REVIEW SCIENTIFIC PAPER
DOI: 10.63395/STEDConf14022025940R223

Submitted: 22.08.2025.
Accepted: 12.09.2025.

DESIGN OF RECOMMENDER SYSTEMS IN E-COMMERCE

Darko Radulović, Marija Blagojević

University of Kragujevac, Faculty of Technical Sciences, Svetog Save 65, 32102 Čačak, Serbia
marija.blagojevic@ftn.kg.ac.rs

Coresponding author: Marija Blagojević, University of Kragujevac, Faculty of Technical Sciences, Svetog Save 65, 32102 Čačak, Serbia, marija.blagojevic@ftn.kg.ac.rs

ABSTRACT

This research presents the design of a recommender system for an online store. The system utilizes user behavior data and purchase history to generate personalized product suggestions. Its main components include a recommender server and client modules. The recommender server performs the core recommendation logic independently of the online store, while the client modules integrate with the store's interface to display recommendations to users. For data analysis and recommendation generation, the system employs a combination of K-Means clustering and collaborative filtering. K-Means clustering groups users based on similarities in their purchase history, while collaborative filtering suggests products based on the purchases of other users within the same cluster. The paper also addresses implementation challenges encountered during system development, including the selection of appropriate libraries and performance optimization. The results indicate that the system provides effective product recommendations and shows strong potential for future enhancements, such as the incorporation of additional user data and exploration of more advanced recommendation algorithms. Future work may focus on algorithm optimization, dataset expansion, and experimentation with emerging technologies.

Keywords: recommender system, e-commerce, K-Means clustering, collaborative filtering, personalized recommendations

INTRODUCTION

In modern business, online stores face the challenge of personalizing the customer experience and increasing sales efficiency. Recommender systems represent one of the key solutions to this problem as they enable the display of products that are most relevant to each user based on their previous activities and interests. This paper focuses on the design and implementation of a recommender system for an online store, with the aim of improving the user experience and optimizing the purchasing process.

The aim of this paper is to investigate and implement techniques for generating personalized product recommendations using data about users and their purchase history. Based on these data, the system will provide recommendations that can increase the chances of additional sales as well as enhance customer satisfaction.

To implement the system, two key techniques were used: K-Means clustering and collaborative filtering. K-Means clustering is an algorithm that groups users into clusters based on the similarity of their purchases, while collaborative filtering enables the generation of recommendations based on the activities of other users within the same cluster. These techniques allow recommendations to be based on the behavior of similar users, thereby achieving a high level of personalization and relevance.

The paper provides a detailed explanation of the methodology used for data analysis, system design, and system implementation, with an emphasis on challenges encountered and solutions applied during development.

MATERIALS AND METHODS

K-Means clustering is one of the most popular unsupervised learning algorithms, used to group data into clusters based on their similarity (Lu et al., 2012). The main idea behind K-Means clustering is to divide the data into k clusters, where each cluster represents a group of data points that are similar to each other, while being different from those in other clusters. The algorithm consists of several key steps:

- Initialization: Selection of the initial number of clusters (k) and random selection of initial cluster centers.
- Cluster assignment: Each data point is assigned to the closest cluster center.
- Update centers: The center of each cluster is recalculated as the mean of the data within that cluster.
- Iteration: The process of assigning data to clusters and updating centers is repeated until convergence is achieved, i.e. until the positions of the cluster centers stabilize.

In the context of recommender systems, K-Means clustering is used to group users based on their purchase history. Users within the same cluster share similar preferences and behaviors, which enables more accurate and effective product recommendations based on the activities of users from the same cluster (Basu et al., 1998).

Advantages of K-Means clustering:

- Simplicity and efficiency: K-Means is easy to implement and it is computationally efficient, especially when applied to large datasets.
- Scalability: The algorithm is scalable and capable of processing large amounts of data.
- Ease of interpretation: K-Means facilitates the interpretation of user groups, as each user is assigned to a single cluster, and the center of the cluster can offer insights into the characteristics of users within that group.

Disadvantages of K-Means Clustering:

- Predefined number of clusters: The number of clusters (k) must be specified in advance. Selecting an appropriate k can be challenging, as too few or too many clusters can result in poor recommendations.
- Sensitivity to initialization: K-Means is sensitive to the initial placement of cluster centers, which can lead to suboptimal results if the initialization is not well chosen.
- Lack of flexibility in cluster shape: K-Means assumes that the clusters are spherical and evenly sized, which can be problematic when dealing with non-linear or irregularly shaped data distributions.

Collaborative filtering is a technique used to recommend products based on the behavioral data of other users. This method relies on the assumption that users who have exhibited similar preferences or behaviors in the past tend to have similar interests in the future (Lu et al., 2012). Collaborative filtering can be divided into two main types (Schafer et al., 2001):

- User-based filtering: Recommendations are generated based on similarities between users. If User A has similar preferences to User B, the products purchased by User B has can be recommended to User A.
- Product-based filtering (Item-based): Recommendations are generated based on similarities between products. If a user has already purchased product X, other products frequently purchased together with X may be recommended.

In both cases, the analysis of user behavior, such as previous purchases, product ratings or clicks on products, is crucial in order to create relevant recommendations (Chen et al., 2004).

Advantages of collaborative filtering:

- Simplicity: Collaborative filtering does not require a detailed understanding of the product or user, as it relies only on the history of interactions and behavior.
- Personalization: Recommendations are highly tailored to each user, as they are based on their previous activities and preferences.

Disadvantages of collaborative filtering:

- Cold start problem: For new users or new products, collaborative filtering may struggle to generate recommendations because due to insufficient data about users or products.
- Data sparsity: In large systems, user and product data can often be very sparse (e.g., a user may have purchased only a few products), which can lead to poor recommendations.
- Scalability: Collaborative filtering can become computationally intensive in a system with large numbers of users and products, especially when using user-based filtering.

By combining K-Means clustering and collaborative filtering, it is possible to create a recommender system that uses the advantages of both approaches and minimizes their disadvantages (Aljunid, & Huchaiah, 2022).

RECOMMENDER SYSTEM DESIGN

The design of the recommender system for e-commerce is based on a two-component architecture, consisting of a server and a client component (Lourenco and Varde, 2020). Each of these components plays a specific role in the process of generating and displaying recommendations to users.

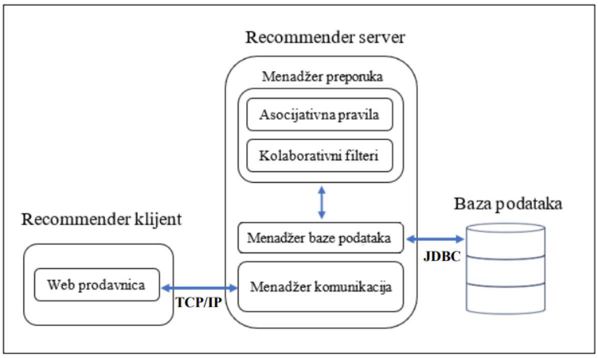

Figure 1. Recommender system design.

The server-side component represents the core logic of the system, responsible for processing data and generating recommendations. This component operates independently of the online store itself, which enables flexibility and easier integration with different platforms. The server uses data about users, their purchase history and preferences to generate personalized recommendations using K-Means clustering and collaborative filtering. The server's algorithms ensure that recommendations are based on relevant data and optimized for each user.

The client-side component is integrated into the online store and is responsible for interacting with the user. It communicates with the server to gather recommendations and display them to users. The client component is typically embedded within the store's user interface, such as personalized recommendation sections on the home page or within product pages. Communication between the server and the client is provided through an API, which enables sending referral requests and receiving responses in a format suitable for user-facing display.

The server component plays a key role in the recommendation generation process. Based on data about users and their previous purchases, the server uses algorithms to analyze and predict which products will be most interesting to individual users. By applying K-Means clustering, the server groups users with similar behavioral patterns, which enables personalized recommendations based on user activity within the same cluster. In addition, the server implements collaborative filtering to further enrich recommendations, using data on other users' interactions with products. This method allows the system to recommend products that other users, who are similar to the current user, have already bought or rated positively.

For each request sent by the client component, the server analyzes the available data and creates a list of recommended products. This process is done in real time, which allows the recommendations to be relevant and updated according to the current behavior of the user on the site.

Communication between the client and server components is provided through an API (Application Programming Interface), which enables efficient data exchange. The client component sends a request to the server to generate recommendations based on specific user data (such as purchase history, preferences, and browsing behavior), and the server responds with a list of products that match the user's requirements.

Communication is usually done in JSON (JavaScript Object Notation) format, a widely adopted standard for structured data transfer that allows for easy data handling on the client side. The client component then displays the recommended products on the user's interface in the form of interactive elements such as lists, carousels or product cards.

This interaction between the server and the client is crucial for providing a personalized and dynamic user experience. The server component provides the power of data analysis, while the client component ensures that users can easily access recommendations that match their interests and needs.

RECOMMENDER SYSTEM IMPLEMENTATION

For the implementation of the recommender system, a set of widely used tools and libraries is employed to enable efficient data processing, application of algorithms and user interface development. The key technologies used include:

- Pandas and NumPy: Used for data manipulation and analysis. These libraries facilitate simple and efficient data loading, filtering, and preprocessing, as well as working with large data sets in spreadsheet format.
- Scikit-learn: Used for implementing K-Means clustering and applying data-driven learning methods. Scikit-learn provides simple tools for training and evaluating models, as well as implementing algorithms such as clustering, regression, and filtering.
- Surprise: A library specialized in creating recommender systems. It is used to implement various collaborative filtering algorithms, such as user-based and item-based filtering. This library enables easy testing of different methods and their evaluation.
- Flask: This web framework is used to develop the server that manages the recommendation generation logic. It supports easy development of a RESTful API that communicates with client components in an online store and enables the generation of recommendations in real time.
- HTML, CSS, JavaScript: On the client side, basic web technologies are used to create the interface, with the aim of providing users with a simple and intuitive way to view recommended products.

Working with data is a key component in the implementation of a recommender system. At the initial stage, data about users and their purchase histories are loaded into a DataFrame using the Pandas library. The data undergoes a cleaning and normalization process [6]. The dropna() method is used to remove missing values, and the scaling() method is used to normalize numerical values.

For K-Means clustering, user purchase histories are encoded into numerical vectors, which are then used as input to the clustering algorithm. The data is structured in such a way that user similarities can be effectively analyzed and then assigned to appropriate clusters (Schafer, 2005).

For collaborative filtering, data about products purchased by users, as well as ratings and interactions with products (if available) are used. This data is organized into a matrix of users and products, where rows represent users and columns represent products, with the corresponding values indicating the level of user interest, such as purchases or ratings (Tareq et al., 2020).

The server component is implemented using the Flask framework, which provides a RESTful API that allows clients (online stores) to send requests to generate referrals. When a user visits a page, the client sends a request containing relevant user information (purchase history, demographic data, etc.), and the server generates recommendations using K-Means clustering and collaborative filtering (Sarvar et al., 2001). The results are then sent back to the user in the form of a JSON object containing a list of recommended products.

The client component is implemented using basic web technologies (HTML, CSS, JavaScript). Its primary function is to send user data to the server and display the received recommendations in the form of dynamically generated product cards. Additionally, the client interface facilitate further user interaction with the system, such as product browsing and rating (Candillier et al, 2007).

System testing includes several stages:

- Functionality testing: All basic system functionalities are tested, including the generation of recommendations and interactions between the server and client components. Tests include checking the accuracy and relevance of recommendations based on the user's purchase history.
- Performance evaluation: Metrics such as precision, recall and F1-score are used to evaluate the accuracy of recommendations. Moreover, the response time of the system is tested in real conditions, in order to ensure that the recommendations are generated in the shortest period of time.
- User Interface Testing: Tests are conducted with real users to ensure that the user interface is easy to use, and that the referral process is clear and intuitive. User feedback is used to further improve the interface design.

During the implementation of the system, several technical challenges were identified and addressed to enhance system performance and user satisfaction:

- Improving recommendation accuracy: One of the biggest challenges was achieving satisfactory recommendation accuracy. By combining K-Means clustering and collaborative filtering, the accuracy is improved, but there is still a need for additional parameter optimization (number of clusters, selection of filtering metrics).
- Management of large data sets: To ensure efficient processing of large-scale user and product data, optimized search and data filtering algorithms have been implemented. Techniques such as batch processing (splitting data into mini-batches) and parallel data processing were introduced to improve computational performance and reduce processing time..
- Scalability of the system: To support a growing number of users and products, the system was designed with scalability in mind. Caching mechanisms for recommendation results and a load balancing strategy on the server side were implemented. This allows system scalability without losing performance.

Through these challenges, the system has been gradually optimized to provide a better user experience and faster referral generation.

CONCLUSIONS

This paper presented an implementation of a recommender system in an online store, using a combination of K-Means clustering based on association rules and collaborative filtering to generate personalized product recommendations to users. Through data analysis and algorithm implementation, the system has achieved significant success in improving the accuracy of recommendations, especially in cases where users have similar purchase histories. By using K-Means clustering to group users based on the similarity of their purchases, and collaborative filtering to recommend products based on the behavior of other users, the system successfully provides relevant recommendations.

Although the implementation of the system gives satisfactory results, there are also challenges, especially related to the cold start problem, when new users do not have enough data for accurate personalization of recommendations. In future, improvements are possible in the form of using more advanced algorithms, such as deep learning, and additional data sources that would enable even more precise and effective personalization.

Further research can focus on the optimization of existing algorithms, the integration of new technologies and the analysis of new methods that will enable a better user experience and greater accuracy of recommendations. Moreover, the introduction of real-time data analysis can contribute to faster and more accurate data processing, which will further improve the performance of the recommender system.

ACKNOWLEDGMENT

This study was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, and these results are parts of the Grant No. 451-03-137/2025-03/200132.

DECLARATIONS OF INTEREST STATEMENT

The authors affirm that there are no conflicts of interest to declare in relation to the research presented in this paper.

LITERATURE

Aljunid, M. F., & Huchaiah, M. D. (2022). IntegrateCF: Integrating explicit and implicit feedback based on deep learning collaborative filtering algorithm. *Expert Systems with Applications*, *207*, 117933. https://doi.org/10.1016/j.eswa.2022.117933

Basu, C., Hirsh, H., & Cohen, W. (n.d.). *Recommendation as Classification: Using Social and Content-Based Information in Recommendation*. Retrieved May 26, 2025, from https://cdn.aaai.org/Workshops/1998/WS-98-08/WS98-08-002.pdf

Candillier, L., Meyer, F., & Boullé, M. (2007). Comparing State-of-the-Art Collaborative Filtering Systems. *Lecture Notes in Computer Science*, 548–562. https://doi.org/10.1007/978-3-540-73499-4_41

Chen, P.-Y., Wu, S., & Yoon, J. (2004). The Impact of Online Recommendations and Consumer Feedback on Sales. *ICIS 2004 Proceedings*, *58*. https://aisel.aisnet.org/icis2004/58/

Lourenco, J., & Varde, A. S. (2020). *Item-Based Collaborative Filtering and Association Rules for a Baseline Recommender in E-Commerce*. IEEE Xplore. https://doi.org/10.1109/BigData50022.2020.9377807

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., & Zhou, T. (2012). Recommender systems. *Physics Reports*, *519*(1), 1–49. https://doi.org/10.1016/j.physrep.2012.02.006

Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the Tenth International Conference on World Wide Web - WWW '01*, 285–295. https://doi.org/10.1145/371920.372071

Schafer, J. B. (2009). The Application of Data-Mining to Recommender Systems. *IGI Global EBooks*, 45–50. https://doi.org/10.4018/978-1-60566-010-3.ch008

Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, *5*, 115–153. https://doi.org/10.1023/a:1009804230409

Tareq, S., Noor, H. & Bepery. C (2020). Framework of dynamic recommendation system for e-shopping. *International Journal of Information Technology*, *12*(1), 135–140. https://doi.org/10.1007/s41870-019-00388-6