InteraRec: Interactive Recommendations Using Multimodal Large Language Models

Saketh Reddy Karra[a], Theja Tulabandhula[a]

[a] *University of Illinois Chicago, 601 S Morgan St, Chicago, IL 60607, United States*

arXiv:2403.00822v1 [cs.IR] 26 Feb 2024

Abstract

Weblogs, comprised of records detailing user activities on any website, offer valuable insights into user preferences, behavior, and interests. Numerous recommendation algorithms, employing strategies such as collaborative filtering, content-based filtering, and hybrid methods, leverage the data mined through these weblogs to provide personalized recommendations to users. Despite the abundance of information available in these weblogs, identifying and extracting pertinent information and key features necessitates extensive engineering endeavors. The intricate nature of the data also poses a challenge for interpretation, especially for non-experts. In this study, we introduce a sophisticated and interactive recommendation framework denoted as *InteraRec*, which diverges from conventional approaches that exclusively depend on weblogs for recommendation generation. This framework captures high-frequency screenshots of web pages as users navigate through a website. Leveraging state-of-the-art multimodal large language models (MLLMs), it extracts valuable insights into user preferences from these screenshots by generating a user behavioral summary based on predefined keywords. Subsequently, this summary is utilized as input to an LLM-integrated optimization setup to generate tailored recommendations. Through our experiments, we demonstrate the effectiveness of *InteraRec* in providing users with valuable and personalized offerings.

Keywords: Large language models, Screenshots, User preferences, Recommendations

Email addresses: `skarra7@uic.edu` (Saketh Reddy Karra), `theja@uic.edu` (Theja Tulabandhula)

1. Introduction

In the evolving landscape of e-commerce, the need to understand user preferences for offering personalized content is increasingly crucial for online platforms to maximize revenue, build customer loyalty, and maintain competitive advantage. Typically, the nuanced information that can explain the user browsing behavior on these platforms is stored as weblogs (Rosenstein, 2000). Numerous state-of-the-art recommendation systems, employing strategies such as collaborative filtering, content-based filtering, and hybrid methods, leverage the data mined through these weblogs to provide personalized recommendations. This data-driven approach ensures that recommendations align closely with user preferences, contributing to a more seamless and satisfying shopping journey and ultimately increasing overall platform revenue.

Raw weblogs contain a wealth of information on key browsing session details, including start and end times, visited pages, and clickstream data. Beyond capturing fundamental details, these weblogs delve into user-specific data, encompassing identifiers and IP addresses. However, interpreting the raw web log data directly can be challenging for non-experts, requiring them to navigate its complexity and dissect it to build recommendation models. Often, sophisticated data engineering techniques are necessary to extract the relevant features and datasets needed for training these models. With recent advancements in artificial intelligence, particularly within the domain of LLMs (Zhao et al., 2023), a compelling case emerges for developing a principled framework that could present a superior solution to the current dependency on weblogs.

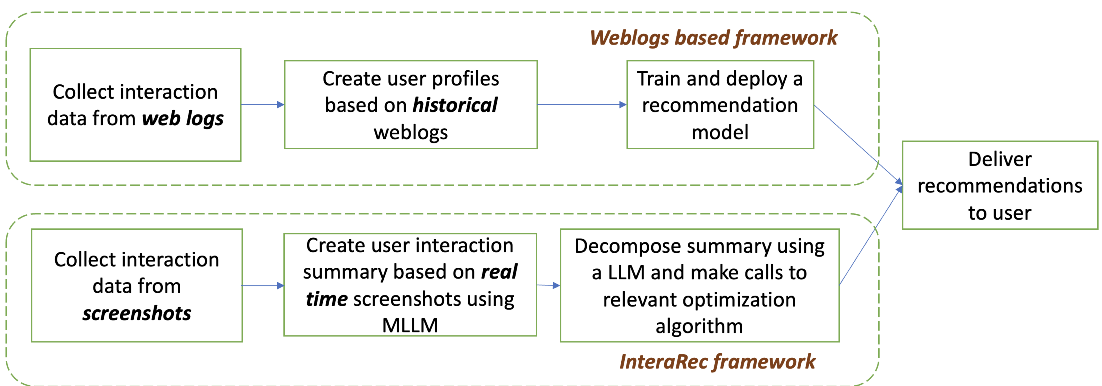

Figure 1: A comparison of weblog-based framework and *InteraRec* framework (refer to Figure 2 for full system description)

.

Inspired by the above, we propose an innovative approach leveraging information from screenshots of users' internet browsing activities. Our approach leverages MLLMs adept at seamless processing and generating content across diverse modalities. By opting for screenshots as inputs instead of relying on weblogs, the system benefits from heightened interpretability. The visual nature of screenshots provides a lucid and transparent representation of user actions, significantly enhancing the comprehension of the inferences made by the LLM. Unlike weblogs, which may introduce complexities associated with various features, the use of screenshots simplifies the inputs, offering a more straightforward and intuitive approach as shown in Figure 1. Furthermore, this method facilitates concurrent, real-time recommendations, as screenshots can be captured almost instantaneously.

In this paper, we introduce *InteraRec*, a novel screenshot-based user recommendation system. We capture high-frequency screenshots of web pages as users navigate through them in real-time. Leveraging the capabilities of MLLMs, *InteraRec* processes these screenshots to extract meaningful insights into user behavior and execute pertinent optimization tools to deliver personalized recommendations. By seamlessly integrating visual data, language models, and optimization tools, *InteraRec* transcends the limitations of existing systems, promising a more personalized and effective recommendation system for users.

2. Related Work

In this study, we expand upon two key streams of research: (a) LLMs for user behavior modeling and (b) Tools and their integration with LLMs. We briefly discuss some of the related works below.

LLMs for user behavior modeling. LLMs are known for their capability to dynamically generate diverse facets of user profiles by analyzing their historical viewing and transaction data. (Chen, 2023) generated user profiles from TV viewing history to retrieve candidates from an item pool, subsequently leveraging LLM for item recommendations. (Liu et al., 2023) employed LLMs to create user profiles, exploring topics and regions of interest based on user browsing history, and integrated the inferred user profile to enhance recommendations. Unlike the approaches above that rely on text data from user history for recommendation generation, our novel proposal creates dynamic user profiles by leveraging screenshots of user interactions using MLLMs.

Tools and their integration with LLMs. : Researchers have made significant strides in using LLMs to tackle complex tasks by extending their capabilities to include planning and API selection for tool utilization. For instance, Schick et al. (2023) introduced the pioneering work of incorporating external API tags into text sequences, enabling LLMs to access external tools. TaskMatrix.AI Liang et al. (2023) utilizes LLMs to generate high-level solution outlines tailored to specific tasks, matching subtasks with suitable off-the-shelf models or systems. HuggingGPT Shen et al. (2023) harnesses LLMs as controllers to manage existing domain models for intricate tasks effectively. Lastly, Qin et al. (2023) proposed a tool-augmented LLM framework that dynamically adjusts execution plans, empowering LLMs to complete each subtask using appropriate tools proficiently. Li et al. (2023) introduced the Optiguide framework, leveraging LLMs to elucidate supply chain optimization solutions and address what-if scenarios. In contrast to the aforementioned approaches, *InteraRec* harnesses the power of LLMs to generate item recommendations by converting user behavior summaries into mathematical formulations that can be readily interpreted by an optimization solver.

3. The *InteraRec* Framework

Addressing the challenge of offering real-time recommendations to users is a multi-step process. It typically commences with thorough data collection and analysis, followed by the careful selection and training of appropriate user behavior models. After a rigorous evaluation, the trained model is deployed to generate optimal recommendations. The user is then empowered to make a decision, choosing whether to act upon or disregard the presented recommendations.

Our framework, *InteraRec*, depicted in Figure 2, begins by systematically capturing screenshots of a user's browsing activity on a designated webpage at regular intervals. Subsequently, an MLLM, with its inferential capabilities, translates user interaction behavior into summary information comprising predefined keywords. This information is then processed by an LLM equipped with function-calling ability, transforming the summary into specific constraints and executing optimization tools following a validation check. The generated solutions are communicated to the user via the LLM through a user interface. The process outlined above is organized into multiple stages: 1) Screenshot generation, 2) Behavioral summarization, and 3) Response generation.

3.1. Screenshot generation

InteraRec initiates an automated script to systematically capture high-frequency, high-quality screenshots of the web pages users navigate during a browsing session. Notably, the screenshots are confined to the user's current viewing area on the screen

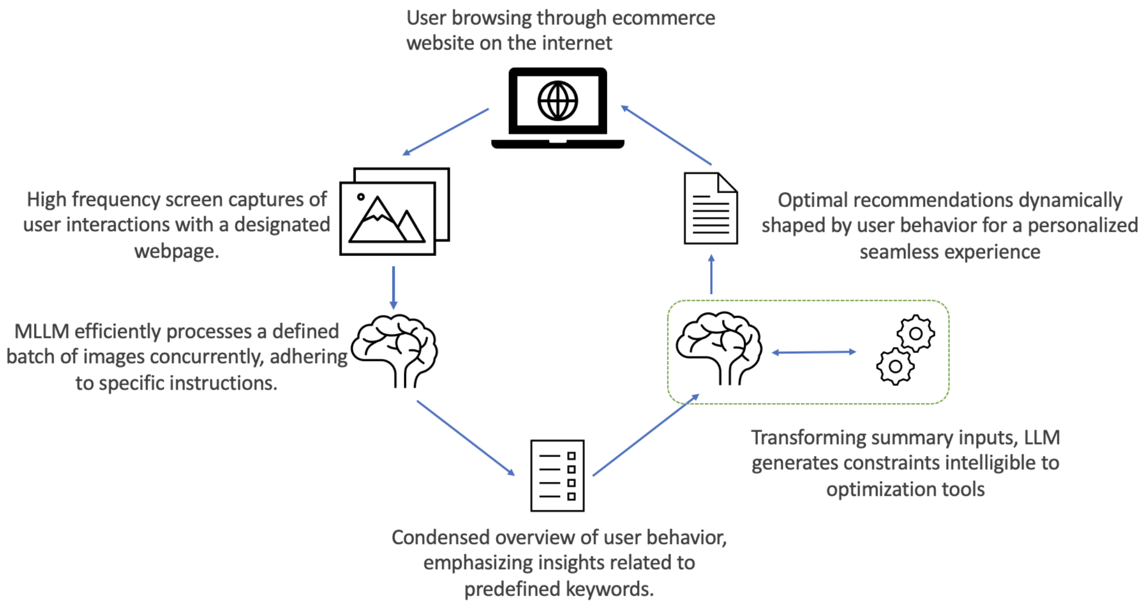

Figure 2: Overview of *InteraRec* framework delivering real-time personalized recommendations

rather than encompassing the full scrollable content of each webpage. This allows for efficient targeting of the visible content that users actively engage with. *InteraRec* then stores these representative screenshots in a database for further processing and analysis.

3.2. Behavioral summarization

In this stage, *InteraRec* sequentially processes a finite number of screenshots in real-time using an MLLM to analyze and provide detailed responses. Specifically, *InteraRec* instructs the MLLM to succinctly summarize user interaction behavior across predefined categories such as product characteristics, lowest and highest prices, brand preferences, product specifications, user reviews, comparisons, and promotions as shown in Figure 3. This precise instruction is crucial to filter the information and capture elements directly aligned with user interests. Generic instructions like "describe the user behavior on screenshots" or "explain the differences in screenshots" yielded responses that lacked specificity and failed to reflect user preferences. If the model cannot deduce information for a given category, *InteraRec* generates a response indicating 'does not know' or 'not applicable.' Additionally, *InteraRec* ensures that the model presents information in a summarized JSON format, facilitating its use as a constraint or filter for subsequent processing.

```
response = client.chat.completions.create(
model="gpt-4-vision-preview"
messages=[
{
"role": "user"
"content": [
{
"type": "text"
"text": "What can you infer from the images below \
with regards to a user preference in the following categories? \
Product Characteristics, \
Lowest Price, \
Highest Price, \
Brand Preference, \
Product Specifications, \
User Reviews and Testimonials, \
Comparisons, \
Promotions. \
Write a response that contains the above information. \
if any of the categorical information is unavailable, \
mark it as not available "}...],...}
```

Figure 3: Guidelines for MLLM to generate a summary of user interactions using predefined keywords

3.3. Response generation

The keyword-based summary generated from the previous stage is rich in information, forming a valuable resource for generating recommendations. The relevant information to be extracted from the summary depends on the capabilities of recommendation methods that can incorporate them. In this study, *InteraRec* harnesses the *InteraSSort* framework (Karra and Tulabandhula, 2023) to deconstruct the user behavior summary into relevant constraints and uses them to solve an assortment optimization problem to generate optimal recommendations. Specifically, *InteraRec* leverages the function-calling capabilities of the LLM to decompose the user interaction summary into relevant constraints using appropriate functions as shown in Figure 4. Subsequently, rigorous validation checks are conducted, encompassing range and consistency assessments of the decomposed constraints. *InteraRec* maintains an extensive database containing parameters for discrete choice models estimated using historical purchase transactions. By leveraging these choice model parameters and additional constraints as arguments, *InteraRec* executes optimization scripts employing tools like optimization solvers to generate optimal solutions. Ultimately, *InteraRec* empowers the LLM to incorporate these results as input, generating user-friendly personalized product recommendations.

```
function_descriptions = [
    {
    "name": "get_user_recommendations",
    "description": "Generate dynamic  recommendations based \
                    the summary of user behavior",
    "parameters": {
    "type": "object", "properties": {
        "lowest_price": {
            "type": "integer",
            "description": "get lowest price preference of user."
        ,},
        "highest_price": {
             "type": "integer",
             "description": "get highest price preference of user."
        ,},
        "color": {
            "type": "string",
            "description": "get color preference of user",
        }, }, }, }, ]
```

Figure 4: Prospective function for decomposing user interaction summaries.

4. Illustration

In this section, we discuss the components needed to run our experiments, followed by an illustrative example.

4.1. Assortment planning

The assortment planning problem involves choosing an assortment among a set of feasible assortments ($\mathcal{S}$) that maximizes the expected revenue. Consider a set of products indexed from 1 to n with their respective prices being $p_1, p_2, \cdots p_n$. The revenue of the assortment is given by $R(S) = \sum_{k \in S} p_k \times \mathbb{P}(k|S)$ where $S \subseteq \{1, ..., n\}$. The expected revenue maximization problem is simply: $\max_{S \in \mathcal{S}} R(S)$. Here $\mathbb{P}(k|S)$ represents the probability that a user chooses product k from an assortment S and is determined by a choice model.

The complex nature of the assortment planning problem requires the development of robust optimization methodologies that can work well with different types of constraints and produce viable solutions within reasonable time frames. In our experiments, we adopt a series of scalable efficient algorithms (Tulabandhula et al., 2022) for solving the assortment optimization problem.

4.2. Multinomial logit (MNL)

The MNL model (Luce, 2012) is one of the most extensively studied discrete choice models and is frequently utilized across various marketing applications. The parameters of the MNL model are represented by a vector $\mathbf{v} = (v_0, v_1, \cdots v_n)$ with $0 \leq v_i \leq 1 \ \forall i$. Parameter $v_i, \ 1 \leq i \leq n$, captures the user preference for purchasing product i. Under this model, the probability that a user chooses product k from an assortment S is given by $\mathbb{P}(k|S) = v_l/(v_0 + \sum_{k' \in S} v_{k'})$.

4.3. LLM

We employ the `gpt-4-vision-preview` (GPT-4V) and `gpt-3.5-turbo` variants from the GPT model series as our MLLM and LLM, respectively. Both the models are publicly accessible through the OpenAI API [1].

4.4. Illustrative example

Through this illustrative example, we present the dynamic process of generating personalized, real-time recommendations for users navigating the *Asos* fashion apparel website. The recommender system (*InteraRec*) initiates by capturing screenshots every 2 seconds, as depicted in Figure 5, and stores them in the database. Following this, *InteraRec* employs GPT-V to process these images in batches of 10 due to its input token size limitation and generate a summary based on provided instructions, as illustrated in Figure 6(see Appendix A for more examples). In our approach, we focus on extracting the following set of constraints, specifically the

[1] `https://platform.openai.com/`

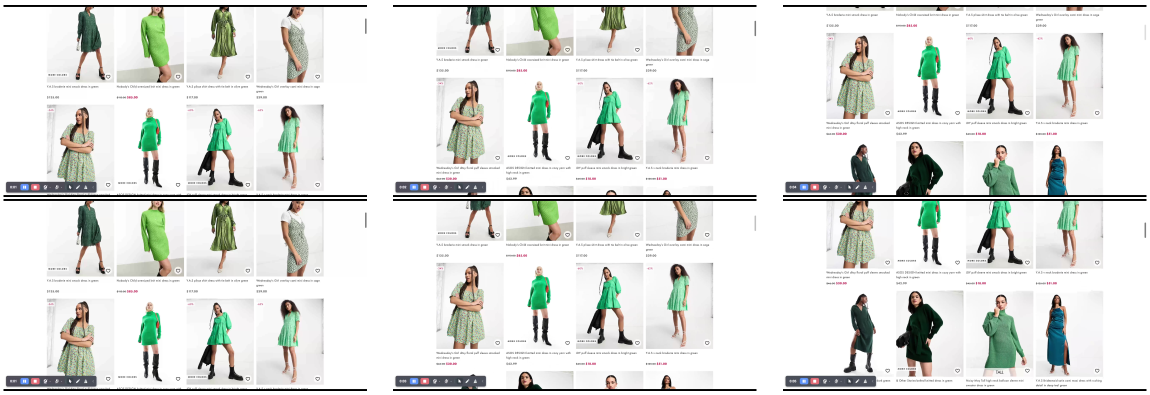

Figure 5: Six screen captures showcasing user interactions with the Asos.com website, each captured at a 1-second interval.

Given the images shown, it's important to note that while certain inferences can be made about product preferences and price sensitivity among other things, these are assumptions based on a set of images featuring green dresses of different styles and prices. Here is a possible interpretation in JSON format: “json { *“Product Characteristics”*: “User appears to be interested in dresses, specifically in shades of green, with various styles including mini, midi, and maxi lengths.”, *“Lowest Price”*: “18.00$ for the JDY puff sleeve mini smock dress in bright green”, “Highest Price”: “144.00$ for the Y.A.S Bridesmaid satin cami maxi dress with ruching detail in deep teal green”, *“Brand Preference”*: “Variety of brands shown, such as Y.A.S, Wednesday's Girl, Nobody's Child, suggesting no clear preference for a single brand.”, *“Product Specifications”*: “Not Available”, *“User Reviews and Testimonials”*: “Not Available”, *“Comparisons”*: “The user may be comparing dresses based on style and discounts, with several items showing reduced prices.”, *“Promotions”*: “Several dresses are listed with percentage discounts, indicating that promotions may influence user preferences.” } ”
The response indicates an interest in green dresses, sensitivity to promotions (as indicated by discounted items), a range of price points indicating a potentially broader budget range or comparison shopping, and a variety of brands.

Figure 6: Extracted summary from *Asos* website activities revealing a user preference for green dresses, diverse brand interest, and price sensitivity.

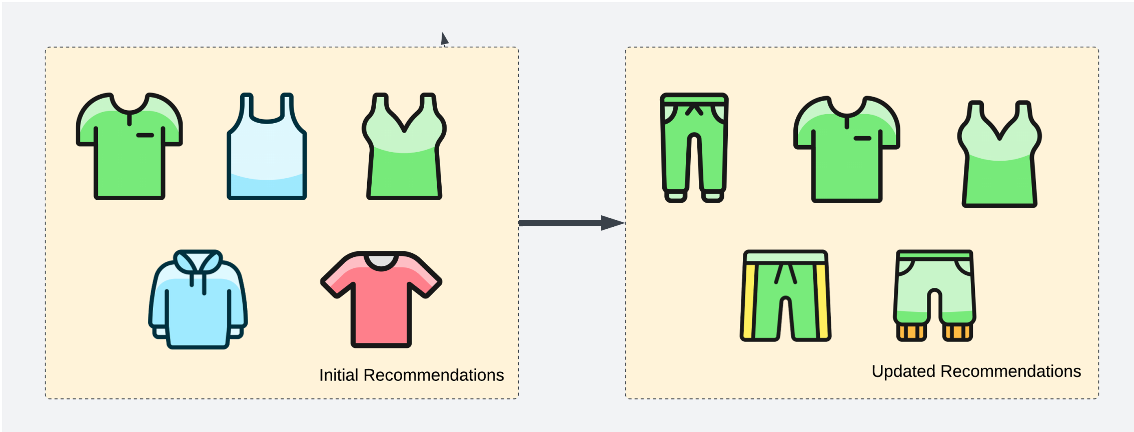

Figure 7: *InteraRec* Real-time recommendations incorporating the dynamic user behavior.

color and price range of products. Subsequently, the system parses the input, extracting details such as color ('green'), lowest price as '18$ and highest price as 144$') by leveraging the function-calling ability of the `gpt-3.5-turbo` model. Using the parsed information as arguments, *InteraRec* executes the MNL optimization script and communicates outcomes through the interface. We can observe the impact of incorporating user behavior in the form of updated recommendations limited to green apparel, as shown in the Figure7. This methodical, step-by-step approach ensures a consistent refinement of recommendations through continuous analysis of successive image batches.

5. Conclusion

In this paper, we introduce *InteraRec*, an interactive framework designed to craft personalized recommendations for users browsing an e-commerce platform. Overcoming the limitations of extensive weblog engineering, *InteraRec* utilizes screenshots to capture and understand user behavior. Harnessing the capabilities of both LLM and MLLMs, *InteraRec* leverages insights from the screenshots and intelligently executes suitable optimization tools, translating user behavior into concise, easily interpretable recommendations for users. Overall, *InteraRec* heralds a new era in recommendation systems driven by visual data.

References

M. Rosenstein, What is actually taking place on web sites: e-commerce lessons from web server logs, in: Proceedings of the 2nd ACM conference on Electronic commerce, 2000, pp. 38–43.

W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, arXiv preprint arXiv:2303.18223 (2023).

Z. Chen, Palr: Personalization aware llms for recommendation, arXiv preprint arXiv:2305.07622 (2023).

Q. Liu, N. Chen, T. Sakai, X.-M. Wu, A first look at llm-powered generative news recommendation, arXiv preprint arXiv:2305.06566 (2023).

T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, T. Scialom, Toolformer: Language models can teach themselves to use tools, arXiv preprint arXiv:2302.04761 (2023).

Y. Liang, C. Wu, T. Song, W. Wu, Y. Xia, Y. Liu, Y. Ou, S. Lu, L. Ji, S. Mao, et al., Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis, arXiv preprint arXiv:2303.16434 (2023).

Y. Shen, K. Song, X. Tan, D. Li, W. Lu, Y. Zhuang, Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface, arXiv preprint arXiv:2303.17580 (2023).

Y. Qin, S. Hu, Y. Lin, W. Chen, N. Ding, G. Cui, Z. Zeng, Y. Huang, C. Xiao, C. Han, et al., Tool learning with foundation models, arXiv preprint arXiv:2304.08354 (2023).

B. Li, K. Mellou, B. Zhang, J. Pathuri, I. Menache, Large language models for supply chain optimization, arXiv preprint arXiv:2307.03875 (2023).

S. R. Karra, T. Tulabandhula, Interassort: Interactive assortment planning using large language models, arXiv preprint arXiv:2311.12241 (2023).

T. Tulabandhula, D. Sinha, S. Karra, Optimizing revenue while showing relevant assortments at scale, European Journal of Operational Research 300 (2022) 561–570.

R. D. Luce, Individual choice behavior: A theoretical analysis, Courier Corporation, 2012.

Appendix A.

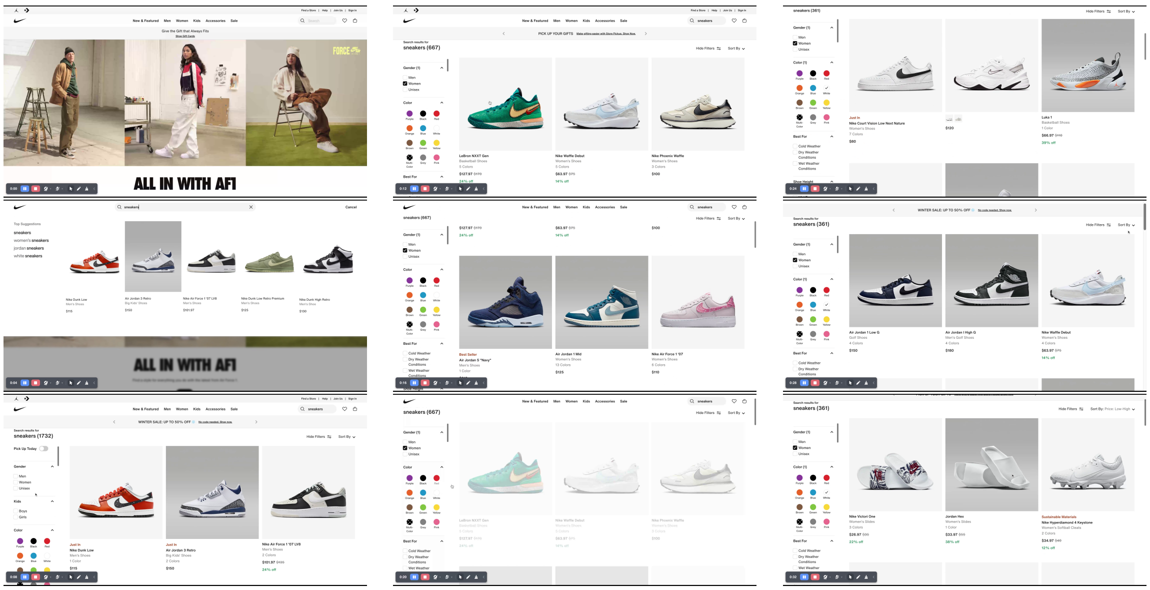

Figure A.8: Nine screen captures showcasing user interactions with the Nike.com website, each captured at a 4-second interval.

"json *"Product Characteristics"*: "The products featured are mostly casual and sports footwear, with a focus on style and functionality.", *"Lowest Price"*: "$63.97, as seen on the Nike Waffle Debut Womenś Shoes.", *"Highest Price"*: "$180, as seen on the Air Jordan 1 High G Menś Golf Shoes.", *"Brand Preference"*: "The brand showcased is Nike, including its Air Jordan line, indicating a preference for this brand.", *"Product Specifications"*: "Gender": "Men, Women, and Unisex options available", "Color": "Multiple color options are available for selection", "Best For": "options like Cold Weather, Dry Weather Conditions, Wet Weather Conditions are available, although specific preferences are not indicated in the images." , *"User Reviews and Testimonials"*: "Not available", *"Comparisons"*: "The images showcase different models and prices, suggesting that comparisons can be made within the selection, but specific comparative analysis is not displayed.", *"Promotions"*: "Promotions are visible, such as ẂINTER SALE: UP TO 50% OFFánd discounts on specific products.""

Figure A.9: Extracted summary of user preferences shaped by activities on the Nike.com website.

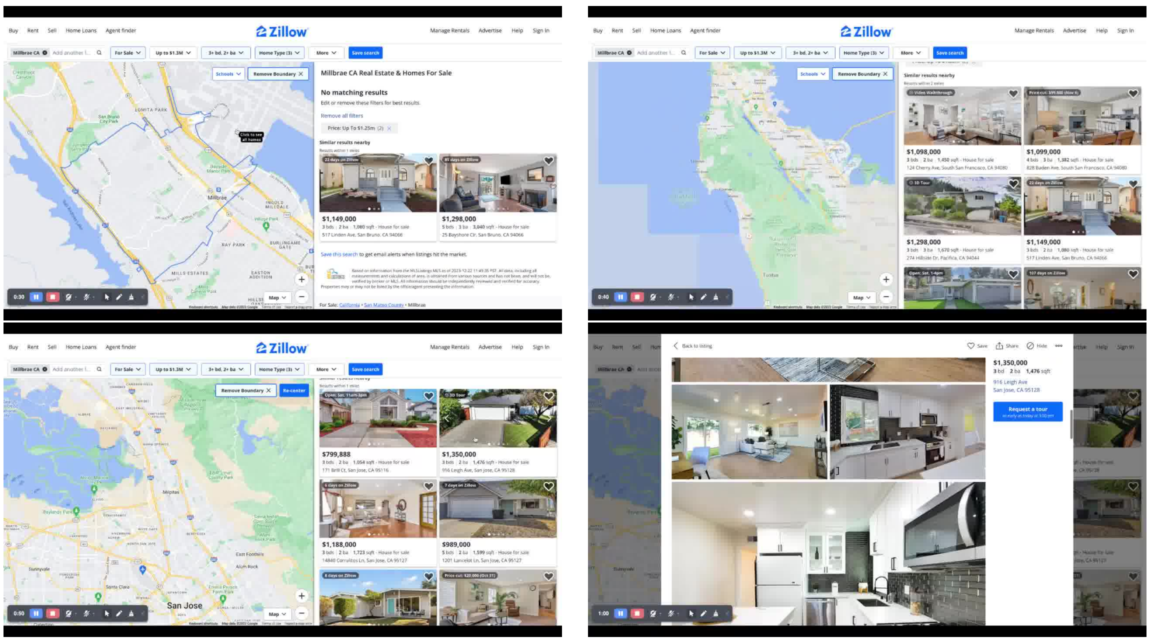

Figure A.10: Four screen captures showcasing user interactions with the Zillow.com website, each captured at a 10-second interval.

"Based on the provided images, it appears that the user is searching for real estate properties for sale on a website, likely with certain preferences. The images show a website with real estate listings that include price ranges, the number of bedrooms and bathrooms, square footage, and location. Using the data visible in the images, I will attempt to create JSON-formatted data that represents the userś inferred preferences within the limitations of the images. "json *"Product Characteristics"*: "Location": "Millbrae, CA; South San Francisco, CA; Pacifica, CA; San Jose, CA", "Property Type": "House", "Bedrooms": "3+", "Bathrooms": "2+" , *"Lowest Price"*: "$799,888", *"Highest Price"*: "$1,350,000", *"Brand Preference"*: "not available", *"Product Specifications"*: "Square Footage": "Low": ",054 sqft", "High": "3,040 sqft" , *"User Reviews and Testimonials"*: "not available", *"Comparisons"*: "Available": true, "Criteria": "Price, Location, Size, Bedrooms, Bathrooms" , *"Promotions"*: "Price Cut": "$20,000 (Oct 31) on a property listed for $989,000" ''Please note that the information above is constructed from the context provided by the images and might lack accuracy in details that are not visible or provide insufficient context. Some categories like B́rand Preferenceánd Úser Reviews and Testimonialsd́o not apply in the context of real estate listings and therefore are marked as "not available." Also, the promotion information is based on limited data indicating a price cut on a specific property, as seen in the images."

Figure A.11: Extracted summary of user preferences shaped by activities on the Zillow.com website.