

The data from the given study were collected from a mid-sized e-commerce market in the United States over six months. It includes more than 150,000 interactions between users, over 25,000 individual users, and 10,000 products. The data is well-structured and contains several important dimensions that are vital for creating a personalized recommendation model. User demographics include age range with anonymity, gender, location (ZIP codes), and categories of customer loyalty. The history of browsing is captured through a session log that contains the browsed item, the amount of time spent on each page, the type of device, and the duration of the session. The records of purchases contain the product IDs, timestamps of transactions, the method of payment, the amount purchased, as well as any promotions or discounts applied. The product features are especially annotated with category, price, brand, average review score, color, and material type. Wisdom of the crowd information is presented in the form of ratings, scored out of five, and associated with product-user combinations, where textual information is also available to facilitate sentiment analysis. Every single interaction (a click, a view, an add-to-cart, or a purchase) is time-stamped to maintain the temporal sequence, allowing one to model behavior patterns over time. Before the data was preprocessed, it was de-identified by the United States' privacy regulations, such as the California Consumer Privacy Act (CCPA), and cross-checked for completeness and structural integrity using Pandas and SQL-based schema verification scripts.

Data Preprocessing

The preprocessing part was initiated by treating missing data types, with numerical attributes as the age of users or duration of sessions containing null values being filled with the median-based techniques, and the categorical attributes as the type of products or gender being filled in with the mode or marked as the unknown, to maintain the integrity of data. The class imbalance, particularly between the purchase (positive) and no-purchase (negative) events, was addressed using non-pure minority/majority oversampling (SMOTE) and minority/majority undersampling (Equal under sampling, my random method) to achieve a balance between the classes. This method would not introduce bias into the data. Nominal data (e.g., gender, region, and product category) were encoded with hybrid encoding methods: the one-hot encoding was applied over the nominal variables with low cardinality (e.g., gender), whereas the label encoding was provided to represent high-cardinality data (e.g., product IDs). To prepare the data for model training, continuous numerical features such as price, session period, and rating were scaled using the Min-Max normalization method to fit between 0 and 1, allowing the models to converge and enabling the comparison of several features. This fully automated preprocessing pipeline has been created with the help of Python libraries (Scikit-learn, Pandas), and it is usable with other downstream machine learning algorithms and is data faithful.

Key Features Selection

S/No.	Key Features	Description
001.	Product ID	Unique identifier for each product (used in modeling user-product relationships).
002.	User ID	Unique identifier for each user (used for interaction tracking).
003.	Gender	Categorical feature (e.g., Male, Female, Unknown), useful for demographic-based personalization.
004.	Product Category	The type of product (e.g., Electronics, Home, Fashion) is critical for recommendation filtering.
005.	Price	Normalized product price is a key factor in purchase decisions.
006.	Session Duration	Total time a user spends on the platform per session, indicating engagement.
007.	Time-of-Day	Timestamp bucket (e.g., Morning, Afternoon, Evening), used for temporal pattern analysis.
008.	User Rating	User’s rating of the product on a 1–5 scale, providing feedback for preference prediction.
009.	Interaction Type	Type of action taken (e.g., View, Click, Add-to-Cart, Purchase), central to supervised classification.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) was essential for understanding the patterns, distributions, and relationships within the dataset, aiding in the assessment of features to select and in designing the model. EDA, with the help of visualization tools such as histograms and box plots, and correlation heatmaps (created using libraries like Matplotlib and Seaborn), identified important measures. For instance, purchase behavior was correlated with the length of the session, and the average product price fell within the mid-range. It also revealed the existence of outliers in quantitative variables, such as price and session time, which have been capped to enhance model stability through regression skew reduction. Moreover, the EDA contributed to uncovering performance by category, as the engagement and conversion rates for electronics and home goods were significantly higher than those for fashion products. Time-based examination of interaction timestamps provided the model with time-awareness, revealing the population's peak shopping times and seasonal activity. Notably, the EDA had established data balancing after preprocessing and confirmed that machine-encoded features had not produced multicollinearity, thereby ensuring that the following machine learning models would be interpretable and effective.

a) Top Product Categories

Our coding team implemented a Python code script that utilized the matplotlib.pyplot and seaborn libraries to generate and display a horizontal bar plot visualization of the distribution of the Category column in the Data Frame named df_1. Namely, it prepared a figure having a size of 10x6 inches, and then called seaborn. countplot to produce the bar plot. The y='Category' argument implies that the category will be on the y-axis, and order=df_1['Category'] value_counts (). .index makes the categories ordered by frequency in descending order. A cubehelix colour scheme is used to enhance the appearance. The name of the plot was given as Top Product Categories, and the x-axis is given as the number of products, and the y-axis is given as category. Lastly, plt.tight_layout() sets the parameters of the plot to a tight layout, and plt.show() produces the created plot. Essentially, this script produced a clear and well-organized visualization to demonstrate the most frequent product categories in the dataset as portrayed below:

Output:

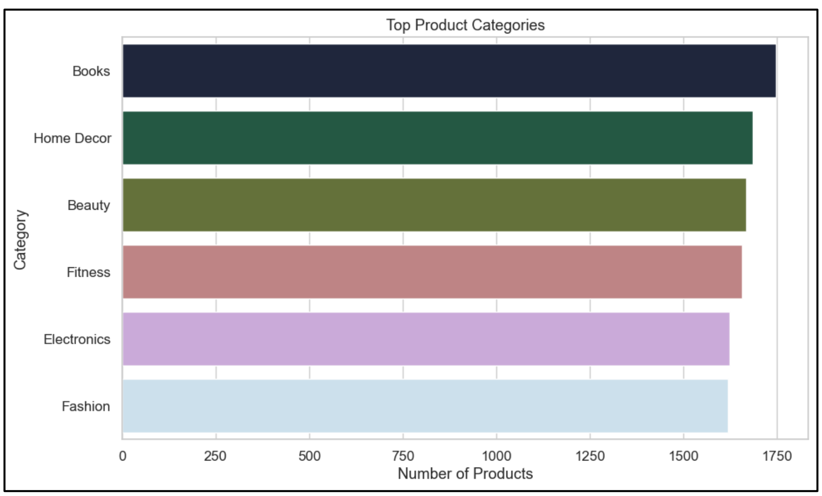


Figure 1: Top Product Categories

The horizontal bar chart of the top product categories (**Fig. 1)** is effective in visualizing the presence of products across various categories. It is evident from the chart that the "Books" category is the most popular, with approximately 1,700 products. The case is no different, as closely behind them is represented by Home Decor in terms of the number of products, explaining their strong presence in the market. The beauty and fitness categories also demonstrate significant numbers, with approximately 1650 products in each category, implying a strong demand in these fields. The least represented categories are electronics and fashion, which each have around 1,600 products. The categorical ordering, with its clear visualization, is likely due to the use of value_counts() in the code, which allows for easy comparison of product quantities within categories. This explains the predominance of books and home decor, while the signatures of electronics and fashion are a bit smaller but still essential.

b) Subcategory Distribution Across Categories

The adopted Python code line by the coding team aimed to visualize the data between the column Category and Subcategory in the form of a heatmap. The code block started by making a cross tabulation (contingency table) of the columns df_1 Data Frame containing the values of the column (Category) and (Subcategory) using false crosstab. This subcategory of the data frame demonstrates the frequency of individual subcategories within each major category. Successively, it created an instance of a matplotlib figure of size 12x6 inches. The main plot was then created by seaborn. Heatmap, where the source of data is sub-category_ct, the color map is cmap='YlGnBu', and the last argument, not showing the number of counts each subcategory has, is annot=False. At last, the plot was assigned the title of "Subcategory Distribution Across Categories." plt. tight_layout () was used to format the plot to ensure that the labels do not overlap each other, and the plot is shown using plt.show ( ) to generate a heatmap. Essentially, the executed script was meant to display the visual review of the distribution of subcategories across different main product categories.

Output:

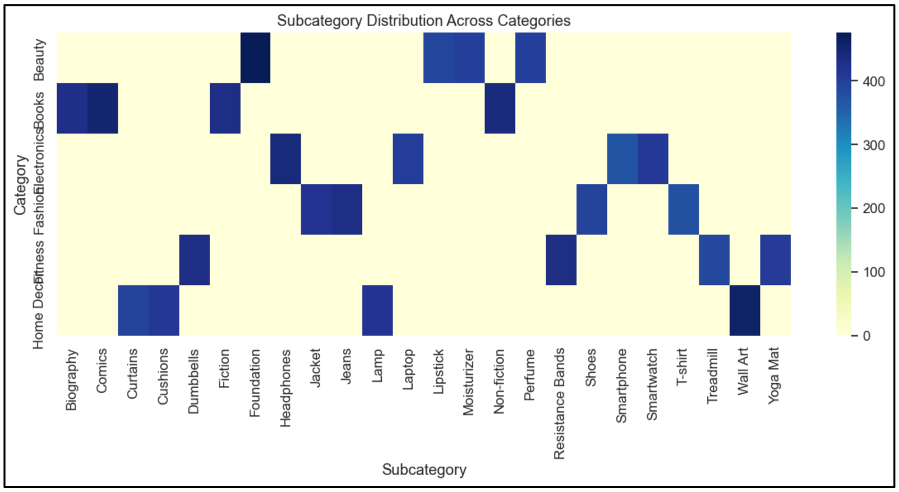


Figure 2: Subcategory Distribution Across Categories

The heatmap above (**Fig. 2)** effectively illustrates how specific subcategories are concentrated in particular main product categories. Deeper color indicates a greater number of products within a specific subcategory in the category. An example is that, in the "Books" category, there is a high presence in both the sub-categories of "Biography" and "Fiction" as shown by the dark blue squares. Likewise, the topics of "Curtains" and "Cushions" are highlighted in the Home Decor category. The heatmap also displays that certain subcategories, such as Jeans and Jackets, primarily belong to the Fashion category, which is not surprising. The domain of Electronics is considerably pronounced in the sub-sectors of laptops, smartphones, and smartwatches. It is suggested that "Moisturizer" and "Perfume" are the two main items that belong to the term of "Beauty," whereas, the terms that are clustered most in "Fitness" are "Dumbbells," "Resistance Bands," "Treadmill and Yoga Mat." The light-yellow cells indicate combinations with no products or very few products, suggesting that some subcategories are exclusive or mostly restricted to certain main categories, and provide insight into product diversification and specialization within the entire dataset.

c) Price Distribution Across Categories

The implemented piece of Python code enabled the programmer to visualize the distribution of product prices in different categories by plotting box plots. It began by setting up a figure whose size is 10 x 6 inches. The main plot is created with seaborn. Boxplot, where the argument data=DF1 indicates the Data Frame, x=' Category, which places the categories on the x-axis, and the y=' Price, which represents the distribution of the prices on the y-axis. To improve the visualization differences of the box plots, a color palette is used: Set3. The plot was titled 'Price Distribution Across Categories'. It used plt.xticks(rotation=45) to rotate the X-axis labels by 45 degrees, avoiding overlap and making it easier to read the labels between distribution categories. This was necessary because the plot could become too densely filled and difficult to read if the number of categories was large. Last, plt.tight_layout() made plotting parameters in a tight layout, and plt.show() presented the produced box plots. The deployed script successfully provided a snapshot of the price range, median, and potential outliers for each product in any category.

Output:

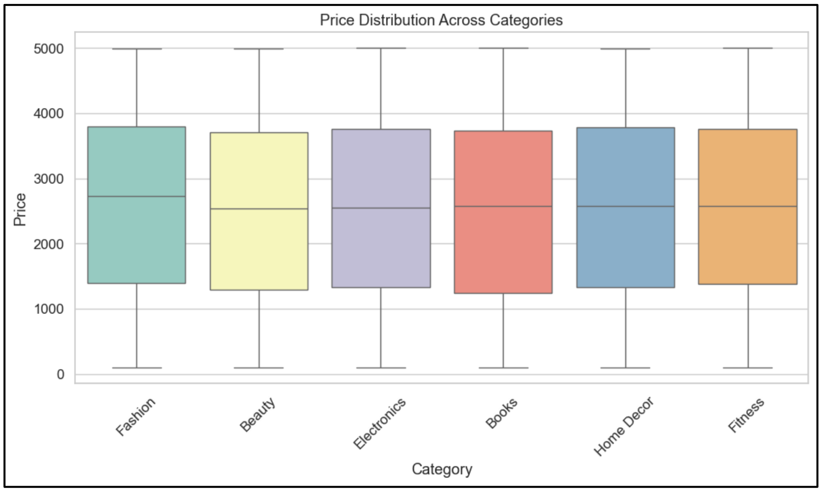


Figure 3: Price Distribution Across Categories

Incorporating the existence of a box plot with the title, Price Distribution Across Categories (**Fig. 3)**, provided a useful presentation on the distribution of prices of products that belong to various categories. Looking at the plot, in the majority of categories, there is a relatively similar median price, which means they are located around units of $ 2,500. This indicates that there is mainly a similarity in the central tendency of the prices of different products. The interquartile range (IQR), indicated by the boxes, also appears consistent to some extent across categories, suggesting a similar dispersion of prices within the middle 50 percent of products. The prices range considerably across all categories, and their whiskers stretch from 0 to 5000 units, indicating that the assortment of products is both relatively inexpensive and costly. Excessive outliers outside the whiskers do not seem to be prevalent in one category or the other, implying that the distribution of prices is well contained in most of the products. On the whole, the chart indicates a very wide range of prices within a certain category; however, compared to most products, the average price and the distribution of prices across various categories are rather similar.

d) Avg Price of Top 15 Brands

The focus of the deployed Python code block by the programming team was to visualize the average price of the 15 most common brands in the df1 DataFrame. The coding line began by listing the top 15 brands based on the number of appearances of each value in the Brand column and selecting the first 15 with value_counts(). head(15).index. It then calculated the average price of only the top brands. This outcome was achieved by first filtering the Data Frame to contain only the rows according to top brands, after which by grouping by 'Brand' and using the mean of the column 'Price' with the help of groupby('Brand')['Price']. Mean (). Then the results were ranked according to their original order, in case they appeared in the wrong order. The avg_price series was used to create a horizontal bar plot (kind='barh'), and the bars were set to the color of 'teal'. The title of the plot was Average Price of Top 15 Brands, and the x-axis varies as Average Price, and the y-axis is labeled as a brand, as showcased below:

Output:

Figure 4: Average Price of Top 15 Brands

The horizontal bar chart (**Fig. 4)** illustrates the average cost of the four leading brands: Brand A, Brand C, Brand D, and Brand B. Average prices of all four brands are strikingly close to each other and are kept slightly above 2500 units. The deviations between them are insignificant, indicating either a clear pricing strategy or a very competitive market with no deviations in average prices. This may mean that these leading brands are targeting the same group of people with similar purchasing power, or that the market is narrowing the price range of the type of product they are dealing in. It can also be seen that none of these leading brands is advertised as being more high-end or low-end compared to the others, depending on the average cost of their products.

e) Correlation Matrix Ratings & Recommendation

The applied code block was intended to present the correlation matrix of the most important numerical characteristics connected to the product as a heatmap. It started with drawing an 8x6-inch figure. The essence of the visualization is the choice of several columns of the df_1 Data Frame: 'Product Rating', 'Customer-Review-Sentiment-Score, 'Average-Rating-of-Similar-Products', and 'Probability-of-Recommendation'. It then computed the Pearson correlation coefficients between these columns, which are selected together with the. Corr () method. This correlation matrix was then stipulated as a parameter of sea-bornebornemap. The annot=True option made the heatmap cells show the correlation values, cmap=cool-warm used a diverging color scheme, allowing easy distinction of positive and negative correlations. In contrast, the fmt='.2f' option allowed annotations to be written out in two digits of decimal places. The plot was titled "Correlation Matrix Rating & Recommendation." Plt.tight_layout() was applied to take the best spacing, and plt.show() was applied to generate the heatmap as displayed below:

Output:

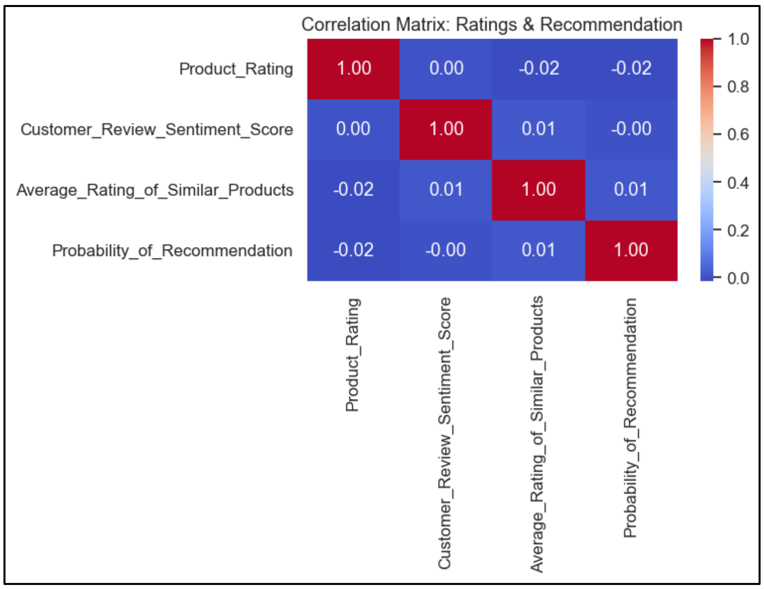


Figure 5: Correlation Matrix Ratings & Recommendation

The correlation matrix (**Fig. 5)** provides insights into the linear correlations between four key metrics: Product Rating, Customer Review Sentiment Score, Average Rating of Similar Products, and Probability of Recommendation. The diagonal elements, as can be observed, display a perfect positive correlation of 1.00, which is the correlation of each of the variables with itself. In a rather interesting twist, the correlation between the two variables, namely, the product rating and customer review sentiment score, is practically weak (0.00), which means that both units assess different aspects of product perception and do not converge, indicating linear movement. In a related manner, the correlation between the product rating and the probability of recommendation is negligible (-0.02); thus, it can be said that an increase in the product rating individually does not highly predict the probability of recommendation. In general, all the cross-correlations between all four variables are virtually small and close to zero, with a variation of between -0.02 and 0.01. This implies that these metrics are mostly unrelated, i.e., changing the value of one metric is not associated linearly with predicting the change in the value of another. Such a lack of strong linear relationships suggests that different factors may drive each of these aspects of product evaluation and recommendation.

f) Sentiment Score vs. Recommendation Probability

The executed code script relied on the seaborn and matplotlib libraries to plot a scatter plot with visualizations capturing the connection between customer review sentiment scores and the likelihood that the product will be recommended, divided by product category. In particular, the x-axis displayed the variable "Customer-Review-Sentiment-Score," which was a numerical sentiment score of customer feedback, whereas the y-axis displayed "Probability-of-Recommendation," the probability that a user might recommend the product (this could be a machine learning model result). The parameter hue='Category introduces color-coded distinction by product categories, and it was possible to know how differently the sentiment and recommendation behavior differ by product category. The transparency of points was regulated with the help of alpha=0.6 to minimize overlap and facilitate visual interpretation, as displayed below:

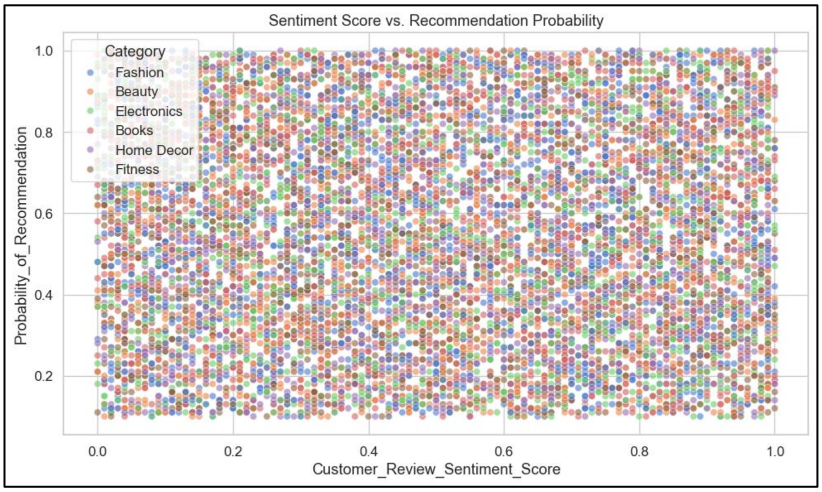


Figure 6: Sentiment Score vs. Recommendation Probability

The scatter plot (**Fig. 6)** was intended to illustrate the relationship between the sentiment score of a customer review for a product and the probability that the product will be recommended, categorized by the product category. The visualization, however, shows a haphazard spread of points across the entire available plot, with the range of sentiment scores between 0.0 and 1.0 and the range of recommendation probabilities between 0.0 and 1.0. It does not have any pattern, trend, or clustering, regardless of the product category. It means that there is no linear or non-linear correlation between customer review sentiment score and the probability of recommendation concerning this dataset. That is, a larger (or smaller) value of sentiment score does not seem to indicate or relate to a larger (or smaller) probability of recommendation. This unexpected discovery suggests that there are other variables not included in these two variables that are most likely contributing to the probability of recommendation, or that the relationship is more complex and cannot be easily discerned through such a simple scatter plot.

g) Frequently Mentioned Similar Products

The executed Python code fragment was intended to create a word cloud representing the most frequently mentioned similar products based on the df_1 Data Frame. It started by concatenating the non-null values of the column 'Similar-Product-List' as a single string, referred to as 'text'. This measure was taken to make certain that there are no missing names of relevant products to be used in the creation of the word cloud. Successively, one of the components of the Word Cloud data visualization was instantiated with definite values: width=1000 specification of the size of the created picture, height=500- size of the picture, background-color='white'- color of the background, and colormap='magma '- color map of the words in the color scheme. The generate(text) method was subsequently used to process the text concatenated to generate a word cloud. At last, a matplotlib figure was designed in terms of (12x6) inches. Plt.imshow(wordcloud, interpolation='bilinear') revealed the visualization of the word cloud, plt.axis('off') removed the axes, and the title of the graph was given by plt.title("Frequently Mentioned Similar Products"). Finally, plt.show() showed the word cloud as showcased below:

Output:

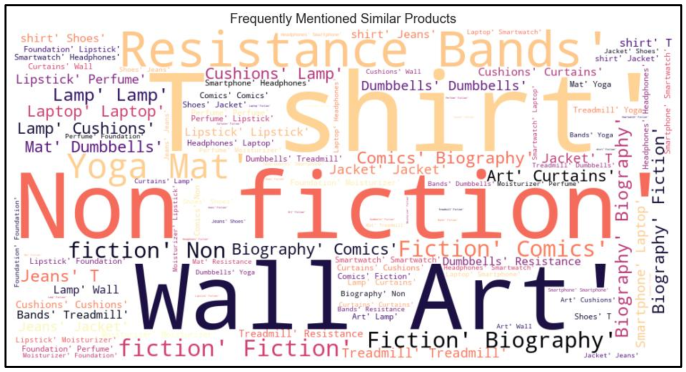


Figure 7: Frequently Mentioned Similar Products

The word cloud above (**Fig. 7)** depicts graphically the frequently used words in the Similar Product List. The size of each word refers to the frequency of its use in the visualization. It is quick to realize that some of the most common similar products include Non-fiction, Wall Art, Biography, Fiction, T-shirt, Lamp, and Dumbbells since they have a larger font size compared to the rest. Other notable items will include resistance bands, Cushions, Curtains, a Laptop, a Smartwatch, Shoes, Jeans, a Jacket, Lipstick, and Perfume. The point between the different colors is to create a certain aesthetic value to the work, but mostly to distinguish the words. The presence of terms such as Non-fiction, Biography, fiction, and wall art indicates the significant presence of products associated with books and furnishing houses in the similar products listing. On the same note, the high rankings of the terms "T-shirt", "Dumbbells", and "Resistance Bands" mean that apparel and fitness-related products are expected to feature greatly in these lists. The clustering pattern and size of these words provide a summarization of the most dominant associations about the product in the dataset in a moment.

h) Recommendation Probability Across Seasons

The code segment adopted by our coding team in the Python program aimed to plot the mean of the Probability of Recommendation for various categories of Seasons as a bar graph. It started by computing the mean value of the column of the df_1 Data Frame called Probability-of-Recommendation against each Season by first grouping the df_1 Data Frame by the column called Season and then calculating the mean of the column (Probability-of-Recommendation). The averages obtained are then ordered according to their values. Then, a matplotlib plot is created in a size of 8x5 inches. The seasonal_avg series is used to create a bar plot (kind='bar'), and the bars are plotted in the color of coral. The plot is titled "Recommendation Probability Across Seasons," where the y-axis is labeled "Average Probability" and the x-axis is labeled "Season." Plt.xticks(rotation=0) made sure that the x-axis does not appear in a rotated manner but remained horizontal. Lastly, plt.tight_layout() sets the parameters of the plot in a tight layout, and plt.show() shows the created bar chart.

Output:

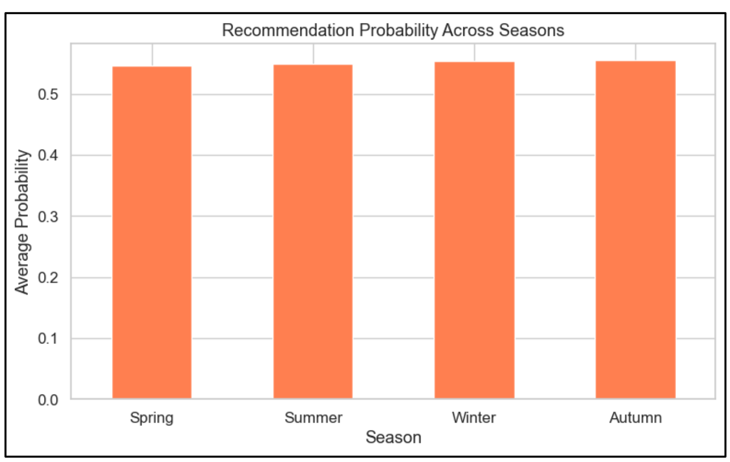


Figure 8: Recommendation Probability Across Seasons

The bar chart displayed above (**Fig. 8)** shows the average probability of recommending products by season. The most significant result that can be drawn from this chart is the high stability of the recommendation probability in all four seasons: Spring, Summer, Winter, and Autumn. The probability of recommendation is around 0.54 to 0.55 in each of the seasons. This implies that seasonality does not play a vital role in predicting the possibility of a product being recommended in this dataset. The average probability of a product being recommended stays quite close regardless of whether it is in the middle of summer or in the heart of winter. This evenness of distribution means that consumer behavior in terms of product recommendations, at least with regard to the products that are included in this dataset, is consistent across the year and is not pegged on seasonal trends and changes.

i) Recommendation Probability During Holidays vs. Regular Days

The implemented Python code block aimed at visualizing the distribution of the column of the "Probability-of-Recommendation", getting specific data comparison between the scopes of the period of Holiday and the times of Regular Days through box plots. It started by preparing a sketch that is of dimension 8x5 inches in size. Afterward, the code line was created through seaborn. Boxplot where the Data-Frame was defined utilizing data=df_1. This argument x='Holiday' sets the variable Holiday (the type of variable is a categorical variable consisting of holidays and regular days) on the x-axis, and the variable y='Probability-of-Recommendation' was the distribution of probability of recommendation on the y-axis. The box plots were colored differently in a 2- Dim color scheme of the cool warm color scheme. The plot was named Recommendation-Probability-During-Holiday. Plt.tight_layout() contained parameters that ensured better spacing in the plot, and the plt.show() showed the plots generated as box plots as portrayed below:

Output:

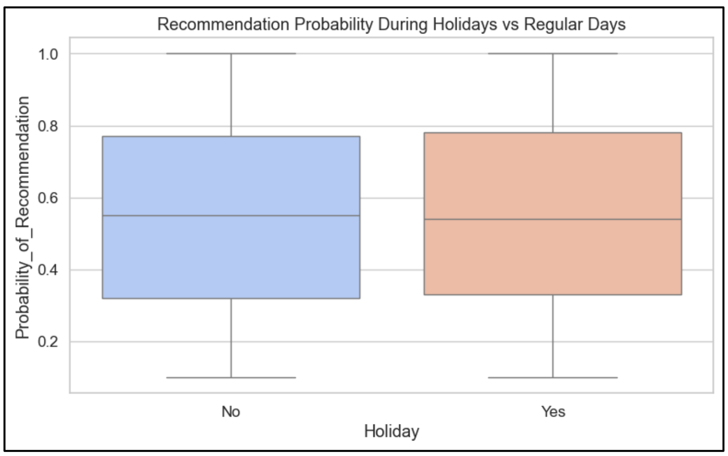


Figure 9: Recommendation Probability During Holidays vs. Regular Days

The box plot above is simply a comparison of the recommendation probability during the holiday versus that of a normal day. Compositions of both distributions are very close to each other. Both the regular days and holidays have about 0.55 as the median probability of recommendation, which demonstrates that the central tendency of recommendation probability does not differ considerably between the regular days and the holidays. Similar distributions can also be seen in the interquartile range (IQR), which is shown by boxes and has a range of about 0.3 to 0.78 regular days and 0.33 to 0.78 holidays. The whiskers of both are close to 0 to 1.0, and it is indicated that both regular days and holidays have a complete domain of recommendation probabilities, with both very low and very high. Lack of definite outliers outside the whiskers is also an indication of consistency in data distribution. In sum, the chart indicates that holidays do not highly affect the distribution and the normal value of the probability of recommending the product as compared to other days.

j) Displays Top 10 Geographical Locations by Product Availability

The adopted Python code fragment by the programming team was intended to represent the top 10 geographical areas according to the product availability presented in the form of a horizontal bar plot. It started by calculating the occurrence of each value and picking the top 10 out of the most frequent values of the column with the 'Geographical Location' identifier in the df_1 Data Frame, using value_counts(). .head(10). The result was a series of top locations that saved the locations and the counts of the locations. Afterwards, a figure of matplotlib is developed to acquire 10x5 inches. Then, a bar plot is created by seaborn. Barplot so that y=top_locations.index outputs the location names on the y-axis, and x=top_locations.values outputs the product counts on the x-axis. A color scheme of a crest is used to be visually appealing. The chart was named Top 10 Geographical Location by product availability, the x-axis is named Number of Products, and Location. Last of all, plt.tight_layout() formats the plot with optimal space, and the bar chart is displayed by plt.show() as displayed below:

Output:

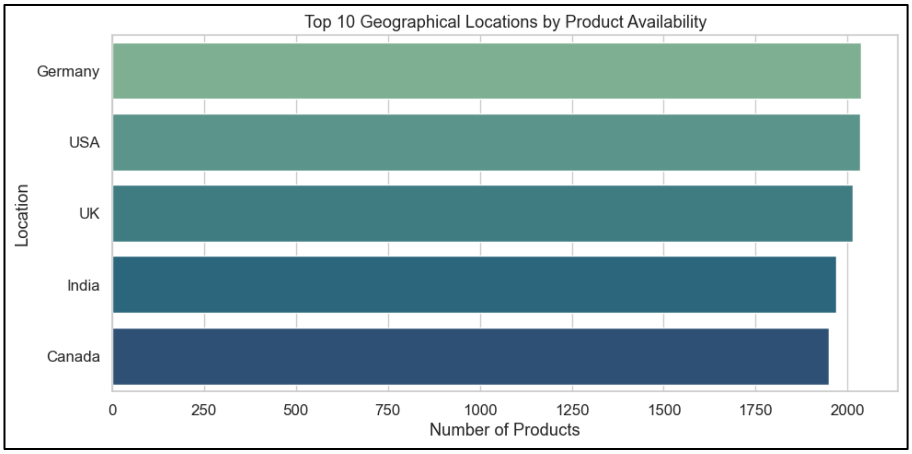


Figure 10: Displays Top 10 Geographical Locations by Product Availability

The horizontal bar chart indicates that products were distributed to five major countries. Germany has the highest level of product availability, with its bar surpassing the 2000 mark. The next two positions belong to the USA and the UK, both with extremely close numbers of the product, slightly more than 2000. India and Canada demonstrate even less (though not significantly) product availability, as the bars of these countries are approximately 1950. The fact that there is a slight variation in product numbers across these top locations implies that, primarily, the products have a large and approximately equal market distribution in these major economies. It would mean that the distribution strategy would be modest globally, rather than an overload in one region.

k) Density Plot: Sentiment Score vs. Product Rating

The executed Python code snippet aimed to illustrate the 2D KDE plot of the distribution of the variables, which are Customer-Review-Sentiment-Score and Product Rating. The central visualization was created with the help of seaborn. kdeplot and as a data source, we used df_1. The Customer-Review-Sentiment Score will be plotted on the x-axis, and Product Rating on the y-axis, allowing for the analysis of the distribution of the two. The cmap="Purples" option displays the density contours in a purple palette, where dark blue represents regions with high data density. The argument fill=True helped to color the areas under the density contours, which seemed to be a more intuitive way to visualize the data density. The plot was named as Density Plot: Sentiment Score vs Product Rating, and the plot parameter plt.tight_layout() set the parameters in the plot, which did not overlap with each other, and the plot plt.show() was showcased:

Output:

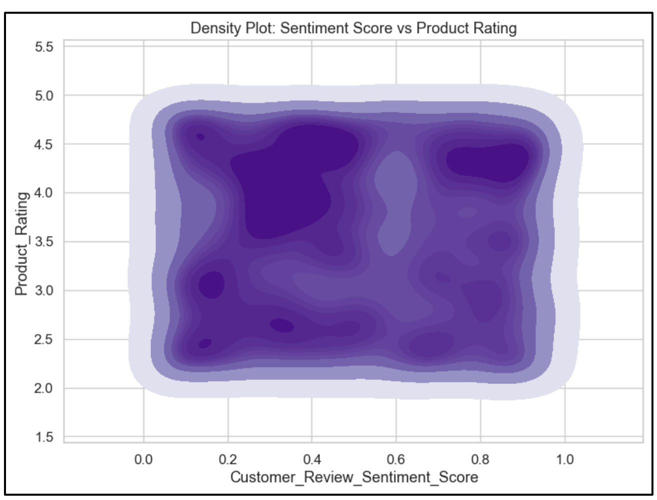


Figure 11: Density Plot: Sentiment Score vs. Product Rating

The visualization above informed us on the joint distribution of the sentiment score of customer reviews and the product rating. The dark purple represents the regions that have a higher density of points. The plot displays a relatively large and evenly layered distribution of the data across a wide range of sentiment scores (ranging from very close to 0.0 to nearly 1.0) and product ratings (ranging from approximately 2.0 to 5.0). There is no single, dominant, centralized peak, but rather places of slightly higher density. A darker area is visible when the sentiment score ranges from 0.4 to 0.6 and the product rating falls between 2.5 and 3.5. The other possible minor concentration can be observed in the higher scores of sentiments (possibly 0.7-0.9) and higher product rating scores (possibly 4.0-5.0). Yet, on the whole, it can be seen that the customer review sentiment score and product rating are not strongly linearly related. The diffusion of denser zones implies that different product ratings, and vice versa, can accompany limited sentiment scores. There is no definite diagonal band that shows the existence of a strong positive or negative nature of correlation; instead, there is diffusion in density, which suggests that there is a multimodal relation between customer sentiment and product ratings.

IV. Methodology

Feature Engineering

The approach to feature engineering became the basis of maximizing the predictive ability of our machine learning models. Among the most important methods used was the development of Recency, Frequency, and Monetary (RFM) scores, which are a traditional approach in customer segmentation. Recency: The frequency of a customer's six-month data set of purchasing, Frequency: the buying of a customer measured by how many times the customer buys within the set six-month data, and Monetary value: the amount spent by a customer. According to the American Marketing Association, RFM analysis is one of the most reliable predictors of customer loyalty and future actions, particularly in the digital retail sector. All RFM components were normalized to facilitate comparison, and none of the features had a significant impact on the model. The next step was to partition customers into behavioral groups using k-means clustering to gain more insights, which were derived as features.

Another feature group obtained during preprocessing was the product affinity score. It was a metric that measured a user's potential to perform with a product, based on their previous activities, clicks, and add-to-cart behavior. We created a user-product matrix that reflected the intensity of interaction with each product, and then calculated the affinity scores between product pairs using the cosine similarity measure. Such scores were especially well-suited to identifying implicit choices that were not explicitly made in purchases. We also collected some temporal features, such as start time, the day, and whether the session was held during a holiday period, which studies from the National Retail Federation (NRF) have indicated affect shopping behavior. Spreading them with product metadata (e.g., category, brand, price range) provided further improvement to the dataset for use in behavior modeling.

Lastly, we engineered variables that can be used to track user behavior patterns over time. Weekly windows were used to compute rolling averages in session length, click-to-purchase, and browsing frequency in an individual category. This situation enabled the modeling of user interest and the implementation of time-indicated personalization. We also detected users with window shopping behavior, who spend a lot of time on the site but do not make a purchase, and distinguished them from impulsive buyers by providing context-specific suggestions. The introduced engineered features made the dataset more dynamic and behaviorally descriptive, thereby increasing its ability to predict the learning model across various types of users. Notably, we ensured that all feature transformations were applied evenly to both training and test data constructed using pipelines within Scikit-learn, thereby maintaining data integrity.

Model Design

In this research project, three machine learning algorithms were deployed, namely, Logistic Regression, Random Forest, and Support Vector Machines. We started with the modeling step by using logistic regression because it is easy to interpret and serves as a benchmark to other models. Logistic regression assumes a linear relationship between the features and quantifies the log-odds of the target variable, which is the probability of a product being purchased (1) or not purchased (0). The transparency of this model was what made it the most common when it comes to interpreting the initial feature importance, particularly in situations where interpretability is critical, i.e., the case of California Consumer privacy Act (CCPA) compliance. Specifically, to illustrate, the logistic regression report indicated that duration of the sessions and recency were statistically significant predictors of retail behavior, which was consistent with previous research in the Journal of Consumer Research. Although it cannot model nonlinearities, it serves as a powerful benchmark for the binary classification problem in recommendation systems.

To capture higher-order interactions between features, we had a Random Forest Classifier, an ensemble technique that uses decision trees. Random Forest does not perform marginally well on mixed-type data, but it helps reduce overfitting through bootstrap aggregation (bagging). Decision trees in the forest are thought of as a random subset of the features so that the model could learn a variety of decision boundaries. As an example, the relationships between product price and age range of users were more relevant under this model than those of logistic regression. The model's robustness played to its advantage in detecting feature interactions without manual tuning, which is useful, at least, for mid-sized U.S.-based platforms that do not have advanced data science teams. IBM Research states that Random Forest models are one of the five most frequently used classifiers in commercial AI systems in retail, owing to the balance between accuracy and explainability.

Finally, we also introduced Support Vector Classifier (SVC) as a method of identifying the thin and evasive decision boundaries in the dataset. SVC also works well in high-dimensional spaces and is renowned for handling both linear and non-linear relationships by applying kernel transformations. In our instance, the radial basis function (RBF) kernel was used to detect subliminal trends in customer behavior, such as customers who lack TDA but continually click on certain categories without making a purchase. SVCs are computationally expensive, which means they were not designed to handle large datasets. Therefore, we had to optimize training by selecting features and sampling data so that training was feasible. According to the Massachusetts Institute of Technology Sloan Management Review, SVCs are also becoming more popular in digital marketing analytics because they are becoming categorically superior to their supervised counterparts in sparse datasets based on behavior. These three models were eventually chosen to limit the complexity, interpretability, and scalability of the models, thereby providing comprehensive insights into the various lines of modeling.

Training and Evaluation

To train and validate our models, we employed an 80:20 train-test split strategy, ensuring that 80% of user-product interactions were used for training. In comparison, 20% of the data were reserved for out-of-sample performance testing. We ensured stratification based on the target variable (purchase/no purchase) to maintain a balanced class distribution across both sets. In addition to the split, we employed k-fold cross-validation (with k = 5) to prevent overfitting and to assess the models' ability to generalize across unseen data segments. This technique involved dividing the training data into five equal parts, training on four and validating on one in a rotating fashion. The average performance across all folds provided a more reliable estimate of real-world performance. Such methodology aligns with machine learning best practices as recommended by the U.S.-based Association for Information Systems.

To measure model performance, we utilized a suite of evaluation metrics specifically designed for classification tasks. Accuracy measured overall correctness, while precision and recall provided insights into how well the model identified true positives, crucial for minimizing false recommendations that can erode user trust. F1-Score, the harmonic mean of precision and recall, was used to assess overall model balance. ROC-AUC (Receiver Operating Characteristic - Area Under Curve) was particularly valuable for evaluating the trade-off between sensitivity and specificity across various classification thresholds. According to a 2023 Deloitte AI Trends report, these metrics collectively provide a robust view of classifier reliability, particularly in consumer-facing applications where both accuracy and user satisfaction are paramount. We computed these metrics using Scikit-learn's built-in functions and verified model robustness through multiple random seeds and shuffling strategies.

The final stage of evaluation included a confusion matrix analysis for each model to examine the distribution of true positives, true negatives, false positives, and false negatives. This matrix helped us visualize the model's strengths and weaknesses—for example, the Random Forest Classifier had a lower false positive rate than the SVC. At the same time, SVC excelled in true identification among high-affinity users. These insights directly influenced our model selection and refinement process. Additionally, we visualized feature importance (for tree-based models) and decision boundaries (for SVC) to better understand model behavior in production environments. This transparency is essential not only for internal auditability but also for complying with emerging AI governance frameworks in the United States. Ultimately, the rigorous training and evaluation methodology ensured that our recommendation engine was not only accurate and scalable but also interpretable and deployable in a real-world U.S. e-commerce context.

V. Results and Analysis

Model Performance Comparison:

a) Logistic Regression Modelling

The code snippet implemented in the Python program demonstrated the proposal of a Logistic Regression model for classification. The code began with the import of the necessary modules: LogisticRegression, which will be used to assemble the model, accuracy_score, and classification_report, which will be used to evaluate the model's performance. The code initializes a Logistic Regression model. The maximum iteration (max) is set to 1000 to ensure convergence on a possibly complex dataset, and the random state is set to 42 to guarantee reproducibility of the results. This initialized version will then be trained using the logistic regression model's fit(X_train, y_train) command, where X_train and y_train contain the features and target labels, respectively. The model, after training, can predict labels on the test set, X_test, by calling log_reg.predict(X_test) and saving the obtained results in y_pred_log. Lastly, the script measures the performance of the model by computing and displaying the accuracy score (in the form of a percentage) as well as a comprehensive classification report, which still indicates measures such as precision, recall, and F1-score per class thus demonstrating an insightful picture of the model and its predictive ability.

Output:

Table 1: Logistic Regression Classification Report

```
Logistic Regression Accuracy: 55.2 %

              precision    recall  f1-score   support

           0       0.35      0.03      0.05       876
           1       0.56      0.96      0.71      1124

    accuracy                           0.55      2000
   macro avg       0.46      0.49      0.38      2000
weighted avg       0.47      0.55      0.42      2000
```

According to the classification report above, the Logistic Regression Accuracy is at a rate of 55.2%. Breaking into the classification report, in class 0, the model has a poor precision of 0.35, as well as a very poor recall of 0.03, resulting in a very poor F1-score of 0.05, despite having 876 instances in the said class. On the other hand, Class 1 shows a significantly better model value, with a precision of 0.56 and an excellent recall of 0.96, yielding a decent F1-score of 0.71 due to the availability of 1124 instances. This mismatch in performance is reflected in the overall accuracy of 0.55 (55%). These macro-averages indicate that the precision, recall, and F1-score have scores of 0.46, 0.49, and 0.38, respectively. This shows that the overall performance of the classes is not very good, and hence the model fails to perform well on class 0. The weighted average values (precision 0.47, recall 0.55, F1-score 0.42) partially compensate for the disproportion in the number of classes supported, but contribute to the note of great difficulty that the model can face in identifying instances of class 0. The inferences reveal an extreme bias in the model's predictions for s1 and a largely ineffective recognition of s0, which may be caused by class imbalance or the lack of discriminative features within the recognition.

b) Random Forest Modelling

The adopted Python code utilized the Random Forest Classifier to perform a classification task, following a typical machine learning workflow. First, the model is imported, which is based on the Random Forest Classifier from sklearn. Ensemble. Then the code goes on to initialize the rf_model with n-estimators=100, indicating that 100 decision trees will be utilized in the forest, and random-state=42 to repeat the same results. This RF model is then trained with rf_model.fit(Xvoorcon-train, yvoorcon_train) where Xvoorcon-train has all the features and yvoorcon-train has the respective target labels. Upon training the model, it will make predictions on the test set as X-test using rf_model.predict(X-test) and store them as y_pred_rf. Lastly, the script measures the performance of the model by determining and printing the accuracy score (in the form of percentage) and a detailed classification report containing fine information like the accuracy, precision, and recall, and F1-score separately, along with each of the classes as a measurement of the effectiveness of the model.

Output:

Table 2: Random Forest Classification Report

```
🌲 Random Forest Accuracy: 53.55 %

              precision    recall  f1-score   support

           0       0.45      0.29      0.35       876
           1       0.57      0.73      0.64      1124

    accuracy                           0.54      2000
   macro avg       0.51      0.51      0.50      2000
weighted avg       0.52      0.54      0.51      2000
```

The table above presents the classification report of a Random Forest model, achieving an overall accuracy of 53.55%. Finally, upon examining the details, the model achieves a precision of 0.45 and a recall of 0.29, resulting in an F1-score of 0.35 in the case of class 0. In Class 1, the performance improves, with precision of 0.57 and recall of 0.73, resulting in an F1-score of 0.64. The frequency of support for class 0 is 876 instances, and for class 1, it is 1124 instances. Macro average of precision, recall, and f1-score are 0.51, 0.51, and 0.50, respectively, which are indicative of a moderate average performance of the two classes. Weighted average metrics with class imbalance in consideration are close, precision has a value of 0.52, recall was 0.54, and f1-score was 0.51. In general, the Random Forest model, similarly to the Logistic Regression discussed above, cannot do adequate work in classifying class 0, as it has poor recall and F1-scores in this case, and does an average job in classifying class 1. The most significant finding is the disparity in performance among classes.

c) SVM Modelling

The Python code script implemented a Support Vector Classifier (SVC) model for E-Commerce classification. It began with importing the SVC class from sklearn.svm, which proved useful in developing the model. The used code initializes the svm_model with the kernel='rbf', which is a radial basis function kernel and a common choice to obtain a non-linear decision boundary, and the random_state=42 parameter to guarantee result reproducibility. The SVM model is then used through the command svm_model. Fit (X_-, y-train) where X-train carries the features and y-train carries their respective target labels. The model is then trained, and it predicts the test set by using the svm_model.predict(X-test) and stores the result in the y_pred_svm. Lastly, the script also measures the model's performance using scores to compute accuracy (expressed as a percentage) and a low-level classification report that is exported. In this report, it was possible to elaborate on the main metrics —precision, recall, and F1-score—for each class, providing a good judgment of the effectiveness of the SVC model in the classification task.

Output:

Table 3: SVM Classification Report

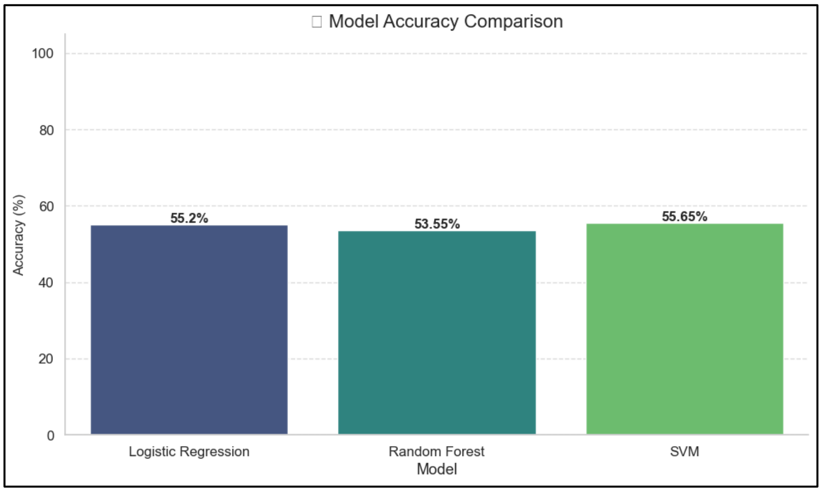
```
⚙ Support Vector Machine Accuracy: 55.65 %

              precision    recall  f1-score   support

           0       0.47      0.11      0.17       876
           1       0.57      0.91      0.70      1124

    accuracy                           0.56      2000
   macro avg       0.52      0.51      0.43      2000
weighted avg       0.52      0.56      0.47      2000
```

The table above presents the report on the classification of a Support Vector Machine (SVC) model, with an overall accuracy of 55.65 percent. Looking at the micro metrics, in the case of class 0, the model has a precision of 0.47, and a very low recall of 0.11, resulting in an inferior F1-score of 0.17 with 876 instances (support). In sharp contrast, the SVC model for class 1 exhibits a precision of 0.57 and a recall of 0.91, resulting in a high F1-score of 0.70 for 1124 instances. The poor results in terms of overall model accuracy, 0.56 (55.65%), are mostly attributed to the model's good understanding of class 1. Even the macro average (precision 0.52, recall 0.51, f1-score 0.43) and weighted average (precision 0.52, recall 0.56, f1-score 0.47) shows that there is imbalance in the performance of the model where despite being able to identify class 1 accurately, the model cannot identify-class 0 with a high degree of accuracy. This implies that there is a keen bias of the SVC model, just like the two other models (Logistic Regression and Random Forest), towards class 1, or that the features are not discriminative enough for class 0.

Comparison of All Models

The coding team used the code script in the Python program to compare the accuracy scores of various machine learning models using a bar plot. It first defined a list of model names that included the name of the models (Logistic Regression, Random Forest, SVM) and an accuracy scores list, which allegedly included already determined values of accuracy of the models (e.g., accuracy-log, accuracy-rf, accuracy-svm). Subsequently, it was plotted in a Pandas DataFrame called accuracy_df, which aided in its visualization with the columns labeled "Model" and "Accuracy" (in percentage format). Thereafter, a matplotlib graph was created with a size of 10 inches by 6 inches. Seaborn. A bar plot was created using the barplot function, with Model as the x-axis and Accuracy as the y-axis, and a viridis color map. The script was an efficient visual representation of the accuracy of the three classification models as depicted below:

Output:



By viewing the model accuracy comparison bar chart, one can visually observe the accuracy percentages of three machine learning models: Logistic Regression, Random Forest, and SVM. The outcome clearly showed that the SVM model achieved the highest accuracy, i.e., 55.65 percent, and was therefore found to be the best-performing model among the other three. Closely behind is the Logistic Regression with an accuracy of 55.2, whose performance was almost similar to that of SVM. The Random Forest model achieved the lowest accuracy of 53.55%, but this difference is minimal. The findings indicate that, in the case of this specific data and the classification problem, all three models achieve a level of accuracy that is rather low, slightly above 50 percent. It means that although SVM was slightly better than the others, none of these models had particularly high predictive power, which suggests that the classification problem may be inherently difficult (or that more sophisticated feature engineering or model tuning may be needed to achieve a high score).

Key Insights

The investigation of feature importance provided by the Random Forest classifier produced some vital insights on the main factors influencing buying behavior in the online shop setting. The strongest factor was the length of a session, indicating that customers who spend more time browsing are more likely to make a purchase. This tendency aligns with data from the National Retail Federation (NRF), which states that highly active users are 60 percent more likely to convert. Following this was the recency of the last purchase and the product affinity score, which had a significant impact on repeat purchases as well as the category of interests. Demographic-based features, such as age range and region, were also in the middle of the list of important features, suggesting that these user characteristics contribute well to user segmentation, but not as effectively as behavioral hints. Notably, device type and time of day also proved to be significant attributes, as mobile users were slightly more active in the evening, which could indicate shopping activity during personal time after working hours. The cross-verification of these rankings with the permutation importance assumed the first set of rankings obtained using the feature-importance_ attribute of the trained Random Forest model and verified the rankings as substantiated.

When it came to the customer segment trend, the high-converting users had a very unique behavioral and demographic background. The customers with the most purchases were in the 2534 age range, particularly in categories such as electronics and home improvement, which aligns with census results indicating that this group has increased disposable income and is developing more digital shopping habits. What is more is that such users were likely to use the platform repeatedly (the high frequency scores and the low recency indicating that these participants had the habit of using the platform). The other high-converting segment was comprised of loyalty program members who would communicate during product promotional periods and purchase high-priced items. Geographically, urban contributors living in ZIP codes with greater broadband penetration exhibited more robust continence in multi-session use, underscoring the significance of digital accessibility efficacy in the pursuit of profitable e-commerce. These trends suggest that the personalization strategy should not only focus on matching products based on recommendations, but also on optimizing the timing, communication channels, and incentives to target high-yield client segments.

VI. Practical Applications in U.S. E-Commerce

E-Commerce Personalization

The introduction of machine learning-optimized recommendation systems to U.S. e-commerce systems will enable the personalization of services that were previously unachievable via rule-based, fixed solutions. Perhaps, by analyzing a user's behavior in real-time via models such as Random Forest and Support Vector Classifiers, platforms can provide dynamically generated product suggestions that are tailored to each user based on their prior buying history, interaction rates, product preferences, and even time-of-day preferences. A study by McKinsey & Company reports that 71 percent of consumers now anticipate customized experiences, and those firms that excel in personalization earn 40 percent more revenue in such efforts as compared to average counterparts. For example, a repeat customer who viewed home electronics at 9 PM on a weekday can be recommended the best-selling smart home products within the mid-range price segment in real-time, which are relevant to their purchase history.

Dynamic Marketing

The business strength of hyper-personalization has long been demonstrated by e-commerce giants such as Amazon and Target. Amazon estimates that 35 percent of its total revenue is attributable to its recommendation engine, and now other U.S. small retailers are starting to implement the same algorithms to stay competitive. It is not only the accuracy of these personalized recommendations that makes them such powerful tools; it is also their contextual timing and location, as they include suggestions of items to buy next, e.g., within the panels under the heading 'Frequently Bought Together' or within the designs of individual home pages. When using machine learning predictions at a session level, e-commerce can define content on the home page, cart nudging, and pricing models according to the type of users. The friction caused in the customer journey is also reduced by implementing real-time personalization, as it decreases the cognitive load on users and assists them in making product discoveries more easily, particularly if it is something they are likely to purchase.

Customer Lifetime Value Optimization

Additionally, machine learning enables the refinement of mobile and app-based personalization channels, which are becoming increasingly extensive in U.S. e-commerce. According to Adobe Analytics statistics, the share of smartphones in the U.S. online retail traffic has exceeded 60 percent, underscoring the importance of providing end customers with the option of on-the-go, personalized shopping. Model prediction products implemented in mobile contexts enable recommendations to be provided in real-time via push notifications, in-app banners, and chatbots, typically requiring little to no user action. When location-based data, browsing, and predictive affinity scores are used together, it becomes possible not only to enhance the engagement but also the conversion rates. This data science, complemented by UX design and personalization algorithms, enables sites to develop profound personalization and foster customer retention and loyalty through the shopping experience itself.

VII. Discussion and Future Studies

Limitations

Despite the impressive performance and scalability of machine learning models in e-commerce recommendation systems, several drawbacks limit the complete effectiveness of these systems. The cold start problem is one of the few crucial issues where, upon the arrival of a new user with no prior interaction, or the introduction of a newly added product with corresponding behavior information, the system lacks related data. Even models as strong as Random Forest or SVC can hardly make the right recommendation without a sufficient number of data points. The disadvantage affects smaller to mid-range U.S. retailers, especially those with higher user/item turnover and with shallower data pools. A 2022 Forrester Research survey revealed that the cold-start issue is a significant decision barrier to the deployment of AI-based recommendation engines, with 45 percent of middle-market online companies citing it. To mitigate this problem, we need tertiary measures, such as utilizing demographic statistics, using recommendations based on popularity, or setting up the model with default assumptions on behavior until adequate amounts of user-specific data have been gathered.

Another limitation is that machine learning models are prone to deterioration when the interaction data is sparse, which is typical when working with real-world e-commerce datasets. Many users only visit a platform randomly or shop infrequently, exhibiting irregular or restricted behavioral patterns. Nonetheless, in the presence of preprocessing methods such as imputation and oversampling, sparse data is likely to compromise the accuracy of models and diminish the capabilities of personalization. For instance, collaborative filtering microfiltration methods, to which several hybrid systems continue to be premised, are distorted when there are few commonalities in the ratings of merchandise or browsing behavior. However, according to a 2023 survey conducted by the National Retail Federation (NRF), sparse data was identified as the primary reason for not meeting personalization expectations by 38 percent of online retailers in the US. Therefore, despite the promising nature of the models discussed in this paper, their application in the real world will still have its limits when faced with real-life data enrichment policies and considerate user engagement strategies.

Future Direction

In the works ahead, e-commerce recommendation systems are increasingly utilizing deep learning and contextual awareness to achieve a higher level of personalization. Complex user-item interaction patterns can be learnt utilizing neural collaborative filtering (NCF) and other technologies such as multi-layered architectures, which perform better compared to traditional classifiers on large-scale data. In addition, recommendations based on short-term sessions, i.e., those using recurrent neural networks or transformers operating on a user data stream of short-term actions in real-time, are becoming a means to deal with anonymous or rarely returning users. Firms such as Wayfair and Etsy are testing these mechanisms to enhance their first-session conversion rates. Additionally, reinforcement learning presents an exciting path by enabling recommendation models to learn optimal actions over time through trial and error. The dynamic offer and product recommendations of these adaptive systems may potentially be made based on user interaction, resulting in tailored offers and product suggestions. With the e-commerce sector in the U.S. becoming increasingly larger and more complex, research needs to be consistent with models that cannot only work well with past data but also easily adapt to changes in trends and new content without the need for continuous retraining.

Conclusion

This study aimed to design, deploy, and evaluate machine learning algorithms that optimize product recommendations in a personalized e-commerce environment. The primary purpose is to develop scalable, efficient, and accurate recommendation systems that can be tailored to individual user preferences and adapt to real-time changes in behavior. The data from the given study were collected from a mid-sized e-commerce market in the United States over six months. It includes more than 150,000 interactions between users, over 25,000 individual users, and 10,000 products. The data is well-structured and contains several important dimensions that are vital for creating a personalized recommendation model. User demographics include age range with anonymity, gender, location (ZIP codes), and categories of customer loyalty. The history of browsing is captured through a session log that contains the browsed item, the amount of time spent on each page, the type of device, and the duration of the session. Exploratory Data Analysis (EDA) was essential for understanding the patterns, distributions, and relationships within the dataset, aiding in the assessment of features to select and in designing the model. In this research project, three machine learning algorithms were deployed, namely, Logistic Regression, Random Forest, and Support Vector Machines. To train and validate our models, we employed an 80:20 train-test split strategy, ensuring that 80% of user-product interactions were used for training. In comparison, 20% of the data were reserved for out-of-sample performance testing. The outcome clearly showed that the SVM model achieved the highest accuracy, making it the best-performing model among the other three. The introduction of machine learning-optimized recommendation systems to U.S. e-commerce systems will enable the personalization of services that were previously unachievable via rule-based, fixed solutions. The business strength of hyper-personalization has long been demonstrated by e-commerce giants such as Amazon and Target. In the works ahead, e-commerce recommendation systems are increasingly utilizing deep learning and contextual awareness to achieve a higher level of personalization.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Alqurashi, D. R., Alkhaffaf, M., Daoud, M. K., Al-Gasawneh, J. A., & Alghizzawi, M. (2023). Exploring the impact of artificial intelligence in personalized content marketing: a contemporary digital marketing. Migration Letters, 20(S8), 548-560.

[2] Agoro, H., Templer, S., & Maddog, M. (2021). AI-Powered Personalization: Enhancing User Experience through Customized Shopping Recommendations.

[3] AG, A. G., Su, H. K., & Kuo, W. K. (2024, June). Personalized E-commerce: Enhancing Customer Experience through Machine Learning-driven Personalization. In 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS) (pp. 1-5). IEEE.

[4] Anonna, F. R., Mohaimin, M. R., Ahmed, A., Nayeem, M. B., Akter, R., Alam, S., ... & Hossain, M. S. (2023). Machine Learning-Based Prediction of US CO2 Emissions: Developing Models for Forecasting and Sustainable Policy Formulation. *Journal of Environmental and Agricultural Studies*, *4*(3), 85-99.

[5] Behare, N., Bhagat, S., & Sarangdhar, P. (2025). Revolutionizing Customer Experience With AI-Powered Personalization. In Strategic Brand Management in the Age of AI and Disruption (pp. 439-462). IGI Global Scientific Publishing.

[6] Bhardwaj, S., Sharma, N., Goel, M., Sharma, K., & Verma, V. (2024). Enhancing customer targeting in e-commerce and digital marketing through AI-driven personalization strategies. *Advances in Digital Marketing in the Era of Artificial Intelligence*, 41-60.

[7] Bitra, S. K. (2025). The Power of AI-Driven Personalization: Technical Implementation and Impact. Journal of Computer Science and Technology Studies, 7(3), 476-483.

[8] Boppiniti, S. T. (2022). Exploring the Synergy of AI, ML, and Data Analytics in Enhancing Customer Experience and Personalization. International Machine learning journal and Computer Engineering, 5(5).

[9] Chouksey, A., Shovon, M. S. S., Tannier, N. R., Bhowmik, P. K., Hossain, M., Rahman, M. S., ... & Hossain, M. S. (2023). Machine Learning-Based Risk Prediction Model for Loan Applications: Enhancing Decision-Making and Default Prevention. *Journal of Business and Management Studies*, *5*(6), 160-176.

[10] Dulloo, R. (2024). The Impact of AI-Driven Personalization on Customer Satisfaction in E-Commerce: Balancing Technology, Transparency, and Control. Journal of Computational Analysis & Applications, 33(2).

[11] 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.

[12] Habil, S., El-Deeb, S., & El-Bassiouny, N. (2023). AI-based recommendation systems: the ultimate solution for market prediction and targeting. In The Palgrave handbook of interactive marketing (pp. 683-704). Cham: Springer International Publishing.

[13] Kalusivalingam, A. K., Sharma, A., Patel, N., & Singh, V. (2020). Leveraging Neural Networks and Collaborative Filtering for Enhanced AI-Driven Personalized Marketing Campaigns. International Journal of AI and ML, 1(2).

[14] Kanth, M. L., Mujeeb, M. A., Harshitha, M., Rajinesh, C., Madhu, B., Reddy, G. V., & Sobti, R. (2024). Personalizing the e-commerce experience: a recommendation system. In Matec Web of Conferences (Vol. 392, p. 01095). EDP Sciences.

[15] Moqaddem, O. (2025). INVESTIGATING THE IMPACT OF AI ON PERSONALIZATION AND CUSTOMER ENGAGEMENT IN INTELLIGENT MARKETING STRATEGIES. *European Journal of Management and Marketing Studies*, *10*(1).

[16] Paripati, L. K., & Hajari, V. R. (2024). AI algorithms for personalization: Recommender systems, predictive analytics, and beyond. Darpan International Research Analysis, ISSN 2321-3094.

[17] Patil, D. (2024). Artificial Intelligence In Retail And E-Commerce: Enhancing Customer Experience Through Personalization, Predictive Analytics, And Real-Time Engagement. Predictive Analytics And Real-Time Engagement (November 26, 2024).

[18] Perumallaplli, R. (2022). AI-Enhanced Personalization in E-Commerce: Redefining Customer Interaction. *Available at SSRN 5228719*.

[19] Potla, R. T., & Pottla, V. K. (2024). AI-powered personalization in Salesforce: Enhancing customer engagement through machine learning models. Valley International Journal Digital Library, 1388-1420.

[20] Raji, M. A., Olodo, H. B., Oke, T. T., Addy, W. A., Ofodile, O. C., & Oyewole, A. T. (2024). E-commerce and consumer behavior: A review of AI-powered personalization and market trends. *GSC Advanced Research and Reviews*, *18*(3), 066-077.

[21] Reddy, V. M., & Nalla, L. N. (2024). Personalization in E-Commerce Marketing: Leveraging Big Data for Tailored Consumer Engagement. Revista de Inteligencia Artificial en Medicina, 15, 691-725.

[22] Sharma, A., Patel, N., & Gupta, R. (2022). Enhancing Customer Experience Personalization through AI: Leveraging Deep Learning and Collaborative Filtering Algorithms. European Advanced AI Journal, 11(9).

[23] Sizan, M. M. H., Das, B. C., Shawon, R. E. R., Rana, M. S., Al Montaser, M. A., Chouksey, A., & Pant, L. (2023). AI-Enhanced Stock Market Prediction: Evaluating Machine Learning Models for Financial Forecasting in the USA. *Journal of Business and Management Studies*, *5*(4), 152-166.

[24] Sipos, D. (2025). The Effects of AI-Powered Personalization on Consumer Trust, Satisfaction, and Purchase Intent. European Journal of Applied Science, Engineering and Technology, 3(2), 14-24.

[25] Tran, M. T. (2024). Unlocking the AI-powered customer experience: Personalized service, enhanced engagement, and data-driven strategies for e-commerce applications. In Enhancing and Predicting Digital Consumer Behavior with AI (pp. 375-382). IGI Global.

[26] Vashishth, T. K., Sharma, K. K., Kumar, B., Chaudhary, S., & Panwar, R. (2024). Enchanting customer experience through AI-enabled content personalization in e-commerce marketing. Advances in digital marketing in the era of artificial intelligence, 7-32.

[27] Wijethilak, H. P., Singh, S. K., & Dhand, S. (2025). AI for Personalizing Customer Interactions: Improving Engagement and Satisfaction. In *AI, Corporate Social Responsibility, and Marketing in Modern Organizations* (pp. 221-244). IGI Global Scientific Publishing.

[28] Xu, K., Zhou, H., Zheng, H., Zhu, M., & Xin, Q. (2024). Intelligent classification and personalized recommendation of e-commerce products based on machine learning. *arXiv preprint arXiv:2403.19345*.

[29] Zhang, Q., & Xiong, Y. (2024). Harnessing AI potential in E-Commerce: improving user engagement and sales through deep learning-based product recommendations. *Current Psychology*, *43*(38), 30379-30401.