PREMIER JOURNAL OF SCIENCE ORIGINAL RESEARCH

Check for updates

Hybrid GPT and Neural Models for Personalized E-Commerce: A Novel Framework for Adaptive and Transparent Product Recommendations

Pavan Gunda and Thirupathi Rao Komati

Department of Computer Science and Engineering, GITAM School of Technology, GITAM, Deemed-to-be University, Visakhapatnam, India ROR

Correspondence to: Pavan Gunda, gsraghupavan9@gmail.com

Additional material is published online only. To view please visit the journal online.

Cite this as: Gunda P and Komati TR. Hybrid GPT and Neural Models for Personalized E-Commerce: A Novel Framework for Adaptive and Transparent Product Recommendations. Premier Journal of Science 2026;16:100250

DOI: https://doi.org/10.70389/PJS.100250

Peer Review

Received: 25 October 2025

Last revised: 13 December 2025

Accepted: 17 December 2025

Version accepted: 2

Published: 31 January 2026

Ethical approval: N/a

Consent: N/a

Funding: N/a

Conflicts of interest: N/a

Author contribution: Pavan Gunda and Thirupathi Rao Komati – Conceptualization, Writing – original draft, review and editing

ABSTRACT

In order to transform e-commerce customization, the study presents a product recommendation system that makes use of OpenAI's Generative Pre-trained Transformer (GPT) model's better resources in conjunction with neural integration. Traditional systems often rely on collaborative filtering or content-based models. This may lack precision and adaptability to individual needs. The proposed method overcomes these limitations by integrating hierarchical classification. Hybrid similarity scoring and GPT-based annotation capabilities provide more precise and interpretable rules. The multi-step technique ensures that the machine leads to different types of statistics which includes textual product descriptions and generated metadata to gain meaningful insights. This is because pre-processing techniques determine the specific format of numbers and facts,while embeddings built from neural models provide rich semantic knowledge about relationships between products. The weighted scoring mechanism dynamically adjusts to the choices made by individuals. This increases the machine's capability to generalize to the needs of a man or woman. In addition, GPT-inspired factors promote consideration and transparency in explaining the underlying reasons in natural language. The results of the experiment validated the device's ability to offer advanced customization and user satisfaction. By combining domain-specific data with existing IA strategies, this structure sets a new benchmark for precision-driven, people-centric ecommerce applications.

Keywords: GPT-based explainable recommendations, Hybrid similarity scoring, Neural embedding integration, Hierarchical category extraction, Adaptive e-commerce personalization

Introduction

ProductGPT is a revolutionary concept that harnesses the power of artificial intelligence to transform the product comparison and recommendation landscape. At the heart of ProductGPT lies the integration of ChatGPT, a cutting-edge language model that enables the generation of human-like explanations and bridges the gap between mathematical metrics and natural language. This synergy empowers ProductGPT to provide users with personalized, transparent, and explainable product recommendations, thereby redefining the user experience.

ChatGPT plays a pivotal role in ProductGPT by generating intuitive explanations that elucidate the reasoning behind product recommendations. By decoding complex mathematical metrics, such as cosine similarity and Euclidean distance, ChatGPT translates technical jargon into accessible language, enabling users to make informed decisions. Furthermore, ChatGPT enhances personalization by incorporating user preferences and feedback, ensuring that product recommendations are tailored to individual needs.

The fusion of ChatGPT and mathematical metrics in ProductGPT gives rise to a robust and user-centric system. By bridging the gap between mathematics and natural language, ProductGPT provides a unique and innovative approach to product comparison and recommendation. As we delve into the specifics of ChatGPT's role in ProductGPT, we will explore how this integration revolutionizes the product recommendation landscape, enabling users to navigate complex product landscapes with ease and confidence.

ChatGPT's Role in Generating Explanations

ChatGPT plays a key role in making the recommendation system not only functional, but also useful and it can also be interpreted.[1] It provides personalized explanations for product recommendations. The system processes complex mathematical calculations, such as similarity scores obtained by combining products. Classification, price, and category, but these are abstract and not intuitive for users. With ChatGPT integration, these calculation results are translated into user-friendly narratives.[2] For example, the get_explanation function asks ChatGPT for the following basic inputs: Product name. Product resources (such as prices, discounts, ratings) and user profile settings. This information is used to manage personalized descriptions, such as "This product was recommended because of a deep discount. Corresponds to the price range you want and is in the desired category." Such narratives ensure users understand how their preferences $U_p = \{C, R, P, D\}$ where C is category, R is rating, P is price, and D is discount — are matched with product features $P_f = \{C', R', P', D'\}$. By bridging computational reasoning with natural language, ChatGPT enhances transparency, trust, and user satisfaction.

Decoding Mathematical Metrics with ChatGPT

The recommendation system relies on mathematical metrics such as cosine similarity. Euclidean distance and mixed integration to calculate the similarity between user preferences and product resources.[3] For example, hybrid coalescence similarity is calculated as follows:

$$S_{\text{hybrid}} = \frac{1}{1 + \left\| E_1 - E_2 \right\|} \cdot S_p \tag{1}$$

Guarantor: Pavan Gunda
Provenance and peer-review: Unsolicited and externally peer-reviewed
Data availability statement: N/a

Where E_1 and E_2 are product embeddings, $||E_1-E_2||$ are their Euclidean distance, and S_p is the price similarity score, because this mathematical method guarantees accuracy. As a result, it lacks interpretability for non-technical users. ChatGPT acts as an interpreter, translating these scores into understandable language. For example, ChatGPT can trigger a description such as "This product is highly recommended, because they are very similar to the selected category. Offer attractive prices and have great reviews from other customers." This change allows users to make informed decisions while appreciating the complex calculations behind them.

Enhancing Personalization and User Experience

ChatGPT customizes the recommendation process by describing or aligning two product resources with user-defined needs.[4] Assume a user profile U_p specify preferences for products with minimum rating of R_u, maximum price of P_u, and target discount of D_u. The recommendation system evaluates the weighted similarity score:

$$S_{\text{total}} = w_c \cdot S_c + w_p \cdot S_p + w_r \cdot S_r + w_e \cdot S_e \quad (2)$$

Where w_c, w_p, w_r, and w_e are weights of the category, price, rating and embedding. ChatGPT sets the context for this score by presenting a story such as: "This product meets user needs with ratings up to 4.5, huge discounts of 30%, and prices within a certain range." By aligning numerical results with human language, ChatGPT helps ensure users feel that recommendations are tailored.[5] To meet their specific needs, this greatly increases engagement and satisfaction. Additionally, the ability to explain instructions in detail increases confidence in the system's decisions.

Bridging Mathematics and Natural Language with ChatGPT

Integrating ChatGPT into the recommendation system bridges the gap between mathematical rigor and user accessibility.[6] For example, similarity between product embeds is calculated using cosine similarity:

$$\text{Cosine Similarity} = \frac{\vec{A}.\vec{B}}{\left\|\vec{A}\right\|.\left\|\vec{B}\right\|} \quad (3)$$

Where, $\vec{A},\vec{B}$ represents product feature vectors. Additionally, numerical features such as price and classificationare normalized and compared using techniques such as Euclidean distance. These calculations create quantitative insights. Because its technical nature can alienate users, ChatGPT translates these insights into relevant stories, such as "This product aligns closely with your needs due to its excellent reviews, competitive price and relevance to the chosen category." This combination of computational precision and linguistic clarity is an example of a strong, user-centred IA system. Users are guaranteed to benefit from the complex algorithms without being overwhelmed by complexity.

Literature Survey

Yang Yin et al.[7] has proposed a hybrid NLP methodology for fine-tuning of LLM to identify the accurate culture products. Especial the data was chosen in symbolic semantic info which displays as cultural symbols; these are created in three phases: physical, ethereal, and behavioural. For clear understanding culture images for Generative Pre-trained Transformer (GPT) 3.5 approach an additional data was gathered from Palace Museum. Uncoded symbols are designed for couple of idea makings design thoughts and other was decoded with full design. To clarify these design principles a high level common language is utilized. For these designs a cultural meaning was designed with high clarifiers. By processing the model all symbols are included to the training and while extraction the outcome was changed to meaningful cultures. This data-driven approach seeks to link historical cultural knowledge with modern product design.

Satish Kathiriya et al.[8] couple of approaches are designed for extracting creation phases and optimization of product ads with LLM and Multi-model info. The data includes write up and visual info such as images and metrics, which creates search engines as user friendly. Initial stage of processing data to model and pre-process the data with suitable image was reflected from the collected data. Then, LLMs are improved so that they can understand this data better and give detailed statements that take into account the user needs and industry trends. This model is mainly utilized with GPT-3.5 and 4. This approach also utilizes NLP for the extraction of meaningful words for detailing, SEO was improved. As the description was generated the appropriate match was generated an image with related to product and it makes user friendly.

Yiwen Shi et al.[9] suggested method is based on automatically summarising studies that look at how food affects people from NDA review papers. It uses LLM including GPT-4 to make short outlines that are based on facts through a repeated asking approach. There are three steps to this process. First, the model makes a first report of a food effect report at an advanced level. In the second step, keyword-focused prompts make sure that important pharmacokinetic measures like AUC, Cmax, and Tmax are considered. Lastly, a question that limits the length of the answer makes sure that the summary is short, usually no more than two or three lines. The human reader and LLM work together over and over to make the results better and better, both in terms of truth and readability. To compare the success of GPT-4 and ChatGPT, evaluations use both automatic measures ROUGE and evaluations by FDA professionals. To reduce manual PSG development for generic medicine evaluation, FDA personnel will examine credible, draft-quality summaries.

Alexander Brinkmann et al.[10] have introduced an established method to get trait values from e-commerce product descriptions that aren't organised using LLMs. The goal is to raise the AVE of the output by using the zero-shot and few-shot features of LLMs, especially when there isn't a lot of marked training data available. They show different prompt forms for teaching LLMs how to do AVE jobs. These include written and JSON-based versions of the goal model. We are using the GPT-3.5, GPT-4, and Llama-3 models in the tests to see how well these reminder templates work in both zero-shot and few-shot situations. The way looks at how well LLMs can pull trait values without a lot of task-specific training and compares them to cutting-edge PLMs like BERT. The F1-score is one of several evaluation measures used to check how well the models can correctly extract attribute values, even for attributes that haven't been seen yet. Additionally, the way looks into what happens when you add example values, do in-context learning demos, and fine-tune the models.

Xinyuan Wang et al.[11] integrates LLM with graph-based data to improve proposal systems. The system aims to make it easier to predict how a person and an item will connect by mixing the text-generating abilities of LLMs with the relationship modelling abilities of GNNs that use a graph structure. The main strategy is creating an attentive LLM backbone with a graph structure that records both first- and second-order relationships between people and objects. This way uses GNNs to get edge data within nodes, like people and things, and add this connection information to the LLM's attention system. To help the model understand the many linkages inside the recommendation system, pre-training is carried out utilising crowd-contextual stimuli, such as user and item descriptions and interactions. The model is then fine-tuned using personalised predictive hints to make it even better at making accurate, personalised suggestions. By combining graph data and written data, the method gives a fuller picture of user tastes and item characteristics, which lets it make more accurate suggestions.

Chenhao Fang et al.[12] a novel use of an LLM ensemble technique for e-commerce product attribute value retrieval. The writers suggest an ensemble framework to take advantage of the fact that different LLMs, like GPT, PaLM, and Llama, have different strengths and weaknesses when it comes to different jobs. Using crowd sourcing methods as inspiration, the ensemble approach gives weights to the results of several LLMs based on how well they guess the values of product attributes. The method improves and speeds up the process of assigning weights by using the Dawid-Skene Approach, which is often used with crowd-labeled data. The ensemble method takes the results from all the models and gives more weight to the LLMs that do a better job while reducing the weight of the models that do a worse job. By utilizing unorganized text the product has achieve high accuracy and retrieved reliable data related to products. Walmart's internal datasets showed enhanced "age" and "gender" extraction using the approach.

Haixun Wang et al.[13] one new way to use LLM to improve e-commerce results. Before, search engines used organised data, which meant that they could index and retrieve data from sources that were not structured, like customer reviews. To change this, the study suggests turning structured and semi-structured data, like product catalogues and categories, into written explanations that can be used with LLM training. To train or fine-tune LLMs, the system uses texts with encoded product IDs that have been marked up. This lets them act as a single input for search and selection jobs. With this combination, the model can use domain-specific knowledge and give solutions that are relevant to the situation. According to the design, the framework will combine databases with LLM features, which will allow for the instant recovery of changing data like prices and stock levels. Creating unique IDs for database items, using templates to make labelled texts, and using self-supervised learning to finetune the model are all parts of the approach.

Konstantinos I. Roumeliotis et al.[14] improves product recommendation systems using K-means clustering, CBF, and hierarchical clustering and GPT-4. These algorithms learn patterns and provide personalised suggestions on product datasets, which GPT-4 evaluates. Beyond assessment, GPT-4 uses natural language comprehension to improve product feature semantics and recommendation accuracy. To simplify implementation, e-commerce operators may use CSV files for training and distributing suggestions using a flask-based API. This API emphasises non-technical usability with model training & real-time predictions. GPT-4 automates review using binary ratings and reasoning, boosting cost and scalability over human evaluation. E-commerce companies may customise models for their datasets while preserving efficiency and relevance using the adaptable technique. Our framework improves unsupervised learning and LLM integration for scalable, accurate, & easy to use product system recommendations (Table 1).

Proposed Methodology

The product recommendation system constructed the usage of GPT showcases several modern functions that distinguish it from traditional structures. The integration of OpenAI's GPT for producing motives brings a brand new level of personalization and transparency to suggestions, enabling users to recognize why precise merchandise has been suggested. Unlike traditional collaborative filtering or content-based models, this machine introduces a hybrid similarity metric that blends class-based totally matching, charge-discount alignment, and neural embeddings for textual product descriptions. The use of embeddings permits the device to understand nuanced relationships between merchandise, even when they lack common functions. Furthermore, the gadget consists of hierarchical category extraction (e.g., high-degree, secondary-level categories) for refined filtering. Another novelty lies inside the weighted similarity scoring mechanism, which may be dynamically adjusted to in shape user choices or domain-precise desires. By incorporating diverse similarity metrics, the model ensures extra precise and

Table 1 | Analysis on existing systems

Author	Algorithm	Merits	Demerits
Yang Yin et al.	GPT-3.5, PLM	Fine-tuning the model for automatic retrievals	Utilized dataset contains less content
Satish Kathiriya et al.	LLM	Accuracy and discovery of items are enhances for user experiences	Computation performance was not up to the mark for real-time apps
Yiwen Shi et al.	Iterative prompting with GPT4 & ChatGPT	It improves the understanding of drugs name by iterating the text	Token size was limited
Alexander Brinkmann et al.	LLM	Holds more training data with efficiency and robustness	Comparison performance does not have huge differences
Xinyuan Wang et al.	Integrate LLM	By combining two orders the prediction was efficient	Need more memory for storing paths
Chenhao Fang et al.	LLM-Ensemble	Values for attributes are extracted efficiently were it works accurately even offline	Little time-consuming while extracting dynamical learning weights
Haixun Wang et al.	LLM	Conversion of structure to unstructured data was efficient	Cost-efficient
Konstantinos I. Roumeliotis et al.	CBF, Cutting-edge gpt4LLM, K-means Cluster, Hierarchical Clustering	Compared to all approaches CBF has achieved high accuracy	Overfitting

diverse recommendations. The interpretability of the improved version through GPT bridges the distance between machine-generated decisions and human interpretability. This makes it a modern machine for contemporary e-commerce systems.

The recommendation system uses Amazon dataset (amazon.csv). The dataset contains 1,465 rows and 16 columns. The attributes in the data set are product_id, product_name, category, discounted_price, actual_price, discount_percentage, rating, rating_count, about_product, user_id, user_name, review_id, review_title, review_content, img_link, product_link. In the dataset amazon.csv, only rating_count has 2 missing values. All the other columns have 0 missing values.

Novelty

The proposed model represents a significant improvement over the old approaches. Although older versions may rely on traditional similarity measures such as cosine similarity to raw numerical features or category matching. But the new model suggests a hybrid approach that combines neural input for textual data. This lets in for a deeper understanding of product descriptions and relationships. Additionally, the hierarchical separation of lessons inside the new model allows for multi-level filtering. This becomes no longer present in the antique device. GPT integration is every other top notch thing for interpretation. This makes guidelines more obvious and intuitive.

Additionally, a brand new weighted similarity metric offers flexibility to spotlight sure functions primarily based on user preferences, Whilst the old model Static or much less precise calculations may be used. The machine also includes extra similarity signs inclusive of price discounts and scores. The similarities make the new device more accurate, versatile and compliant with present day e-commerce requirements, together with solving the restrictions of the old gadget greater effectively.

The below flowchart shows a system that uses user input, data preparation, feature extraction, similarity calculations, and a GPT explainability module to provide product suggestions. "User Inputs" via a web user interface (UI) are the first step in the process, and they are handled by a "Input Layer." Prior to being transformed into a format appropriate for analysis, the data is pre-processed, missing values are addressed, and numerical characteristics are standardized. After that, the system pulls characteristics from the data that are related to things like ratings, text embeddings (768-dimensional), and pricing and discounts.

Calculating similarities based on a variety of factors—category similarity (taking hierarchies into account), price and discount similarity, rating similarity, and text embedding similarity—is the main component of the procedure. After that, a "Weighted Similarity Calculation" is created by combining these distinct similarities. A "Recommendation Ranking" stage uses the computed similarities to rank possible recommendations. A crucial element is a "Explainability Module," which makes use of GPT (perhaps the Generative Pre-trained Transformer) to justify the suggestions. The "Final Recommendations" are then shown to the user. Overall, the suggested recommendation system uses OpenAI's GPT to incorporate contemporary AI approaches (Figure 1).

In Product GPT, weights are learned during model training through gradient-based optimization, where each weight represents the strength of influence between neurons in the neural network. Given an input sequence of product-related tokens $x = (x_1, x_2,..., x_n)$, each token is first converted into an embedding vector eie_iei. In the self-attention mechanism, attention weights are computed using the formula.

$$\text{Attention}\,(Q, K, V) = \text{softmax}\frac{QK^T}{\sqrt{d_k}}V$$

Where, $Q = eW_Q$, $K = eW_K$ and $V = eW_V$, and W_Q, W_K, W_V are trainable weight matrices learned from data. These weights are updated by minimizing a loss function $L(y,\hat{y})$, commonly cross-entropy loss, using backpropagation and gradient descent:

$$W_{t+1} = W_t - \eta\frac{\partial L}{\partial W_t}$$

 DOI: https://doi.org/10.70389/PJS.100250 | Premier Journal of Science 2026;16:100250

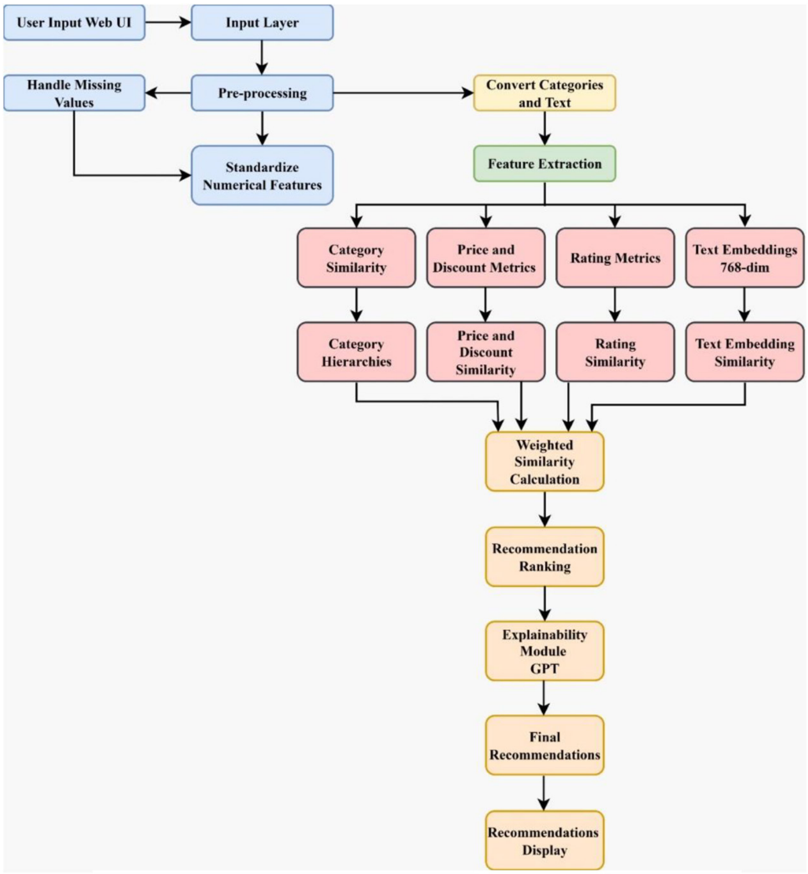

Fig 1 | Block diagram for proposed recommendation system

where η is the learning rate. Over many iterations, the model adjusts these weights to better capture relationships between product features, descriptions, and user preferences, enabling accurate predictions and recommendations.

Data Pre-Processing

Pre-processing helps ensure that datasets are clean, consistent, and ready for advanced analysis.[15] Missing values in numeric columns are handled by defining the median value:

$$x_{\text{filted}} = \begin{cases} x & \text{if } x \neq \text{NaN} \\ \text{median}(x) & \text{if } x = \text{NaN} \end{cases} \tag{4}$$

Categorical columns are imputed with the mode:

$$c_{\text{filted}} = \begin{cases} c & \text{if } c \neq \text{NaN} \\ \text{mode }(c) & \text{if } c = \text{NaN} \end{cases} \tag{5}$$

To standardize numerical data for cosine similarity, we apply:

Where z is the value after standardization, x is the actual value, μ is the mean, and σ is the standard deviation. For textual fields, embeddings (E) are generated using neural models:

$$E = f_{\text{embedding}}(T) \tag{6}$$

Where $f_{\text{embedding}}$ is a neural network-based function (e.g., BERT or GPT embeddings), and T is the product description.[16] These embeddings allow the model to efficiently encode the semantic relationships between products. When combined with hierarchical class separation, these steps ensure data consistency and provide a robust model for the input format. It helps make predictions more accurate and to improve user experience (Figure 2).

Feature Extraction and Hierarchical Category Management

In the approach, the next step after preparing the data is to extract the categories, focusing on hierarchical categories.[17] The system divides the category space into high-level categories, secondary level and tertiary-level categories in function separator get_high_level_category(category) using the delimeter "|". The first section is selected as the high level category. Mathematically, if the category string is as specified C so higher level classes are:

$$\text{high\_level\_categroy}(c) = \text{split}(c, \text{“|”})[0] \tag{7}$$

For the secondary and tertiary levels, similar logic applies:

$$\text{secondary\_level\_category}(c) = \text{split}\ (c, \text{“| ”})[1] (\text{if exists}) \tag{8a}$$

$$\text{tertiary\_level\_category}(c) = \text{split}\ (c, \text{“|”})[2]\ (\text{if exists}) \tag{8b}$$

These extracted items are then added to the dataset as new categories. This allows the model to consider product distribution at different levels. This structure is useful for more granular filtering and recommendations.[18] Because items in higher groups are grouped and sorted more efficiently, a hierarchical structure allows the model to identify patterns in the data based on different levels of variance. Further improve the quality of recommendations by including deeper relationships between constructs.

Hypothetical Input Data

Hypothetical input data are summarized in Table 2.

Extraction Logic:

$$\text{high\_level\_categroy}(c) = \text{split}(c, \text{“|”})[0]$$

$$\text{secondary\_level\_category}(c) = \text{split}\ (c, \text{“|”})[1]\ (\text{if exists})$$

$$\text{tertiary\_level\_category}(c) = \text{split}\ (c, \text{“|”})[2]\ (\text{if exists})$$

Table 2 | Hypothetical input data

Product ID	Category
101	Electronics | Smartphones | Accessories
102	Home | Furniture | Living Room
103	Electronics | Laptops

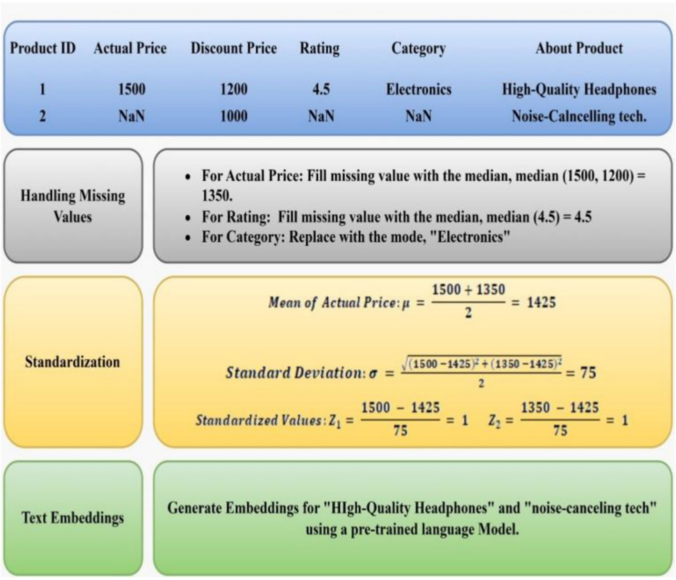

Fig 2 | Pre-processing of elements

Extracted Categories:

1. For Product 101:
   - High-level: Electronics
   - Secondary-level: Smartphones
   - Tertiary-level: Accessories
2. For Product 102:
   - High-level: Home
   - Secondary-level: Furniture
   - Tertiary-level: Living Room
3. For Product 103:
   - High-level: Electronics
   - Secondary-level: Laptops
   - Tertiary-level: None (no 3rd segment)

Final Table

Let's go through more detailed numerical examples for each of the five steps with expanded calculations which ensure to show all intermediate steps and formulas (Table 3).[19]

Feature Extraction and Hierarchical Category Management

Hypothetical Input Data

Hypothetical input data are summarized in Table 4.

Table 3 | Example product categories

Product ID	High Level	Secondary Level	Tertiary Level
101	Electronics	Smartphones	Accessories
102	Home	Furniture	Living Room
103	Electronics	Laptops	

Table 4 | Hypothetical input data

Product ID	Category
101	Electronics | Smartphones | Accessories
102	Home | Furniture | Living Room
103	Electronics | Laptops

Extraction Logic:

$$\text{high\_level\_categroy}(c) = \text{split}\left(c, \text{“|”}\right)\left[0\right]$$

$$\text{secondary\_level\_category}(c) = \text{split}\ (c, \text{“|”})[1](\text{if exists})$$

$$\text{tertiary\_level\_category}(c) = \text{split}\ (c, \text{“|”})[2](\text{if exists})$$

Extracted Categories:

1. For Product 101:
   - High-level: Electronics
   - Secondary-level: Smartphones
   - Tertiary-level: Accessories
2. For Product 102:
   - High-level: Home
   - Secondary-level: Furniture
   - Tertiary-level: Living Room
3. For Product 103:
   - High-level: Electronics
   - Secondary-level: Laptops
   - Tertiary-level: None (no 3rd segment)

Final Table

Example product categories are summarized in Table 5.

Standardization of Numerical Features

Standardization of numerical features is an important step to make the variables comparable by converting them to a common scale. The numerical features in the dataset, such as actual_price, discounted_price, discount_percentage, rating, and rating_count, are standardized using the formula for Z-score normalization:

$$z = \frac{x - \mu}{\sigma} \tag{9}$$

Where x is the original value, μ is the mean, and σ is the standard deviation of the feature. The purpose of this adjustment is to ensure that all numeric are standardized, so that the similarity measure is not affected by large differences between features and able to work effectively. This standard is important for similarity - the base models participate equally in the calculations. After setting the standard the dataset is more suitable for similarity-based analysis. This leads to more accurate recommendations. Normalization reduces bias caused by disproportionately large or small values in a data set. This will improve the generalizability of the model.

Text Embedding Generation

Another important feature in this example is the text entry for the field about_product. Embeddings

Table 5 | Example product categories

Product ID	High Level	Secondary Level	Tertiary Level
101	Electronics	Smartphones	Accessories
102	Home	Furniture	Living Room
103	Electronics	Laptops	

are mathematical representations of meaningful data results. The function get_product_embedding (product_description) tracks the process of creating an embedding. If T stands for product specification, Embedding E is obtained from a pre-trained model, e.g. BERT or GPT embedding generation formula can be specified as follows:

$$E(T) = f_{\text{embedding}}(T)$$

Where $f_{\text{embedding}}$ is a neural network-based function (e.g., BERT or GPT embeddings), and T is the product description. These embeddings permit the machine to compare the semantic similarity of different merchandise; the closer pals are in vector space. The extra semantically similar the products can be, the greater similar they will be. This approach allows the gadget to advise products that don't monitor genuine characteristics however are comparable inside the text description. This makes the model greater flexible and might locate hidden relationships among products. In addition to information, it is able to additionally make use of rich semantic context.[20]

***Hypothetical Input Data*:**

Using a pre-trained embedding model (OpenAI), generate embeddings (Table 6).

Embedding

Generated embeddings for product descriptions are summarized in Table 7.

Hybrid Similarity Computation

The subsequent step includes calculating the similarity between products based on multiple capabilities: class, fee, cut price, score, and embeddings. The similarity among two products is computed the usage of an aggregate of numerous metrics. For class similarity, the formula is straightforward: if each merchandise belongs to the same high-degree category, the similarity is 1, in any other case 0:

$$\text{category\_similarity}(p_1, p_2) = \begin{cases} 1 & \text{if high\_level\_category}(p_1) = \text{if high\_level\_category}(p_2) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Table 6 | Product descriptions used for embedding generation

Product ID	About Product
101	"High-quality smartphone accessories."
102	"Comfortable and modern living room furniture."
103	"Latest laptops with powerful specifications."

Table 7 | Generated embeddings for product descriptions

Product ID	Embedding (E)
101	[0.45, 0.60, 0.32, 0.11]
102	[0.12, 0.35, 0.55, 0.40]
103	[0.80, 0.70, 0.40, 0.25]

For price and discount similarity, the difference in prices and discount percentages is normalized to create a similarity score:

$$\text{price\_discount\_similarity}(p_1, p_1) = \frac{1}{1 + |\text{price}(p_1) - \text{price}(p_2)| + |\alpha - \beta|} \quad (11)$$

Where α is discount_percent(p_1) and β is discount_percent(p_2). For rating similarity, a similar approach is used:

$$\text{rating\_similarity}(p_1, p_2) = \frac{1}{1 + |\text{rating}(p_1) - \text{rating}(p_2)|} \quad (12)$$

For embedding similarity, the Euclidean distance between the embeddings of two products is used:

$$\text{embedding\_similarity}(p_1, p_2) = \frac{1}{1 + |\|E(p_1) - E(p_2)\||} \quad (13)$$

Finally, these individual similarities are combined using weighted summation to generate a final similarity score:

$$\begin{aligned} &\text{total\_similarity}(p_1, p_2) = \\ &w_1.\ \text{category\_similarity}(p_1, p_2) + w_2.\text{price\_discount\_} \\ &\text{similarity}(p_1, p_2) + w_3.\ \text{rating\_similarity}(p_1, p_2) + \\ &w_4.\text{embedding\_similarity}(p_1, p_2) \end{aligned} \quad (14)$$

Where, w_1,w_2,w_3,w_4 are the weights assigned to each feature's similarity. The model then uses this final similarity score to rank the products and recommend those most aligned with the user's preferences (Figure 3).

Final Recommendation and Explainability

After calculating the similarity ratings, the goods are ranked in descending order of similarity. The top tips are selected for display to the user. The machine also carries GPT-based totally explainability. For every encouraged product, the machine generates a textual explanation as to why the product turned into recommended, the use of the formula:

$$\text{explanation} = g_{\text{GPT}}(\text{product\_features}, \text{user\_profile}) \quad (15)$$

Where is OpenAI GPT model that generates explanation primarily based at the product capabilities and the consumer profile. This enhances the transparency and trustworthiness of the recommendation device. Finally, the gadget returns the pinnacle N hints, whole with their factors, pictures, and hyperlinks for further exploration. The hybrid similarity model combined with explainability affords a comprehensive advice framework that caters to the consumer's choices and enhances the buying revel in. By presenting explanations for every advice, the model fosters extra person believe and engagement, for that reason enhancing the general consumer relevant (Figure 4).

Limitations and Future Scope

It is important to recognize that the Product GPT research has a number of limitations. First, it is

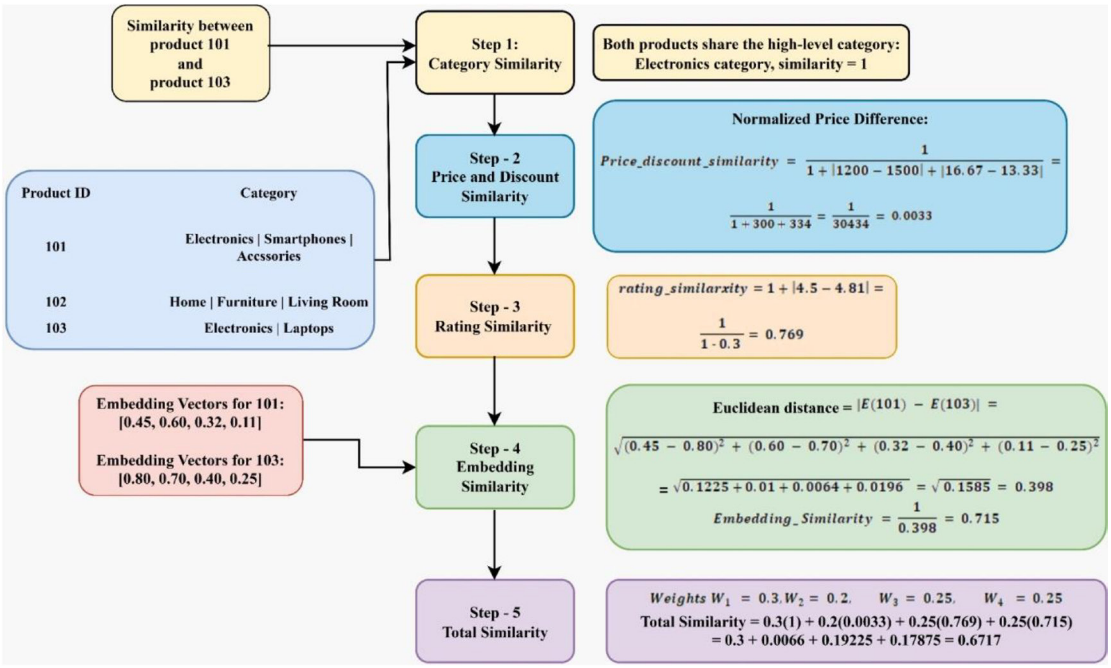

Fig 3 | Similarity computations

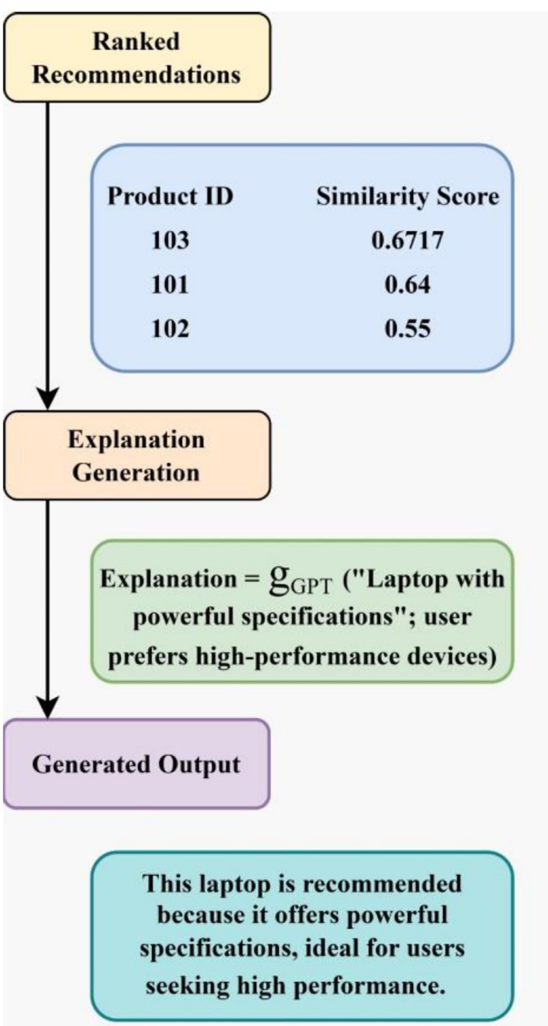

Fig 4 | Explainable AI recommendation

limited by the amount, scope, and quality of the dataset, which could not accurately reflect the variety of goods, clientele, or market circumstances, thereby resulting in biased conclusions and forecasts. Second, Product GPT is mostly dependent on text data from the past (such as reviews or descriptions), which may miss contextual elements like seasonality and geographical patterns, as well as current market dynamics and changing customer preferences. Third, it might be challenging to completely trust outputs without human validation since language models can have interpretability issues and provide reasonable but incorrect results.

Furthermore, there is still a lack of research on ethical issues such data privacy, prejudice transmission, and an excessive dependence on automated suggestions. Expanding datasets across many domains and regions, including real-time and multimodal data (such visuals, price, and user behavior), enhancing model explainability, and creating reliable assessment frameworks should be the key goals of future research. Human-in-the-loop systems, bias mitigation techniques, and the useful effects of Product GPT on business decision-making and consumer trust might all be the subject of future research.

Although Product GPT's explanation quality is generally helpful in that it can produce understandable, human-readable explanations for product recommendations or insights, its faithfulness may be limited because explanations are deduced from learned patterns rather than explicit causal reasoning, which may result in reasonable but incorrect justifications. Overreliance on created explanations, possible hallucinations, and amplification of preexisting biases in product reviews or ratings—such as popularity or demographic bias—are among the risks. Strong data anonymization

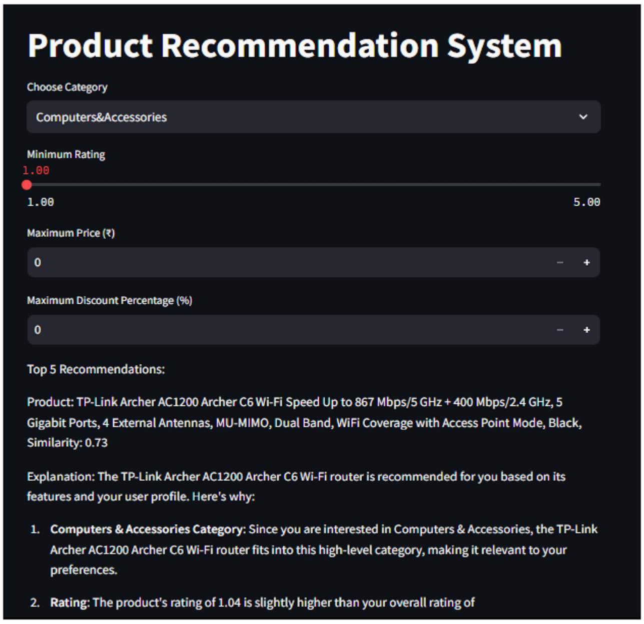

Fig 5 | UI of the application

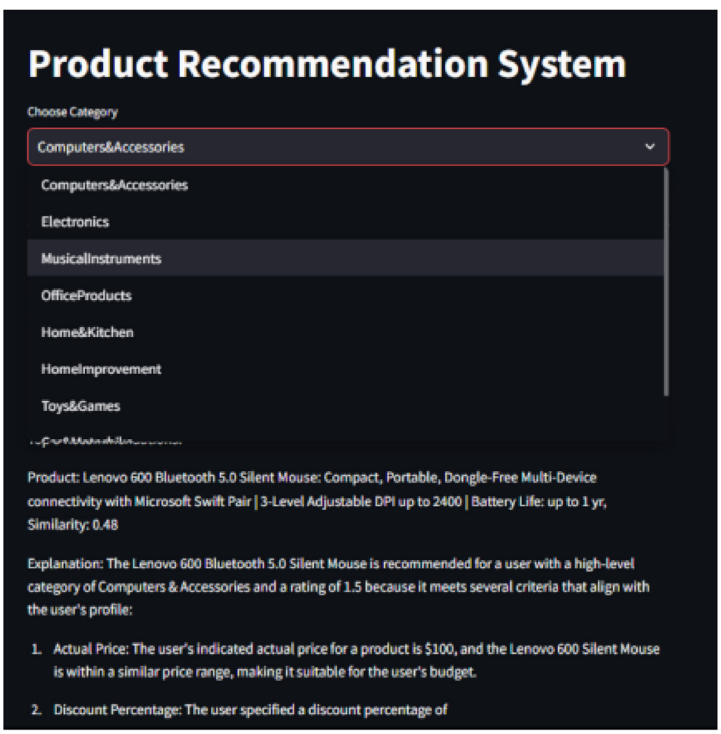

Fig 6 | Categories available

and control are necessary when training on or drawing conclusions from user-generated content that can contain sensitive information. In reality, Product GPT can be expensive to train and implement because of its high computational and storage needs. Additionally, latency may rise for real-time applications, particularly when managing lengthy product descriptions or heavy user traffic, requiring optimization and caching techniques.

Results and Discussions

Figure 5 shows the main UI of the product recommendation application created using Streamlit. The interface is designed to be intuitive and easy to use. It has a header that clearly states the purpose of the application. Users will see several interactive elements such as drop-down menus, sliders, and number entry fields. These components allow users to specify their preferences, such as selecting categories, minimum score required, maximum price and maximum discount percentage. The clean layout ensures that users can easily navigate the interface without any confusion. Interactive widgets provide real-time feedback. It makes the user experience smooth and attractive. The modular interface design reflects careful consideration in structuring user input. This screenshot shows the application's main function of storing user preferences in an orderly manner. It emphasizes the simplicity and accessibility of the design. To support both novice and experienced users, clear division of elements ensures that each input field is easily identified to prevent ambiguity during interactions.[21]

Figure 6 offers an outline of the product classes available in the dataset. The classes are displayed in a dropdown menu, allowing customers to select the excessive-degree class that suits their preferences. These categories are derived from the 'class' column in the dataset and include numerous product kinds. The dropdown capability enables dynamic filtering of hints based totally at the person's preference. Users can fast navigate thru the listing and make selections that replicate their shopping pursuits. The inclusion of categories allows slender down the search space, making hints greater focused and relevant. This feature not handiest enhances user delight but additionally improves the efficiency of the advice engine. The design guarantees that even a huge variety of categories are presented in a prepared and reachable way. By permitting category-specific filtering, the utility guarantees that users acquire consequences aligned with their number one hobbies. This screenshot highlights how the software leverages established statistics to improve personalization and person experience.

In Figure 7, the input fields are highlighted, showing how users can customize their search for recommendations. These inputs include sliders for specifying the minimum acceptable product rating and number fields for defining the maximum price and discount percentage. By combining these inputs with the category selection, the application gathers detailed user preferences to create a comprehensive user profile. This profile serves as the foundation for calculating similarity scores and filtering products. The flexibility of the input options ensures that users can fine-tune their preferences to match specific needs.

For instance, the sliders provide granular control over rating thresholds, while the number inputs allow precise budget constraints. This level of customization makes the application versatile and accommodating to a wide range of use cases. The intuitive placement of input fields ensures that users can quickly configure their

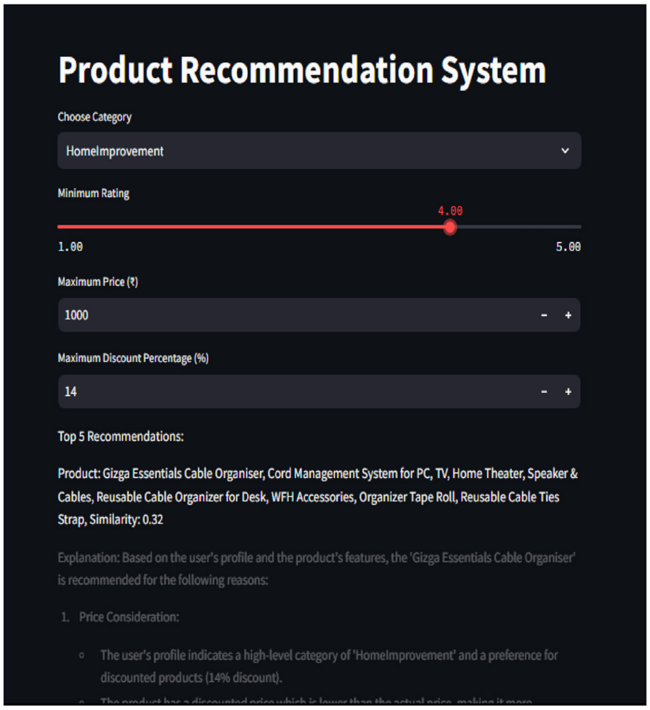

Fig 7 | Input features given to the application

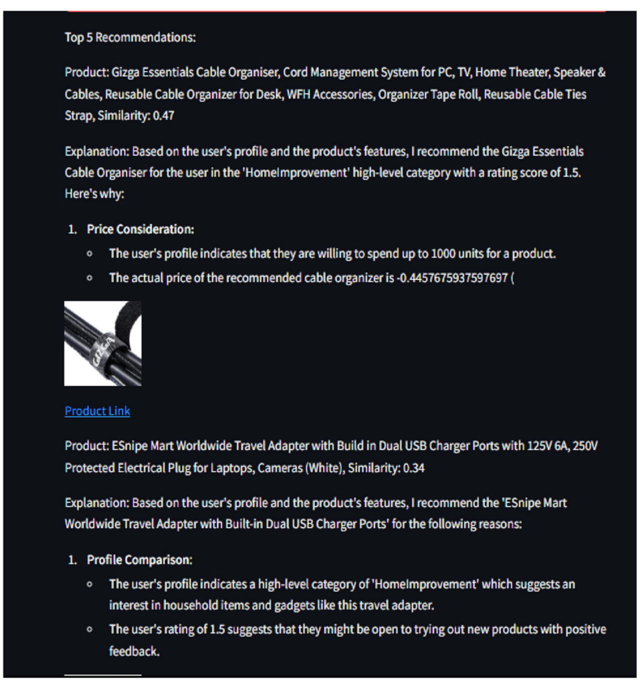

Fig 8 | Top 5 recommendations generated

preferences without requiring additional guidance. Overall, this screenshot illustrates the application's commitment to delivering a tailored user experience by empowering users to define their own criteria for recommendations.

Figure 8 showcases the consequences of the advice set of rules, showing the top five recommended products. Each product is indexed with its call and similarity score, indicating how well it fits the user's preferences. The guidelines are looked after in descending order of similarity, making sure that the most applicable alternatives seem on the top. This view underscores the effectiveness of the similarity metrics used inside the utility, along with category similarity, fee-bargain alignment, and score alignment. The concise presentation of effects makes it smooth for customers to assess their alternatives at a glance. Additionally, the inclusion of similarity rankings provides transparency into the recommendation procedure, constructing accept as true with the customers. The visible readability of the guidelines web page ensures that customers can fast become aware of their desired picks and proceed to explore further info. This screenshot reflects the utility's capacity to synthesize consumer inputs into actionable outputs, highlighting the performance and precision of the underlying set of rules.

Figure 9 provides a detailed view of a recommended product. It consists of critical records which include the product's name, similarity percentage, and a brief clarification of why it turned into encouraged. Additionally, the product's picture and a clickable link to the product page are displayed. The explanation is generated the use of OpenAI's GPT model, adding a layer of transparency to the recommendation system through describing how the product suits the consumer's choices. The inclusion of visuals and direct links enhances consumer engagement, bearing in mind seamless exploration and buying selections. The similarity percentage gives insight into how intently the product aligns with the person's inputs, while the reason gives context for the recommendation. This mixture of functions ensures that users have all the important information to make knowledgeable selections.

The utility's potential to provide complete statistics in a visually attractive manner enhances its usability and effectiveness. This screenshot encapsulates the utility's attention on handing over fee through targeted and customized recommendations.

Figure 10 demonstrates the functionality of the clickable product links, displaying how customers are redirected to the respective product pages. Once a person clicks on a product link, they're taken to the e-commerce platform wherein they are able to view extra details, study critiques, and complete their purchase. This function bridges the space between pointers and motion, permitting users to act on their options with minimal attempt. The easy redirection procedure highlights the practical software of the utility, streamlining the online buying enjoy whilst making sure person satisfaction. The integration of clickable links guarantees that customers can transition seamlessly from exploring hints to making purchasing selections. This screenshot also emphasizes the utility's function as a facilitator in the e-commerce adventure, simplifying the technique of coming across and obtaining favoured merchandise. By supplying a direct course to the product page, the utility eliminates needless steps, enhancing the overall efficiency and comfort of the consumer enjoy.

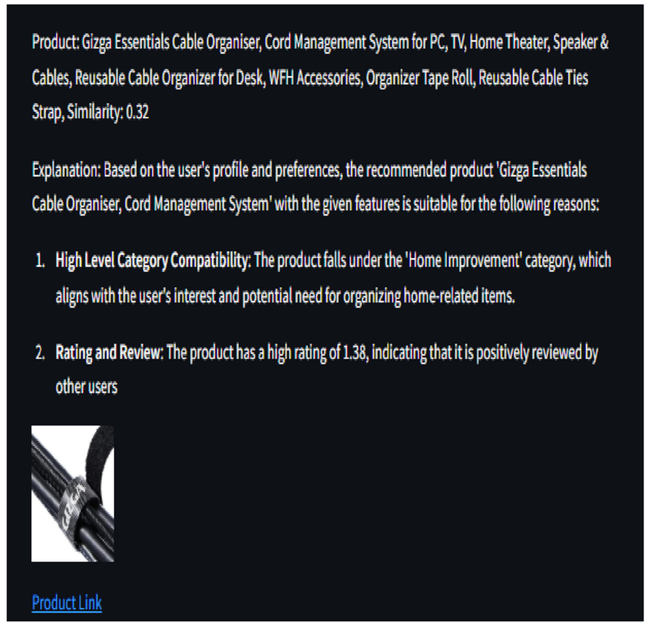

Fig 9 | Product details, similarity percentage, explanation, product image and product link for the recommended product

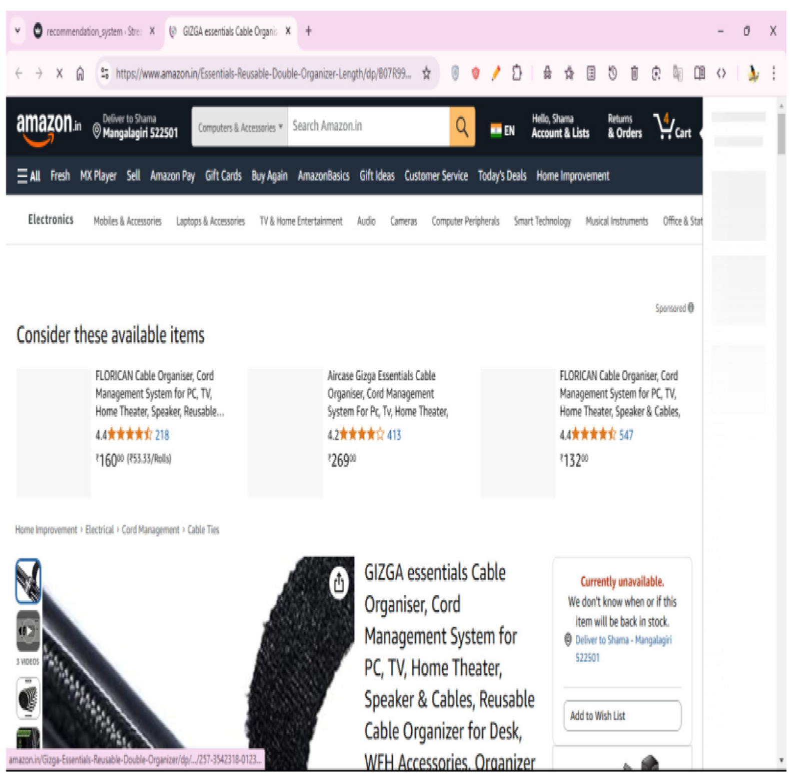

Fig 10 | Forwarded to the product page when clicked on provided link

Figure 11 presents a breakdown of the different types of content generated by GPT-3.5, highlighting its versatility in various applications. Among the five categories: Code Generation, Text Summaries, Explanations, Creative Writing, and Data Analysis, the highest frequency is observed in Explanations, suggesting that GPT-3.5 is extensively used to clarify complex topics, making it a valuable tool for education, research, and professional learning. Code Generation follows closely, indicating the model's effectiveness in assisting programmers by producing functional and efficient code snippets.

Text Summaries also constitute a significant portion of GPT-3.5's output, showing its ability to condense large volumes of information into concise, readable formats, particularly useful for summarizing reports,

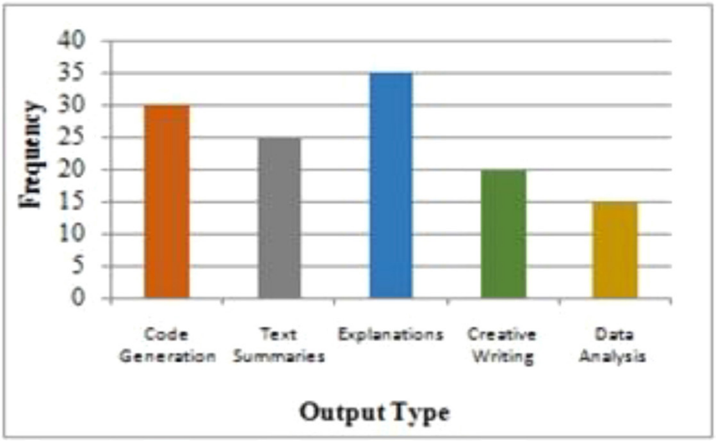

Fig 11 | GPT-3.5 output distribution

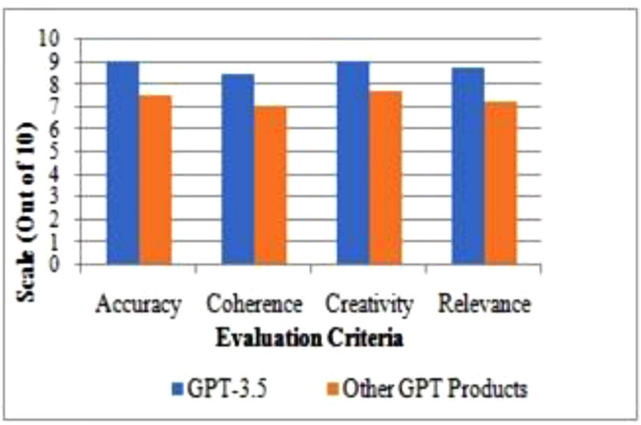

Fig 12 | Comparision of GPT-3.5 performance

academic papers, or news articles. While GPT-3.5 can generate imaginative content and perform data-driven tasks, Creative Writing and Data Analysis are relatively less frequent, indicating that these areas are not its primary strengths or most common use cases. Overall, the distribution reflects GPT-3.5's capability to adapt to multiple domains, with a strong emphasis on technical and educational applications.

Figure 12 compares GPT-3.5's performance against other GPT models, based on four evaluation criteria: Accuracy, Coherence, Creativity, and Relevance. GPT-3.5 consistently outperforms other models across all criteria, demonstrating its superiority in generating high-quality responses. It scores significantly higher in Accuracy, reflecting its ability to provide factually correct and reliable information. GPT-3.5 also excels in Coherence, ensuring logically structured and contextually appropriate responses, and scores well in Creativity, producing original and engaging content.

Furthermore, it shows a strong advantage in Relevance, providing responses aligned with user intent and offering meaningful insights. Overall, the graph highlights GPT-3.5's superior linguistic and analytical capabilities, making it a preferred choice for applications requiring precision, clarity, and contextual understanding, and demonstrating how improved AI models can significantly enhance the quality of generated content across multiple domains.

Considered amazon.csv is a public dataset. Compared to many publicly available datasets, the Amazon.csv dataset is more suited for Product GPT as it includes extensive, domain-specific product information such thorough descriptions, ratings, and customer reviews that closely resemble actual e-commerce transactions. Product GPT can efficiently learn semantic associations between products and user sentiment thanks to its structured yet text-heavy style, which is frequently lacking or superficial in general public datasets. The model's outputs are more useful and trustworthy for tasks involving product comprehension and recommendation as the dataset also represents real-world customer behavior and purchasing patterns.

Conclusion

The proposed recommendation system effectively integrates modern AI techniques. To meet the challenges of traditional e-commerce platforms by leveraging the interpretive power of OpenAI's GPT model and the structural insights of hierarchical class separations, the machine thus achieves an excellent balance between precision and scalability. The hybrid similarity scoring mechanism enhances the version's ability to distinguish different relationships between products while neural integration provides a richness of meaning that is lacking in conventional similarity measures. The dynamic weighting of similarity measures guarantees flexibility and scalability. Meet the needs of many people and various space requirements. Incorporating GPT-based annotation capabilities not only improves the user experience but also improves user experience, but also add consideration Provide clear and interpretable reasons for each recommendation. This transparency promotes consumer engagement and loyalty. Filling a critical gap in traditional recommendation structures, the device's modular and scalable design also allows for perfect integration into modern ecommerce workflows. By combining data based on semantic text analysis, the structure can therefore provide a precise, versatile, and actionable dictionary. Future trends may focus on increasing scalability. Reduce computational costs and integrate real-time observation mechanisms. This groundbreaking technique creates a solid foundation for the next generation of user-centred intelligent recommendation systems.

References

1 Kumar A, Lakkaraju H. Manipulating large language models to increase product visibility. arXiv [Preprint]. 2024. https://doi.org/10.48550/ARXIV.2404.07981
2 Shahin M, Chen FF, Hosseinzadeh A, Maghanaki M, Eghbalian A. A novel approach to voice of customer extraction using GPT-3.5 Turbo: linking advanced NLP and Lean Six Sigma 4.0. Int J Adv Manuf Technol. 2024;131:3615–30. https://doi.org/10.1007/s00170-024-13167-w
3 Cheng Y, Chen J, Huang Q, Xing Z, Xu X, Lu Q. Prompt Sapper: a LLM-empowered production tool for building AI chains. ACM Trans Softw Eng Methodol. 2024;33(5):1–24. https://doi.org/10.1145/3638247
4 Das S, Saboo R, Vadrevu CSK, Wang B, Xu S. Applications of LLMs in E-commerce search and product knowledge graph: the DoorDash case study. In: Proceedings of the 17th ACM international

conference on web search and data mining, Merida Mexico; 2024. https://doi.org/10.1145/3616855.3635738
5 Feng P, He Y, Huang G, Lin Y, Zhang H, Zhang Y, et al. AGILE: a novel reinforcement learning framework of LLM agents. arXiv [Preprint]. 2024. https://doi.org/10.48550/ARXIV.2405.14751
6 Gunda P, Komati TR. Enhancing user experience and mitigating BIAS in electronic device recommendations on e-commerce platforms: leveraging NLP and GPT technologies. In: 2024 4th international conference on sustainable expert systems (ICSES), Kaski, Nepal; 2024. p. 149–56. https://doi.org/10.1109/ICSES63445.2024.10763266
7 Yin Y, Ding S, Zhang X, Wang C, Li X, Cai R. Cultural product design concept generation with symbolic semantic information expression using GPT. In: Proceedings Design Research Society Conference 2024. https://doi.org/10.21606/drs.2024.508
8 Kathiriya S, Mullapudi M, Karangara R. Optimizing eCommerce listing: LLM-based description and keyword generation from multimodal data. Int J Sci Res. 2023;12(10):2123–30. https://doi.org/10.21275/SR24304113521
9 Shi Y, Ren P, Wang J, Han B, Valizadeh Aslani T, Agbavor F, et al. Leveraging GPT-4 for food effect summarization to enhance product-specific guidance development via iterative prompting. J Biomed Inform. 2023;148:104533. https://doi.org/10.1016/j.jbi.2023.104533
10 Brinkmann A, Shraga R, Bizer C. ExtractGPT: exploring the potential of large language models for product attribute value extraction. arXiv [Preprint]. 2023. Available from: http://arxiv.org/abs/2310.12537
11 Wang X, Wu L, Hong L, Liu H, Fu Y. LLM-enhanced user-item interactions: leveraging edge information for optimized recommendations. arXiv [Preprint]. 2024. Available from: http://arxiv.org/abs/2402.09617
12 Fang C, Li X, Fan Z, Xu J, Nag K, Korpeoglu E, et al. LLM-Ensemble: optimal large language model ensemble method for e-commerce product attribute value extraction. In: SIGIR '24: Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval. 2024. p. 2910–4. https://doi.org/10.1145/3626772.3661357
13 Wang H, Na T. Rethinking E-commerce search. In: ACM SIGIR forum. Vol. 57, No. 2. New York, NY, USA: ACM; 2024. p. 1–19. https://doi.org/10.48550/arXiv.2312.03217
14 Roumeliotis KI, Tselikas ND, Nasiopoulos DK. Precision-driven product recommendation software: unsupervised models, evaluated by GPT-4 LLM for enhanced recommender systems. Software. 2024;3(1):62–80. https://doi.org/10.3390/software3010004
15 Gunda P, Komati TR. Integrating self-attention mechanisms for contextually relevant information in product management. Int J Comput Exp Sci Eng. 2024;10(4). https://doi.org/10.22399/ijcesen.651
16 Li P, Castelo N, Katona Z, Sarvary M. Frontiers: determining the validity of large language models for automated perceptual analysis. Mark Sci. 2024;43(2):254–66. https://doi.org/10.1287/mksc.2023.0454
17 Ilagan JB, Ilagan JR, Zulueta PY, Rodrigo MM. Optimizing conversational commerce involving multilingual consumers through large language models' natural language understanding abilities. In: Degen H, Ntoa S, editors. Artificial intelligence in HCI. HCII 2024. Lecture Notes in Computer Science. Vol 14736. Cham: Springer; 2024. https://doi.org/10.1007/978-3-031-60615-1_4
18 Smilarubavathy G, Mohammed Afzal KA. Empowering product purchase decisions through AI-driven review analysis and summarization. In: 2024 international conference on inventive computation technologies (ICICT), Lalitpur, Nepal; 2024. p. 155–62. https://doi.org/10.1109/ICICT60155.2024.10544601
19 Jagatap A, Merugu S, Comar PM. Improving search for new product categories via synthetic query generation strategies. In: Companion Proceedings of the ACM Web Conference 2024, Singapore; 2024. p. 29–37. https://doi.org/10.1145/3589335.3648299
20 Roumeliotis KI, Tselikas ND, Nasiopoulos DK. LLMs in e-commerce: a comparative analysis of GPT and LLaMA models in product review evaluation. Nat Lang Process J. 2024;6:100056. https://doi.org/10.1016/j.nlp.2024.100056
21 Erlansyah D, Mukminin A, Julian D, Negara ES, Aditya F, Syaputra R. Large language model comparison between GPT-3 and PaLM-2 to produce Indonesian cultural content. East Eur J Enterp Technol. 2024;4(2):19–29. https://doi.org/10.15587/1729-4061.2024.309972