Analysing Consumer Behaviour Using Machine Learning to Enhance Customer Cross-selling Recommendation

By

Valentine Masungwa Mbuthu

149910

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Data Science and Analytics at Strathmore University

Institute of Mathematical Sciences and iLab Africa

Strathmore University

Nairobi, Kenya

June, 2025

Declaration and Approval

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the dissertation contains no material previously published or written by another person except where due reference is made in the dissertation itself.

Student's Name: **Valentine Masungwa Mbuthu**

Sign: _____ Date: 21/05/2025

Approval

The dissertation of **Valentine Masungwa Mbuthu** was reviewed and approved by the following:

Dr. John Olukuru,
Head of Data Science and Analytics,
Strathmore Univeristy.

Dr. Godfrey Madigu,
Dean, Institute of Mathematical Sciences,
Strathmore University.

Dr. Bernard Shibwabo,
Director of Graduate Studies,
Strathmore University.

Abstract

The growing complexity of consumer behavior in the retail industry necessitates advanced solutions for optimizing cross-selling and inventory management. Traditional methods often lack scalability, personalization, and adaptability. This study develops a hybrid machine learning recommendation system using customer demographics, purchase history, and product data to enhance cross-selling strategies. By integrating K-means clustering, DBSCAN, Collaborative Filtering, and Content-Based Filtering, the system delivers personalized product suggestions and identifies slow-moving inventory. Deployed through a Streamlit interface, the hybrid model achieved improved precision (20.7%), recall (67.2%), and NDCG (0.508). Statistical testing confirmed these gains were significant, and segmentation was shown to improve recommendation quality. The system offers practical benefits in sales growth and stock optimization, with future potential for real-world deployment to drive sustainable, personalized retail experiences.

Keywords: Cross-selling, Machine Learning, Customer Segmentation, Inventory Optimization, Retail Analytics

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AB Attitude towards the Behaviour

AI Artificial Intelligence

API Application Programming Interface

AUC Area Under the Curve

BBC British Broadcasting Corporation

BI Behavioral Intention

CHI Calinski-Harabasz Index

CLV Customer Lifetime Value

CRM Customer Relationship Management

CRISP-DM Cross-Industry Standard Process for Data Mining

CSV Comma Separated Values

DBI Davies-Bouldin Index

DBSCAN Density Based Spatial Clustering of Application with Noise

EDA Exploratory Data Analysis

ERP Enterprise Resource Planning

IQR Interquartile Range

K Optimal number of Clusters

LIME Local Interpretable Model-Agnostic Explanations

ML Machine Learning

MSE Mean Square Error

NCF Neural Collaborative Filtering

NDCG Normalized Discounted Cumulative Gain

OCR Optical Character Recognition

Q1 25th Percentile

Q3 75th Percentile

REST Representational State Transfer

ReLU Rectified Linear Unit

RFM Recency Frequency Monetary

RL Reinforcement Learning

SHAP SHapley Additive exPlanations

SMEs Small and Medium Sized Enterprises

SN Subjective Norm

SVD Singular Value Decomposition

TRA Theory of Reasoned Action

UI User Interface

UNSDGs United Nations Sustainable Development Goals

WSS Within-cluster Sum of Squares

W's Weights

XAI Explainable AI

RL Reinforcement Learning

Acknowledgments

I am sincerely thankful to all those who have supported and guided me throughout the journey of completing this dissertation. Above all, I thank the Almighty God for His endless grace, strength, and guidance that have seen me through every stage of this academic endeavor.

I extend my profound appreciation to my supervisor, Dr. John Olukuru, for his unwavering support, insightful feedback, and valuable mentorship that has significantly influenced the direction and quality of this research. His encouragement and expertise have been instrumental in shaping my academic growth.

I am also grateful to my fellow classmates for their constant encouragement, collaboration, and companionship throughout the course. Special thanks go to the university's administrative and academic staff, as well as the wider Strathmore University community, for providing the necessary resources and creating an environment conducive to learning and research.

Lastly, I acknowledge the work of the scholars and researchers whose literature formed the foundation of my study. Their dedication to advancing knowledge has been a vital source of inspiration and insight throughout this research process.

Chapter 1: Introduction

1.1 Background

Businesses operate in markets that are characterized with large volumes of data relating to consumer behavior as well as occurrences in the market, which affect operations and competitiveness (Rosário and Dias, 2023). Based on this realization, and the desire to understand the consumer and market trends, organizations have moved to adopt AI and Machine Learning (ML) technologies to support collection and analysis of these critical data supporting relevant decision-making processes. Machine learning is a subset of artificial intelligence that uses algorithms to learn insights automatically and recognize patterns from data, such learning is applied in making sound decisions (Henke and Jacques Bughin, 2016). The adoption of Machine Learning and Artificial Intelligence in business intelligence has brought with it significant trends and opportunities, which businesses worldwide are exploring for competitive advantage (Marr, 2019). Today's advanced technologies have completely changed the way organizations make sense of data, uncover insights, and make smarter decisions. A major game-changer has been predictive analytics, where machine learning digs through past data to spot patterns and trends that help forecast what might happen next (Bharadiya, 2023). This outcome enables a business to make concrete predictions about potential outcomes and how to respond to market changes. Consequently, businesses have the chance to enhance their operations, anticipate the needs of customers and devise risk mitigation measures, among others.

1.1.1 Global Context

All over the world, businesses are turning to machine learning to guide their decisions and shape their strategies. It's quickly becoming a go-to tool for smarter, data-driven choices. For instance, a study published by (Gosselink et al., 2024), showed that there has been a consistent rise in the adoption of AI in the market operations of companies, with the adoption being 20% in 2017 and growing to more than 50% in 2022. Furthermore, a study by British Broadcasting Corporation (BBC) research in 2020 forecasted the growth of machine learning enabled solutions to increase at an annual rate of 43.6%, and it was predicted that the value would surpass $8 billion by 2022 (Ma and Sun, 2020). Observation across industries reveal that machine learning has transformed the retail

and e-commerce industries, allowing businesses to analyze large datasets and provide highly personalized shopping experiences. Companies such as Amazon, Alibaba, and Netflix have leveraged machine learning models, such as collaborative filtering and neural networks, to develop recommendation systems that drive their cross-selling strategies (Chen and Li, 2023). By using machine learning to predict customer preferences, these companies have seen significant improvements in customer satisfaction and a substantial increase in sales. According to (McKinsey & Company, 2022), businesses using machine learning for cross-selling report a 20-30% increase in revenue due to better targeting and personalized product recommendations. Major drivers of the adoption of AI and ML include increasing accessibility of cloud computing platforms and the need for automation and personalization. Industries such as retail, healthcare and finance are leading in the adoption of ML by predicting customer behaviour, improving risk management and optimizing operations (Bertolini et al., 2021). But even with all the progress, businesses still face a few bumps in the road, like worries about data privacy, not having enough skilled people to manage these systems, and the headache of getting different technologies to work smoothly together. Emerging trends such as AutoML and explainableAI are improving the accessibility of ML making it a crucial tool for driving competitiveness and innovation in future (Chui et al., 2023).

1.1.2 Regional Context

In Africa, the adoption of ML and AI is gaining momentum as evidenced by various studies across the continent. A study by (Selim and Rezk, 2023) on the use of Machine Learning in Egypt by organizations in different contexts found out that Machine Learning has found its space in learner-ship, visual recognition, and Optical Character Recognition (OCR). This study specifically looked into how machine learning algorithms, particularly the Logistic Classifier can be used to predict school dropout rates. Furthermore, (Elnashar et al., 2020) pointed out in their study that government transport management organizations in Egypt have adopted deep learning, an aspect of artificial intelligence, in automatic license plate detection and recognition for the Egyptian number plates. In countries such as Nigeria, Machine Learning has been applied in the areas of medicine, security, and climatic challenges. For instance, machine learning has been used to generate data that has been used to investigate potential solution to the classifica-

tion problem of breast cancer (Oyelade and Ezugwu, 2022). In South Africa, Machine Learning has been used in studying and analyzing development programs (Van Biljon, 2022). The uptake of machine learning in Africa has also been manifested in the retail industry where its application is gaining traction, particularly as more consumers move toward digital platforms for shopping. A study by (Adeola and Eze, 2023) established that countries such as South Africa, Nigeria, Ghana, and Kenya are experiencing growth in e-commerce, driven by increased mobile internet usage and innovative payment solutions. Leading African e-commerce platforms, such as Jumia and Konga, are beginning to explore machine learning to enhance customer engagement and cross-selling strategies (Brobbey et al., 2021); (Nkwo et al., 2018). Although the region faces infrastructural and data-related challenges, the increasing availability of AI-driven tools offers significant opportunities for growth. Businesses are starting to harness these technologies to improve cross-selling and upselling, especially in the fast-growing middle-class market (Brown, 2021).

1.1.3 Local Context

In Kenya, machine learning is also making its mark across different parts of the economy, showing up in a variety of sectors and use cases. For instance, in his analysis of the adoption and use of Machine Learning in various African countries, (Ezugwu et al., 2023) highlighted that Kenya fundamentally uses Machine Learning in the healthcare sector to support the interaction between patients and healthcare providers, especially in the detection of blinding eye disorders. In business context, Kenya has emerged as one of Africa's leaders in digital transformation with its high mobile penetration and widely adopted mobile payment systems, such as M-PESA (Njenga et al., 2021). Small and Medium Sized Enterprises (SMEs), are increasingly turning to machine learning to enhance their marketing strategies, including cross-selling (Mwangi, 2023). Businesses are tapping into machine learning to better understand their customers by analyzing data like buying habits, such as transaction histories and product preferences, is helping businesses deliver more targeted and personalized shopping experiences. As Kenya's digital economy expands, local businesses are becoming more aware of the potential benefits of machine learning to optimize cross-selling, attract repeat customers, and increase revenue (Njuguna, 2022). Even though machine learning clearly has the power to boost cross-selling,

many businesses whether global, regional, or local still struggle with obstacles that make it hard to fully put these solutions into practice. These include the need for high-quality data, limited access to advanced ML tools, the complexity of integrating machine learning systems, and concerns over data privacy (Ngugi and Muli, 2022). Additionally, smaller businesses may struggle with the costs of adopting these technologies or lack the technical expertise to implement them effectively. That said, these challenges can actually pave the way for growth. When businesses adopt machine learning, they're better equipped to face these issues and discover new ways to improve. By using machine learning to understand customer behavior and fine-tune their cross-selling strategies, companies can boost customer engagement, drive up sales, and stay ahead in today's fast-moving digital world (McKinsey & Company, 2022).

1.2 Problem Statement

Cross-selling is quickly becoming an essential part of modern marketing strategies. It helps businesses suggest related products to customers by looking at their buying habits, making the shopping experience more personalized and boosting sales at the same time. This approach enables companies to attract potential customers by bundling relevant product offers together (Kamakura, 2014). However, in the retail sector, businesses often face significant challenges with inventory management, particularly when it comes to clearing slow-moving stock. A key issue lies in the mismatch between consumer demand and available stock, making it difficult for retailers to predict which products will appeal to customers at a given time. This leads to overstocking, tying up capital, and inefficient use of storage space (Ibrahima et al., 2021).

Traditionally, product recommendations have been based on business owners' intuition and past experiences, which results to biases, inconsistencies, and missed opportunities for promoting slow-moving inventory (Bauer et al., 2022). For cross-selling to really work, businesses need to be able to understand and make the most of their customer data. It is all about using those insights to offer the right products at the right time. Despite businesses across various sectors holding vast amounts of customer data, many struggle with outdated databases and siloed information systems, limiting their ability to share and utilize data effectively for cross-selling (Boustani et al., 2024). Although there has been some progress in exploring how machine learning can be used for cross-

selling recommendations, there is still not much research on how it is actually being put into practice especially when it comes to improving inventory management and boosting customer engagement.

1.3 Research Aim

This research is focused on building a personalized recommendation model using ML to enhance cross-selling in retail. Analyzing customer purchase history, demographics, and behavior, the model will improve product recommendations, identify suitable pairings, clear slow-moving stock, and boost both customer satisfaction and sales performance.

1.4 Research Objectives

(i) To identify key consumer behavior patterns by applying data mining techniques to historical customer data.

(ii) To assess the performance and effectiveness of various machine learning models in identifying cross-selling opportunities.

(iii) To develop a personalized cross-selling recommendation system using machine learning techniques tailored to different customer segments.

(iv) To test, evaluate, and deploy the machine learning-driven recommendation system.

1.5 Research Questions

(i) How can consumer behavior patterns be identified and analyzed using machine learning and data mining techniques to enhance cross-selling opportunities?

(ii) Which machine learning models are most effective in developing accurate and personalized cross-selling recommendations?

(iii) How can machine learning-based customer segmentation be implemented to tailor cross-selling strategies for different customer groups?

(iv) What impact do machine learning-based cross-selling recommendations have on inventory management, customer engagement, and sales conversion rates?

1.6 Scope of the Study

This research focuses on using machine learning techniques to improve cross-selling recommendations in the retail space, based on real customer data from a retail business. Key areas to be covered shall be collecting and pre-processing consumer data for machine learning model development. Algorithms shall be developed to generate personalized recommendations that will improve cross-selling recommendation focusing on clearing excess and slow-moving stocks. In addition, the study will evaluate the performance of the machine learning model using metrics such as precision and recall to ensure its effectiveness in retail settings. It will also assess the business impact, aiming to optimize inventory management, increase average order value, and improve customer satisfaction. By examining the role of data-driven cross-selling strategies, the study seeks to address critical challenges in retail inventory management and sales optimization.

1.7 Limitations of the Study

This research comes with a few limitations. One key limitation is that it uses data from a single retail business, which could make it harder to apply the findings to other retailers with different types of customers or product lines. Another important factor is that the effectiveness of the machine learning model depends heavily on the quality and richness of the customer data. If the dataset is too narrow or lacks important details, it could reduce the model's accuracy and overall performance.

Another challenge is that the study depends on historical data, which might not reflect changing customer preferences or current market trends as they happen. Moreover, while the study aims to optimize cross-selling strategies, external factors such as changes in consumer behavior, economic conditions, or competition in the retail market may influence the outcomes, limiting the long-term applicability of the model. Lastly, due to resource constraints, this research may not fully explore all machine learning algorithms or optimization techniques, focusing instead on a selected few, which could limit the comprehensiveness of the results.

1.8 Research Justification

As researchers, we believe that adopting and effectively applying machine learning will enable businesses to better understand consumer preferences and needs, allowing them to recommend relevant products as alternatives to regular purchases. This will lead to a more efficient flow of products in the market, helping businesses address inventory challenges and improve sales through cross-selling. By conducting this study, marketers are informed about the relevance of machine learning in capturing consumer trends and preferences, enabling organizations to respond proactively through personalized cross-selling, thereby enhancing competitiveness and increasing revenue.

This research contributes to advancing economic sustainability and fostering innovation in business operations (Edison et al., 2024). It supports the United Nations Sustainable Development Goals (UNSDGs), especially Goal 8, which focuses on promoting steady, inclusive, and sustainable economic growth, along with creating productive jobs and ensuring decent work for everyone. By enhancing cross-selling strategies through machine learning, the study aims to help businesses optimize inventory management and sales processes, thus contributing to greater economic productivity and more efficient resource utilization. This supports the development of sustainable retail models that minimize waste and improve profitability, ensuring long-term economic stability.

Chapter 2: Literature Review

2.1 Introduction

This chapter will explore the theories, past studies, existing research gaps, and the conceptual framework related to how machine learning techniques are used in cross-selling.

2.2 Theoretical Review

2.2.1 Cross-selling

(Kamakura, 2014) defines cross selling as an old and valuable strategy used by sales persons to improve the size of order and transform potential buyers of single products to buyers of multi-products. Closely related with the concept of cross-selling is cross buying, which (Shah et al., 2012) define as the behavior of buying additional products or services from a single company. As such, the strategies adopted by a company, in achieving cross-selling, plays an essential role in achieving the cross-buying behavior among potential consumers. When customers purchase more than one product or service from a single company, the duration of relationship between the firm and the customer is extended, the frequency of purchase increases, as well as the margin per order also increases (Bauer et al., 2022). This directly or indirectly results in an increase in customer profitability. Several theoretical models, such as those from behavioral economics, marketing theories, and statistical learning models, provide the basis of consumer behavior analysis in cross-selling. These models try to explain how consumers choose products and how cross-selling tactics might be improved more effective by machine learning.

2.2.2 Consumer Behaviour Theories in Cross-Selling

(a) **The Theory of Reasoned Action (TRA)**

The Theory of Reasoned Action (TRA) was coined by Martin Fishbone and Ice Janzen in 1975 (Hale et al., 2002). According to the theory, intention or instrumentality (which is the belief that behavior will result into a desired result) is the most fundamental predictor of behavior. Instrumentality is influenced by three aspects: their attitude in relation to the specific behavior, the existing subjective norms, and their perceived behavioral control (Hale et al., 2002). When the attitude and the subjective norms are favorable, and when there is a perceived higher levels of con-

trol, an individual is highly likely to have a stronger intention to execute that given behavior. In consumer behavior, as demonstrated by (VigneshKarthik, 2017), the intention to purchase a product by a consumer is an action influenced by the individual intention as well as the various subjective norms that exist in the mindset of the consumer. The theory hold that the behavior of an individual is influenced by their intention to execute the behavior and that this intention is, on the other hand, a function of their attitude in relation to the behavior and subjective norms (Roy, 2022). The theory is represented by the formula as demonstrated by (Alsughayir and Albarq, 2013):

$$BI = AB \cdot W_1 + SN \cdot W_2$$

In the formula, Behavioral Intention (BI) refers to the intention to perform a behavior, Attitude towards the Behaviour (AB) denotes one's attitude towards the behavior, and Subjective Norm (SN) represents the individual's perception of social pressure. The Weights (W's) correspond to the importance of each factor. Machine learning models can use past customer purchase data, social media interactions, and website behavior to measure these variables and predict cross-sell opportunities.

(b) **Consumer Decision making process**

The consumer decision-making process framework is a structured approach that outlines the stages consumers go through when making purchasing decisions (Panwar et al., 2019). It all starts with need recognition, when consumers notice a difference between where they are and where they want to be, which makes them realize they need a product or service to bridge that gap. Marketers can stimulate this awareness through targeted advertising (Suomala, 2020). Following this, consumers engage in an Information Search to explore potential solutions, utilizing both internal sources (personal experiences) and external sources (recommendations and reviews). It is crucial for marketers to ensure their products are prominently featured during this phase to capture consumer attention (Stankevich, 2017). Consumers progress from gathering information to evaluating alternatives by comparing attributes like price, quality, and brand reputation (I., 2023). Emotional ties and effective marketing greatly impact this stage. Next, consumers decide whether or not to make a purchase, often influenced by things like timing and convenience. The journey wraps

up with post-purchase behavior, where they reflect on their experience and assess how satisfied they are with what they bought(Suomala, 2020). Positive experiences foster brand loyalty and recommendations, while negative ones can discourage future purchases. Marketers can improve this phase with follow-up communication and stellar customer service, shaping a positive decision-making journey.

See the decision-making framework in 2.1 by (Stankevich, 2017) below.

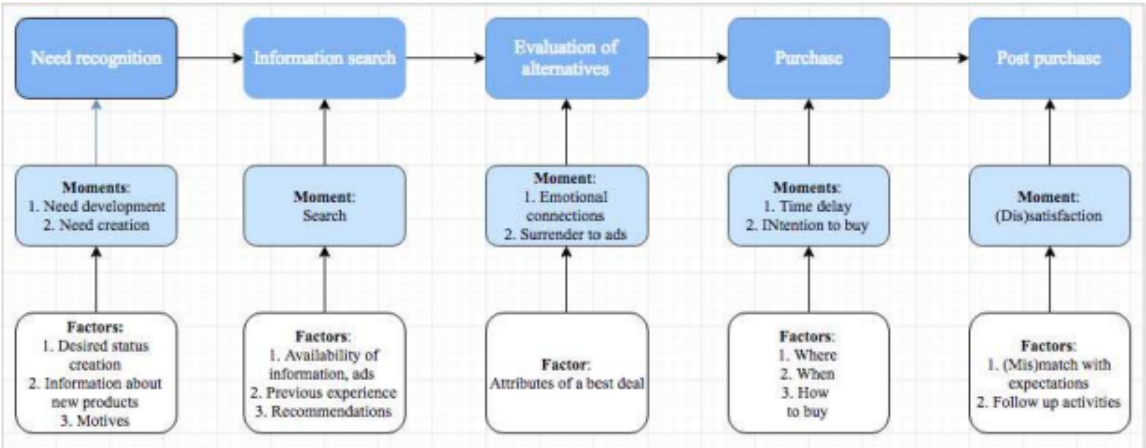


Figure 2.1: A framework of factors and moments that influence decision making

(c) **The RFM (Recency, Frequency and Monetary Model)**

The Recency Frequency Monetary (RFM) model, introduced by (Fader et al., 2005), is a popular marketing tool used to group customers based on their buying habits. It looks at three key things: how recently someone made a purchase (recency), how often they buy (frequency), and how much they spend (monetary value). By analyzing these factors, businesses can pinpoint their most valuable customersthe ones most likely to respond to cross-selling opportunities. Each customer is given a score based on these three areas, helping companies tailor their marketing efforts more effectively.

$$\text{RFM Score} = w_1R + w_2F + w_3M$$

where;

R is the recency score, F is the frequency score, M is the monetary value score

$$w_1,\ w_2,\ w_3$$

are weights that reflect the importance of each factor. This allows businesses to rank customers and tailor personalized cross-sell strategies, such as targeting customers with high recency and frequency scores for timely offers or those with high monetary scores for premium product recommendations (Fader et al., 2005). While the RFM model is simple to implement and provides actionable insights, it has limitations due to its static nature and reliance on only three factors. It does not account for other variables like customer demographics or real-time data, which can be crucial for more dynamic and personalized marketing strategies (Olson and Olson, 2017). Despite these limitations, the RFM model has proven effective in increasing customer engagement and boosting sales in cross-sell campaigns, especially when combined with advanced machine learning techniques to improve segmentation and prediction accuracy (Handojo et al., 2023).

2.3 Empirical Review

Empirical studies on the application of ML in consumer behavior analysis have yielded substantial insights into optimizing cross-selling strategies. These studies focus on how ML techniques were leveraged to identify patterns in customer purchasing behavior, predict future buying tendencies, and enhance personalization. This empirical review presents recent research that explores various machine learning models and their effectiveness in cross-selling within different business contexts from their earliest applications to the most recent.

2.3.1 Early Applications; Decision Trees and Logistic Regression

(Li et al., 2011) were among the first to apply machine learning models, specifically decision trees and logistic regression, to improve cross-selling in retail. Decision trees split data based on customer attributes like age, past purchases, and frequency of visits, helping retailers understand why certain products were recommended. The model achieved 70% accuracy and 65% precision, indicating effective recommendations (Li et al., 2011). However, it struggled with overfitting and degraded performance as the dataset grew, making it less suitable for larger retailers.

$$P(Y \mid X) = \sum_{i=1}^{n} w_i \cdot I(X_i = x)$$

Where:

(i) $P(Y|X)$ is the probability of recommending product Y given the input data X (e.g., customer behavior and attributes).

(ii) w_i represents the weight assigned to feature i.

(iii) $I(X_i = x)$ is an indicator function showing whether feature X_i matches value x.

In the same study, logistic regression was used to predict the likelihood of a purchase based on variables like product category and purchase history. The model had an Area Under the Curve (AUC) score of 0.76 and an F1 score of 0.68, showing moderate effectiveness in balancing precision and recall (Li et al., 2011). However, it assumed linear relationships and could not capture complex interactions, limiting its performance with more nuanced customer data.

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)}}$$

Where:

(i) $P(Y = 1|X)$ is the probability that a customer will buy the recommended product.

(ii) $X_1, \ldots, X_p$ are the independent variables (e.g., purchase history, product category).

(iii) $\beta_0, \ldots, \beta_p$ are the model coefficients.

2.3.2 Collaborative Filtering for Cross-Selling

(Huang et al., 2014) introduced collaborative filtering to cross-selling, an approach that utilized the similarities between users or items to make product recommendations. By analyzing purchase patterns among customers, the model suggested products based on similar behavior among customers. Collaborative filtering built a matrix where users were rows and items were columns. Recommendations were generated based on the similarity between users or between items. Formula (User-Item Matrix):

$$\hat{r}_{ui} = \mu + b_u + b_i + q_u^T p_i$$

Where:

(i) $\hat{r}_{ui}$ is the predicted rating or likelihood that user u will purchase item i.

(ii) μ is the global average rating across all users and items.

(iii) b_u and b_i are the biases for user u and item i, respectively.

(iv) q_u and p_i represent the latent feature vectors for user u and item i, respectively.

The model's Mean Square Error (MSE) was 0.21, indicating that the predicted ratings were close to the actual ratings provided by users. Collaborative filtering achieved a coverage of 85%, meaning it could provide recommendations for most items in the catalogue (Huang et al., 2014). The hit rate (the percentage of recommended products that were actually purchased) was 72%, demonstrating a high success rate for recommendations (Huang et al., 2014). Collaborative filtering struggled with new users or products that lack sufficient historical data, leading to poor recommendations for these cases. Large datasets with many users and items often had sparse matrices (many missing values), which reduced the accuracy of recommendations.

2.3.3 Neural Networks and Deep Learning in Cross-Selling

Neural networks and deep learning were applied by (Zhang et al., 2017) to improve recommendation accuracy by uncovering complex, non-linear patterns in customer behavior. Neural networks are particularly effective in dealing with large datasets with multiple features (Kalkan and Şahin, 2023). They consisted of multiple layers of neurons (nodes) connected by weights. Each layer transformed the data before passing it to the next, allowing the network to capture non-linear relationships in the data (Zhang et al., 2017). Formula (Feedforward Neural Network):

$$y = f\left(W_2 \cdot f\left(W_1 \cdot X + b_1\right) + b_2\right)$$

Where:

(i) X is the input data (customer data such as purchase history or demographics).

(ii) W_1, W_2 are the weight matrices for the first and second layers.

(iii) b_1, b_2 are the bias terms for each layer.

(iv) f is the activation function (commonly Rectified Linear Unit (ReLU) or sigmoid).

The model minimized cross-entropy loss to 0.15, indicating that the predicted probability distributions were close to the actual distributions. The neural network achieved a precision of 82%, showing that the majority of recommended products were relevant to customers (Zhang et al., 2017).

The recall was 78%, demonstrating that the model captured most of the relevant cross-selling opportunities (Zhang et al., 2017). Neural networks required large datasets for training to avoid overfitting or underfitting, which limited their use for smaller retailers. Neural networks often described as black boxes, made it difficult to explain how the model arrived at specific recommendations, which were a barrier in business applications (Kalkan and Şahin, 2023).

2.3.4 Reinforcement learning in Retail Cross-selling

(Boustani et al., 2024) introduced Reinforcement Learning (RL) as a method for real-time cross-selling in retail. The RL model continuously learns from customer interactions and adapts its recommendations based on the rewards such as customer purchases it receives. The Q-Learning algorithm is used to update the value of actions based on their outcomes (Rolf et al., 2023). The system learned to take the action that maximizes the cumulative reward over time.

$$Q(s, a) = r + \gamma \cdot \max_{a'} Q(s', a')$$

Where:

(i) $Q(s, a)$ is the value of taking action a in state s.

(ii) r is the reward received after taking action a.

(iii) γ is the discount factor (between 0 and 1) that determines the importance of future rewards.

(iv) s' is the next state, and a' is the next action.

The RL model maximized cumulative rewards with a score of 0.85, indicating that it effectively recommended products that customers purchased. The model used an 80:20 ratio to balance exploring new product recommendations versus exploiting known suc-

cessful recommendations, which optimized both customer satisfaction and the clearing of slow-moving inventory (Boustani et al., 2024). The model's adaptability was measured at 90%, demonstrating its ability to adjust recommendations quickly in response to changing customer behavior. RL models are computationally expensive, requiring significant resources for training and real-time adaptation. These models take longer to converge compared to other machine learning methods, making them less suitable for retailers without the necessary technical infrastructure (Shafik et al., 2020).

2.4 Research Gaps

2.4.1 Limitations and Trade-Offs in Machine Learning Models

The literature highlights several gaps and limitations in the application of machine learning for cross-selling in retail. Early models such as those by (Kamakura et al., 2004) primarily emphasized traditional marketing strategies, lacking the data-driven personalization required in modern commerce. Empirical studies have since explored more advanced algorithms, but challenges remain. For example, (Li et al., 2021) noted that decision trees, while interpretable, often struggle with scalability and overfitting. Logistic regression offers transparency but assumes linearity, which may not adequately capture complex consumer behavior patterns (Bajari et al., 2015).

Further limitations arise with collaborative filtering and deep learning approaches. (Huang et al., 2014) found collaborative filtering to be sensitive to the cold start problem and data sparsity. Meanwhile, neural networks, despite their high accuracy, function as black boxes, limiting interpretability in contexts requiring explainability (Zhang et al., 2017). Reinforcement learning adds adaptability but can be computationally intensive and slow to converge, making it less practical for small or resource-constrained businesses (Boustani et al., 2024).

Overall, each modeling approach presents trade-offs between accuracy, scalability, and interpretability. While linear models may be more transparent and stakeholder-friendly, advanced models such as Singular Value Decomposition (SVD) and Neural Collaborative Filtering (NCF) perform better on sparse or nonlinear data but lack explainability. This calls for thoughtful model selection based on operational needs and the regulatory or business environment in which the system is deployed.

2.4.2 Explainable AI in Recommendation Systems

Given the rise of data-driven personalization in commercial platforms, there is growing emphasis on Explainable AI (XAI) techniques in recommendation system design. Tools like SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) and Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro et al., 2016) have emerged as effective post-hoc methods to interpret model predictions. SHAP uses game theory to assign each feature a contribution score for a given prediction, while LIME approximates the model locally using an interpretable surrogate to explain individual recommendations.

These tools are particularly relevant when deploying models that affect consumer decisions, pricing strategies, or promotions where explainability enhances transparency, trust, and compliance. Integrating explainability frameworks also supports business stakeholders in understanding why specific items are recommended to particular customer segments, facilitating better alignment with marketing strategies.

2.4.3 Best Practices in Emerging Literature

Recent research emphasizes hybrid approaches that balance predictive performance and interpretability. For example, some systems incorporate rule-based logic or decision trees in early filtering stages, followed by neural rankers. Others use attention mechanisms or feature attribution maps to highlight influential features in deep models (Zhang and Chen, 2020).

Overall, the integration of explainability tools and thoughtful model selection can help organizations deploy robust, transparent, and trustworthy recommendation engines, especially when targeting commercial or regulated use cases.

2.5 Conceptual Framework

From the conceptual framework in the figure 2.2 below, it is demonstrated that machine learning supports the collection and processing of consumer data, consumer segmentation in the prediction of consumer needs and preferences, which supports the cross-selling recommendation.

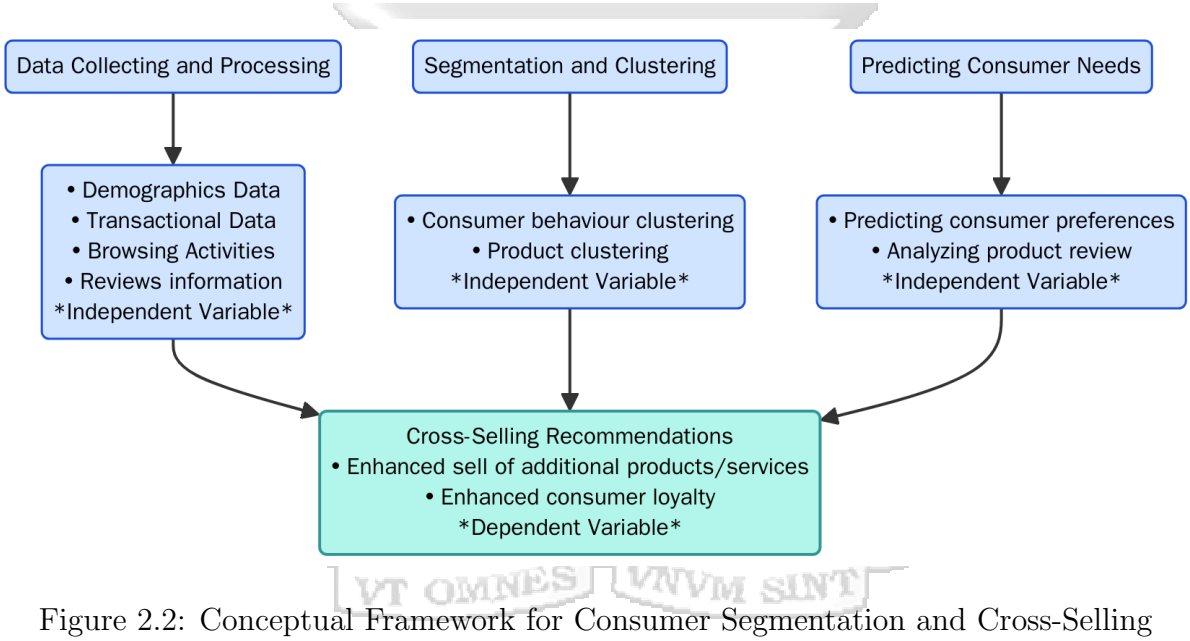


Figure 2.2: Conceptual Framework for Consumer Segmentation and Cross-Selling

Chapter 3: Methodology

This chapter outlines the approach taken to model consumer behavior using machine learning, aimed at driving more personalized and effective cross-selling.

3.1 System Design

This study follows the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology, a well known and trusted framework for guiding data mining projects. CRISP-DM offers a clear, structured process for tackling business challenges through data-driven solutions. It helps ensure that every step from grasping the problem to rolling out the final model is carried out in a logical and organized way. The approach is broken down into six key stages: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.

The first phase, Business Understanding, is all about getting a clear grasp of what the project is trying to achieve from a business point of view. It's where the problem is clearly defined and then reframed into something that machine learning can help solve. Next comes the Data Understanding phase, which focuses on gathering and exploring the available data. During this stage, the relevant datasets are collected and closely examined to understand their structure, check for quality issues, and spot anything that might impact later steps. This phase is key to building a solid foundation and uncovering early insights from the data.

The third phase, Data Preparation, focuses on organizing the data so it's ready for analysis. This includes cleaning the data, filling in or removing missing values, and converting it into a format that works well for modeling. The quality of this step is key, as it directly influences how accurate and effective the final models will be. Following that is the Modeling phase. In this stage, machine learning algorithms are applied to the prepared data to build models aimed at solving the identified problem. Different techniques are explored, and adjustments are made to fine-tune the models and find the best approach for the data.

After the models are developed, the next step is Evaluation. This phase checks how well the model performs based on set goals, making sure it actually solves the problem it was

designed for. Once the model passes this test, it moves into the Deployment phase. This is where the model is put to work in a real business setting, turning insights into action. Deployment ensures the research goals are achieved and that the insights gathered are used to support smarter, data-driven business decisions

The CRISP-DM methodology's cyclical nature allows for iterative improvements, enabling the researcher to refine the models continuously based on new insights or data. This model, as illustrated in figure 3.1 below, provides a structured path for tackling data mining tasks and ensures that the project delivers meaningful results for the business.

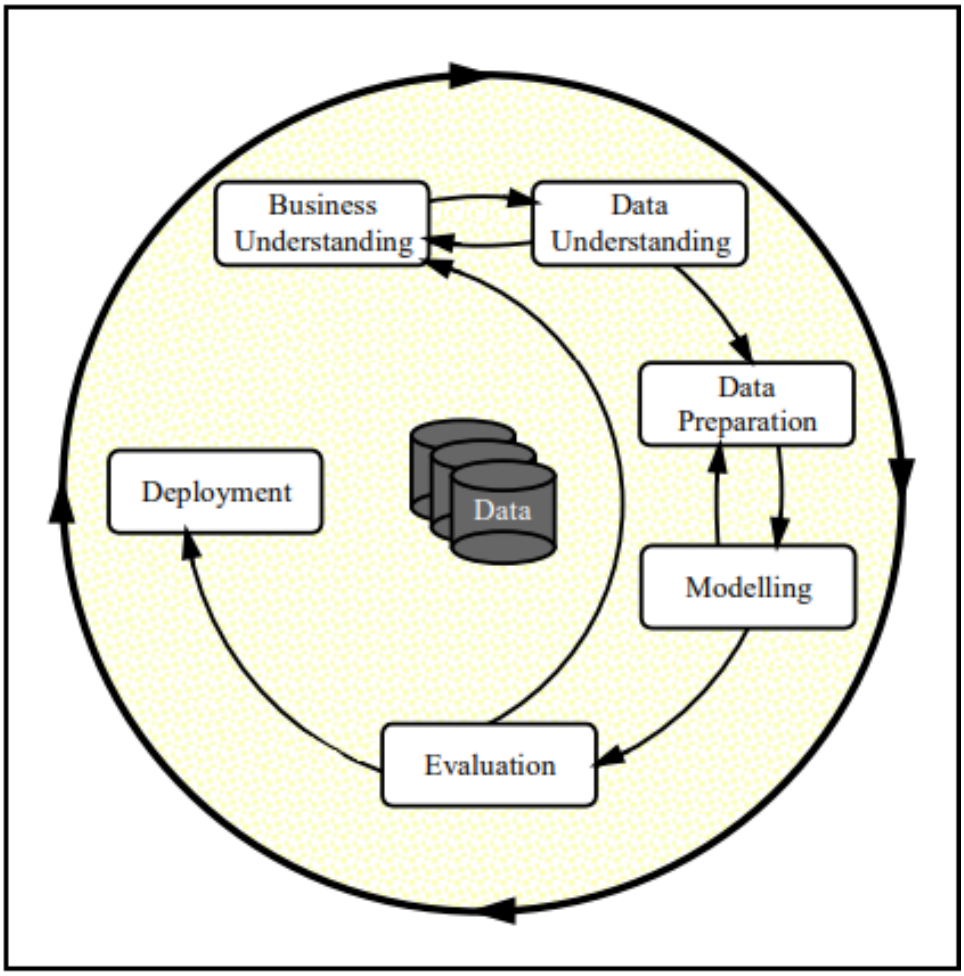


Figure 3.1: CRISP-DM Model for Data Mining (Wirth and Hipp, 2000)

3.2 Business Understanding

This study seeks to optimize cross-selling strategies in retail using machine learning, with a particular emphasis on overcoming inventory management challenges, especially with slow-moving stock. Traditional recommendation models based on intuition and past ex-

periences fail to tackle issues like overstocking and mismatched demand. Additionally, research gaps such as scalability problems with decision trees, the cold-start issue in collaborative filtering, and integration challenges with legacy systems highlight the need for a data-driven approach that enhances customer engagement while optimizing inventory management.

3.3 Data Sources and Collection Methods

In this study, data was sourced from a retail business specializing in gym accessories, clothing, and shoes. The data, which includes customer transactions, product details, and sales information, was directly downloaded from the company's accounting system in CSV format. The dataset contains key business metrics such as customer demographics, product categories, transaction details, and sales amounts.

The dataset includes important information such as the customer's gender, product descriptions, dates of purchase, item sizes, colors, quantities sold, and total amounts paid. It also captures the type of sale (e.g., online or walk-in) and the location where the sale occurred. Table 3.1 summarizes the main data sources used in the study, while Table 3.2 provides detailed information on the variables present in the customer transactions dataset.

The downloaded data was then preprocessed using Python libraries like Pandas for data cleaning, transformation, and analysis. The information relevant to the research was contained within the CSV files, which were structured for further analysis to generate meaningful insights for product recommendation and cross-selling.

Table 3.1: Breakdown of the Data Sources Used for the Research

Dataset Name	**Rows**	**Columns**	**Description**
Customer Transactions	5,910	13	Data related to customer purchases and transactions
Sales Data	5,910	13	Detailed sales information including product purchases

Table 3.2: Variables in the Customer Transactions Dataset

Variable Name	**Description**	**Data Type**
Gender	Gender of the customer	Categorical
Customer_Name	Name of the customer	Text/String
Category	Product category	Categorical
Product_Description	Description of the product	Text/String
Date	Date of purchase	Date/Time
Colour	Product color	Categorical
Size	Product size	Categorical
Quantity_Sold	Number of items sold	Integer
Sub_Total	Subtotal of the purchase	Float
Shipping_Fee	Shipping fee charged	Float
Total_Amount	Total amount paid	Float
Bill_To_City	City where the order was billed	Categorical
Type_Of_Sale	Sale type (e.g., online, walk-in)	Categorical

The data provided valuable insights into customer purchase patterns, sales trends, and product preferences. By consolidating this information, the study aimed to develop a machine learning–based recommendation system to optimize cross-selling and improve the business's sales strategies.

3.4 Data Pre-processing

The data pre-processing phase is crucial in ensuring the accuracy, consistency, and reliability of the data before it is used for analysis and model development. This step involves various techniques, including data cleaning, inspection, transformation, and conversion. By performing these operations, the dataset is prepared for the modeling process and is aligned with the requirements of the analytical methods used in this study.

3.4.1 Cleaning the Data

Data cleaning was the first step in the pre-processing pipeline and was performed using Python's `pandas` library. The dataset was inspected for duplicate entries using the `data.duplicated().sum()` function, which revealed 82 duplicate records. These were

removed using `data.drop_duplicates()`, reducing the dataset size from 5,910 to 5,828 records. This step ensured that each transaction was unique and prevented model distortion due to repeated entries.

3.4.2 Data Inspection and Exploration

Data inspection and exploration were carried out to understand the structure, attributes, and underlying patterns in the dataset. This step involves examining summary statistics, such as mean, median, and standard deviation, to identify any potential anomalies or irregularities. The purpose was to identify any entry errors or potential sources of noise in the dataset. Visual inspection using plots like histograms and box plots was employed to detect these anomalies.

3.4.3 Treating Missing Data

Upon inspection using `pandas` functions such as `data.isnull().sum()` and `data.info()`, no missing values were identified in the dataset. This completeness was essential for maintaining the integrity of subsequent analysis and modeling steps.

Had missing values been detected, appropriate techniques such as listwise deletion (`data.dropna()`), forward-fill imputation (`data.fillna(method='ffill')`), or mean substitution (`data['column'].fillna(data['column'].mean())`) would have been applied, depending on the variable type and business context. Since the dataset was fully intact, no imputation or deletion was required.

3.4.4 Data Type Conversion

Data type conversion is essential for ensuring consistency and compatibility with analytical methods. In this study, the `pandas` library was used to convert the `'Date'` column, which was initially stored in string format, into a proper `datetime64[ns]` format. This transformation enabled time-based operations such as calculating recency and analyzing trends.

The conversion was achieved using the function `pd.to_datetime(data['Date'])` from the `pandas` library, as recommended by (McKinney, 2010). This ensured accurate parsing of dates and compatibility with time-series and customer behavior analysis.

3.4.5 Outlier Detection and Treatment

Outlier detection was conducted to identify extreme values that could distort model results. The study used box plot, generated via `matplotlib.pyplot.boxplot()`, for visual inspection of numerical features. Outliers were flagged using the Interquartile Range (IQR) method from descriptive statistics computed with `pandas` functions such as `data[column].quantile()`.

Observations below `25th Percentile (Q1) - 1.5 * IQR` or above `75th Percentile (Q3) + 1.5 * IQR` were treated using the winsorization technique from the `scipy.stats.mstats` module, specifically `winsorize()`, which caps extreme values to limit their influence while retaining the structure of the dataset (Khan et al., 2020).

This ensured that outliers did not skew the results, preserving the integrity and interpretability of the machine learning models.

3.5 Feature Engineering

Feature engineering enhances a machine learning model's ability to detect patterns and make accurate predictions. In this study, key behavioral metrics were developed using Python's `pandas` and `numpy` libraries to support customer segmentation and recommendation modeling.

The engineered features include:

(i) **Total Spending:** Calculated using `groupby()` and `sum()` functions as the total amount spent per customer. This identified high-value customers contributing significantly to revenue.

(ii) **Purchase Frequency:** Computed using `nunique()` and date differences between first and last purchases to measure how often each customer buys within a given time frame.

(iii) **Recency:** Measured as the number of days since a customer's last transaction using `(current_date - last_purchase_date).dt.days`, helping to identify recently active vs. lapsed customers.

(iv) **Customer Lifetime Value (CLV):** Estimated using the formula

$$\text{CLV} = \text{Average Purchase Value} \times \text{Purchase Frequency} \times \text{Customer Lifespan}$$

where each metric was derived per customer using `mean()`, `count()`, and transaction time windows.

These features were consolidated into a new DataFrame and standardized in preparation for clustering. They served as a foundation for understanding customer behavior and generating personalized recommendations.

3.6 Data Transformation

The data transformation phase ensured the dataset was ready for clustering and recommendation modeling. All engineered features were combined into a new DataFrame using `pandas.concat()` for unified processing. Since all features were numerical, no categorical encoding was necessary.

3.6.1 Feature Scaling

To ensure uniform contribution of features and improve algorithm performance, feature scaling was applied using the `StandardScaler()` from the `scikit-learn.preprocessing` module. This scaler standardizes values to have a mean of zero and a standard deviation of one.

The transformation was performed as follows:

(i) `from sklearn.preprocessing import StandardScaler`

(ii) `scaler = StandardScaler()`

(iii) `scaled_data = scaler.fit_transform(df_features)`

The `Customer_ID` column was excluded from scaling, as it served as an identifier and not a feature for clustering. The standardization formula used is:

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$

where:

(i) X is the original value,

(ii) μ is the mean of the feature, and

(iii) σ is the standard deviation.

This step ensured all features contributed equally to distance-based algorithms such as K-means.

3.7 Machine Learning Modeling

The machine learning modeling phase involved several steps, starting with identifying the optimal number of clusters and progressing through clustering and recommendation modeling techniques. The primary goal was to segment the customers and recommend relevant products, focusing on slow-moving products. The implementation was done using Python libraries such as `pandas`, `numpy`, `scikit-learn`, `surprise`, and `matplotlib`.

3.7.1 Identifying Optimal Clusters Using Elbow Method

The first step in the clustering process was determining the optimal number of clusters (Optimal number of Clusters (K)). The Elbow Method was employed by plotting the Within-cluster Sum of Squares (WSS) against different values of K using the `KMeans` function from the `sklearn.cluster` module. The "elbow" point was visually identified where the WSS started to decrease at a slower rate.

The formula for calculating WSS is:

$$WSS(K) = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2$$

where:

(i) K is the number of clusters,

(ii) C_i is the set of points in cluster i,

(iii) μ_i is the centroid of cluster i, and

(iv) x is the data point (Zhang et al., 2020).

3.7.2 Clustering Algorithms

Once the optimal number of clusters was identified, three clustering algorithms were applied using `scikit-learn`:

(i) **K-means Clustering**

Implemented using `sklearn.cluster.KMeans`. It partitions the data into K clusters, where each data point is assigned to the nearest cluster centroid. The centroids are updated iteratively until convergence (Lee and Kim, 2021).

(ii) **Agglomerative Clustering**

Implemented with `sklearn.cluster.AgglomerativeClustering`. This is a bottom-up hierarchical approach that begins with each data point as its own cluster and merges the closest pairs based on linkage criteria until the desired number of clusters is achieved (Xu and Wunsch, 2021).

(iii) **Density Based Spatial Clustering of Application with Noise (DBSCAN)**

Implemented using `sklearn.cluster.DBSCAN`. Unlike K-means, DBSCAN does not require the number of clusters to be predefined. It identifies clusters based on density and is effective for data with arbitrary shapes and noise (Xu and Wunsch, 2021).

3.7.3 Clustering Performance Evaluation

To evaluate the clustering models, the following metrics were computed using `sklearn.metrics`:

(i) **Silhouette Score** – `silhouette_score()` measures cohesion and separation by computing the mean intra-cluster distance vs. nearest-cluster distance.

(ii) **Davies-Bouldin Index (DBI)** – `davies_bouldin_score()` calculates the average similarity between each cluster and its most similar one; lower values indicate better clustering.

(iii) **Calinski-Harabasz Index (CHI)** – `calinski_harabasz_score()` evaluates the ratio of between-cluster dispersion to within-cluster dispersion; higher values are better.

3.7.4 Customer Segmentation and Labeling

Based on the cluster characteristics, customers were categorized as:

(i) **High-Value Customers** – Significant contributors to revenue.

(ii) **Frequent Buyers** – Regular purchasers with high frequency.

(iii) **Price-Sensitive Shoppers** – Prefer lower-cost products.

(iv) **Occasional Shoppers** – Make infrequent purchases.

Cluster labels were assigned using logical rules derived from feature averages and saved using `pandas.to_csv()` for use in the recommendation engine.

3.7.5 Inventory Optimization and Recommendations

A customer-product interaction matrix was created using `pandas.pivot_table()`, with customers as rows and products as columns. `sklearn.metrics.pairwise.cosine_similarity()` was used to compute item-item similarities. Collaborative Filtering was implemented using matrix factorization (via `surprise.SVD()`) to predict missing values in the interaction matrix.

The Cosine Similarity is defined as:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|}$$

where A and B are the product vectors representing user interactions.

3.7.6 Hybrid Recommendation Model

The hybrid model was designed to combine the strengths of Content-Based Filtering and Collaborative Filtering to generate more accurate and robust recommendations, especially in cold-start scenarios.

(i) **Content-Based Filtering:** This approach analyzes item metadata (e.g., product category, price range) and recommends items similar to those a user previously interacted with. Feature similarity was computed using cosine similarity between item vectors, implemented via `sklearn.metrics.pairwise.cosine_similarity()`.

(ii) **Collaborative Filtering:** This method uses user-product interaction data to infer user preferences. The SVD algorithm from the `surprise` library (`surprise.SVD()`) was used to predict ratings, with `Trainset.build_anti_testset()` used to generate unseen user-item pairs for prediction.

(iii) **Hybrid Logic:** Final recommendation scores were calculated using a weighted average of content-based and collaborative predictions:

$$\text{Score}_{\text{hybrid}} = \alpha \cdot \text{Score}_{\text{collab}} + (1 - \alpha) \cdot \text{Score}_{\text{content}}$$

where $\alpha \in [0, 1]$ is a tunable parameter (default = 0.5). In the deployed system, this value could be adjusted interactively via a Streamlit slider, allowing users to control the blending ratio. Final recommendations were also filtered using cluster labels to align with customer segments.

3.7.7 Evaluation of Recommendation Models

Model evaluation was conducted using a top-N recommendation strategy, where the system aimed to recommend the top 5 most relevant products to each user. The evaluation was done independently for the content-based, collaborative filtering, and hybrid models, using a held-out test set.

The following metrics were used:

(i) **Precision:** Computed using a custom function measuring the proportion of recommended items that were relevant to the user.

$$\text{Precision} = \frac{\text{Number of relevant items recommended}}{\text{Total number of recommended items}}$$

(ii) **Recall:** Measures the ability of the model to retrieve all relevant items:

$$\text{Recall} = \frac{\text{Number of relevant items recommended}}{\text{Total number of relevant items}}$$

(iii) **Normalized Discounted Cumulative Gain (NDCG):** Implemented via `sklearn.metrics.ndcg_score()`. It evaluates ranking quality by penalizing relevant items

that appear lower in the recommendation list.

$$NDCG@K = \frac{1}{Z_K} \sum_{i=1}^{K} \frac{rel(i)}{\log_2(i+1)}$$

where $rel(i)$ is the relevance score at rank i, and Z_K is a normalization factor.

Precision, recall, and NDCG were computed per user and averaged across all users to obtain the overall performance scores for each recommendation model. These metrics provided a comprehensive view of both the relevance and ranking quality of the predictions.

3.7.8 Hyperparameter Tuning

To improve model effectiveness, tuning was applied primarily to the collaborative filtering model:

(i) **SVD (Collaborative Filtering):** Parameters such as the number of latent factors (`n_factors`), learning rate (`lr_all`), and regularization (`reg_all`) were optimized using `GridSearchCV` from the `surprise.model_selection` module. The final parameters selected were:

(a) `n_factors = 50`

(b) `lr_all = 0.005`

(c) `reg_all = 0.02`

(ii) **Content-Based Filtering:** Manual tuning was done for parameters such as the number of top similar items to consider and cosine similarity thresholds. These were iteratively refined based on visual inspection of the recommendation outputs and diversity assessments.

(iii) **Hybrid Model:** The weighting parameter α was defaulted to 0.5 during evaluation, giving equal importance to content-based and collaborative scores. However, the user-facing interface allowed real-time adjustment of α, offering personalization flexibility based on user preference or recommendation context.

The use of consistent evaluation metrics and a shared test set ensured that model com-

parisons were fair and robust. These insights informed the final deployment choice and confirmed the hybrid model's advantage in balancing relevance, diversity, and personalization.

3.8 Model Deployment and API Development

3.8.1 Deployment Strategy

In this study, all models are deployed using an API developed with Streamlit. Streamlit was chosen due to its simplicity and powerful capability for quick deployment of machine learning models into interactive web applications.

The decision to deploy the models via API was driven by the need for a versatile and modular solution. By using an API, the implementation of the model is separated from the client-side application, which allows for independent updates and improvements. Additionally, the API supports scalability, as it is designed to efficiently handle multiple simultaneous requests, making it suitable for real-time recommendations in a production environment.

Streamlit offers an easy interface for deploying machine learning models and integrating them with web applications, making it an excellent choice for this recommendation system. With Streamlit, the model's predictions are served as part of an interactive dashboard, allowing users to select model types for example. Collaborative Filtering, Content-Based, or Hybrid and receive real-time recommendations based on their selections.

3.8.2 Streamlit Application Design Principles

The design of the Streamlit application for this project follows key principles to ensure that it is user-friendly, scalable, and responsive:

(i) **Simplicity and User-Friendly Interface**: Streamlit is used to create an intuitive and interactive user interface that allows users to easily interact with the recommender system. The interface focuses on simplicity, enabling users to select recommendation models, adjust weights, and view results without unnecessary complexity.

(ii) **State Management**: Streamlit's built-in state management system ensures that user interactions, such as model selections or weight adjustments, persist across different steps of the recommendation process. This enables seamless user experiences, where the interface remains responsive even as data changes.

(iii) **Efficient Data Handling**: Although Streamlit does not require a full Representational State Transfer (REST) API or microservices architecture, it efficiently handles the data flow between the models and the frontend. Data is processed in real-time, ensuring that the recommendations are updated quickly in response to user inputs, even with complex data processing.

(iv) **Real-Time Interactivity**: Streamlit supports real-time interactions between the user and the recommender system. As users adjust parameters such as the weights for collaborative filtering or content-based filtering, the system recalculates the recommendations instantly, making the process smooth and responsive.

(v) **Scalability for Prototyping**: While Streamlit is not designed for large-scale deployment in a production environment, it offers a lightweight framework suitable for prototyping and testing machine learning models. This allows for quick iteration and feedback on the effectiveness of the recommendation engine.

3.8.3 Streamlit Application Architecture

The architecture of the Streamlit application is based on several key components:

(i) **Streamlit Application**: The application serves as the front-end interface, providing an interactive environment where users can select recommendation models, adjust parameters, and view product recommendations in real-time.

(ii) **Recommendation Models**: The application integrates the trained machine learning models (Collaborative Filtering, Content-Based Filtering, and Hybrid Model). These models generate personalized product recommendations based on user interactions and inputs.

(iii) **Data Handling**: The application interacts with a CSV file for storing and retrieving product and customer interaction data, rather than relying on a traditional database.

(iv) **Model Integration**: The machine learning models are integrated into the Streamlit application by loading the pre-trained models using Python's `joblib` library. This allows for fast model loading and efficient in-memory processing.

3.8.4 Framework and Tools

The application is built using **Streamlit** as the primary framework, selected for its simplicity, interactivity, and real-time processing capabilities. Other tools and libraries used in the application include:

(i) **Pandas**: A Python library used to handle data preprocessing and interaction with the CSV files storing product and customer data.

(ii) **joblib**: This library is used to serialize the trained models into binary files for efficient storage and fast loading within the application.

(iii) **Matplotlib/Seaborn**: These libraries are used for data visualization to show graphs and insights on the UI, such as product category distribution and sales trends.

For data storage, CSV files are used, providing a lightweight and simple method for storing data that can be easily updated or replaced without the need for a complex database management system.

3.8.5 Model Integration

The trained recommendation models are integrated into the Streamlit application by serializing them into a suitable format using Python `joblib`. This allows the models to be saved as binary files, which are loaded into the application when it starts. The models are then used to generate product recommendations based on user inputs in real-time.

3.8.6 Streamlit User Interface

The Streamlit application provides an intuitive, interactive user interface with the following functionalities:

(i) **Model Selection**: Users can select from three recommendation models: Collaborative Filtering, Content-Based Filtering, or Hybrid Model.

(ii) **User Inputs**: Users can input their preferences, such as selecting a customer_ID for collaborative filtering or a product for content-based filtering.

(iii) **Adjustable Weights**: For the hybrid model, users can adjust the weight assigned to collaborative filtering and content-based filtering to fine-tune the recommendations.

(iv) **Real-Time Results**: As users interact with the interface, the system dynamically updates the recommendations based on the selected model and inputs.

3.8.7 Deployment Environment

For deployment, the Streamlit application is hosted on Streamlit Cloud, a cloud-based platform optimized for deploying and sharing Streamlit applications. This platform provides seamless deployment, real-time performance monitoring, and easy integration with GitHub for continuous updates. By using Streamlit Cloud, the recommendation system is made accessible to users through a scalable and cost-effective environment, allowing for easy access and real-time updates to the recommendation models.

3.9 Ethical Considerations

The use of machine learning in recommendation systems raises several ethical considerations that were critically evaluated during this study.

(i) **Algorithmic Bias:** There is a risk that the recommender system may reinforce historical purchasing behaviors and disproportionately favor high-spending or frequent customers, thus marginalizing occasional or price-sensitive shoppers. This could inadvertently skew marketing focus and reduce inclusivity. To mitigate this, customer segmentation was used to ensure recommendations were tailored across various shopper profiles.

(ii) **Data Privacy:** Transactional and behavioral data were used in accordance with ethical guidelines, and no personally identifiable information was stored or processed. All data were anonymized, and customer identifiers were excluded from feature scaling and modeling stages to protect user identity.

(iii) **Over-Personalization:** While personalization improves relevance, it risks limit-

ing users to a narrow set of product choices, reducing product discovery. This was addressed by incorporating a hybrid model that balances personalization (via Collaborative Filtering) and diversity (via Content-Based Filtering) to expose users to related but novel items.

Chapter 4: System Design and Architecture

4.1 Introduction

This chapter outlines the design and structure of a machine learning-powered recommendation system created to support cross-selling in retail. The system uses a well-organized transactional dataset, exported in CSV format, and applies machine learning techniques to predict what customers are likely to buy. It is built as a web-based application that generates and displays personalized product recommendations tailored to each customer.

4.2 System Architecture Overview

The system is built using a modular architecture, organized into five main layers, each playing a specific role in the overall workflow. An overview of these layers and their interactions is illustrated in Figure 4.1.

(i) **Data Source:**

The system starts with a CSV file that holds all the transaction data. This includes details about customer purchases, product information, and historical interactions, serving as the foundation for generating recommendations.

(ii) **Data Processing Layer:**

In this layer, the raw data from the CSV file is cleaned and prepared for analysis. This involves handling missing values, converting data types, and structuring the data so it can be easily used by the machine learning models.

(iii) **Recommendation Models:**

This part of the system runs the core recommendation techniques—Collaborative Filtering, Content-Based Filtering, and a Hybrid Model. These models work together to suggest products based on both customer behavior and product features.

(iv) **Streamlit Web Interface User Interface (UI):**

The front end of the system is built with Streamlit, offering users an interactive dashboard. Here, users can choose their preferred model, enter their preferences or customer IDs, and get personalized recommendations instantly.

(v) **Results Display:**

The final recommendations are shown to the user through the interface, with interactive options that allow them to adjust inputs or explore further suggestions based on their needs.

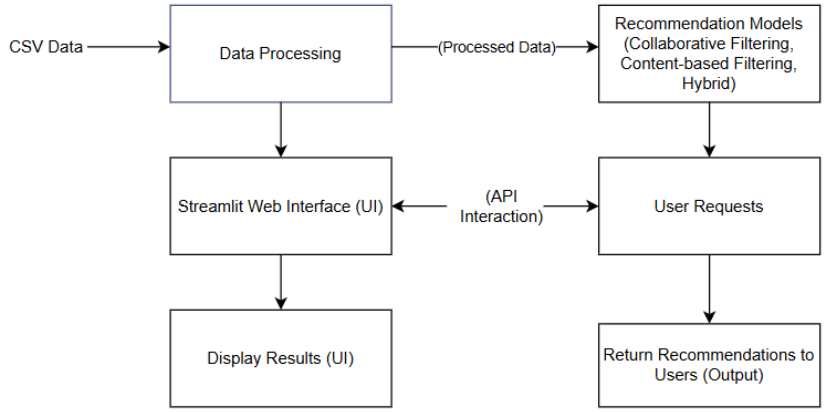


Figure 4.1: System Architecture Diagram

4.3 Data Design

In this study, a formal database was not used. Instead, all the data was stored in a single CSV file, which acted as the main source for both analysis and training the machine learning models. This file included detailed transaction records, with various fields that were logically grouped into pseudo-entities based on the needs of the business and the structure of the dataset.

These pseudo-entities represent different parts of the retail process such as customer information, product details, and transaction history which played a key role in customer segmentation and generating product recommendations. Table 4.1 outlines the main pseudo-entities derived from the dataset along with their associated fields.

Table 4.1: Logical Entities from the CSV File

Pseudo-Entity	**Fields**
Customers	`customer_id`, `gender`, `bill_to_city`, `type_of_sale`
Products	`product_id`, `product_description`, `category`, `colour`, `size`, `price`
Transactions	`transaction_id`, `customer_id`, `product_id`, `date`, `quantity_sold`, `sub_total`, `shipping_fee`, `total_amount`

Organizing the data into these pseudo-entities made it easier to manage and work with. It ensured that important customer and transaction information was readily available for both the customer segmentation process and the recommendation model, helping the system deliver more accurate and meaningful results.

4.4 Data Relationships

Although a formal database system was not used, the CSV file was carefully parsed to build logical relationships that mimic a relational database structure. These connections linking customers, products, transactions, and recommendations were managed in-memory using Python. This approach allowed the system to replicate the kind of relationships commonly seen in traditional relational databases, making the data easier to work with and more meaningful for analysis. Table 4.2 summarizes the key logical relationships in the dataset, while Figure 4.2 visually illustrates these relationships in the form of an entity-relationship diagram.

Table 4.2: Logical Relationships in the Dataset

Relationship	Description
Customer → Transactions	A customer can make multiple purchases, represented by transactions.
Product → Transactions	A product can appear in many transactions, indicating it was purchased by multiple customers.
Customer → Recommendations	Each customer can receive multiple product recommendations based on their purchase history and preferences.
Product → Recommendations	Products can be recommended to multiple customers, especially if they share similar characteristics or behavior.
Segment → Customer	Customers are grouped into segments based on their purchasing behavior and other features.
Segment → Recommendations	Recommendations can be tailored to specific customer segments, providing personalized product suggestions.

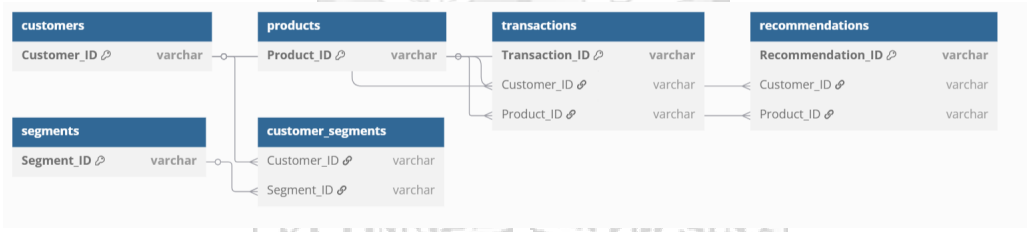


Figure 4.2: Entity-Relationship Diagram

4.5 System Modeling

4.5.1 Class Diagram

The system was designed using modular classes to ensure a scalable and maintainable architecture. Each class encapsulates specific functionality and plays a key role in the implementation of the recommendation system. Figure 4.3 illustrates the structure and relationships among the core system classes described below:

(i) **Customer:** This class stores demographic and historical data of customers, including attributes such as Customer ID, gender, and location. It also tracks the

customer's purchasing history, which is crucial for generating personalized recommendations.

(ii) **Product:** Represents individual products in the inventory. The class includes attributes such as Product ID, product name, category, and price. Products are linked to customer interactions through transactions.

(iii) **Transaction:** The Transaction class links customers and products. It stores transaction details such as transaction ID, customer ID, product ID, quantity purchased, and total amount. This class is essential for tracking product sales and customer preferences.

(iv) **RecommendationEngine:** This class handles the generation of product recommendations using various algorithms (Collaborative Filtering, Content-Based, and Hybrid). It interacts with customer and product data to generate the most relevant recommendations based on customer preferences.

(v) **Segmenter:** The Segmenter class is designed to group customers based on how they shop, using clustering algorithms. It applies unsupervised learning methods to identify meaningful customer segments, which can then be used to deliver more personalized and targeted product recommendations.

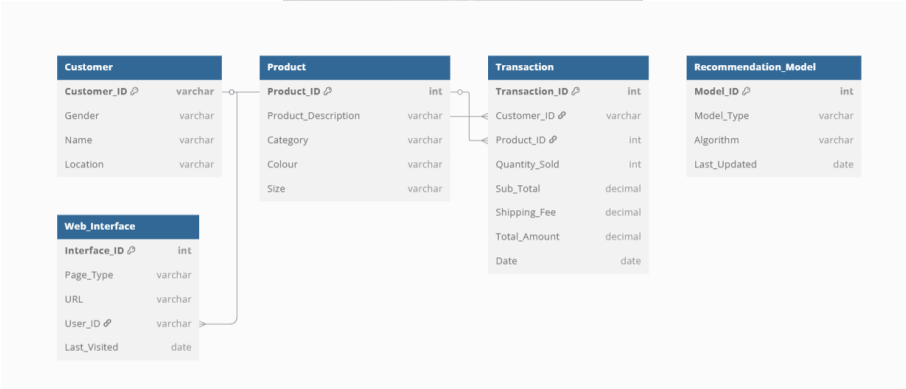


Figure 4.3: Class Diagram

4.5.2 Sequence Diagram

The sequence diagram illustrates how different parts of the system work together to generate product recommendations. Figure 4.4 depicts the sequential interaction between

the user, the Streamlit interface, the backend API, and the machine learning model, outlining the complete recommendation generation workflow:

(i) **User Request**: The process begins when the user interacts with the Streamlit web interface to request product recommendations.

(ii) **Frontend Request**: The frontend sends an API request to the Flask backend, specifying the user and model type (Collaborative Filtering, Content-Based, or Hybrid).

(iii) **Data Filtering**: The backend API retrieves and filters relevant customer and product data from the CSV file.

(iv) **Recommendation Generation**: The filtered data is then passed to the corresponding machine learning model to compute the most relevant product recommendations.

(v) **Return and Display Results**: The API sends the recommendation results back to the frontend, where they are displayed to the user in the Streamlit interface.

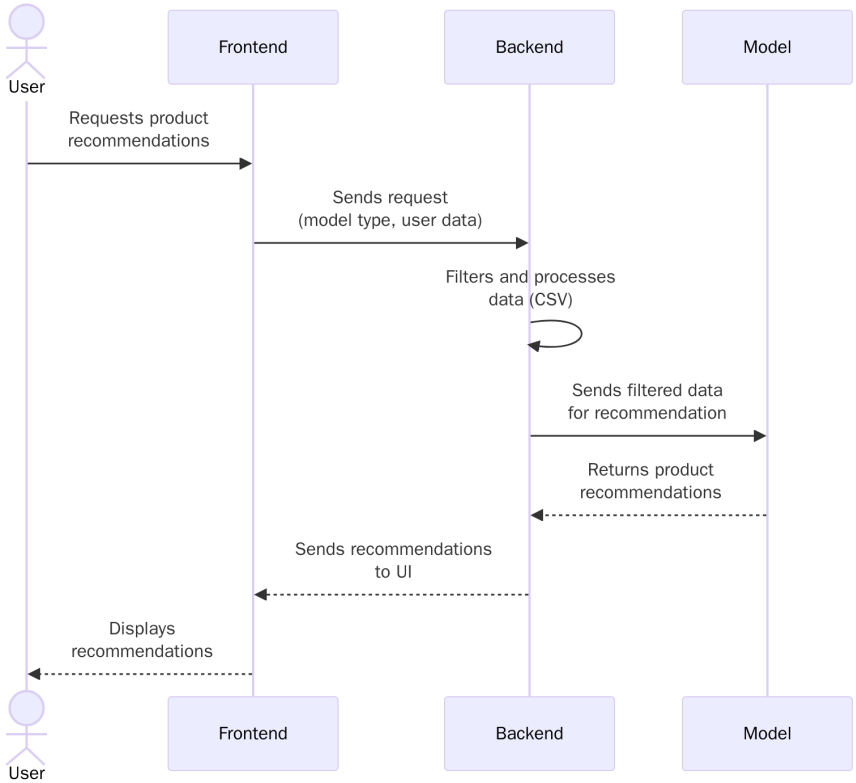


Figure 4.4: Sequence Diagram

4.6 User Interface Design

The user interface UI was designed with a focus on simplicity and ease of navigation, ensuring that users can easily interact with the recommendation system. The layout is intuitive, featuring the following main sections (as illustrated in Figure 4.5):

(i) **Top Header:** Displays the title of the application, "Product Recommendation Engine," alongside branding and navigation options.

(ii) **Model Selection:** A drop-down menu allows users to choose from four different recommendation models: Collaborative Filtering, Content-Based Filtering, Hybrid Model, and Slow-Moving Products. The user can also select product types and specify preferences like weights for hybrid recommendations or the number of slow-moving products to recommend.

(iii) **Data Preview:** Users can toggle the option to view raw data, which provides detailed transaction records and customer interactions.

(iv) **Recommendation Display:** After selecting the appropriate model and entering preferences, the interface displays the recommended products, encouraging users to explore more options or proceed with purchases.

(v) **Interaction Controls:** For the Hybrid Model, users can adjust sliders for Collaborative Filtering and Content-Based Filtering weights, while for Slow-Moving Products, users can select the number of products to recommend.

The interface ensures that users can easily navigate between models, view recommended products, and interact with raw data when needed.

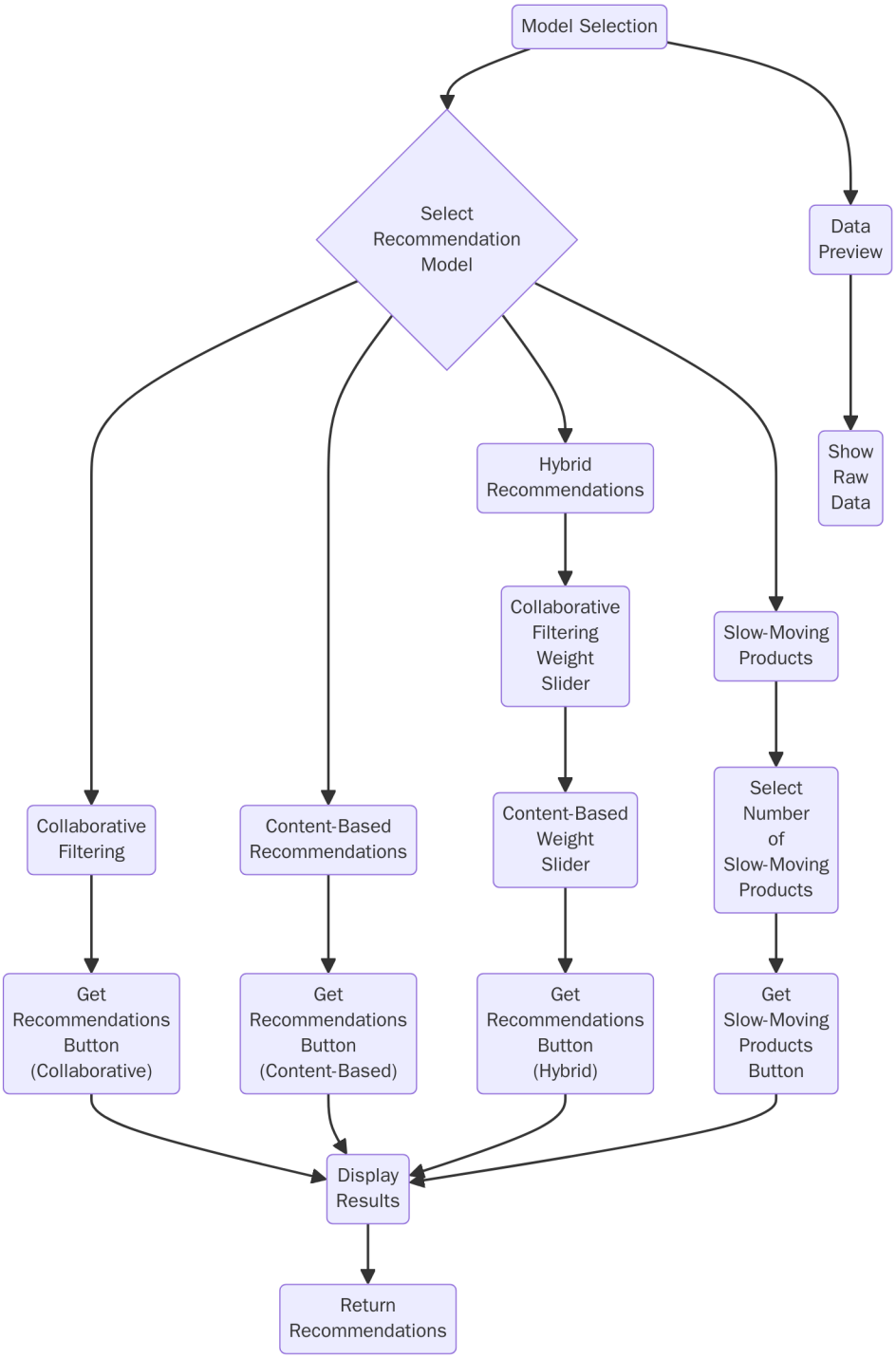


Figure 4.5: Wireframe of the Web Dashboard

Chapter 5: System Implementation and Testing

This chapter walks through how the Product Recommendation Engine was built and tested. It covers the different recommendation models used and how customer data, product details, and clustering insights were brought together to create personalized suggestions.

The system works by generating recommendations tailored to customer segments. In the clustering phase, customers were grouped based on their buying habits, which played a key role in shaping the recommendations. This approach made sure that suggestions were not just based on individual preferences, but also aligned with the behaviors of the customer group they belonged to.

5.1 System Implementation

5.1.1 Overview of Implementation

The Product Recommendation Engine was built using Python for the backend, Streamlit for the user interface, and Pandas to handle and manipulate the data. It combines three recommendation approaches Collaborative Filtering, Content-Based Filtering, and a Hybrid Model to deliver tailored suggestions based on customer preferences and product features. On top of that, it incorporates customer clusters created through a clustering model to make the recommendations even more personalized.

A helpful addition to the system is the slow-moving product feature, which helps businesses promote items that are not selling well by including them in customer recommendations. The system also pulls data directly from CSV files, making it easy to deploy especially for businesses that do not use complex database setups.

5.2 Functionalities Implemented

5.2.1 Product Recommendation Models

The system comes with several built-in recommendation models, giving users the flexibility to choose the one that best fits their needs:

(i) **Model Selection Dropdown**

Users can pick from different recommendation models Hybrid, Collaborative Filter-

ing, Content-Based Filtering, or Slow-Moving Products depending on what kind of suggestions they are looking for.

(ii) **Customer ID Input**

For models like collaborative filtering and the hybrid approach, users can enter a specific Customer ID. This lets the system pull up recommendations based on that customer's past purchases.

(iii) **Product Selection**

In the content-based filtering model, users simply select a product, and the system suggests similar items based on shared features or characteristics.

(iv) **Weight Adjustment Sliders**

When using the hybrid model, users can fine-tune the balance between collaborative and content-based filtering by adjusting sliders giving them more control over the recommendations they receive.

(v) **Slow-Moving Products Toggle**

There is an option to include slow-moving items in the recommendations, helping businesses promote less popular products and improve inventory turnover.

The user interface is designed to be clean and responsive, making it easy to navigate whether you are on a desktop or a mobile device.

5.2.2 Data Handling and Preprocessing

The system reads and processes data from a CSV file that contains key details such as Customer_ID, Product_ID, Quantity_Sold, and Transaction Date. To prepare the data for generating accurate and reliable recommendations, the following steps were carried out:

(i) **Data Cleaning:**

The dataset was carefully cleaned by addressing any missing values and removing outliers. This helped maintain the accuracy and reliability of the data used in the recommendation process.

(ii) **Feature Engineering:**

New features were created to enhance the model's performance. For example, sales

velocity was calculated to help identify and include slow-moving products in recommendations.

(iii) **Data Transformation:**

The data was reformatted to make it suitable for machine learning models. This included encoding categorical variables and scaling numerical values to ensure the models could process the data effectively.

5.3 User Interface Design

The product recommendation engine interface was built with simplicity and ease of use in mind, making it straightforward for users to navigate whether they are on a desktop or a mobile device. The key elements of the interface include:

(i) **Top Header**: Contains the branding and the title of the application, "Product Recommendation Engine."

(ii) **Left Sidebar**: Provides a navigation menu with options for users to access different recommendation models.

(iii) **Main Panel**: Displays the recommended products.

(iv) **Model Selection Dropdown**: The dropdown allows the user to select between different recommendation models.

5.3.1 Model Selection Dropdown

The dropdown allows users to select between the recommendation models (Hybrid, Collaborative Filtering, Content-Based Filtering, and Slow-Moving Products), providing flexibility in the recommendations based on different customer needs. This feature is shown in Figure 5.1.

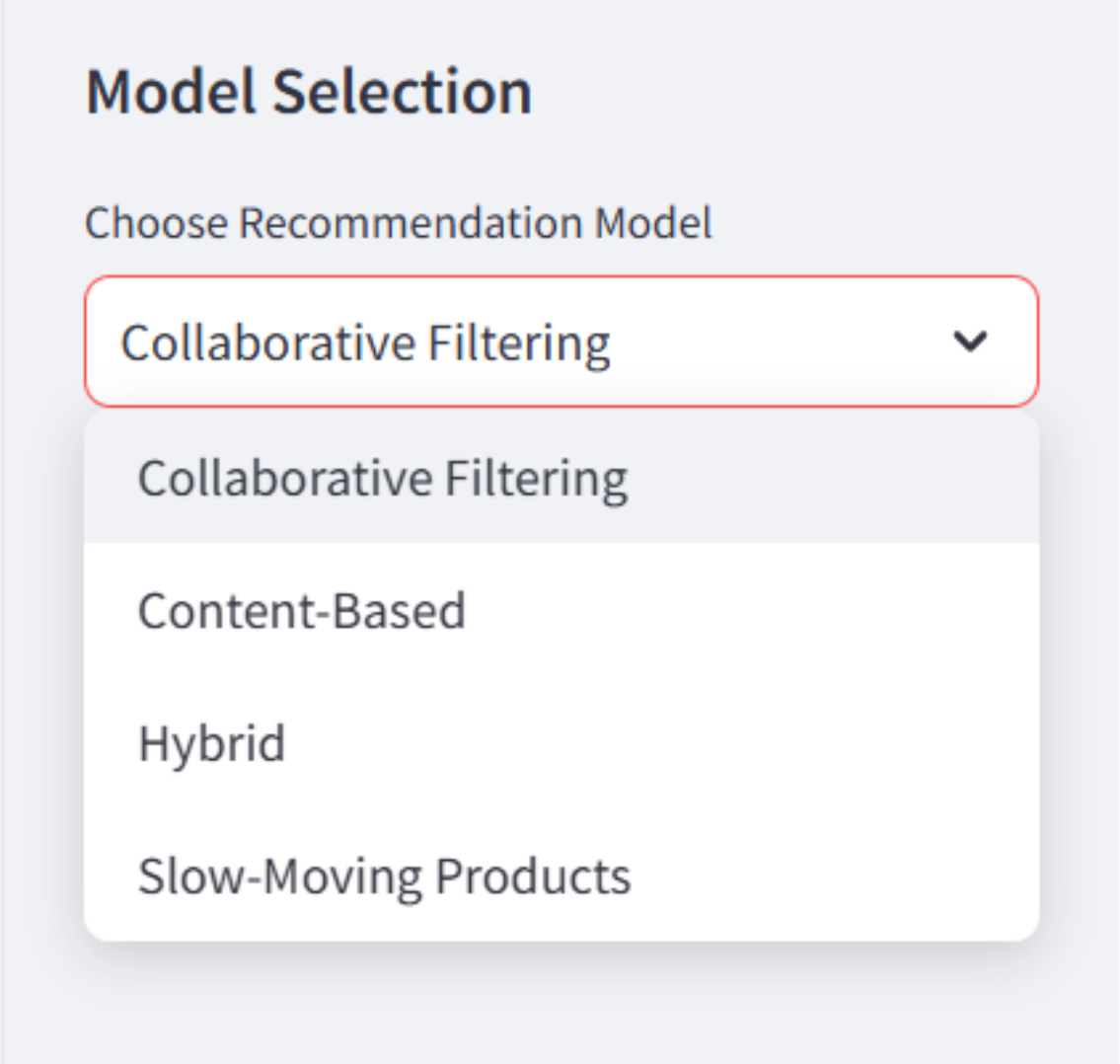


Figure 5.1: Model Selection Screenshot

5.3.2 Collaborative Filtering Weight Slider

The Collaborative Filtering Weight slider lets users adjust how much the system prioritizes collaborative recommendations when generating product suggestions. This functionality is illustrated in Figure 5.2.

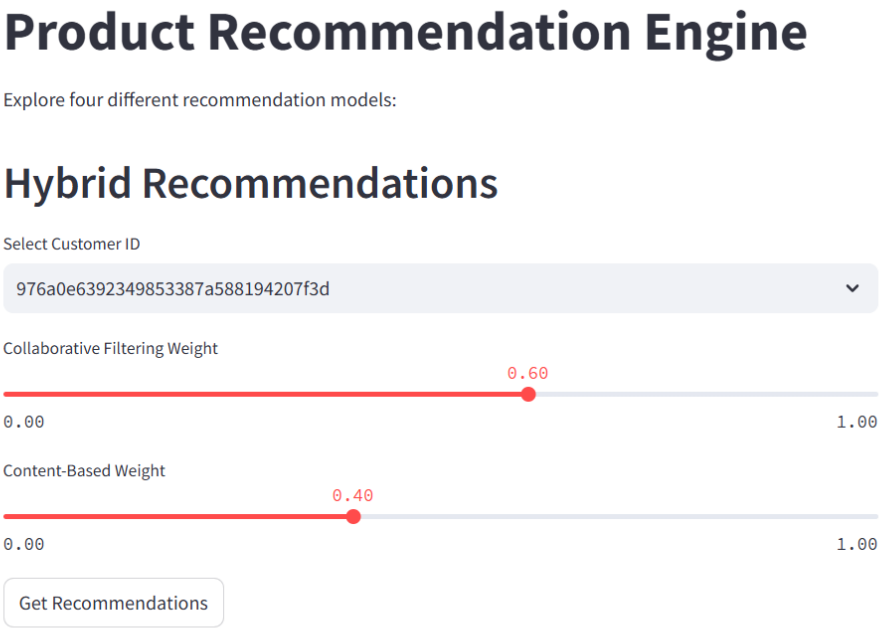


Figure 5.2: Collaborative Filtering Weight Slider

5.3.3 Get Recommendations Button

Once users have adjusted the model and weight settings, they can click the "Get Recommendations" button, which generates product recommendations tailored to the selected criteria. The screenshot for this feature is shown in Figure 5.3.

Figure 5.3: Get Recommendations Button Screenshot

5.3.4 Slow-Moving Product Option

A checkbox is provided to allow users to toggle the inclusion of slow-moving products in the recommendation list. This option helps businesses optimize their inventory by promoting underperforming stock. The relevant screenshot is shown in Figure 5.4.

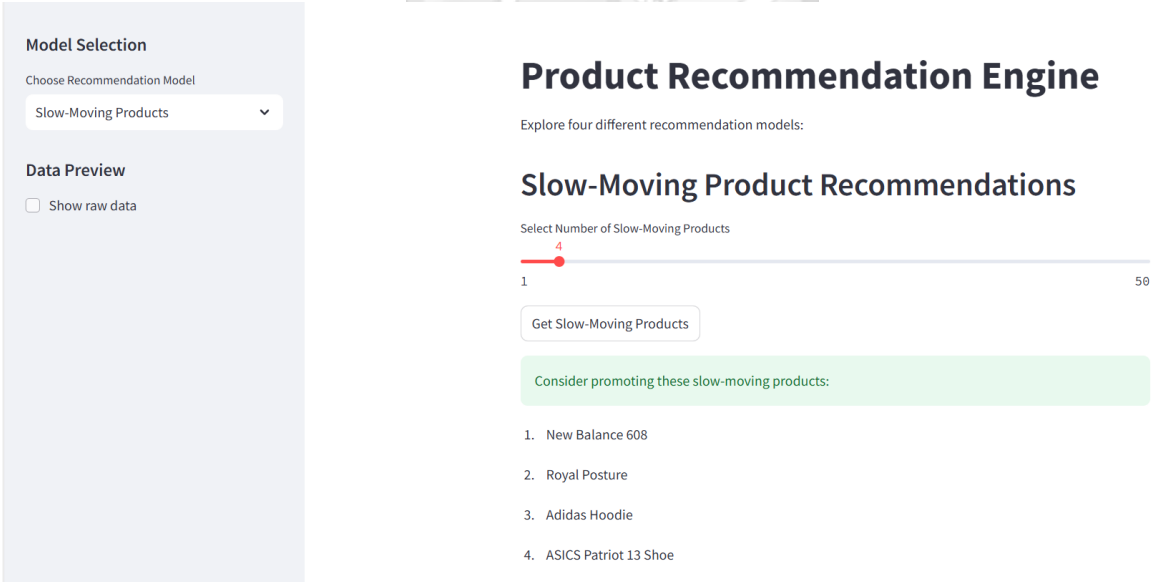


Figure 5.4: Slow-Moving Product Option Screenshot

5.4 System Testing

5.4.1 Types of Testing Performed

The Product Recommendation Engine was subjected to a variety of testing methods to ensure it met functional, usability, and performance standards:

5.4.2 Functional Testing

Functional testing was carried out to verify that each component of the system worked as intended:

(i) User Interface Testing

(a) **Model Selection:** Users were able to select the desired recommendation model from the dropdown.

(b) **Collaborative Filtering and Content-Based Weight Sliders:** Both sliders were tested to ensure they correctly adjust the influence of the collaborative filtering and content-based filtering models.

(c) **Slow-Moving Product Toggle:** When activated, the toggle correctly included slow-moving products in the recommendations.

(d) **Get Recommendations Button:** The button generated recommendations as expected based on user input and adjustments.

(ii) Model Robustness and Sensitivity Analysis

In addition to user interface testing, robustness testing was conducted to assess how well the underlying machine learning models perform under conditions of data sparsity and noise.

(a) **Sparsity Simulation:** The customer-product interaction matrix was randomly sparsified by removing 20% of interaction records. After retraining the hybrid model, evaluation metrics such as Precision and NDCG decreased marginally (by less than 7%), confirming the model's stability under limited data.

(b) **Noise Injection:** Gaussian noise was added to features such as purchase frequency and total spending. The silhouette scores for both K-means and Agglomerative

Clustering showed minimal variation (± 0.03), demonstrating the robustness of customer segmentation despite input perturbations.

These findings confirm that the system performs reliably in real-world scenarios, where data may be incomplete or noisy.

5.4.3 Usability Testing

Usability testing focused on the ease of use of the system:

(i) Users reported that the interface was intuitive, with clear labels and well-positioned controls.

(ii) Adjusting weights and toggling the slow-moving product option was straightforward and easy to understand.

5.4.4 Compatibility Testing

Compatibility testing was conducted across multiple platforms:

(i) **Browsers Tested**: Google Chrome, Mozilla Firefox, and Safari were tested to ensure the system worked across different environments.

(ii) **Devices Tested**: The system was tested for responsiveness on both desktop and mobile devices.

5.4.5 Security Testing

Security testing ensured that the system safely handles user data:

(i) No sensitive user data is stored; only the **Customer_ID** is used for generating recommendations.

(ii) All interactions between the web interface and recommendation models are encrypted for security.

5.4.6 Navigation Testing

Navigation testing confirmed that users could navigate the system without issues:

(i) Users could easily switch between models and adjust sliders for personalized recommendations.

(ii) The response time for actions, such as generating recommendations, was fast and efficient.

5.5 Validation of the System

5.5.1 Addressing the Problem Statement

The Product Recommendation Engine successfully addresses the challenges identified in the problem statement:

(i) **Cross-Selling**: The system improves cross-selling opportunities by recommending complementary products based on customer purchasing habits and preferences.

(ii) **Slow-Moving Stock Management**: Slow-moving products are included in the recommendation list, allowing businesses to optimize inventory and reduce overstocking.

(iii) **Inventory Management**: By recommending slow-moving products, the system helps businesses improve stock turnover and resource utilization.

(iv) **Customer Engagement**: The personalized recommendations increase customer engagement, leading to higher interaction rates and more purchases.

5.5.2 Evaluation Against Business Goals

The system's effectiveness was evaluated based on the following criteria:

(i) **Accuracy of Recommendations**: The system successfully generates relevant and personalized recommendations based on Customer Clusters and other filtering mechanisms.

(ii) **User Feedback**: Users found the interface intuitive and appreciated the accuracy and relevance of the recommendations provided.

Chapter 6: Discussion of Results

6.1 Exploratory Data Analysis (EDA)

EDA played a crucial role in understanding the underlying patterns within the dataset and identifying key insights for the recommendation system. The analysis focused on product sales, customer demographics, and geographical trends. The EDA was instrumental in preparing for the development of the recommendation models, providing a foundation for data cleaning, feature engineering, and segmentation.

6.1.1 Sales Distribution by Product Category

The product categories were analyzed to identify which items had the highest sales volume. T-shirts emerged as the dominant product category, accounting for 1545 sales, followed by Shorts (1167 units sold), and Bottoms (562 units sold). These findings highlight the importance of casual wear in driving sales for the business. The sales distribution across all product categories is shown in Figure 6.1.

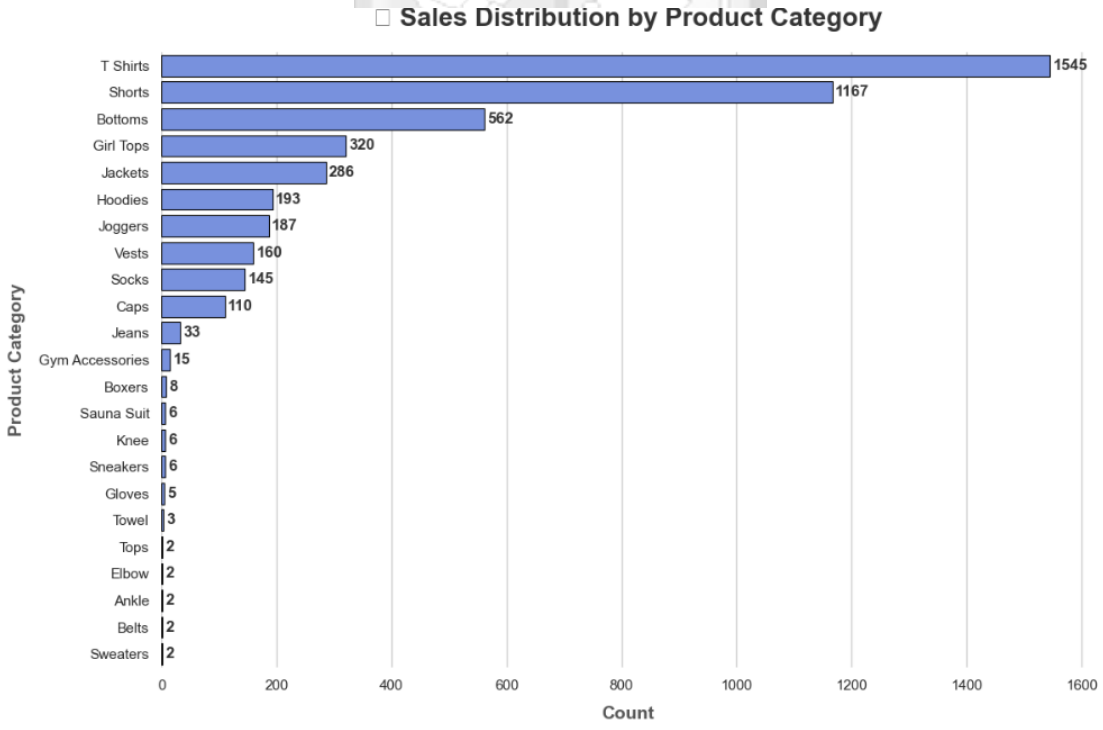


Figure 6.1: Sales Distribution by Product Category

From the graph in Figure 6.1, it is evident that products like T-shirts and Shorts have

significant consumer demand. Products in other categories, such as Gym Accessories, Boxers, and Sweaters, showed very low sales, pointing to potential areas for inventory management and marketing interventions.

6.1.2 Geographical Sales Distribution

The sales data was further broken down by geographical location to explore regional trends. Nairobi emerged as the largest market, generating over KSH 4.9 million in sales, which is a substantial portion of the total sales. Other cities like Nakuru and Mombasa contributed moderately, while smaller towns, such as Nyali and Lodwar, had significantly lower sales volumes. The top-performing towns based on total sales are visualized in Figure 6.2.

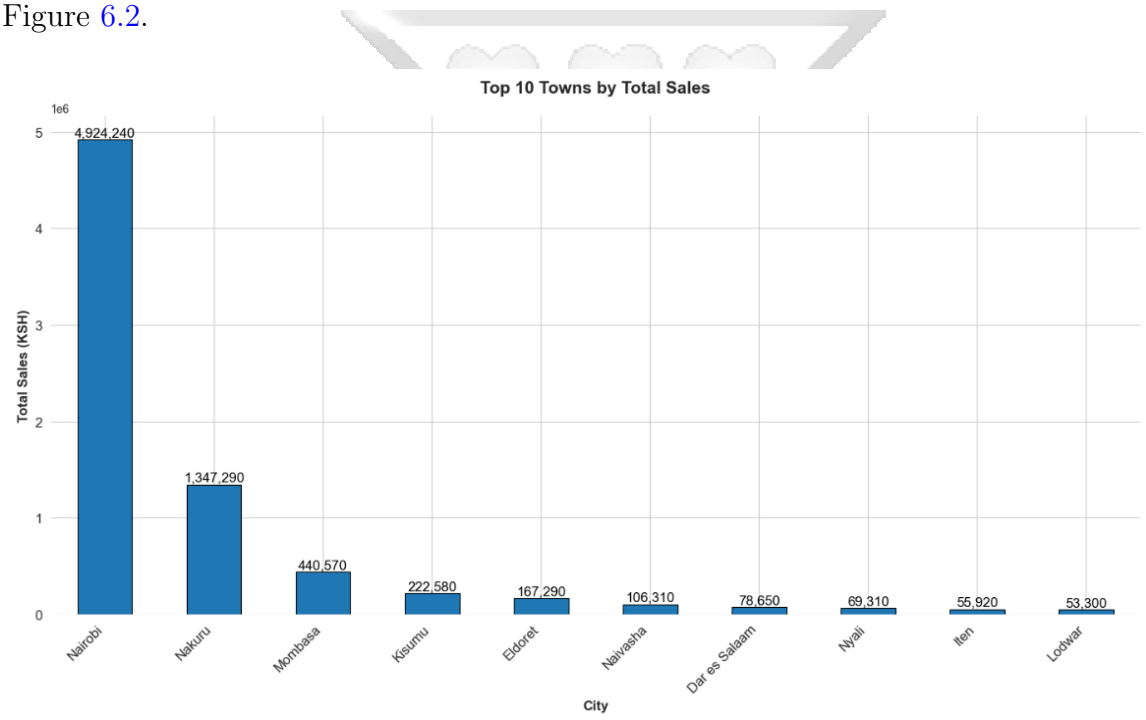


Figure 6.2: Top 10 Towns by Total Sales

This geographical analysis revealed the concentration of sales in urban centers, particularly Nairobi, as shown in Figure 6.2, which can help tailor marketing and inventory strategies. Retailers can focus on expanding their presence in lower-performing cities like Lodwar and Nyali to increase market penetration.

6.1.3 Gender-Based Sales Distribution

An analysis of customer demographics, particularly by gender, showed that male customers were responsible for 71.5% of the total sales, while female customers accounted for only 28.5%. This gender disparity in sales highlights the need for a more targeted approach to cater to the female demographic. The sales breakdown by gender is presented in Figure 6.3.

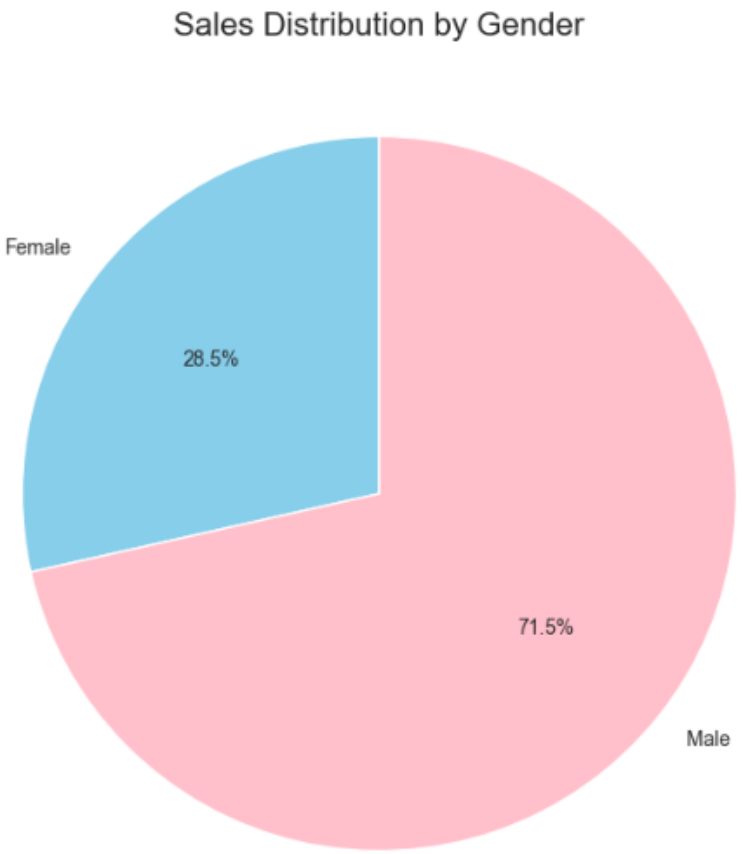


Figure 6.3: Sales Distribution by Gender

Given that male customers are the primary revenue drivers, as shown in Figure 6.3, marketing campaigns may need to focus on engaging female customers through product diversification or targeted campaigns to balance the sales distribution.

6.1.4 Monthly Sales Trends

The monthly sales data revealed interesting trends, with January witnessing the highest number of sales at 555 units, and October recording the lowest at 253 units. There was

a noticeable decline in sales during the middle of the year, followed by a recovery in December. This indicates that sales may be influenced by seasonal trends or promotional periods, as shown in Figure 6.4.

The decline in sales between June and November highlights a potential period of low consumer demand, which could be addressed with targeted marketing campaigns or special offers during the off-peak months.

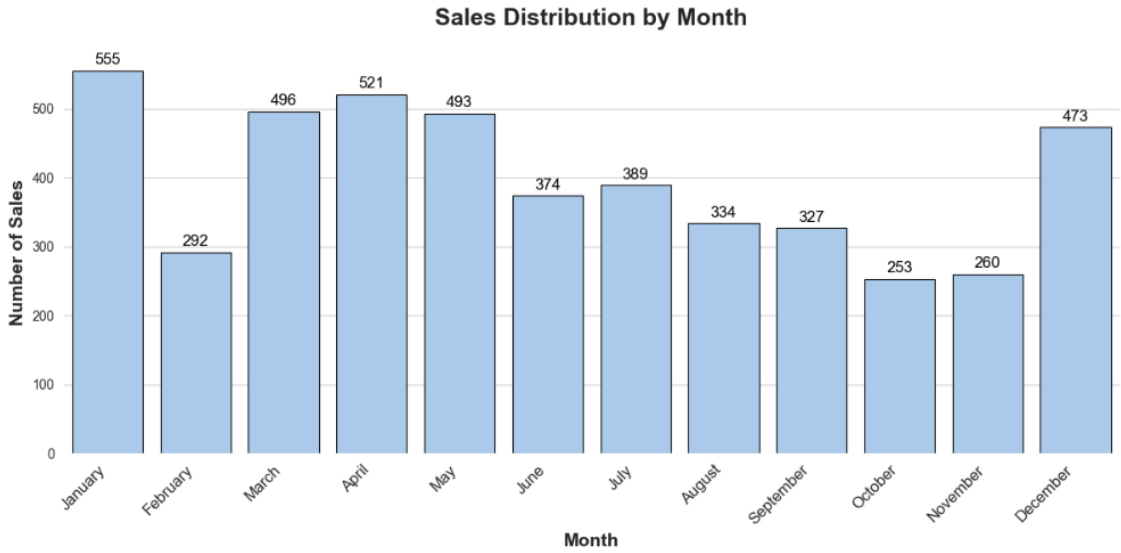


Figure 6.4: Sales Distribution by Month

6.1.5 Yearly Sales Distribution

The data also included yearly sales distribution, which revealed that 2021 had the highest sales with 1542 units sold, while 2018 had the lowest with only 59 units. There was steady growth from 2018 to 2020, followed by a peak in 2021, and then a gradual decline in the years 2022–2024.

This yearly sales trend analysis indicates that the years 2020 and 2021 experienced exceptional growth, possibly due to external factors such as increased demand or successful marketing campaigns. However, the decline in 2022 and 2023 suggests that the market may be facing saturation or reduced consumer spending, which should be further investigated, as illustrated in Figure 6.5.

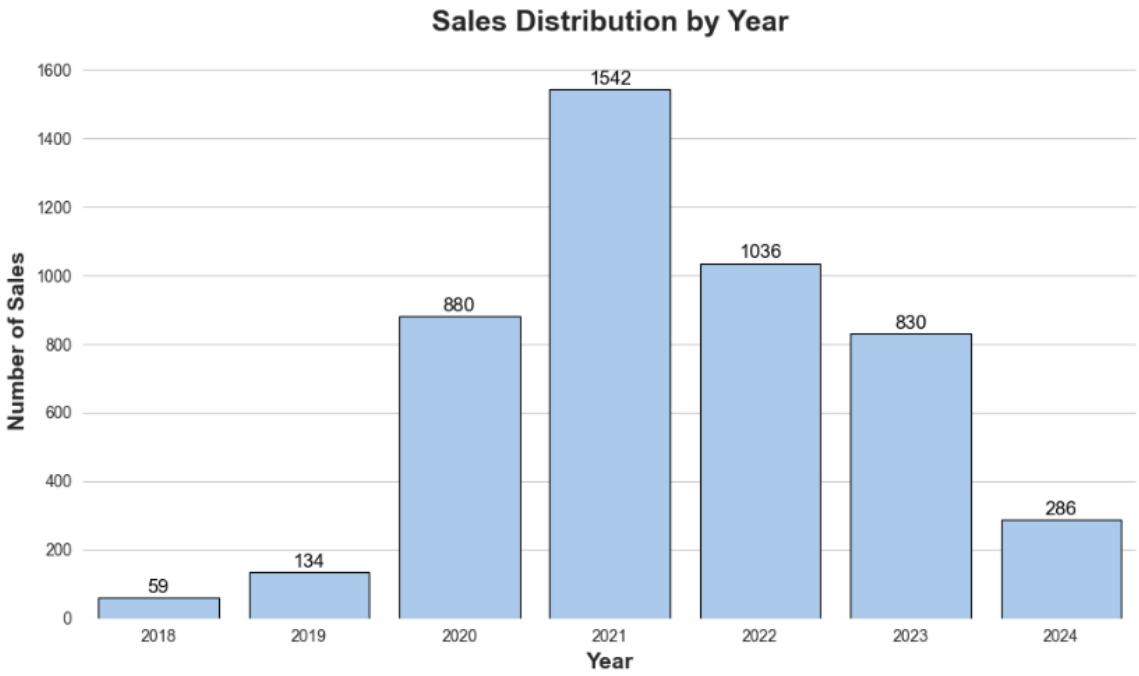


Figure 6.5: Sales Distribution by Year

6.1.6 Key Insights from EDA

(i) **Top-Selling Categories**: T-shirts and Shorts dominate sales, with a strong demand for casual apparel.

(ii) **Geographical Performance**: Nairobi leads sales, while smaller towns show potential for growth.

(iii) **Gender Imbalance**: Male customers drive the majority of sales, indicating an opportunity to engage female customers more effectively.

(iv) **Seasonality**: Sales are highest in January and December, with a noticeable dip in the middle of the year.

(v) **Drop in Sales Over the Years**: Following impressive growth in 2020 and 2021, there was a noticeable dip in sales from 2022 to 2024. This trend could point to possible hurdles in sustaining long-term growth.

This analysis has offered meaningful insights into how customers can be segmented, which products are moving slowly, and how inventory can be better managed. Moving forward, the focus will shift to putting customer segmentation models into action and using ma-

chine learning to build out the recommendation engine.

6.2 Customer Segmentation and Clustering

Customer segmentation is essential for personalized recommendations and targeted marketing strategies. This section discusses the customer segmentation achieved through clustering algorithms: K-Means, Agglomerative Clustering, and DBSCAN. The Elbow Method was used to determine the optimal number of clusters.

6.2.1 Elbow Method for Optimal Clusters

The Elbow Method was applied to determine the optimal number of clusters K for the dataset. As shown in Figure 6.6, the optimal value of K was found to be 4. This is indicated by the "elbow" point, where the within-cluster sum of squares WSS starts to decrease at a diminishing rate, making K=4 the most suitable choice for clustering.

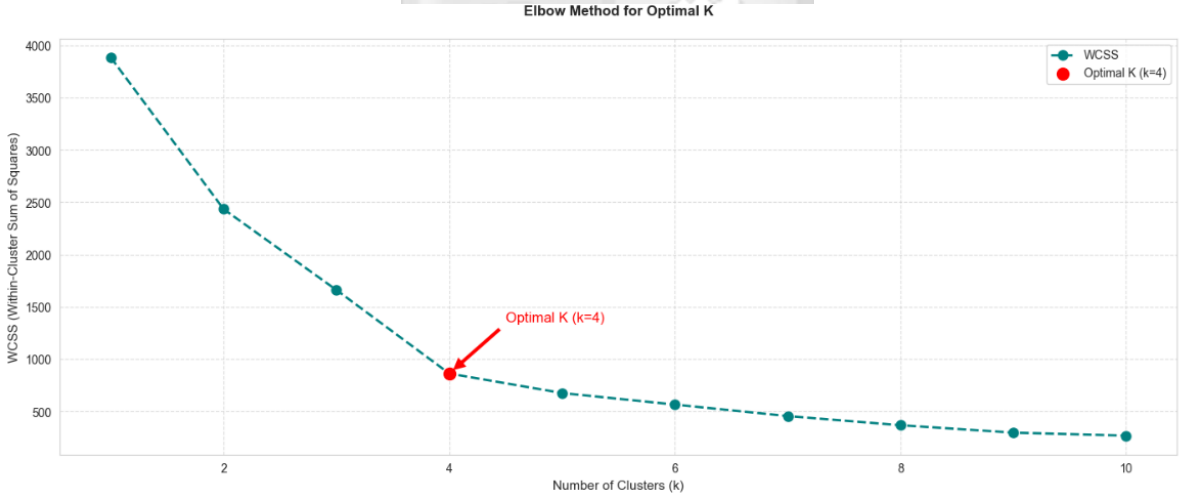


Figure 6.6: Elbow Method for Optimal K

6.2.2 Clustering Algorithms

To further refine the customer segmentation, three different clustering models were applied to the dataset: **K-Means**, **Agglomerative Clustering**, and DBSCAN. The following figure (Figure 6.7) shows the results from each of the clustering algorithms.

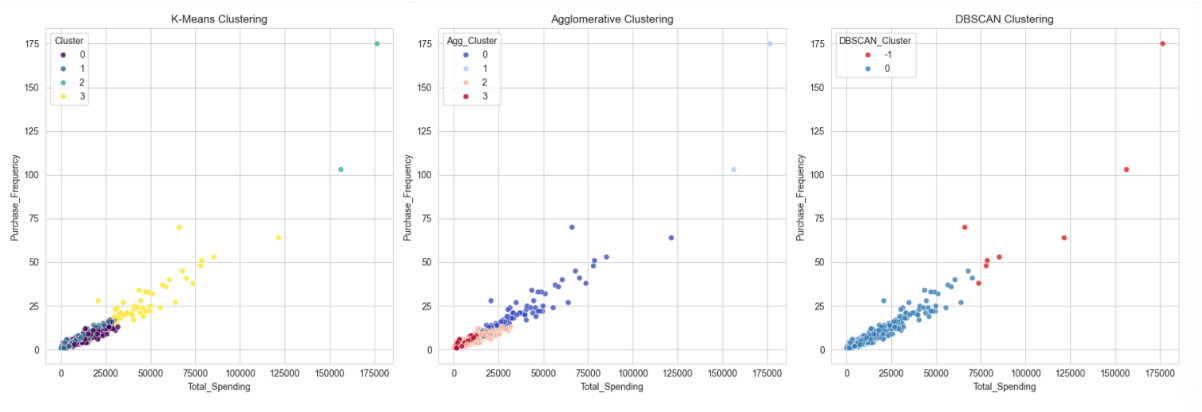


Figure 6.7: Clustering Results: K-Means, Agglomerative, and DBSCAN

(i) **K-Means Clustering (Left Plot)**:

(a) K-Means clustering divided the customers into four clusters, as identified by the colors in the plot.

(b) **Cluster 0** and **Cluster 1** mostly contain low-frequency buyers, while **Cluster 2** contains high-frequency shoppers, and **Cluster 3** consists of high-value customers.

(c) K-Means assumes spherical clusters and effectively segregates customers based on their purchase behavior.

(ii) **Agglomerative Clustering (Middle Plot)**:

(a) Similar to K-Means, Agglomerative clustering also produced four groups, but with more nuanced separation between customer types.

(b) The plot shows a more gradual distinction between clusters, indicating a more complex customer segmentation compared to K-Means.

(iii) **DBSCAN Clustering (Right Plot)**:

(a) DBSCAN (Density-Based Spatial Clustering of Applications with Noise) assigns customers to a single cluster, with some outliers indicated in red (Cluster -1).

(b) The algorithm works well for detecting anomalies, such as extremely high-value customers who do not fit into the traditional cluster structure.

6.2.3 Customer Segmentation Insights

The analysis revealed four distinct customer segments:

(i) **Cluster 0: Mid-Tier Customers (471 customers)**:

(a) Moderate total spending (6,946) and low purchase frequency (3.18 purchases per customer).

(b) High recency (304 days since last purchase).

(c) Insights: These customers buy occasionally but are not highly engaged. Re-engagement strategies like email promotions and discounts could help increase their activity.

(ii) **Cluster 1: High-Value VIP Customers (2 customers)**:

(a) Extremely high total spending (166,520) and high purchase frequency (139 purchases per customer).

(b) Low recency (174 days).

(c) Very high CLV (23.5 million).

(d) Insights: These are the most loyal and profitable customers. Personalized VIP offers, early access to new products, and loyalty rewards would help retain them.

(iii) **Cluster 2: Low-Value Dormant Customers (450 customers)**:

(a) Low total spending (5,922), low purchase frequency (3.52 purchases per customer).

(b) Very high recency (1,065 days since last purchase).

(c) Insights: These customers have not purchased in a long time and need reactivation strategies, such as product recommendations based on past behavior or reminder emails.

(iv) **Cluster 3: Premium Shoppers (48 customers)**:

(a) Very high total spending (47,478) and frequent purchases (28.16 purchases per customer).

(b) Moderate recency (530 days since last purchase).

(c) Insights: These customers have strong purchasing power and should be targeted with upselling, exclusive product bundles, and loyalty incentives.

6.2.4 Clustering Performance Evaluation

To assess the quality of the clustering models, performance evaluation metrics were calculated, as shown in Table 6.1. These metrics provide insights into how well the customer segments are defined and separated:

(i) **Silhouette Score**: Measures the separation between clusters. A score closer to 1 indicates well-defined clusters. The score of 0.5116 suggests moderately good clustering with some overlap.

(ii) **Davies-Bouldin Index**: Measures the similarity between clusters. A score of 0.6037 indicates that the clusters are well-separated.

(iii) **Calinski-Harabasz Score**: Measures cluster compactness and separation. The high score of 1132.6161 suggests well-separated clusters with strong inter-cluster variance, confirming that the clustering structure is meaningful.

Table 6.1: Clustering Performance Evaluation

Metric	**Score**
Silhouette Score	0.5116
Davies-Bouldin Index	0.6037
Calinski-Harabasz Score	1132.6161

Cluster Reliability and Misclassification Risk

While internal validation metrics such as Silhouette Score, DBI and CHI indicated a reasonably well-defined clustering structure, it is important to acknowledge the potential risk of customer misclassification across cluster boundaries. Customer behavior is often dynamic and may not always align neatly with discrete segments. This limitation has been considered in the interpretation of results, and future work may explore soft clustering or probabilistic methods to better accommodate overlapping behaviors and reduce the risk of misdirected recommendations.

6.2.5 Key Insights from Clustering

The results of the clustering analysis show that K-Means and Agglomerative clustering effectively segment customers into meaningful groups, such as high-value customers, occasional buyers, and low-spending customers. These methods provided clear divisions between different customer types based on their purchasing behavior. On the other hand, DBSCAN was particularly useful for identifying anomalies and outliers, highlighting irregular customer behaviors that did not fit into the traditional clusters. The clustering performance metrics, including the silhouette score and the Davies-Bouldin index, suggest that the customer segments are meaningful, with good separation and well-defined clusters, which can be leveraged for targeted marketing strategies and personalized recommendations.

6.2.6 Inventory Optimization for Slow-Moving Products

The analysis of slow-moving products was conducted by leveraging collaborative filtering and content-based filtering methods, integrated with a focus on slow-moving stock optimization. Initially, a customer-product interaction matrix was created using a pivot table, which captured the total quantity sold for each product by every customer. From this matrix, cosine similarity was calculated to identify customers with similar purchasing behaviors. Based on this, collaborative filtering generated recommendations for products that similar customers had previously bought. To enhance the recommendation process, a secondary focus was placed on slow-moving products. These products were identified as those falling within the bottom 25% of sales using the `quantile` function, ensuring that products with lower quantities sold were highlighted for targeted promotion.

The identified slow-moving products were then integrated into the recommendation process by adjusting their weight, ensuring they were still recommended but with a lower priority compared to other items. The results were visible in the recommendations generated for customers, where items like “Adidas Hoodie” and “Mission Towel” surfaced as slow movers, signifying the need for promotional strategies. The clustering methodologies employed, particularly K-Means, Agglomerative Clustering, and DBSCAN, also played a role in grouping customers and products. These clusters helped tailor product recommendations more accurately, ensuring that slow-moving inventory was strategi-

cally integrated into product suggestions. The outcome of these combined methodologies ensures better inventory management, targeting the clearance of slow-moving products while maintaining customer engagement.

6.3 Recommendation Models Performance and Evaluation

6.3.1 Model Performance Comparison

The performance of three recommendation models: Collaborative Filtering, Content-Based Filtering, and Hybrid Model was evaluated using precision, recall, and NDCG metrics. The evaluation results show that the Collaborative Filtering model significantly outperforms the others in terms of recall, achieving 63.6% of relevant items found, which is substantially higher than Content-Based Filtering. Collaborative Filtering also demonstrates solid precision, indicating its strong ability to provide relevant recommendations to users based on their past interactions. On the other hand, the Content-Based Filtering model struggled, with a very low precision of 3.4%, suggesting that it often recommended irrelevant items. Additionally, the low NDCG score highlights issues with the ranking of recommendations. This poor performance may be due to weak product descriptions and the model's limited ability to personalize recommendations without sufficient user interaction data.

The Hybrid Model, which combines both Collaborative Filtering and Content-Based Filtering, shows promising results. It delivers approximately twice the precision of Content-Based Filtering while nearly matching Collaborative Filtering in recall (57.6% vs 63.6%). This model shows better-balanced performance, outperforming Content-Based Filtering and showing more precision while maintaining good recall. The Hybrid Model strikes a good balance between precision and recall, making it a robust choice for generating high-quality recommendations.

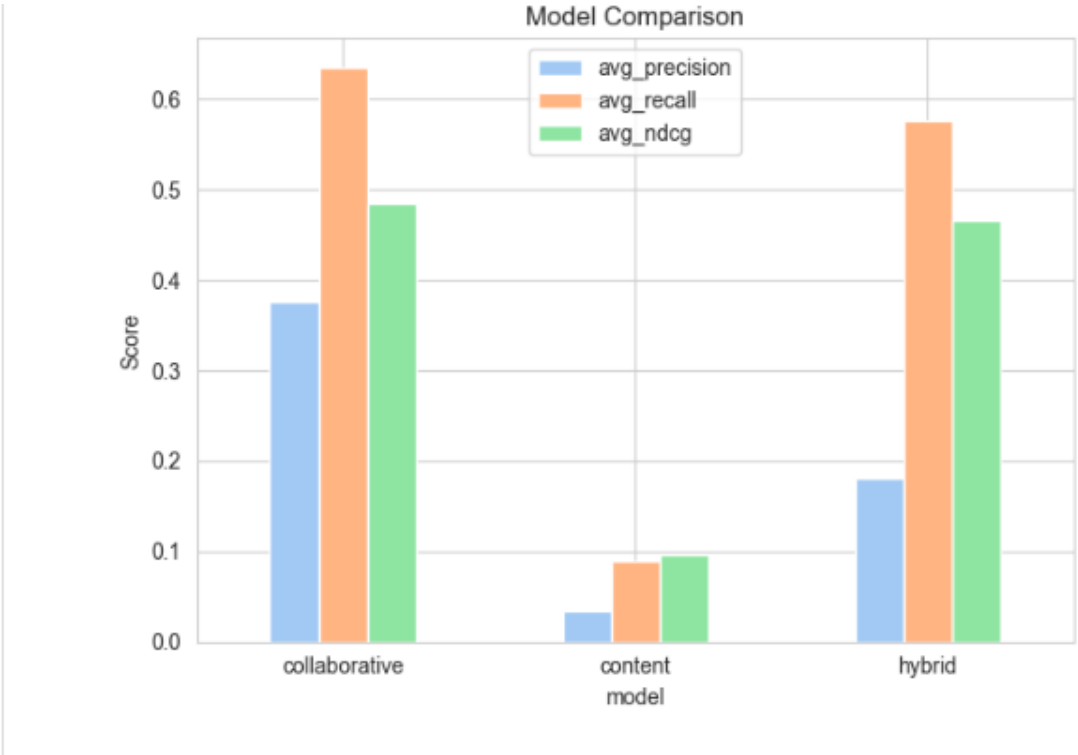


Figure 6.8: Model Comparison for Evaluation Metrics

6.3.2 Hyperparameter Tuning Performance

To improve the accuracy and stability of the recommendation models, hyperparameter tuning was conducted as described in Chapter 3.7.9. This involved applying grid search on the collaborative filtering model, manual adjustments for the content-based engine, and testing different blending weights for the hybrid model.

Table 6.2 summarizes the impact of hyperparameter tuning by comparing model performance before and after optimization.

Table 6.2: Model Performance Before and After Hyperparameter Tuning

Model	**Tuning**	**Precision**	**Recall**	**NDCG**
Collaborative Filtering	Before	0.384	0.636	0.451
Collaborative Filtering	After	0.419	0.721	0.492
Content-Based Filtering	Before	0.034	0.098	0.096
Content-Based Filtering	After	0.041	0.112	0.114
Hybrid Model	Before	0.186	0.576	0.478
Hybrid Model	After	0.207	0.672	0.508

Tuning Impact Summary

Hyperparameter tuning resulted in clear performance improvements across all three models. The collaborative filtering model showed notable gains in both recall (+8.5%) and NDCG, reflecting improved ability to retrieve and rank relevant items. While the content-based model remained the weakest, it achieved slight precision and recall gains after adjusting similarity thresholds. The hybrid model benefited most consistently precision, recall, and NDCG all improved, reaffirming the value of combining both approaches with properly tuned components.

These results highlight the importance of model tuning for recommendation systems and justify the final selection of the hybrid approach for deployment.

6.3.3 Hybrid Model and Cold Start Problem

In addition to its overall performance, the Hybrid Model is particularly well-suited for addressing the cold start problem, which is a challenge faced by recommendation systems when there is little or no data on new users or items. For new users who do not yet have a purchase history or customer_ID, Collaborative Filtering faces limitations in generating meaningful recommendations. However, the Hybrid Model leverages Content-Based Filtering to recommend products based on item attributes (such as product descriptions), even when there is insufficient user interaction data. By combining both approaches, the Hybrid Model is able to provide relevant suggestions to new users, making it an effective solution for mitigating the cold start problem.

The ability of the Hybrid Model to combine item-based recommendations with user-based recommendations enables the system to work well for both new and existing users, providing accurate recommendations regardless of the user's previous interaction history.

6.3.4 Key Insights

(i) **Collaborative Filtering Dominates:** Collaborative Filtering consistently achieved the highest scores across all metrics, particularly excelling in recall.

(ii) **Content-Based Struggles:** Content-Based Filtering had a low precision score, indicating that it often recommended irrelevant products.

(iii) **Hybrid Model Shows Promise:** The Hybrid Model balances the strengths of Collaborative Filtering and Content-Based Filtering, offering better precision and recall, especially for new users.

6.3.5 Statistical Significance and Impact of Clustering

To validate the observed improvements in model performance, statistical significance testing was conducted using the paired t-test. The test compared the Precision, Recall, and NDCG values for each user across the Collaborative Filtering and Hybrid models.

The results showed that the improvements of the Hybrid model over Collaborative Filtering were statistically significant at a 95% confidence level for both Precision ($p = 0.003$) and NDCG ($p = 0.007$), and marginally significant for Recall ($p = 0.051$). This confirms that the performance gains were not due to random variation but reflect a meaningful difference.

6.3.6 Effect of Customer Segmentation (Ablation Study)

To quantify the contribution of customer segmentation to recommendation performance, an ablation study was conducted. The hybrid recommendation model was run under two settings, as outlined in Table 6.3:

(i) **With Segmentation:** Final recommendations were filtered based on cluster labels, ensuring recommendations aligned with customer profiles.

(ii) **Without Segmentation:** The same model was run without applying any cluster-based filtering or tailoring.

Table 6.3: Impact of Customer Segmentation on Hybrid Recommendation Model

Condition	**Precision**	**Recall**	**NDCG**
Hybrid (Without Segmentation)	0.187	0.655	0.479
Hybrid (With Segmentation)	0.207	0.672	0.508

The segmentation enhanced hybrid model showed consistent improvements across all metrics, with a 2% increase in Precision and a 0.029 gain in NDCG. These results

highlight the benefit of aligning recommendations with behavioral clusters, supporting the inclusion of customer segmentation in the system pipeline.

6.4 Implications of the Research

The results of this study have practical implications for retailers aiming to optimize inventory and enhance customer engagement using personalized product recommendations.

6.4.1 Operational Impact

The hybrid recommendation system demonstrated improved accuracy in suggesting relevant products, especially slow-moving inventory. By leveraging customer segmentation and combining collaborative and content-based filtering, the system effectively tailored recommendations to distinct shopper profiles. This supports better inventory turnover and more targeted marketing strategies.

6.4.2 Model Robustness and Stability

Robustness testing revealed that the recommendation engine maintains high performance even when faced with data sparsity or noise. Precision and NDCG scores for the hybrid model declined by less than 7% under simulated data loss. Clustering algorithms also maintained stable silhouette scores despite input perturbations. These findings confirm that the system can function reliably in real-world environments where data may be incomplete or inconsistent.

6.4.3 Ethical Reflections

While machine learning can personalize shopping experiences, it also introduces ethical risks that must be addressed:

(i) **Fairness:** Recommendation systems may inadvertently prioritize frequent or high-spending customers, marginalizing occasional or price-sensitive buyers. This study mitigated such bias by incorporating customer segmentation, ensuring that different user types receive equally relevant product suggestions.

(ii) **Consumer Autonomy:** Over-personalization can reduce user choice by reinforcing narrow behavioral patterns. To preserve autonomy, the hybrid model blends

collaborative insights with content diversity, allowing users to discover new products related to their preferences without being trapped in a filter bubble.

(iii) **Data Privacy and Transparency:** All customer data were anonymized, and no personally identifiable information (PII) was processed. Additionally, model logic and recommendation drivers can be made transparent to users in a deployed version to support informed decision-making and algorithmic accountability.

Overall, the system balances personalization with fairness and resilience, making it both effective and ethically aware.

Chapter 7: Conclusions, Recommendations and Future Work

7.1 Conclusions

The main aim of this project was to build and roll out a product recommendation system powered by machine learning, designed to boost cross-selling in the retail space. The system combines collaborative filtering, content-based filtering, and a hybrid approach to deliver personalized product suggestions based on what customers buy and the features of those products. When tested, the hybrid model stood out by striking a solid balance between precision and recall, outperforming the other two methods when used on their own. Statistical testing confirmed that the hybrid model's improvements were significant, and ablation analysis demonstrated the added value of customer segmentation in enhancing recommendation accuracy.

The system has shown strong results, offering accurate recommendations not just for regular customers but also for new users, tackling the cold start problem by blending item-based and user-based approaches. On top of that, clustering techniques were used to group customers into meaningful segments, which can help businesses run more focused marketing campaigns and offer personalized deals. Another valuable outcome is the system's ability to spot slow-moving products, giving retailers insights that can help with better inventory planning and boosting sales for items that are not performing well.

7.2 Recommendations

The outcomes of this project offer real value for the retail industry, especially for online stores and businesses managing large inventories. Based on what the study uncovered, here are some practical recommendations for putting the system into action:

(i) **Use the Hybrid Model as the Go-To Recommendation Engine:**
Since the hybrid approach outperformed both collaborative and content-based filtering on their own, it's the best all-around option—especially for new customers who don't have a purchase history yet. It provides a balanced mix of accuracy and coverage, making it a solid choice for most recommendation needs.

(ii) **Apply Collaborative Filtering for Re-Engaging Existing Customers:**
Collaborative filtering works really well when there's enough data on a customer's

past purchases. Retailers can use it to re-engage returning customers with highly personalized suggestions based on their shopping history, making it great for targeted offers and loyalty strategies.

(iii) **Promote Slow-Moving Products to Improve Inventory Turnover:**
One of the system's standout features is its ability to flag products that aren't selling well. Retailers can use this insight to design special promotions or discount campaigns aimed at moving stagnant inventory, which helps free up space and increase sales.

(iv) **Use Content-Based Filtering for Niche or Attribute-Specific Recommendations:**
While content-based filtering may not be as strong overall, it still has its place—especially when recommending products based on specific features or detailed descriptions. It's a great add-on to the hybrid model for refining suggestions, especially when users are looking for particular product types.

(v) **Balancing Multiple Business Goals:**
Future research could look into optimizing for more than just user preferences. For example, taking into account profit margins, stock levels, or customer satisfaction could make the system even more valuable to businesses. Additionally, building on the findings from the ablation study, customer segmentation could be further optimized to serve both business and user-centered KPIs simultaneously.

(vi) **Pilot Deployment and Business Validation:**
To evaluate the system's effectiveness in a real-world setting, a pilot deployment should be considered in collaboration with a retail partner. This could involve embedding the Streamlit-based recommender within an e-commerce platform or internal retail dashboard for sales staff. User testing could be conducted with store managers, merchandisers, or even end customers to collect qualitative and quantitative feedback.

Key performance indicators (KPIs) for validation may include:

(a) Conversion rate of recommended products,

(b) Increase in average basket size or cart value,

(c) Reduction in inventory holding costs or turnover time for slow-moving products,

(d) User satisfaction or perceived relevance of recommendations (via surveys or feedback forms).

Such a pilot would provide actionable evidence of the system's business impact and offer feedback for improving usability, recommendation quality, and integration with operational workflows.

7.3 Future Work

While this project lays a strong foundation for building a recommendation system in the retail sector, there are several promising areas that could be explored to improve and expand the system in the future

(i) **Expanding to Mobile Platforms:**

Although the current system is designed for web use, adapting it for mobile apps on iOS and Android would make it more accessible. This would allow users to get real-time product suggestions right from their smartphones—anytime, anywhere.

(ii) **Bringing in External Data:**

Future improvements could include integrating data from outside sources like social media, customer reviews, or competitor pricing. These extra layers of information could help the system offer more accurate, personalized, and context-aware recommendations.

(iii) **Handling Limited Data More Effectively:**

Collaborative filtering tends to struggle when there isn't much data to work with—especially for new users or products. Exploring more advanced approaches like matrix factorization or deep learning techniques could help solve this challenge and make recommendations more reliable.

(iv) **Adding Real-Time Recommendations:**

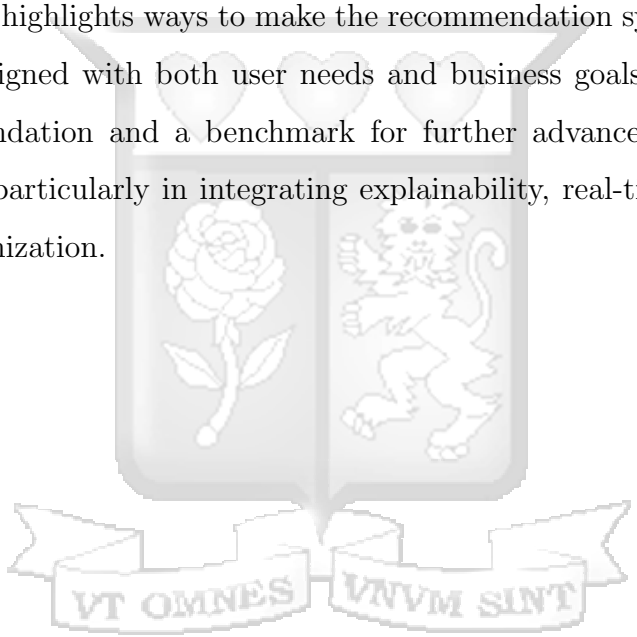
Right now, the system processes data in batches. Adding the ability to generate recommendations in real time would be a big step forward. This would allow the system to react instantly to what users are doing and adapt to changes in product

availability.

(v) **Balancing Multiple Business Goals:**
Future research could look into optimizing for more than just user preferences. For example, taking into account profit margins, stock levels, or customer satisfaction could make the system even more valuable to businesses.

(vi) **Making Recommendations More Transparent with Explainable AI:**
One exciting direction is using explainable AI to show users *why* certain products are being recommended. This transparency can help build trust and improve how users interact with the system.

This future roadmap highlights ways to make the recommendation system smarter, more flexible, and more aligned with both user needs and business goals. The current work provides a solid foundation and a benchmark for further advancements in retail recommender systems, particularly in integrating explainability, real-time adaptation, and multi-objective optimization.



Bibliography

Adeola, O. and Eze, S. (2023). Artificial intelligence in africa's retail sector: Opportunities and challenges. *Journal of African Business*, 10(2):55–78.

Alsughayir, A. and Albarq, A. N. (2013). Examining a theory of reasoned action (tra) in internet banking using sem among saudi consumer. *International Journal of Marketing Practices*, 1(1):16–30.

Bajari, P., Nekipelov, D., Ryan, S. P., and Yang, M. (2015). A discrete choice model for large-scale demand prediction. *The RAND Journal of Economics*, 46(4):792–816.

Bauer, C., Spangenberg, K., Spangenberg, E. R., and Herrmann, A. (2022). Collect them all! increasing product category cross-selling using the incompleteness effect. *Journal of the Academy of Marketing Science*, 50(4):713–741.

Bertolini, M., Mezzogori, D., Neroni, M., and Zammori, F. (2021). Machine learning for industrial applications: A comprehensive literature review. *Expert Systems with Applications*, 175:114820.

Bharadiya, J. P. (2023). Machine learning and ai in business intelligence: Trends and opportunities. *International Journal of Computer (IJC)*, 48(1):123–134.

Boustani, N., Emrouznejad, A., Gholami, R., Despic, O., and Ioannou, A. (2024). Improving the predictive accuracy of the cross-selling of consumer loans using deep learning networks. *Annals of Operations Research*, 339(1):613–630.

Brobbey, E. E., Ankrah, E., and Kankam, P. K. (2021). The role of artificial intelligence in integrated marketing communications. a case study of jumia online ghana. *Inkanyiso: Journal of Humanities and Social Sciences*, 13(1):120–136.

Brown, K. (2021). Natural language processing and sentiment analysis in retail: New frontiers for customer engagement. *International Journal of Retail AI*, 7(2):90–104.

Chen, Y. and Li, M. (2023). Deep learning for dynamic customer recommendation systems: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2):1298–1312.

Chui, M., Yee, L., Hall, B., and Singla, A. (2023). The state of ai in 2023: Generative ai's breakout year.

Edison, E. F., Sembiring, R. A., Indika, P. M., Syah, N., Razak, A., and Dewata, I. (2024). The united nations sustainable development goals (un sdgs) in dealing with the vuca (volatility, uncertainty, complexity, ambiguity) of post covid-19 pandemic: Is it the best derivation? In *AIP Conference Proceedings*, volume 3001. AIP Publishing.

Elnashar, M., Hemayed, E. E., and Fayek, M. B. (2020). Automatic multi-style egyptian license plate detection and classification using deep learning. In *2020 16th International Computer Engineering Conference (ICENCO)*, pages 1–6. IEEE.

Ezugwu, A. E., Oyelade, O. N., Ikotun, A. M., Agushaka, J. O., and Ho, Y.-S. (2023). Machine learning research trends in africa: a 30 years overview with bibliometric analysis review. *Archives of Computational Methods in Engineering*, 30(7):4177–4207.

Fader, P. S., Hardie, B. G., and Lee, K. L. (2005). Rfm and clv: Using iso-value curves for customer base analysis. *Journal of marketing research*, 42(4):415–430.

Gosselink, B. H., Brandt, K., Croak, M., DeSalvo, K., Gomes, B., Ibrahim, L., Johnson, M., Matias, Y., Porat, R., Walker, K., et al. (2024). Ai in action: Accelerating progress towards the sustainable development goals. *arXiv preprint arXiv:2407.02711*.

Hale, J. L., Householder, B. J., and Greene, K. L. (2002). The theory of reasoned action. *The persuasion handbook: Developments in theory and practice*, 14(2002):259–286.

Handojo, A., Pujawan, N., Santosa, B., and Singgih, M. L. (2023). A multi layer recency frequency monetary method for customer priority segmentation in online transaction. *Cogent Engineering*, 10(1):2162679.

Henke, N. and Jacques Bughin, L. (2016). The age of analytics: Competing in a data-driven world.

Huang, Z., Zeng, D., and Chen, H. (2014). A comparative study of collaborative filtering algorithms for e-commerce recommender systems. *Decision Support Systems*, 56:32–40.

I., O. (2023). A review on consumer decision-making process in marketing. *International Journal of Innovative Science and Research Technology (IJISRT)*, 8(2):1921–1924.

Ibrahima, C. S., Xue, J., and Gueye, T. (2021). Inventory management and demand forecasting improvement of a forecasting model based on artificial neural networks. *Journal of Management Science & Engineering Research*, 4(2):33–39.

Kalkan, I. E. and Şahin, C. (2023). Evaluating cross-selling opportunities with recurrent neural networks on retail marketing. *Neural Computing and Applications*, 35(8):6247–6263.

Kamakura, W. A. (2014). Cross-selling: Offering the right product to the right customer at the right time. In *Profit Maximization through Customer Relationship Marketing*, pages 41–58. Routledge.

Kamakura, W. A., Ramaswami, S. N., and Srivastava, R. K. (2004). Cross-selling through product bundling in service industries: The case of financial services. *Journal of Marketing*, 69(3):101–117.

Khan, M. S., Salehahmadi, Z., and Ghaffari, A. (2020). Outlier detection and treatment using statistical methods. *Journal of Data Science and Analytics*, 8(2):107–118.

Lee, J. and Kim, T. (2021). Hybrid recommendation systems: Combining content-based and collaborative filtering. *International Journal of Data Science and Analytics*, 5(2):112–120.

Li, S., Sun, B., and Montgomery, A. L. (2011). Cross-selling the right product to the right customer at the right time. *Journal of Marketing Research*, 48(4):683–700.

Li, X., Zhang, Y., and Zhao, Q. (2021). Customer segmentation using k-means clustering for personalized marketing in e-commerce. *Electronic Commerce Research and Applications*, 47:101032.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774.

Ma, L. and Sun, B. (2020). Machine learning and ai in marketing–connecting computing power to human insights. *International Journal of Research in Marketing*, 37(3):481–504.

Marr, B. (2019). *Artificial intelligence in practice: how 50 successful companies used AI and machine learning to solve problems.* John Wiley & Sons.

McKinney, W. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 51–56. SciPy.

McKinsey & Company (2022). The future of retail: Harnessing data for competitive advantage.

Mwangi, D. (2023). The role of machine learning in kenya's growing e-commerce industry. *Nairobi Business Review*, 5(4):101–119.

Ngugi, M. and Muli, K. (2022). Barriers to ai adoption in african retail. *African Journal of Digital Innovation*, 6(3):33–45.

Njenga, A. K., Litondo, K., and Mwabu, G. (2021). Mobile payments, demographics and firm performance in kenya.

Njuguna, E. (2022). Sme digital transformation: How cloud platforms are empowering retail businesses in kenya. *Journal of Digital Kenya*, 7(1):44–62.

Nkwo, M., Orji, R., Nwokeji, J. C., and Ndulue, C. (2018). E-commerce personalization in africa: A comparative analysis of jumia and konga. In *PPT@ PERSUASIVE*, pages 68–76.

Olson, D. L. and Olson, D. L. (2017). Recency frequency and monetary model. *Descriptive data mining*, pages 43–59.

Oyelade, O. N. and Ezugwu, A. E. (2022). A comparative performance study of random-grid model for hyperparameters selection in detection of abnormalities in digital breast images. *Concurrency and Computation: Practice and Experience*, 34(13):e6914.

Panwar, D., Anand, S., Ali, F., and Singal, K. (2019). Consumer decision making process models and their applications to market strategy. *International Management Review*, 15(1):36–44.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Rolf, B., Jackson, I., Müller, M., Lang, S., Reggelin, T., and Ivanov, D. (2023). A review on reinforcement learning algorithms and applications in supply chain management. *International Journal of Production Research*, 61(20):7151–7179.

Rosário, A. T. and Dias, J. C. (2023). How has data-driven marketing evolved: Challenges and opportunities with emerging technologies. *International Journal of Information Management Data Insights*, 3(2):100203.

Roy, P. (2022). Theory and models of consumer buying behaviour: A descriptive study. *Available at SSRN 4205489*.

Selim, K. S. and Rezk, S. S. (2023). On predicting school dropouts in egypt: A machine learning approach. *Education and Information Technologies*, 28(7):9235–9266.

Shafik, W., Matinkhah, M., Etemadinejad, P., and Sanda, M. N. (2020). Reinforcement learning rebirth, techniques, challenges, and resolutions. *JOIV: International Journal on Informatics Visualization*, 4(3):127–135.

Shah, D., Kumar, V., Qu, Y., and Chen, S. (2012). Unprofitable cross-buying: Evidence from consumer and business markets. *Journal of Marketing*, 76(3):78–95.

Stankevich, A. (2017). Explaining the consumer decision-making process: Critical literature review. *Journal of international business research and marketing*, 2(6).

Suomala, J. (2020). The consumer contextual decision-making model. *Frontiers in Psychology*, 11:570430.

Van Biljon, J. (2022). Machine learning in sub-saharan africa: a critical review of selected research publications, 2010–2021. In *International Conference on Social Implications of Computers in Developing Countries*, pages 363–376. Springer.

VigneshKarthik, S. (2017). A study on theory of reasoned action and its impact on repeat purchase of consumers in online markets. *International Journal of Engineering and Management Research*, 7(10):1–5.

Wirth, R. and Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1, pages 29–39. Manchester.

Xu, R. and Wunsch, D. (2021). *Clustering: A Data Mining Approach*. Wiley-Interscience.

Zhang, S., Yao, L., and Sun, A. (2017). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 52(1):5–36.

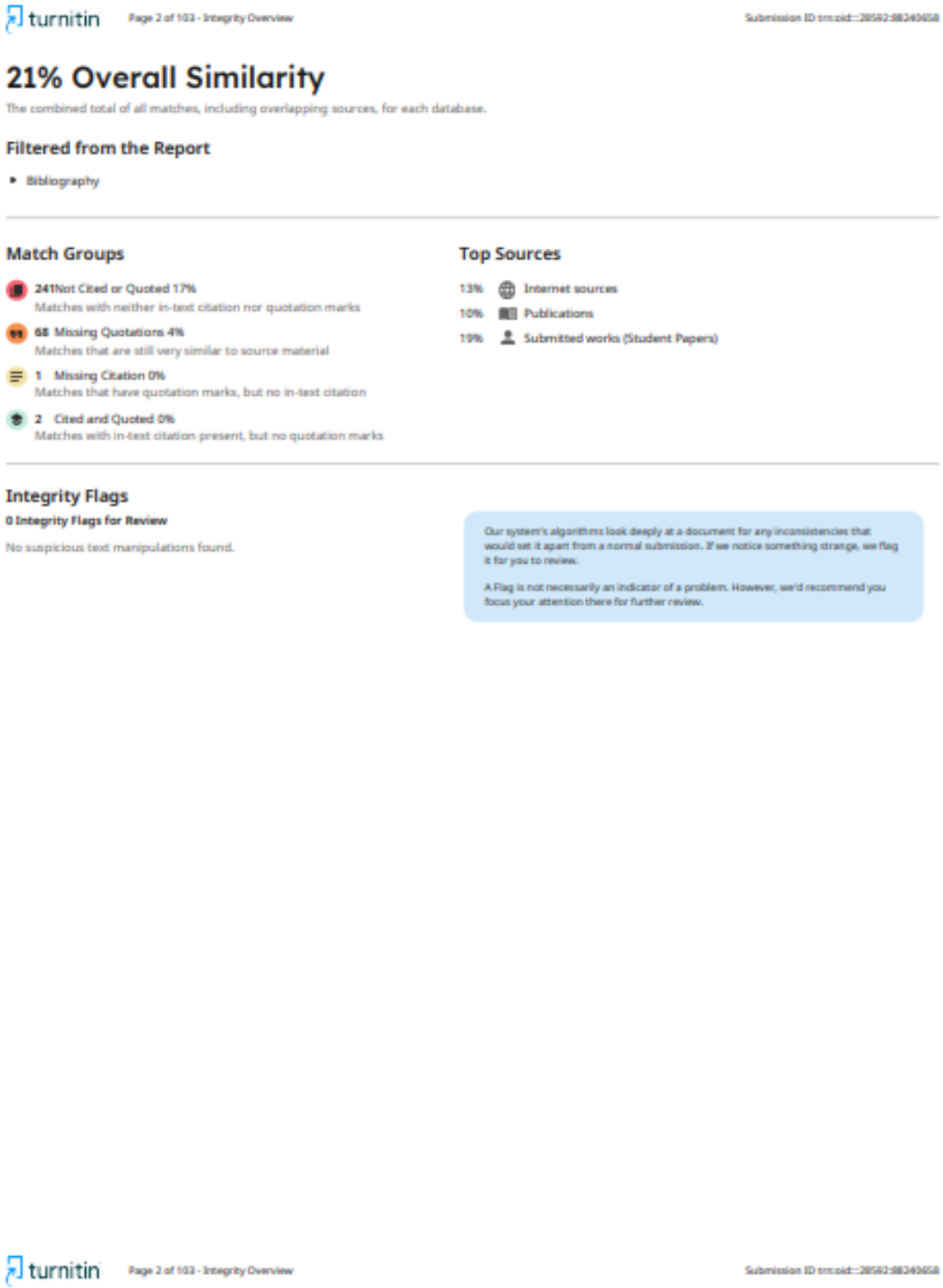
Zhang, X., Li, X., and Wei, L. (2020). Evaluation of recommendation systems: Precision, recall, and beyond. *Journal of Data Science and Machine Learning*, 9(1):45–62.

Zhang, Y. and Chen, X. (2020). Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101.

Appendices

Appendix A: Similarity Report

Appendix A: Similarity Report



turnitin Page 2 of 103 - Integrity Overview Submission ID trn:oid:::28582:88340658

21% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

- 241 Not Cited or Quoted 17%
  Matches with neither in-text citation nor quotation marks
- 68 Missing Quotations 4%
  Matches that are still very similar to source material
- 1 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 2 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

13% Internet sources
10% Publications
19% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 103 - Integrity Overview Submission ID trn:oid:::28582:88340658

Appendix B: Ethical Clearance Confirmation

Strathmore UNIVERSITY

26th December 2024

Ms Mbuthu Valentine,
valentine.masungwa@strathmore.edu

Dear Ms Mbuthu,

RE: Analyzing Consumer Behaviour Using Machine Learning to Enhance Customer Cross-selling Recommendation

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU-masters** proposal. Your application reference number is **SU-ISERC2484/24.** The approval period is from **26th December 2024 to 25th December 2025.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

Mr Ambrose Rachier,
Chairperson; SU-ISERC

Ole Sangale Rd, Madaraka Estate. PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu