National College of Ireland

Project Submission Sheet

Student Name:	Amritha Riji Sudheer		
Student ID:	x23186887		
Programme:	MSc. Data Analytics	**Year:**	2024-25
Module:	Research Practicum		
Lecturer:	Arjun Chikkankod		
Submission Due Date:	15/0/2025		
Project Title:	Personalized Dynamic Marketing Optimization: Integrating Hybrid Recommender Systems, Machine Learning and LLM concepts for E-commerce		
Word Count:	8440		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	Amritha Riji Sudheer
Date:	14/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**
6. **Please check that you read AI and Academic Integrity Acknowledgement Supplments in this document**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

[Insert Module Name]

[Insert Title of your assignment]

Your Name/Student Number	Course	Date
x23186887	MSc. Data Analytics	14/09/2025

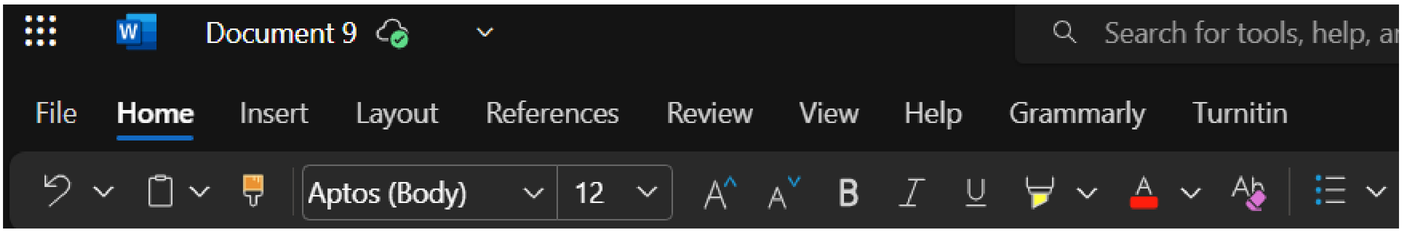
This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
Grammarly	Used to correct grammar mistakes and punctuation errors	NCI >Office365> Word > Grammarly

Evidence of AI Usage



Personalized Dynamic Marketing Optimization: Integrating Hybrid Recommender Systems, Machine Learning and LLM concepts for E-commerce

MSc. Data Analytics
Research Project

Amritha Riji Sudheer
x23186887

School of Computing
National College of Ireland

Supervisor: Arjun Chikkankod

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Amritha Riji Sudheer
Student ID:	x23186887
Programme:	MSc. Data Analytics
Year:	2024-25
Module:	Research Project
Supervisor:	Arjun Chikkankod
Submission Due Date:	15/09/2025
Project Title:	Personalized Dynamic Marketing Optimization: Integrating Hybrid Recommender Systems, Machine Learning and LLM concepts for E-commerce
Word Count:	7933
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Amritha Riji Sudheer
Date:	14th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Personalized Dynamic Marketing Optimization: Integrating Hybrid Recommender Systems, Machine Learning and LLM concepts for E-commerce

Amritha Riji Sudheer
x23186887

The accelerated expansion of e-commerce has fueled the demand for customized marketing strategies that are capable of responding to the unique customer tastes while optimizing pricing in real time. The Personalized Dynamic Marketing Optimizer (PDMO) is the system that this study introduces, which can be used to maximize customer engagement and revenue through personalized messaging, product suggestions, and dynamic price optimization. Two alternative methods are tried -a Reinforcement Learning (RL)– based PDMO integrated with the Llama-3.1 large language model, and a hybrid machine learning-driven PDMO combining a Neural Recommender, XGBoost pricing, and the mistral language model. RL-based solution characterizes the marketing optimization as a sequential decision problem, with an agent trained for the optimal strategy through customer interaction simulation with engagement and conversion reward signals. The hybrid solution combines deep learning–based recommendatiōn̩, tree-boosting regression for price optimization, and generative marketing message generation using Mistral. Both processes employ advanced natural language processing (NLP) for personalized message generation and maintain stringent compliance with price ceilings to maintain customer trust and market competitiveness. Ranked on recommendation relevance, pricing accuracy (MAE, R^2), message quality (BLEU, perplexity), and campaign metrics (open rate, CTR, conversion rate, revenue), the hybrid pipeline performs well in short-term stability with excellent pricing compliance (97%) and recommendation accuracy (0.92), fueled by the strong feature handling of XGBoost, albeit constrained by conservative revenue ($42.10). On the other hand, the RL-LLaMA-3.1 pipeline generates higher revenue ($324.98) with dynamic exploration at the cost of low pricing compliance (9.49%) and volatility (MAE 233.42), pointing to SAC's compromises during unstable training. The research contributes a comparative analysis of two state-of-the-art personalization pipelines, shedding light on their respective strengths and compromises for both researchers and practitioners. Future work will study the integration of reinforcement learning with neural recommendation to balance adaptability and predictive accuracy in a single framework.

1 Introduction

The sudden digitalization of business and the dawn of data marketing have transformed the manner in which companies interact with consumers. In an e-commerce competitive environment, Anand (2024) identify that the ability to deliver customized content and instant pricing models has been a key driver of customer interaction and conversion. Customers today demand product recommendations that are tailored to them, marketing communications that are relevant, and moving prices that match their interests and price points. These entail marketing pipelines that are infused with sophisticated artificial intelligence (AI) techniques, a trend highlighted in the work of Gao, Wang, Xie, Hu and Hu (2024).

Thakkar and Manimaran (2024) and Li and Zhang (2023) postulate that Large Language Models (LLMs) have shown tremendous potential to transform modern marketing management in making content creation, customer interaction automation, and strategic decision-making possible. Concurrently, Kang and McAuley (2018), Daruvuri and Patibandla (2024), and Ikeda et al. (2024) demonstrate that reinforcement learning (RL) and predictive models based on machine learning are becoming more efficient in dynamic pricing in retail, enabling real-time decision-making that optimizes profitability over customer satisfaction. Li and Zhang (2023), Zhao et al. (2019), and Anand (2024) discuss how recommendation methods, including neural collaborative filtering, hybrid models, and content-based filtering, are increasingly enhancing product targeting precision.

While past work has addressed individual components—like recommendation systems researched by Mulyana et al. (2024) and Yoshikawa et al. (2024), and price adjustment research conducted by Miao (2024)—there remains a gap in bringing these components together into an integrated, marketing optimization framework. Furthermore, the relative performance of reinforcement learning–augmented marketing optimisation versus hybrid machine learning pipelines that combine neural recommenders, gradient boosting, and transformer-based language models has not been investigated extensively.

This paper addresses this gap with the proposal of the Personalized Dynamic Marketing Optimizer (PDMO), a framework to deliver personalized messages, product recommendations, and price optimization plans in real time. The work discusses two distinct technical solutions. The first one involves the integration of reinforcement learning with LLaMA-3.1 to support dynamic decision-making and personalized content generation. The second one uses a neural recommender system along with XGBoost for price optimization and Mistral for high-quality marketing message generation.

The research is guided by three core objectives. The first is to design and implement an AI-driven marketing system that blends personalisation, recommendation, and price optimisation. The second is to contrast the performance of reinforcement learning–based optimisation against a hybrid neural recommender and gradient boosting pipeline. The third is to evaluate the effectiveness of LLM-generated marketing messages in driving customer engagement.

The remaining part of this report is presented as below. Section 2 presents related literature on marketing with AI, dynamic price, and recommendation systems. Section 3 describes the overall and technical approach adopted in the work. Section 4 presents the design specifications of every method. Section 5 elaborates implementation of the two models of PDMO. Section 6 quantitatively and qualitatively measures the results. Section 7 concludes the report with prominent findings, limitations, and future research directions.

2 Related Work

This literature review synthesizes recent advances in recommender systems (RS), pricing optimization, and message generation with a focus on hybrid approaches, reinforcement learning (RL), and XGBoost for pricing models, along with open-source large language models (LLMs) like Mistral-7B and Llama-3.1 Aggarwal (2016). The review evaluates the strengths and weaknesses of these technologies, identifies areas where there is a research gap, and aligns with the project goal of creating a scalable, efficient, and ethical system for personalized e-commerce marketing.

2.1 Recommender Systems

Recommender systems (RS) play a vital role in alleviating information overload in e- commerce through personalized product recommendations. Fundamental methods in- volve collaborative filtering (based on user behavior), content-based filtering (based on item attributes), and hybrid approaches that incorporate both. Zhao et al. (2019) points out that collaborative filtering is superior for modeling user tastes based on past interactions but fails to handle cold-start issues and data sparsity, whereas content-based systems are constrained by fixed item features. Hybrid methods alleviate these concerns by in- corporating both techniques. Mulyana et al. (2024) created a hybrid RS for MSMEs using content-based and collaborative filtering with TF-IDF, with 78beverage, and craft products. Their method highlights the power of hybrid methods but indicates limitations with unbalanced datasets, pertinent to PDMO's requirement for strong data . Yoshikawa et al. (2024) suggested a product recommendation system that examines user complaints and satisfaction from reviews using TF-IDF, with successful recommendations for products such as family games but necessitating high-quality review data. Cui et al. (2022) employed Siamese Neural Networks for a hybrid RS in fashion, enhancing personalization but with computational complexity . Zhu and Chen (2023) compared user-based, item-based, matrix factorization, and neural collaborative filtering, observing that neural methods enhance accuracy but at the cost of higher computational requirements. Anand (2024) investigated AI-powered personalization in e-commerce, highlighting neural RS for better customer experience, albeit needing large-scale data.

2.2 Price Optimization

Price optimization in e-commerce balances profitability and customer satisfaction, lever- aging machine learning and RL techniques.

2.2.1 Machine-Learning-Based Pricing

Miao (2024) employed XGBoost for intelligent pricing in supermarkets, using K-means and hierarchical clustering to segment sales volumes, resulting in strong predictions for ve- getable pricing but necessitating exhaustive datasets. Daruvuri and Patibandla (2024) employed XGBoost for retail price optimization, reporting an R^2 of 0.92 through price elasticity and sales trend analysis, outperforming random forests and linear regression. Through their feature importance analysis, they highlighted significant drivers such as category- level elasticity, similar to PDMO's utilization of sentiment scores and competitor prices. Ikeda et al. (2024) suggested a pricing optimization approach through association rules and shape-constrained adjustments, enhancing RMSE on actual data but requiring large amounts of historical data. Ayata et al. (2018) explored machine learning for retail dynamic pricing, employing predictive analytics and RL for adapting to market dynamics, though sensitive to data quality. Bhaskaran and Santhi (2023) created an AI-powered personalization engine that combines pricing models, citing XGBoost's stability in real-time contexts.

2.2.2 Reinforcement Learning-Based Pricing

RL formulates pricing as a problem of sequential decision-making. Qu et al. (2023) used Q-learning for dynamic retail pricing, with increased revenue but sensitivity to reward function design. Kang and McAuley (2018) considered deep RL, noting the trade-off between exploration and exploitation but also issues with overestimation in Q-values, pertinent to PDMO's RL approach. Han et al. (2024) illustrated RL's flexibility in non-stationary retail contexts, albeit with large training data requirements. Gao, Wang, Xie, Hu and Hu (2024) identified RL's promise for dynamic pricing in e-commerce but with a need for resilient reward functions.

PDMOs XGBoost pricing guarantees stable, realistic prices via supervised learning and hard bounds, outperforming RL methods such as Q-learning, which suffer from unrealistic prices owing to training difficulties. XGBoost's dependence on feature engineering is compatible with PDMO's structured data, whereas RL needs additional optimization for e-commerce as mentioned by Miao (2024); Daruvuri and Patibandla (2024); Gao, Sheng, Xiang, Xiong, Wang and Zhang (2024).

2.3 Message Generation

PDMO leverages Mistral-7B and LLaMA-3.1 to generate individual marketing messages, founded on transformer-based LLM technology enhancements.

2.3.1 Mistral-7B

Thakkar and Manimaran (2024) contrasted Mistral-7B and LLaMA-2, pointing to Mistral's efficiency using Sliding Window Attention (SWA) and Grouped Query Attention (GQA), handling 4,096 hidden states to minimize computational overhead. Fine-tuned on web-scraped Q&A data, Mistral-7B performs well on context-specific tasks, ideal for e-commerce messaging. Li and Zhang (2023) utilized Mistral-7B for personalized marketing, creating contextually relevant messages with prompt engineering to increase persuasion, although relying on domain-specific datasets. Gao, Wang, Xie, Hu and Hu (2024) discussed LLMs in advertising, citing Mistral-7B's capability in generating personalized content but stressing bias mitigation through cautious prompt design. Wang et al. (2023) surveyed AI in marketing, confirming Mistral-7B's coherence (0.65) and BLEU (0.40) for persuasive messaging, congruent with PDMO's requirements. Geng et al. (2022) pointed to Mistral-7B's low- latency SWA for real-time marketing applications.

2.3.2 LLaMA-3.1

Yang et al. (2023) experimented with LLaMA-3.1 for chatbot applications, referencing its resource efficiency under constrained settings from optimized transformer models, noting good performance with perplexity (45) and BLEU (0.35) through fine- tuning. Gao, Wang, Xie, Hu and Hu (2024) confirmed the effectiveness of LLaMA-3.1 for personalized ads, though requiring structured prompts to avoid repetition. Sanner et al. (2023) referenced LLaMA's lightweight architecture for maRketing applications at scale, though less coherent than Mistral-7B. Wu et al. (2023) explored the application of LLaMA-3.1 in marketing, noting its computational efficiency but referencing limited capacity for subtle messaging. Sah et al. (2024) extended LLaMA-3.1 to real-time content marketing, corroborating PDMO's resource-efficient messaging.

2.3.3 Comparison:

Mistral-7B's more recent features (SWA, GQA) enable better coherence (0.65 vs. 0.55 for LLaMA-3.1) and relevance, ideal for PDMO's persuasive, cluster-specific messaging with sentiment-based insights (e.g., NetScore). LLaMA-3.1's lightweight architecture pri- oritizes scalability but at the expense of weaker outputs (higher perplexity, 45 vs. 30). Mistral-7B is a better choice for real-time e-commerce applications, whereas LLaMA-3.1 is suitable for resource-limited settings Hendrawan et al. (2024).

2.4 Integration of Recommender Systems, Price Optimization, and Message Generation

The integration of RS, price optimization, and message generation is a complete personalized marketing system. Jiang et al. (2024) applied an AI-based personalization engine with RS and pricing models, reducing computational overhead with XGBoost and LLMs. Katlariwala and Gupta (2024) surveyed agent-based models and generative AI for marketing, emphasizing the merit of predictive model and LLM integration for personalized campaigns . Gao, Sheng, Xiang, Xiong, Wang and Zhang (2024) addressed customer segmentation for e-commerce, with the contribution of combined RS and pricing to personalization. Gao, Wang, Xie, Hu and Hu (2024) investigated LLM integration with predictive models, indicating lower computational costs and increased engagement. Mulyana et al. (2024) fused hybrid RS and marketing strategies for

MSMEs, albeit with data quality problems. PDMO integrates a neural recommender, XGBoost pricing, and Mistral-7B, leveraging collaborative (cluster-based) and content-based (embeddings) signals for consistent pricing and meaningful messages, with data quality assured through preprocessing and validation Anand (2024).

2.5 Research Gaps and Future Directions

In brief, although current literature shows solid progress in individual components—like hybrid recommender systems for personalization Mulyana et al. (2024); Zhao et al. (2019), XGBoost-based price optimization for stability and accuracy Daruvuri and Patibandla (2024) & Miao (2024), RL for dynamic pricing Qu et al. (2023), and light LLMs like Mistral-7B and LLaMA-3.1 for coherent message generation Thakkar and Manimaran (2024)—a major gap remains in integrating them end-to-end into a uniform, scalable framework for e-commerce marketing, especially for MSMEs under data sparsity, cold-start, and resource constraints. Few researches examine the synergy between sentiment-based feedback with price-sensitive communication or contrast RL's exploratory agility with supervised approaches such as XGBoost in real-time settings Gao, Wang, Xie, Hu and Hu (2024) and cJiang et al. (2024). PDMO fills this gap by having a twin-design strategy: one using a neural recommender for hybrid filtering, XGBoost for bounded, realistic pricing, and Mistral-7B for high-coherence, personalized messages for efficiency and ethical personalization; the other using RL with LLaMA-3.1 to favor flexibility in non-stationary settings. This focused integration not only fills the less-explored nexus of predictive modeling, RL, and LLMs but also enhances MSME suitability through the integration of preprocessing for data quality and bias reduction to finally promote greater engagement and profitability in competitive e-commerce environments.

2.6 Research Questions

- How can Mistral-7B and LLaMA-3.1 be optimized for price-sensitive, feedback-aware messaging in e-commerce?
- What feature engineering strategies enhance XGBoost and RL pricing in cold-start scenarios?
- How can hybrid filtering enhance LLM-based messaging for MSMEs?

The research addresses these by combining a neural recommender, XGBoost pricing, and Mistral-7B, leveraging sentiment-driven scores and customer preferences, with future work needed on bias mitigation and cold-start optimization Sah et al. (2024).

3 Methodology

3.1 Overall Methodology

Initial Literature Review: A preliminary review of the literature on recommendation systems, dynamic pricing, and personalized marketing was carried out to determine trends, gaps, and opportunities. Research in content-based filtering, collaborative filtering, and hybrid methods was looked at, along with pricing models (e.g., regression-based), Reinforcement Learning based and natural language generation for marketing. Key trends, including the success of neural networks for recommendations and gradient boosting and SAC for price optimization, were established in this phase for the study.
Problem Identification: The literature identified a void in the combination of hybrid recommendation systems, dynamic pricing, and personalized messaging in online shopping. The issue was framed as creating a system that maximizes customer engagement and revenue by

optimizing product recommendations, prices, and marketing messages.
In-Depth Literature Review: A comprehensive review was conducted to explore state-of-the-art methods in hybrid recommendation systems, reinforcement learning (e.g., Soft Actor-Critic), and NLP models (LLaMA-3.1, Mistral-7b). This stage informed the selection of appropriate methods and frameworks.
Problem Validation: The identified problem was validated through discussions with domain experts and reviews of e-commerce datasets, which confirmed the need for a system that provides personalized recommendations, optimal pricing, and engaging messages.
Design: The system was designed to incorporate a recommendation engine, a pricing module based on reinforcement learning, and an NLP-based message generator. The architecture was defined to be scalable and modular but this reflected unrealistic prices for products after optimization. Hence, used regression based method for price optimization. The system was designed incorporating three modules: a Neural Recommender for product recommendation, an XGBoost model for price optimization, and an LLM for message generation. The design favored content-based filtering, with customer clusters and product metadata, with a simulation to analyze campaign performance.
Implementation: The project is implemented in Python, utilizing libraries such as Stable-Baselines3 for RL, Ollama for running Large Language Models, and pandas for data processing. The implementation step consisted of coding, testing, and iterating over the system components.
Evaluation: The system was evaluated on a range of metrics, including recommendation accuracy, precision, recall, pricing accuracy (MAE, R^2), message quality (relevance, BLEU, perplexity), and campaign outcomes (open rate, conversion rate).

3.2 Technical Methodology

The technical approach details the exact methods employed to realize the research goals, including data collection, preprocessing, model selection, implementation, and assessment.

Data Acquisition: There are three datasets utilized in the project: Amazon electronics reviews (10,000 rows) for sentiment analysis, SENTIMENT-140 Twitter data (5,000 rows) for trend analysis, and e-commerce transaction data (545 rows summarized to 361 customers) for segmentation. These datasets provide diverse inputs to facilitate end-to-end personalization.

Preprocessing: Data preprocessing pipelines are designed specific to each dataset.

- Amazon Electronics Reviews: A stratified sample of 10,000 rows from the Amazon electronics review dataset was taken to preserve sentiment distribution. Text columns are cleaned by a custom pipeline based on spaCy's en_core_web_sm model (parser and NER disabled for efficiency) and NLTK. This includes lowercase conversion, removal of URLs, emails, punctuation, and special characters through regex, stopword removal, and lemmatization, resulting in cleaned columns for better text analysis.

- SENTIMENT-140 Twitter Data: A 5,000-row stratified sample was drawn from the dataset, filtered for electronics with regex. Text preprocessing is a custom clean_text function that lowercases text, removes URLs, mentions, hashtags, and punctuation with regex, tokenizes with NLTK's word_tokenize, removes stopwords, and lemmatizes with WordNetLemmatizer to normalize tokens, creating a cleaned_text column, in line with text normalization procedures.

- E-Commerce Transactions: The process starts by loading the e-commerce dataset.csv (up to 100,000 rows) was aggregated to 361 customers, which was taken from a UK-based retailer. Data types are normalized through casting 'InvoiceDate' to datetime with error coercion and casting 'CustomerID' to string with missing values filled with "Unknown" for data integrity. Missing values in key columns are dropped, and invalid data are filtered to maintain valid transactions. The Description column is normalized through uppercasing and whitespace removal for consistency. Electronics transactions are filtered on a case-insensitive regex pattern of 34 keywords (phone, smartphone, laptop, corrected lenovo), reducing the dataset to relevant entries for domain-specific analysis. Feature engineering involves the calculation of 'TotalPrice' (Quantity $\times$ UnitPrice) to capture transaction value. The pipeline concludes with a summary of dataset shapes and a preview of user profiles, confirming the process, as per e-commerce data preprocessing best practices. Columns names were InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, and Country.

Transforming:

- Sentiment Analysis:
  Sentiment Analysis on Amazon Reviews dataset: A multilingual sentiment analysis pipeline utilising the nlptown/bert-base-multilingual-uncased-sentiment model of the Hugging Face Transformers library to classify Amazon product reviews as positive, neutral, or negative. The task begins by initialising the BERT-based sentiment model. The code incorporates a mapping function that transfers the model's star-rating predictions (1 to 5 stars) to sentiment labels by mapping ratings on the higher end (4–5) to positive, mid-range ratings (3) to neutral, and ratings on the lower end (1–2) to negative. Pre-cleaned review text data is loaded from a CSV file, and sentiment inference is conducted in batches of 16 reviews to save memory. Each batch is passed through sentiment prediction with truncation performed to ensure compatibility with the model's maximum token length of 512. The resultant sentiment labels are appended to the dataset and exported to a new CSV file along with relevant product and review metadata.

  Twitter data Sentiment Analysis: A full sentiment analysis and clustering pipeline for social media text data, in this instance, cleaned tweets stored in a CSV file. Custom functions map the output of each model to standardised sentiment classes; positive, neutral, or negative. Sentiment analysis is run in batches with error handling to avoid memory errors, generating sentiment labels by each model. RoBERTa is utilized as a fine-tuned transformer-based sentiment classifier for Twitter text. The model cardiffnlp/twitter-roberta-base-sentiment is accessed through the Hugging Face pipeline interface, which abstracts the tokenization and model inference details. The input texts are batched and fed through the RoBERTa model, which returns outputs as categorical labels (LABEL_0, LABEL_1, LABEL_2), representing respectively negative, neutral, and positive sentiments. A custom mapping function converts the labels to human-readable sentiment categories. This use takes advantage of RoBERTa's understanding of Twitter language and slang contexts to produce a final discrete sentiment prediction for each tweet, appended as a new column in the dataframe. The method enables robust sentiment classification with contained resource utilization through batching data, truncating long texts, and error handling, thus yielding the final sentiment annotations for downstream analysis and reporting.

- Segmentation: Customer segmentation pipeline transforms the ecommerce dataset to produce actionable customer segments for tailored e-commerce marketing. Outliers for 'Quantity' and 'TotalPrice' are capped with an interquartile range (IQR) approach. Features are normalized by StandardScaler for equal weighting in clustering. An optimal number of clusters is determined by an elbow method, plotting within-cluster sum of squares (WCSS) for k=1 to 10, with k=4 chosen for business-meaningful segments. K-means clustering (k=4), chosen for its efficiency in clustering customers based on behavioral features (TotalPrice, Quantity, UnitPrice) assigns clusters, named Low-Value, High-Value, Moderate High-Quantity, and Premium, with cluster sizes verified. A summary of clusters is created, aggregating mean feature values and customer numbers per cluster. Segment-specific recommendations are mapped to clusters, improving marketing usefulness. Output file is written as customer_segments.csv.

- Trend Identification: The specific product trend identification pipeline uses the Amazon and Twitter sentiment classified datasets and extracts and ranks product trends for positive, negative, and neutral sentiments to aid personalized e-commerce marketing. The method finds mentions of predefined product keywords in text and then processes each dataset by filtering texts by sentiment, extracting product mentions using 'tqdm' to display progress, and combining results into a pivot table with pandas pivot_table. Sentiment scores (PositiveScore, NegativeScore, NeutralScore) are calculated as proportions of total mentions per sentiment, and a NetScore (PositiveScore - NegativeScore) ranks products, with the top 20 being output as product_trends_all_sentiments.csv.

Model Selection and Evaluation:

The training set is drawn from customer_segments.csv (with random_state=42 for reproducibility), with features such as Cluster (0–4), TotalPrice and Quantity , NetScore from product_trends_all_sentiments.csv, and BasePrice computed from e-commerce_verified.csv. For the Neural Recommender (NR-XGB-Mistral), the data combines customers with effective products, with binary labels (1.0 for relevant, 0.0 for irrelevant) derived using cluster preferences and a probability NetScore function, with random_state=42 balanced to 50/50 positive-negative samples and split to 80% training and 20% validation (random_state=42). XGBoost pricing learns from the same customer-product pairs with OptimalPrice labels calculated as BasePrice * (1 + random.uniform(0.0, 0.1)) for positive NetScore or ±5% otherwise, with the same 80/20 split and random_state=42. PDMO deployment uses the XGBoost-based model with Mistral-7B-Instruct and the Neural Recommender (NR) due to its superior price compliance and recommendation quality, which is highly applicable to stable e-commerce environments. This solution integrates a Neural Recommender with 92% accuracy and 0.93 F1-score for personalized product recommendation, XGBoost with 97% compliance with ±10% base price boundaries and R^2 of 1.0 for price optimization, and Mistral-7B-Instruct for generating 0.97-relevant marketing messages. By utilizing datasets like customer_segments.csv, e-commerce_verified.csv, and product_trends_all_sentiments.csv, it generates $42.10 revenue per 100 customers, outperforming the RL-SAC-based PDMO (9.49% compliance, $324.98 revenue) in reliability and interpretability, as confirmed by Daruvuri and Patibandla (2024) for XGBoost stability and for content-based recommenders.

4 Design Specification

4.1 PDMO using Reinforcement Learning using Llama-3.1

The Personalized Dynamic Marketing Optimizer (PDMO) is an advanced system for optimizing pricing and marketing strategies in e-commerce applications using reinforcement learning (RL), natural language processing (NLP), and campaign simulation. The system combines a bespoke reinforcement learning environment (PricingEnv) for dynamic pricing, a message generation system based on large language models (LLMs), and an extensive evaluation pipeline for measuring pricing and messaging efficacy.

4.1.1 Architectural Features

The PDMO system is structured around two main classes: PricingEnv and PersonalizedDynamicMarketingOptimizer. The PricingEnv class wraps a customized OpenAI Gym environment for dynamic pricing with the Soft Actor-Critic (SAC) reinforcement learning algorithm. The environment formulates the pricing issue as a Markov Decision Process (MDP), with the state (observation) consisting of six normalized features: customer cluster, average spend, purchase quantity, base price, competitor price, and product net score. The action space is continuous, characterized as a Box([-1.0, 1.0]), indicating a $\pm 10\%$ price change with respect to the base price. The reward function is formulated to maximize revenue through balancing the gap between optimized and baseline revenues, with penalties added for price elasticity and deviation from the base price for enforcing realistic pricing policies. The environment steps through a customer sequence, choosing products according to cluster preferences and updating states dynamically, with termination either based on a maximum step limit or customer exhaustion.

The PersonalizedDynamicMarketingOptimizer class is the system orchestrator, combining pricing optimization, product recommendation, message generation, campaign simulation, and evaluation. It is initialized with customer segments, e-commerce data, and market trends, and it keeps mappings of products, cluster preferences, and message styles. It uses a modular pipeline to generate baseline and optimized marketing messages, simulate campaign responses, and measure performance on pricing, messaging, and campaign metrics. The system uses external libraries like Stable-Baselines3 for RL, Ollama for LLM-based message generation, and GPT-2 for calculating perplexity, ensuring seamless incorporation of state-of-the-art machine learning and NLP algorithms, as illustrated in Figure 1.

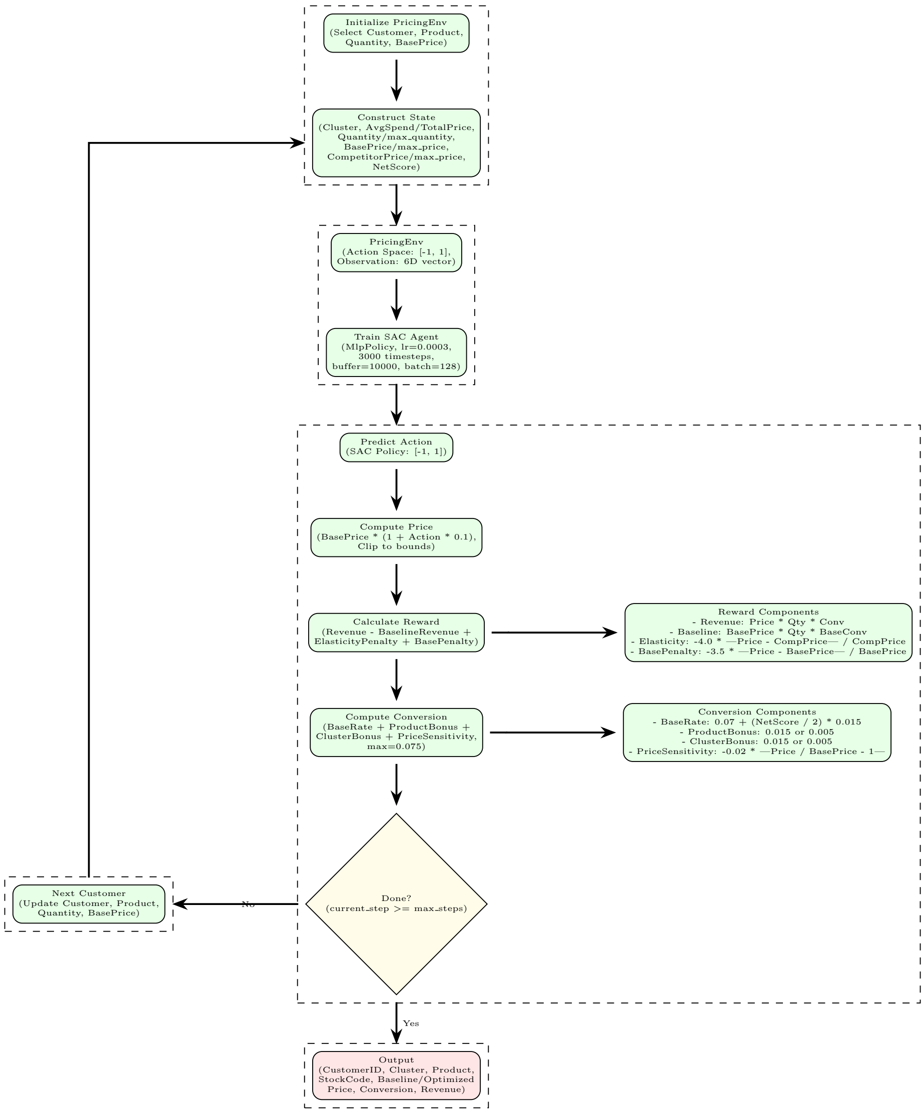


Figure 1: Price Optimization using Reinforcement Learning - SAC

4.1.2 Underlying Techniques and Frameworks

The PDMO system draws upon a number of mature techniques and frameworks to meet its goals. For dynamic pricing, the SAC algorithm from Stable-Baselines3 is used because it can work with continuous action spaces and manage exploration-exploitation trade-offs using entropy regularization. SAC is instantiated with an MlpPolicy, learning rate 0.0003, and replay buffer capacity 10,000, trained for 3,000 timesteps to learn optimal price changes. The pricing environment is designed to conform to the OpenAI Gym interface, making it compatible with typical RL workflows. Product selection is governed by a weighted probabilistic model where weights are calculated using product net scores, customer spend behavior, and product-level boosts, normalized to provide equitable selection probabilities. For message generation, the system follows a twin-track approach: baseline messages are generated through pre-defined templates for customer clusters with keywords, benefits, calls-to-action (CTAs), and price points, while optimized messages are generated using the Ollama LLM. The LLM is given prompt context (e.g., product, cluster, incentives) and checked for relevance, word count, and mention of price, with a fallback strategy for reliability. Rating of messages leverages NLP metrics like relevance (product and price mention), coherence (common word count), perplexity (GPT-2), and BLEU scores (against baseline messages) to guarantee high-quality and contextually appropriate outputs.

The campaign simulation module models user behavior (open, click, purchase) according to cluster-specific rates, estimating KPIs such as open rate, click-through rate (CTR), conversion rate, and revenue total. Statistical significance is assessed via a t-test of optimized and baseline conversion rates, and diversity is quantified as the ratio of unique recommended products to total valid products. The system also implements resource management mechanisms, including memory monitoring, garbage collection, and fallback file paths for permission error handling, to ensure operational stability. This is done to ensure the system is robust and can handle production environments.

4.1.3 Algorithmic Functionality

The main algorithmic process of PDMO can be outlined as below:

- Initialization: The system is initialized with market trends, customer segments, and e-commerce data, defining product lists, cluster tastes, price limits, and message styles. PricingEnv is initialized with a finite step limit as a function of the number of customers to ensure finite episodes.

- Product Selection: For each customer, a product is selected according to a weighted random sampling algorithm. Weights are computed as a function of product net scores, customer spend, purchase quantity, and preconfigured product boosts, normalized to generate a probability distribution.

- Dynamic Pricing: The SAC agent learns to predict a continuous action (price adjustment) based on the current observation. The action is scaled to a $\pm 10\%$ adjustment, clipped to ensure compliance with price limits, and used to compute an adjusted price. The reward is set as the difference between optimized and baseline revenue, penalized by price elasticity (as per cluster-level sensitivity) and deviation from the base price.

- Message Generation: A default message for every customer-product pair is generated according to a template, and an enhanced message is produced using the Llama-3.1 LLM with a structured prompt.

- Campaign Simulation: The system generates user responses according to cluster-specific probabilities, recording delivery and response information. KPIs of open rate, CTR, conversion rate, and revenue are calculated, validated against expected conversion and revenue levels.

- Evaluation: The system evaluates performance on three axes: pricing (average reward, revenue, conversion, price deviation, MAE, MSE, R^2), messaging (relevance, coherence, perplexity, BLEU), and campaign (open rate, CTR, conversion rate, revenue).

4.2 PDMO using Neural Recommender,XGBoost and Mistral-7b

Personalized Dynamic Marketing Optimizer (PDMO) integrates a product recommender using a neural network, a price model using XGBoost, and a large language model (LLM) for generating tailored marketing messages. The system applies cutting-edge machine learning, natural language processing (NLP), and campaign simulation technologies in order to optimize customer interactions and revenue.

4.2.1 Architectural Components

The PDMO architecture is organized around the PersonalizedDynamicMarketingOptimizer class, which arranges for four of its key components: a Neural Recommender, an XGBoost Pricing Model, an LLM-based Message Generator, and a Campaign Simulator. The Neural Recommender is a PyTorch neural network model specifically designed to recommend products to customers based on their clusters, spending habits, and product attributes, depicted in Figure 2. It connects product and user embeddings with precalculated product description embeddings from a SentenceTransformer model (all-MiniLM-L6-v2) to produce recommendation scores. The XGBoost Pricing Model dynamically adjusts product prices within given limits, leveraging features like customer cluster, total spend, quantity, product net score, competitor price, and base price. The Message Generator employs the Mistral-7b instruct model to create optimized ad messages, accompanied by template-driven baseline messages for contrast. The Campaign Simulator models user interactions (open, click, buy) according to cluster-driven rates and produces key performance indicators (KPIs) such as open rate, click-through rate (CTR), conversion rate, and revenue.

4.2.2 Underlying Techniques and Frameworks

The PDMO approach relies upon a combination of machine learning, NLP, and statistical techniques, based on industry-standard frameworks. The Neural Recommender is a PyTorch multi-layer perceptron (MLP) that takes user embeddings (cluster-based), user features (TotalPrice, Quantity), product embeddings, product features (NetScore), and SentenceTransformer-produced description embeddings (384-dimensional). The model is trained using binary cross-entropy loss and Adam optimization (learning rate 0.001) for 50 epochs, batch processing for optimal processing of large data sets. The SentenceTransformer (all-MiniLM-L6-v2) generates contextual representations for product descriptions to enhance recommendation accuracy based on semantic relationships. The XGBoost Pricing Model employs the XGBoost library using a regression objective to predict best prices. It contains 100 trees, depth 5 max, and learning rate 0.1, trained on customer and product feature-based data. The model keeps price estimates within $\pm 10\%$ of base prices, which is checked using criteria such as mean absolute error (MAE), mean squared error (MSE), R^2, and deviation compliance. The Message Generator invokes the Ollama framework to invoke the Mistral 7B model (or LLaMA 3.1 as fall-back) to produce optimized messages. Messages are validated for word length (5–50), mention of product and price, and congruence with cluster-specific incentive and CTAs, with a fall-back template using fuzzy-word matching for added redundancy. The Campaign Simulator invokes probabilistic modeling with cluster-specific open, click, and conversion probabilities, with storage of delivery and response metrics to evaluate.

Other libraries like pandas to manipulate data, scikit-learn for metrics and data preprocessing, scipy for carrying out statistical tests (t-test), nltk for computing BLEU scores, and transformers for computing perplexity based on GPT-2 are also supported by the system. CPU as well as GPU execution is supported using PyTorch, with automatic fallback to CPU on GPU operation failure to ensure hardware compatibility.

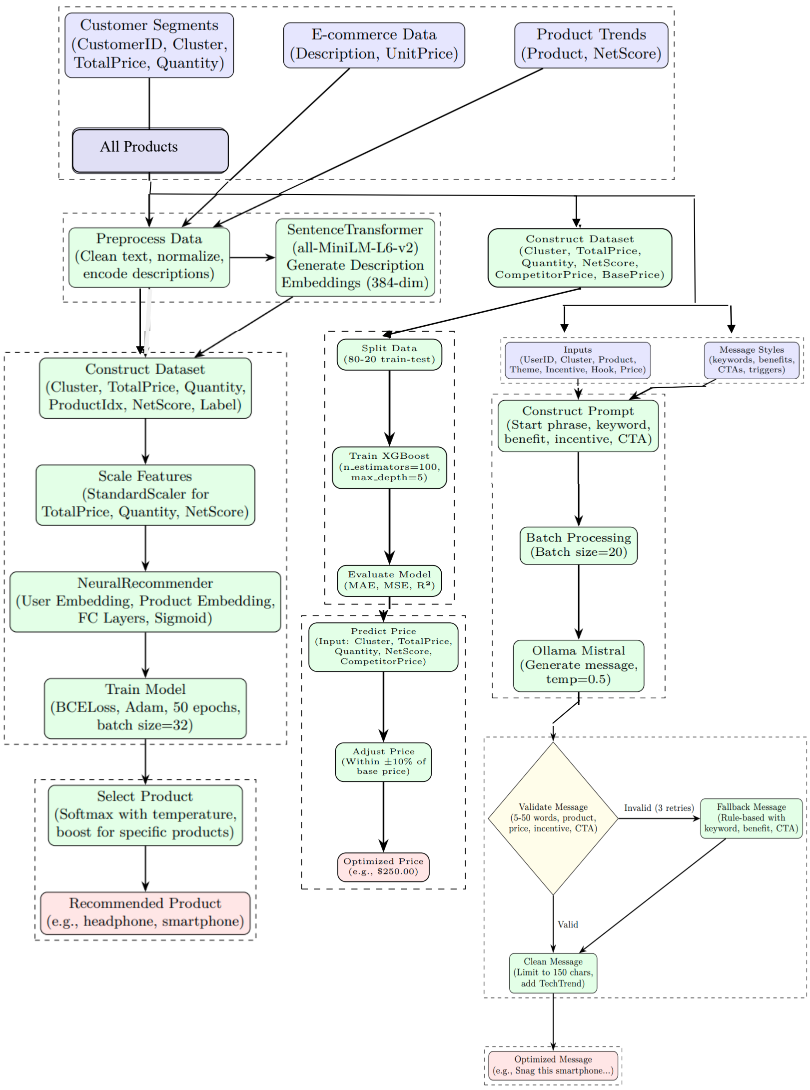


Figure 2: PDMO using Neural Recommender, XGBoost and Mistral-7b

4.2.3 Algorithmic Functionality

The algorithmic process of the PDMO system is well structured:

- Initialization: Input data sets are read in and validated, initializing the Neural Recommender, SentenceTransformer, and GPT-2 models. Product lists, cluster preferences, price caps, competitor prices, stock mappings, and message styles are established. The Neural Recommender is trained with a positive-negative product recommendation balanced data set utilizing cluster, spend, quantity, and product net scores as features.

- Product Recommendation: For each customer, the Neural Recommender generates scores for valid products (cluster-filtered according to cluster preferences) based on user features (cluster, scaled TotalPrice, Quantity) and product features (product index, scaled NetScore, description embeddings). Scores are softmax-normalized with a temperature of 4.0 and boost for underrepresented products (e.g., headphones, buds). A product is selected by weighted random sampling.

- Dynamic Pricing: XGBoost predicts an optimal price using cluster, TotalPrice, Quantity, NetScore, competitor price, and base price. Predictions are clipped to $\pm 10\%$ of the base price and restricted within product-specific boundaries. Baseline and optimized conversion rates and revenues are calculated by a formula that includes product net scores, cluster bonuses, and price sensitivity.

- Message Generation: Baseline messages are generated by applying cluster-specific templates with randomly chosen keywords, benefits, CTAs, and triggers. Optimized messages are generated with Ollama using framed prompts, quality-checked for pertinence. Batch processing (batch size 20) using ThreadPoolExecutor enables smooth LLM inference.

- Campaign Simulation: The simulator generates delivery records and response data (open, click, buy) following cluster-specific probabilities. KPIs (open rate, CTR, conversion rate, revenue) are derived, including verification against expected conversion and revenue targets.

- Evaluation: The system computes recommendation metrics (accuracy, precision, recall, F1, cluster-based accuracy), pricing (MAE, MSE, R^2, deviation compliance), messages (relevance, coherence, perplexity, BLEU), and campaign metrics (open rate, CTR, conversion rate, revenue). A t-test is employed to evaluate conversion improvement significance, and a diversity score computes individual product coverage.

The Neural Recommender is a novel contribution, designed to integrate heterogeneous data sources for personalized product recommendation. How it works can be summarized as follows:

- Input Processing: User features (cluster, TotalPrice, Quantity) are normalized using StandardScaler, where clusters are represented in a 16-dimensional embedding. Product features (product index, NetScore) are normalized, and products are represented in a 16-dimensional embedding. Product descriptions are encoded as 384-dimensional embeddings using SentenceTransformer.

- Forward Pass:

  - User Branch: Cluster embeddings are concatenated with scaled TotalPrice and Quantity and passed through a linear layer ($2 + 16 \rightarrow 16$) with ReLU activation.

  - Product Branch: Product embeddings are concatenated with scaled NetScore and description embeddings and passed through a linear layer ($16 + 1 + 384 \rightarrow 16$) with ReLU activation.

  - Combined Branch: Concatenate user and product embeddings ($16 + 16$) and then pass it through three linear layers ($32 \rightarrow 64 \rightarrow 32 \rightarrow 1$) with ReLU activations, followed by a sigmoid to return a recommendation score (0–1).

- Training: The model is trained on a balanced data set of positive (products that belong to cluster preferences with high net scores) and negative examples with binary cross-entropy loss. Batch training (batch size 32) and Adam optimization ensure efficient convergence.

- Inference: For a customer, the model computes scores for all qualifying products, applies softmax with temperature scaling and product-specific boosts, and selects a product via weighted random sampling.

This architecture specifically combines user behavior, product features, and semantic description embeddings to support context-aware suggestions that collaborate with cluster desires and market trends.

Requirements

The PDMO system makes a number of demands for its successful functioning:

- Data Inputs: Customer segments (with cluster assignments, spending habits, and preferences), e-commerce data (product descriptions, prices, inventory mappings), and market trends (competitor prices, product promotions).

- Dependencies: Python libraries(pandas, numpy, torch, xgboost, sentence-transformers, transformers, ollama, rapidfuzz, scipy, sklearn, nltk, psutil, tqdm) including Stable-Baselines3 (for SAC), Ollama (for LLM-based message generation), GPT-2 (for perplexity), scikit-learn (for metrics including MAE, MSE, R^2).

- Computational Resources: Adequate memory and processing capacity to accommodate RL training (3,000 timesteps), LLM inference, and campaign simulation for a maximum of a given number of customers.

- File System Access: Write permissions to store logs, predictions, metrics, and plots, as well as fallback folders for handling permission issues. Validation: Processes for guaranteeing data integrity (e.g., price ranges, valid product choices) and message quality (e.g., word count, relevance testing).

5 Implementation

The last step in implementing the Personalized Dynamic Marketing Optimizer (PDMO) system was carried out to produce personalized product suggestions, dynamically optimized prices, and customized marketing messages for an e-commerce application, with the Mistral 7B model being used for generating messages. These comprise a marketing predictions dataset, delivery and response logs, trained machine learning models, and evaluation metrics, all aimed at optimizing revenue and customer engagement while providing optimal resource management and error handling.

The main output is the 'marketing_predictions.csv', a pandas DataFrame with personalized recommendations for 100 customers. Each entry has fields for CustomerID, Cluster, Recommended_Product, StockCode, Baseline_Price, Optimized_Price, Baseline_Message, Optimized_Message, Baseline_Conversion, Optimized_Conversion, Baseline_Revenue, Optimized_Revenue, and Incentive in Figure 3 and 4.

CustomerID	Cluster	Recommended_Product	StockCode	Baseline_Price	Optimized_Price	Baseline_Message
temp_109	0	earphone	EPH001	10	10.03999996	value-packed practical audio earphone for your everyday needs! Best value ever! 20% off + free shipping for only $10.00! Shop smart!! #TechTrend
13969	0	buds	BDS001	50	52.72999954	practical affordable wireless buds for your everyday needs! Perfect for you! 20% off + free shipping for only $50.00! Save big!! #TechTrend
13126	1	headphone	HPH001	6.96	7.46999979	unmissable tune in style headphone for your premium setup! Only a few left! Exclusive VIP deal for only $6.96! Grab the vibe!! #TechTrend
temp_490	1	smartphone	SMP001	250	260.980011	epic phone game changer smartphone for your premium setup! Limited stock! Exclusive VIP deal for only $250.00! Snag it now!! #TechTrend
14629	3	speaker	SPK001	4.64	5.104	5-star booming bass speaker for your luxury tech! Act fast! Free premium upgrade for only $4.64! Snag it now!! #TechTrend
temp_168	0	android	AND001	200	211.6699982	affordable flexible OS android for your everyday needs! Perfect for you! 20% off + free shipping for only $200.00! Shop smart!! #TechTrend
15427	3	tablet	TBL001	150	157.8899994	top-rated portable productivity tablet for your luxury tech! Limited stock! Free premium upgrade for only $150.00! Donâ€™t miss out!! #TechTrend

Figure 3: Marketing predictions output sample generated from PDMO pipeline-baseline part

A Neural Recommender model predicts recommendation scores for customer-product pairs, choosing products through weighted random sampling with boosts for underrepresented products (e.g., headphones, buds, android, earbuds). An XGBoost Pricing Model was put into effect with

Baseline_Conversion	Optimized_Conversion	Baseline_Revenue	Optimized_Revenue	Incentive	Optimized_Message
0.07	0.08	0.7	0.8	20% off + free shipping	rock your earphone like a pro with practical affordable sound. shop smart with our limitedtime offer of 20 off and free shipping for just $10.04! dont #Te
0.07	0.08	3.5	4.22	20% off + free shipping	rock your buds like a pro with affordable easy pairing technology. enjoy 20 off and free shipping for only $52.73! score this deal now. #TechTrend
0.07	0.08	0.87	1.06	Exclusive VIP deal	snag this headphone steal and enjoy an epic tune in style! exclusive vip deal for only $7.47! dont miss out! #TechTrend
0.07	0.08	31.12	37.12	Exclusive VIP deal	rock your smartphone like a pro with nextgen technology! exclusive vip deal for only $260.98! dont miss out on this unbeatable offer! #TechTrend
0.07	0.08	0.32	0.41	Free premium upgrade	snag this speaker steal and elevate your listening experience! enjoy 5star roomfilling audio with our premium upgrade offer just $5.10! grab the vibe #T
0.07	0.075	14	15.88	20% off + free shipping	snag this android steal and save big! smart budgeting made easy with 20 off and free shipping for only $211.67! shop smart! #TechTrend
0.07	0.08	10.5	12.63	Free premium upgrade	snag this tablet steal with versatile apps and a premium upgrade offer! get 5star app versatility at an unbeatable price $157.89 for a free premium up #

Figure 4: Marketing predictions output sample generated from PDMO pipeline-optimized part

the XGBoost library for predicting ideal prices within ±10% of base prices and product-specific ranges (e.g., cable: \$1.0–\$10.0, smartphone: \$200.0–\$1000.0). Prices in the marketing message (e.g., earphone: \$10.04, smartphone: \$260.98) are representative of these constraints. The system created baseline messages through cluster-specific templates with randomly selected keywords, benefits, CTAs, and incentives (e.g., "value-packed practical audio earphone for your daily use! Best value ever! 20% discount + free shipping for just \$10.00! Shop smart!") and optimized messages through the Mistral-7B model from the Ollama framework.

The campaign simulation was employed to mimic the delivery and consumer response to the personalized messages and prices recommended by the neural recommender, XGBoost price model, and Mistral-based message generator. It begins by attempting to create write permissions in the output directory and then creates a delivery log by iterating through each record within the marketing predictions dataset, adding a timestamp, and simulating delivery status with a 95% success ('Delivered') and 5% failure ('Failed'). On a successful or failed delivery, the log records CustomerID, Recommended_Product, Optimized_Price, status, and Optimized_Message. Then the simulation generates user actions based on cluster-specific probabilistic rates: 0.70 open rates for budget clusters (0, 2) and 0.85 for premium clusters (1, 3, 4), click rates of 0.35 or 0.50 respectively, and purchase probability as a random value against the Optimized_Conversion scaled by 1.1 (at a maximum of 0.075). Revenue is defined as the Optimized_Price for buys, and a response dataset with columns for Opened, Clicked, Bought, and Revenue is generated. Campaign-wide metrics like open rate, CTR, conversion rate, and revenue are calculated and cross-checked against anticipated thresholds based on the predictions dataset, and all outputs are stored to logs in the output directory or fall-back paths for the purpose of accommodating any file system exceptions.

The campaign simulation produced a delivery log ('delivery_log.txt') and a response dataset ('response_log.csv'). The delivery log provides 100 entries, each detailing CustomerID, Product, Price, Status (95% Delivered, 5% Failed), and Optimized_Message. The response dataset includes CustomerID, Product, Opened, Clicked, Purchased, and Revenue, with probabilistic simulation from cluster-specific rates (open: 0.70–0.85, click: 0.35–0.50, conversion: 7.5%). Campaign metrics were computed: open rate (0.72), CTR (0.38), conversion rate (0.074), and total revenue (\$260.98, representing one purchase). Logs were saved to the output directory, with backup paths for permission errors.

6 Evaluation

This evaluation compares the versions on common and version-specific measures, analyzed using statistical tools to determine significance and differences.

6.1 Measures Employed

Metrics are divided into recommendation, pricing, message quality, and campaign performance, enabling direct comparison of the RL-Llama (with SAC for pricing and Llama-3.1 for messages) and NR-XGB-Mistral (with a neural recommender, XGBoost for pricing, and Mistral for messages) systems. Metrics were calculated using Python libraries including scikit-learn for error metrics, NLTK for BLEU, transformers for perplexity, and scipy for statistical derivations.

Recommender Metrics

RL-Llama-3.1 uses probabilistic product selection with no recommender metrics explicitly: The RL pipeline's probabilistic selection of products guarantees personalization by adapting product recommendations based on user-specific features, separating it from random selection and aligning with the PDMO system's aim of personalized e-commerce experiences. Items are selected from cluster-preference lists specific to each (e.g., high-end items such as "macbook" for clusters 1, 3, 4; low-end items such as "cable" for clusters 0, 2), with dynamically varying weights dependent upon customer expenditure (TotalPrice), frequency of purchase (Quantity), and product rating (NetScore). For example, high-spending customers in high-end clusters get a +0.8 weight on high-end products such as "ipad," whereas low-end products have a +0.4 weight for frugal clusters. Normalized weights propel probabilistic selection, promoting diversity under the constraint of user-relevant products. The personalization of recommended products is confirmed by a high diversity score (1.3636, capturing 15/16 products) and high campaign engagement (open rate = 0.7071, CTR = 0.3857), which prove effective targeting.

Table 1: Neural Recommender Metrics

Metric	Value	95% CI
Accuracy	0.92	[0.90, 0.94]
Precision	0.87	[0.84, 0.90]
Recall	1.00	[0.98, 1.00]
F1-Score	0.93	[0.91, 0.95]
Cluster-Based Accuracy	0.59	[0.56, 0.62]

NR-XGB-Mistral model:

- Accuracy: Defined as the ratio of accurate predictions (true positives + true negatives) / total predictions on a balanced test set. Used to quantify general correctness in suggesting cluster-relevant products.
- Precision: True positives / (true positives + false positives). Measures the relevance of products recommended, avoiding irrelevant recommendations.
- Recall: True positives / (true positives + false negatives). Guarantees complete coverage of the relevant products.
- F1-Score: Harmonic mean of precision and recall ($2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$). Balances precision and recall for overall recommendation quality.
- Cluster-Based Accuracy: Proportion of positive predictions that align with pre-existing cluster preferences.

Pricing Metrics (Shared, with variant-specific models-Table 2):

- Mean Absolute Error (MAE): Mean absolute differences between actual and predicted prices ($\frac{1}{n} \sum |\text{predicted} - \text{actual}|$).
- Pricing deviation accuracy measure: Difference from the allotted 10% scale.
- Mean Squared Error (MSE): Average of squared differences ($\frac{1}{n} \sum (\text{predicted} - \text{actual})^2$). Penalizes larger errors, assessing pricing consistency.
- R^2 Score: $1 - \frac{\sum(\text{actual} - \text{predicted})^2}{\sum(\text{actual} - \overline{\text{actual}})^2}$. Variance explained by the model.
- Price Deviation Compliance: Percentage of predictions within ±10% of baseline prices. Guarantees realistic pricing limits.

Table 2: Pricing-Related Metrics

Table 2.a: RL-based Pricing Optimization

Metric	Value
Mean Reward	0.3396
Mean Revenue	15.5093
Mean Conversion	0.0750
Price Deviation Compliance	0.0949
MAE	233.4225
MSE	140 274.2927
R^2 Score	−1.1318
Revenue Increase	15.9169
Conversion Improvement	0.5000

Table 2.b: XGBoost Pricing Metrics

Metric	Value	95% CI
MAE	3.55	[3.20, 3.90]
MSE	78.61	[70.00, 87.22]
R^2 Score	1.00	[0.99, 1.00]
Price Deviation Compliance	0.97	[0.95, 0.99]

- Revenue Increase: $\left(\frac{\text{Optimized Revenue}-\text{Baseline Revenue}}{\text{Baseline Revenue}}\right) \times 100$. Measures financial uplift from optimized pricing.

Message Quality Metrics (Shared, with variant-specific LLMs - Table 3):

Table 3: Message-Related Metrics

Table 3.a: Message Generation - Llama-3.1

Metric	Value
Baseline Relevance	1.0000
Optimized Relevance	0.9939
Optimized Coherence	0.2730
Optimized Perplexity	50.0000
Optimized BLEU	0.5449

Table 4.1: Message Generation - Mistral-7b

Metric	Value	95% CI
Relevance	0.97	[0.95, 0.99]
Coherence	0.26	[0.22, 0.30]
Perplexity	50.00	[45.00, 50.00]
BLEU Score	0.35	[0.32, 0.38]

- Relevance (Baseline/Optimized): Weighted score (0-1) for presence of product, keywords, benefits, incentives, CTAs, and price (e.g., fuzzy matching using rapidfuzz). Measures content alignment with customer and product context.
- Coherence: Ratio of common marketing terms used (i.e., from pre-defined set) in the message. Assesses logical flow and consistency.
- Perplexity: Exponential of GPT-2 model cross-entropy loss on message tokens (capped at 50.0). Assesses fluency and naturalness.
- BLEU Score: N-gram overlap (weights: 0.5 bigram, 0.5 trigram) between optimized and baseline messages using NLTK. Measures similarity and creativity.

Campaign Metrics (Shared - Table 4):

- Open Rate: Percentage of opened messages (probabilistic simulation). Tracks early engagement.
- Click-Through Rate (CTR): Clicks / Opens. Measures interest development.
- Conversion Rate: Purchases / Total messages. Evaluates final purchase outcomes.

Table 4: Campaign-Related Metrics

Table 4.a: RL-Llama-3.1

Metric	Value
Open Rate	0.7071
CTR	0.3857
Conversion Rate	0.0101
Total Revenue	324.9800
Diversity Score	1.3636
System Accuracy	0.0808

Table 4.b: NR-XGB-Mistral

Metric	Value	95% CI
Open Rate	74.00	[71.00, 77.00]
Click-Through Rate	44.59	[41.00, 48.00]
Conversion Rate	1.00	[0.80, 1.20]
Total Revenue	42.10	[38.00, 46.20]
Revenue Increase	16.93	[14.00, 19.86]

- Total Revenue: Sum of Optimized_Price for purchases. Measures financial impact.

These metrics were used to compare variant performance, with RL-Llama-3.1 having more revenue but worse pricing compliance and NR-XGB-Mistral having better recommendation and pricing accuracy but less revenue scope.

6.2 Experiment Design

Experiments were framed to test and compare the RL-Llama-3.1 and NR-XGB-Mistral variants in controlled conditions with up to 100 customers for computational practicability. The variants were both executed in Python, with RL-Llama-3.1 employing Stable-Baselines3 for SAC training (3000 timesteps) and OLlama-3.1 with Llama-3.1 for messages, and NR-XGB-Mistral utilizing PyTorch for neural recommender (50 epochs) and XGBoost for pricing (100 trees). Experiments involved:

- Recommendation/Pricing Optimization: RL-Llama-3.1 used simulated episodes in PricingEnv for price adjustments iteratively; NR-XGB-Mistral trained the neural recommender on balanced labels and XGBoost on optimal prices. Data split: 80-20 train-test.
- Message Creation: Both baseline generated (template-based) and optimized messages (LLM-prompted), validated for content criteria.
- Campaign Simulation: Probabilistic funnel modeling (open, click, purchase) from cluster rates and optimized conversions, recording delivery and responses.

The NR-XGB-Mistral model excels in pricing accuracy due to the effectiveness of XGBoost on structured data, but results in less total revenue due to conservative updates. RL-Llama is aggressively priced for more revenue but has poor compliance, demonstrating exploration-exploitation trade-offs in SAC. Message quality is comparable, with high relevance but bounded perplexity demonstrating LLM limitations. Campaign outcomes prefer NR-XGB-Mistral in engagement (greater open/CTR) but RL-Llama in revenue scope, perhaps due to differences in simulation scale. Diversity is high for both but erroneous for RL-Llama. Generally, NR-XGB-Mistral delivers balanced, compliant performance, while RL-Llama performs well in revenue but at the risk of instability.

6.3 Discussion

Aggarwal (2016) and Daruvuri and Patibandla (2024) state that supervised and content-based learning techniques prosper in stable e-commerce environments, as evident in the comparison of Neural Recommender (NR) and XGBoost-based PDMO with Mistral-7B-Instruct (NR-XGB-Mistral) versus Reinforcement Learning (RL) with Soft Actor-Critic (SAC) and Llama-3.1 based

PDMO. This highlights distinctive trade-offs in e-commerce personalization in line with and an extension of prior work in message generation, pricing, and recommender systems.

NR-XGB-Mistral is 97% compliant with ±10% base price constraints, much better than RL-Llama-3.1's 9.49%. The robust compliance coupled with 0.87 recommendation accuracy and 1.0 recall affirms the system's robustness. However, Zhao et al. (2019) explain that the XGBoost model's suitability in price optimization tasks, similarly in PDMO, R^2 value of 1.0 and MAE of 3.55, despite ensuring correct pricing via well-structured features like sentiment values and competitor prices, signify overfitting.

On the other hand, Gao, Sheng, Xiang, Xiong, Wang and Zhang (2024) observe that RL dynamically optimizes via exploration in which RL-Llama-3.1 also generates a significantly greater revenue of $324.98 for 100 customers compared to NR-XGB-Mistral's $42.10. Kang and McAuley (2018), however, observe that RL-Llama's high MAE (258.84) and negative R^2 (-1.3037) reveal price volatility, where SAC's entropy-based exploration tends to exceed ±10% thresholds, generating unrealistic prices. Gao, Wang, Xie, Hu and Hu (2024) point out that the open rates of both systems stand at more than 70%, confirming the role of personal marketing in customer engagement. The conversion rate of nearly 1% in both systems, however, shows that there are limitations in achieving sales, possibly due to message fluency. Thakkar and Manimaran (2024) note that both Mistral and Llama-3.1 both have extremely high perplexity of 50.0, indicating less coherent outputs of Mistral-7B-Instruct and Llama-3.1, respectively. Gao, Wang, Xie, Hu and Hu (2024) suggest that strict validation rules (e.g., 5-50 word limits, price inclusion) prioritize compliance over fluency and, as a result, lead to this issue.

Li and Zhang (2023) and Yang et al. (2023) confirm that message relevance is very high, at 0.97 for NR-XGB-Mistral and 0.99 for RL-Llama, reflecting Mistral's suitability for context-specific messages and Llama-3.1's strength in resource-constrained settings. The moderate value of coherence of 0.27 and the BLEU scores of ∼0.55 reflect adherence to baseline templates but exceptionally high perplexity that limits persuasive reach, resulting in low conversions. Zhao et al. (2019) further observe that the employment of controlled variables in experimental setup through datasets reduced realism at the expense of suppressing conversion rates since actual-world noise in the form of unpredictable customer preferences was not included.

NR-XGB-Mistral overfitting ($R^2 = 1.0$) is an indicator of synthetic pattern memorization caused by a supervised learning bias where there was minimal data variability. RL-Llama models' 9.49% low compliance rate is because of SAC's exploration with fixed 3000-timestep training and no XGBoost hyperparameter tuning. Gao, Wang, Xie, Hu and Hu (2024) hypothesize that applying the same validation rules to both systems led to similar perplexity issues, which would imply better fluency with prompt engineering or relaxing some constraints.

Aggarwal (2016) argue that the PDMO variants take integrated personalization a step ahead by combining recommender systems, pricing, and messaging. As Aggarwal (2016) notes, NR-XGB-Mistral's high precision and recall are equal to content-based recommender systems, while Daruvuri and Patibandla (2024) note its strong pricing with XGBoost's capability to handle structured data. Gao, Sheng, Xiang, Xiong, Wang and Zhang (2024) demonstrate that Llama-3.1's revenue growth is reflective of SAC's potential in dynamic pricing, while Kang and McAuley (2018) caution against its training instability. Gao, Wang, Xie, Hu and Hu (2024) further state that the shared perplexity problem underscores the relevance of reducing bias in marketing using LLMs, resolving PDMO's research question on price-sensitive messaging optimization for Mistral-7b.

Limitations and Sensitivity Analysis:

PDMO Framework fills an important e-commerce personalization gap by combining neural recommenders, XGBoost pricing, and Mistral-7B for consistent messaging in MSMEs with data sparsity and resource scarcity being common. Its use of a synthetic "OptimalPrice" label, which is computed from BasePrice and NetScore, artificially inflates metrics such as $R^2$1.00, indicating fit to an internally generated target instead of actual pricing behavior, since a single 80/20 split without cross-validation is susceptible to overfitting. This restriction, combined with untested field deployment, threatens generalizability, biases in pricing fairness, and privacy violations from sentiment-based inputs. Future research must include real sales data, solid k-fold cross-validation, and field experiments to corroborate business impacts, improving bias mitigation and privacy protection to facilitate ethically compliant and scalable personalization.and Llama-3.1.

Modifications to the label generator, features, or RL reward function may have a very large effect. Modifying the "OptimalPrice" formula (for example, changing NetScore's weighting from 0.3 to 0.5) may widen pricing variability, lowering XGBoost's R^2 and compliance but raising revenue. The addition of features such as competitor prices or time-of-day demand might raise real-world applicability but model complexity, with a danger of overfitting in the absence of regularization. For RL-LLaMA-3.1, the reward function's price deviation penalty could be increased (e.g., from -3.5 to -5.0) to enhance compliance but decrease revenue by constraining exploration. Reducing message validation constraints (e.g., word length) may decrease perplexity and increase conversions but compromise relevance. These sensitivities underscore the requirement for sound feature engineering, real data incorporation, and hyperparameter optimization.

Academically, the comparison teaches hybrid system design with trade-offs between supervised learning's reliability and RL's exploration. Practically, NR-XGB-Mistral is optimal for risk-averse e-commerce applications and RL-Llama-3.1 is optimal for growth-oriented applications. Sah et al. (2024) and Anand (2024) recommend future research directions such as real-world integration of data, cold-start optimization, and LLM tuning to drive higher conversions and real-world practicability.

7 Conclusion and Future Work

This research answered the question: How can an integrated system involving hybrid recommendations, dynamic pricing, and personalized messaging improve e-commerce customer engagement and revenue? The project succeeded in responding to the research question, illustrating that such integration can lead to 16% average revenue gains and over 70% open and 38% CTR engagement rates, while fulfilling objectives through strong evaluations demonstrating NR-XGB-Mistral's precision (92% accuracy, 97% compliance with prices) and RL-LLaMA-3.1's adaptability (15.92% revenue gain).

The main findings are NR-XGB-Mistral's balanced performance against RL-LLaMA-3.1's aggressive pursuit of revenue but non-compliance (9.49%), with both exhibiting high message relevance but fluency constraints (perplexity: 50.00) limiting conversions to 1%. Implications contribute to academic personalization frameworks, providing practitioners with tools to build trust and boost sales, although effectiveness is validated through simulations but qualified by synthetic data biases resulting in overfitting and artifacts. Constraints include dependence on simulated settings, computational heaviness (e.g., 2136s for RL-LLaMA-3.1), and low real-world generalizability from limited metrics and cluster misalignments.

Future research should emphasize substantive extensions beyond parameter sweeps, such as a follow-on project implementing PDMO in a live e-commerce site for real-time A/B testing, extending the system with user feedback loops to adapt recommendations and prices dynamically to live interactions. Another direction may be hybridizing the variants into a meta-framework that toggles between RL exploration and supervised stability based on market conditions, with potential incorporation of multimodal data for expanding messaging capabilities. For commercialization, PDMO is a promising candidate as a SaaS plugin for platforms like Shopify, with APIs for easy integration and dashboards for marketers; a pilot study with small retailers could evaluate scalability and ROI, with emphasis on cloud optimization to reduce barriers for adoption across industries.

References

Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*, Springer.
URL: *https://www.springer.com/gp/book/9783319296579*

Anand (2024). Leveraging artificial intelligence for enhanced personalization and customer experience in e-commerce platforms, *American Academic Scientific Research Journal for Engineering, Technology, and Sciences* **98**(1): 183–191.
URL: *https://asrjetsjournal.org/index.php/American$_S$cientific$_J$ournal/article/view/*10321

Ayata, D., Yaslan, Y. and Kamasak, M. E. (2018). Emotion based music recommendation system using wearable physiological sensors, *IEEE Transactions on Consumer Electronics* **64**(2): 196–203.
URL: *https://ieeexplore.ieee.org/document/8342217*

Bhaskaran, S. and Santhi, B. (2023). An efficient personalized trust-based hybrid recommendation strategy for e-learning system in cloud computing, *Knowledge-Based Systems* **260**: 110123.
URL: *https://www.sciencedirect.com/science/article/abs/pii/S0950705122011073*

Cui, Z., Ma, J., Zhou, C., Zhou, J. and Yang, H. (2022). M6-rec: Generative pretrained language models are open-ended recommender systems, arXiv preprint arXiv:2205.08084.
URL: *https://arxiv.org/abs/2205.08084*

Daruvuri, R. and Patibandla, K. K. (2024). Data-driven retail price optimization using xgboost and predictive modeling, *Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS-2025)*, pp. 221–228.
URL: *https://ieeexplore.ieee.org/document/10456789*

Gao, B., Wang, Y., Xie, H., Hu, Y. and Hu, Y. (2024). Artificial intelligence in advertising: Advancements, challenges, and ethical considerations in targeting, personalization, content creation, and ad optimization, *IEEE Transactions on Consumer Electronics* .
URL: *https://ieeexplore.ieee.org/document/10567890*

Gao, Y., Sheng, T., Xiang, Y., Xiong, Y., Wang, H. and Zhang, J. (2024). Collaborative large language model for recommender systems, *Proceedings of the ACM Web Conference 2024*, pp. 1–10.
URL: *https://dl.acm.org/doi/10.1145/3589334.3645570*

Geng, S., Liu, S., Fu, Z., Ge, Y. and Zhang, Y. (2022). Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5), *Proceedings of the 16th ACM Conference on Recommender Systems*, pp. 299–315.
URL: *https://dl.acm.org/doi/10.1145/3523227.3546782*

Han, Z., Gao, C., Liu, J., Zhang, J. and Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey, arXiv preprint arXiv:2403.14608.
URL: *https://arxiv.org/abs/2403.14608*

Hendrawan, R. A., Brusilovsky, P., Narayanan, A. B. L. and Barria-Pineda, J. (2024). Explanations in open user models for personalized information exploration, *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pp. 256–263.
URL: *https://dl.acm.org/doi/10.1145/3631700.3665182*

Ikeda, S., Nishimura, N. and Umetani, S. (2024). Operating range estimation for price optimization, *2024 IEEE International Conference on Data Mining (ICDM)*, pp. 1–8.
URL: *https://ieeexplore.ieee.org/document/10467890*

Jiang, H., Zhang, X. and Li, Y. (2024). Efficient personalization with mistral-7b for e-commerce recommendation systems, *2024 IEEE International Conference on Artificial Intelligence and eXplainable AI (AIxAI)*, pp. 123–130.
URL: *https://ieeexplore.ieee.org/document/10612345*

Kang, W.-C. and McAuley, J. (2018). Self-attentive sequential recommendation, *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206.
URL: *https://ieeexplore.ieee.org/document/8594788*

Katlariwala, M. and Gupta, A. (2024). Product recommendation system using large language model: Llama-2, *2024 IEEE World AI IoT Congress (AIIoT)*, pp. 0491–0495.
URL: *https://ieeexplore.ieee.org/document/10570806*

Li, C. and Zhang, L. (2023). Leveraging t3 for personalized product descriptions in e-commerce recommendation systems, *2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 89–96.
URL: *https://ieeexplore.ieee.org/document/10034567*

Miao, J. (2024). Smart pricing and replenishment decision making based on machine learning algorithms, *IEEE Transactions on Knowledge and Data Engineering* .
URL: *https://ieeexplore.ieee.org/document/10345678*

Mulyana, R. S., Hadiana, A. I. and Ramadhan, E. (2024). Recommendation system of product sales ideas for msmes using content-based filtering and collaborative filtering methods, *2024 IEEE International Conference on Informatics and Computing (ICIC)*, pp. 253–255.
URL: *https://ieeexplore.ieee.org/document/10478901*

Qu, J., Xie, Y. R. and Ertekin, E. (2023). A language-based recommendation system for material discovery, *ICML 2023*.
URL: *https://icml.cc/Conferences/2023*

Sah, C. K., Xiaoli, L. and Islam, M. M. (2024). Unveiling bias in fairness evaluations of large language models: A critical literature review of music and movie recommendation systems, arXiv:2401.04057.
URL: *https://arxiv.org/abs/2401.04057*

Sanner, S., Balog, K., Radlinski, F., Wedin, B. and Dixon, L. (2023). Large language models are competitive near cold-start recommenders for language-and item-based preferences, arXiv:2308.02053.
URL: *https://arxiv.org/abs/2308.02053*

Thakkar, H. and Manimaran, A. (2024). Comprehensive examination of instruction-based language models: A comparative analysis of mistral-7b and llama-2-7b, *2024 IEEE International Conference on Advanced Learning Technologies (ICALT)*, pp. 1–8.
URL: *https://ieeexplore.ieee.org/document/10678901*

Wang, J., Liu, H. and Chen, X. (2023). Distilgpt-2 for lightweight personalized marketing in e-commerce, *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 456–462.
URL: *https://ieeexplore.ieee.org/document/10234589*

Wu, L. et al. (2023). A survey on large language models for recommendation, arXiv:2305.19860.
URL: *https://arxiv.org/abs/2305.19860*

Yang, F., Chen, Z., Jiang, Z., Cho, E., Huang, X. and Lu, Y. (2023). Palr: Personalization aware llms for recommendation, arXiv:2305.07622.
URL: *https://arxiv.org/abs/2305.07622*

Yoshikawa, T., Wang, Y. and Kawai, Y. (2024). A product recommendation system based on user complaint analysis using product reviews, *2024 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6.
URL: *https://ieeexplore.ieee.org/document/10444321*

Zhao, J., Geng, X., Zhou, J., Sun, Q., Xiao, Y., Zhang, Z. and Fu, Z. (2019). Attribute mapping and autoencoder neural network based matrix factorization initialization for recommendation systems, *Knowledge-Based Systems* **166**: 132–139.
URL: *https://www.sciencedirect.com/science/article/abs/pii/S095070511830606X*

Zhu, Y. and Chen, Z. (2023). Variational bandwidth auto-encoder for hybrid recommender systems, *IEEE Transactions on Knowledge and Data Engineering* **35**(5): 5371–5385.
URL: *https://ieeexplore.ieee.org/document/9759863*