European Research Studies Journal
Volume XXVII, Special Issue 2, 2024

pp. 137-149

Self-learning Recommendation System Using Reinforcement Learning

Submitted 18/02/24, 1st revision 16/03/24, 2nd revision 20/04/24, accepted 16/05/24

Paweł Rymarczyk[1], Tomasz Smutek[2], Daria Stefańczak[3], Wiktor Cwynar[4], Sebastian Zupok[5]

Abstract:

Purpose: *This work aims to develop and evaluate a recommendation system using reinforcement learning methodology. This methodology uses RFM (Recency, Frequency, Monetary value) analysis to leverage customer segmentation. It also incorporates contextual aspects of purchasing, such as the day and part of the month, to enhance the accuracy of product recommendations by aligning offers with individual customer preferences and needs.*
Design/Methodology/Approach: *The study involves implementing and simulating customer behavior to initiate a self-learning process, which is crucial for adapting and optimizing recommendations in dynamic markets where direct customer feedback is limited. The methodology includes detailed customer segmentation using data on the last purchase, purchase frequency, and total expenditure from [specific data sources] to identify groups with different purchasing profiles. These profiles are integrated with contextual shopping data to define states representing distinct shopping scenarios. A self-learning process, facilitated by simulations, iteratively optimizes the value function to improve the system's ability to anticipate and meet customer needs.*
Findings: *The research shows that advanced customer segmentation combined with contextual analysis and reinforcement learning significantly improves recommendation system performance. The iterative optimization, which involves [specific process], increases the system's accuracy in anticipating and fulfilling customer needs.*
Practical Implications: *This approach significantly enhances the shopping experience and customer satisfaction by delivering more personalized recommendations. It also provides valuable insights for optimizing marketing and sales strategies across various e-commerce industries, thereby keeping the audience informed about the real-world applications of the research.*
Originality/Value: *This study offers a novel combination of customer segmentation, contextual analysis, and reinforcement learning. It demonstrates that this integrated approach can significantly improve recommendation system efficiency, thereby making a valuable contribution to the field of marketing strategies and customer-focused recommendations.*

[1]*Corresponding Author: Netrix Link sp. z o.o. , Lublin, Poland, e-mail: pawel.rymarczyk@netrix.com.pl;*
[2]*WSEI University, Lublin, Poland, e-mail: Tomasz.Smutek@wsei.lublin.pl;*
[3]*WSEI University, Lublin, Poland, e-mail: Daria.Stefanczak@wsei.lublin.pl;*
[4]*WSEI University, Lublin, Poland, e-mail: Wiktor.Cwynar@wsei.lublin.pl;*
[5]*Wyższa Szkoła Biznesu - National Louis University, e-mail: sebastian.zupok_xl@wp.pl;*

Keywords: *Reinforcement learning, RFM segmentation, recommendation system, contextual analysis.*

JEL codes: *C45, C61, C63, L86, M15, D83.*

Paper type: *Research article.*

1. Introduction

The recommendation system of wholesale customers, intermediaries, or resellers is critical to effective customer relationship management. Collecting data from various areas of intermediaries' activities allows for precise adjustment of the offer to their needs and preferences (Lakshmanan, Ng, and Ling, 2008; Wang, Du, Turk, and Liu, 2018; He, Zhang, Kan, and Chua, 2016). First, information regarding customer identification, such as the name of the intermediary's company and its key, is selected from the database.

In addition, company characteristics are also analyzed, including the type of activity (e.g., wholesaler, specialist store) and the size of the enterprise, expressed, among others, in the number of employees, annual sales, and annual turnover. An important aspect is also the date of establishment of the company, which may affect the communication strategy and approach to the customer (Rymarczyk and Kłosowski, 2018; Rymarczyk, Tchórzewski, Adamkiewicz, Duda, Szumowski and Sikora, 2017; Tyagi *et al.,* 2023; Nermend *et al.,* 2021).

Then, the system takes into account order frequency, order types, and the value of minimum payments, which allows the identification of key trends in customer purchasing behavior. Order history data is also analyzed, such as the year of the first and last order, the month of order, and the most frequently ordered product line.

Order values during and outside promotions are also critical in understanding customers' willingness to take advantage of promotional offers. Another essential element of the analysis is orders in various product categories and the time needed for the customer to decide to purchase a new product after its introduction to the market. This information allows you to assess the customer's flexibility and level of involvement in the product offer (Norena-Chavez and Thalassinos, 2023).

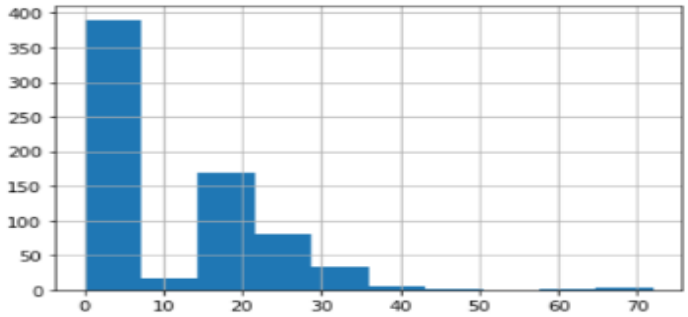
Additionally, shipping cost data and geographic information such as country, state or region, and city are included in the recommendation process. In the context of geographical data, variables such as the GDP per capita of a given area and the population of a given city are also included in the analysis. These additional factors allow for a better understanding of the context in which a given intermediary

operates, which may be crucial when making decisions regarding the commercial offer or marketing strategy (Zampeta and Chondrokoukis, 2023a).

Overall, the wholesale customer recommendation system is based on a wide range of data analyzed holistically, allowing the offer to be effectively tailored to each customer's needs and preferences. Thanks to this approach, companies can increase the efficiency of their commercial activities and strengthen relationships with key business partners.

Continuous data are then grouped into bins for later coding as categorical variables. Grouping is done based on histograms. Most variables are characterized by right-sided solid asymmetry. Below in Figure 1 is an example of a histogram

Figure 1. *Sales in the "Accessories" product category.*



Source: *Own creation.*

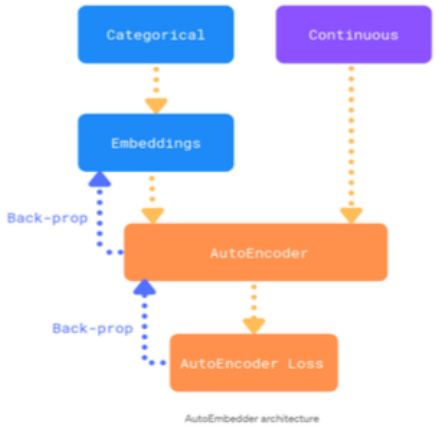
2. Embedding Autoencoder Model

Splitting a dataset into training and testing is a standard practice in machine learning to provide an independent assessment of model performance. In this case, the test set constitutes 20% of the entire data set, which allows for a reliable evaluation of the model's ability to generalize to new data.

An embedding autoencoder model, AutoEmbedder, was used to process categorical variables efficiently. AutoEmbedder is a valuable tool for converting categorical variables into a numerical representation, which enables further processing using machine learning algorithms.

The basic workflow of AutoEmbedder includes several key steps. First, the input data is properly prepared, which provides for removing missing values, normalizing numerical data, and coding categorical variables. Then, an autoencoder model, consisting of an encoder and a decoder, is built. The encoder is designed to transform the input data into a lower-dimensional space. At the same time, the decoder recreates the original data from the lower-dimensional space and returns it to the original space.

Figure 2. *Operational diagram of the embedding autoencoder*



Source: *Own creation.*

After building the model, it is trained on the training data to minimize the loss function. After training, the autoencoder model encoder obtains embeddings for categorical variables. These embeddings are low-dimensional representations of categorical variables that can be used for further analysis.

By using AutoEmbedder in data processing, it is possible to effectively encode categorical variables, which can significantly improve the performance of machine learning models, especially in the case of data containing many categorical variables with different levels of cardinality and cardinality. The operating diagram of this model is shown in Figure 2.

2.1 Encoding the Entire Set and Reducing Dimensionality

After training and verifying the correct operation of the embedding autoencoder, embeddings are generated for all clients in the dataset. Below is a section of the encoded data set, where each embedding is represented by a vector of dimension 132, consisting of 132 natural numbers (Figure 3).

To reduce the dimensionality of the data and facilitate further analysis, principal component analysis (PCA) is used. The number of components is selected in such a way as to explain at least 95% of the variance in the data. As a result, the data set is reduced to 39 variables, which allows retaining a significant part of the information in the data while reducing the dimensionality.

This dimensionality reduction process is crucial in data analysis because it preserves essential features of the data while eliminating redundant information and making the results easier to interpret. Thanks to this, it is possible to process and analyze

large data sets effectively, translating into a better understanding of the problem and more effective decision-making (.

Figure 3. *Data after extracting the main components*

	BusinessType_0	BusinessType_1	BusinessType_2	OrderFrequency_0	OrderFrequency_1	OrderFrequency_2	OrderMonth_0	OrderMonth_1	OrderMonth_2
0	-0.036051	0.012171	0.008651	-0.015262	0.016103	-0.020561	-0.005759	0.002461	-0.005658
1	0.017514	-0.011140	-0.014271	-0.000587	0.006925	-0.002392	-0.005759	0.002461	-0.005658
2	0.001059	-0.020903	0.004655	-0.003255	-0.009592	0.010488	-0.006564	-0.005564	-0.007062
3	-0.036051	0.012171	0.008651	-0.015262	0.016103	-0.020561	0.002322	0.007652	0.009084
4	0.017514	-0.011140	-0.014271	-0.000587	0.006925	-0.002392	-0.006564	-0.005564	-0.007062
...	...	...	...	...	...	...	...	...	...
696	-0.036051	0.012171	0.008651	-0.015262	0.016103	-0.020561	-0.005759	0.002461	-0.005658
697	0.017514	-0.011140	-0.014271	-0.000587	0.006925	-0.002392	0.002322	0.007652	0.009084
698	0.001059	-0.020903	0.004655	-0.003255	-0.009592	0.010488	0.004943	-0.001133	-0.005999
699	-0.036051	0.012171	0.008651	-0.015262	0.016103	-0.020561	-0.005759	0.002461	-0.005658
700	0.001059	-0.020903	0.004655	-0.003255	-0.009592	0.010488	-0.006564	-0.005564	-0.007062

701 rows × 132 columns

Source: *Own creation.*

2.2 Creating a Similarity Matrix

The matrix of similarities between customers plays a key role in creating a graph recommendation system. This matrix has dimensions corresponding to the number of customers, where each value in the ith row and jth column determines the level of similarity between the ith and jth customers. It is essential that these values range from 0 to 1, where 0 means no similarity and 1 means complete similarity.

Various methods are used in the process of creating a similarity matrix. The first one is the calculation of Euclidean distances between clients. Then, based on these distances, similarity is determined by subtracting the distance from one. This method assumes that the closer the points are to each other in space, the greater their similarity. Another method is to calculate cosine similarities.

This technique involves measuring the angle between vectors representing customers. The result of this operation is scaled to the interval [0,1] by adding one and dividing by 2. This method is often used in recommendation systems and text analysis. In addition, it is possible to use your formula to calculate similarities, which may take into account specific factors related to the data or application needs. The choice of a particular method depends on the characteristics of the data and business requirements.

Analyzing customer similarities is a key step in creating a graph recommendation system because it allows you to identify customers with similar preferences and purchasing behavior. Thanks to this, the system can generate accurate recommendations, which translates into better customer experience and greater profits for the company. Similarities were calculated using the formula:

$$2x(1 - F\left(a\ x\ ||x - y||^{\alpha}\right)) \tag{1}$$

where x, y are the feature vectors of the compared customers, ||, || means the quadratic norm, F is the distribution function of the N(0,1) distribution, and the coefficients a and α are selected so as to scale the values of the difference norms to the interval [0,4).

Calculation of similarities using the formula:

$$2(\frac{e^{z}}{e^{z}+1} - 0{,}5) \tag{2}$$

where $z = ||x - y||$, and subtracting 0.5 and multiplying by 2 scales the similarities to the interval [0,1].

After determining the cosine similarity matrix, the next step is to test which similarity calculation method works best. The cosine similarity matrix contains values representing the degree of similarity between customers, with each value scaled to the interval [0,1] to be consistent with the requirements of a graph recommendation system.

Subsequently, the generated similarity matrix is used to calculate the connection weights in the graph. The graph recommendation system is based on the analysis of the graph structure, where the vertices are individual customers, and the edges represent the relationships between them based on their similarity. Testing different methods of calculating similarities allows you to choose the one that best reflects customer relationships and leads to accurate recommendations.

Correctly selecting the appropriate method can significantly affect the effectiveness of the recommendation system and customer satisfaction. Therefore, the analysis of the cosine similarity matrix is a key stage in assessing and optimizing the operation of a graph recommendation system aimed at increasing the effectiveness of recommendations and improving user experience. The similarity matrix is then used in the graph recommendation system to calculate the weights of connections in the graph.

3. Searching Tables with Natural Language Queries Using the TAPAS Library

Semantic parsing, i.e., the process of translating natural language queries into SQL queries, is a solution often used in table search issues. This approach transforms queries formulated in natural language into appropriate SQL queries, which can then be executed in the database. For example, the question "How many champions are there with exactly one title?" can be translated into an SQL query like "select

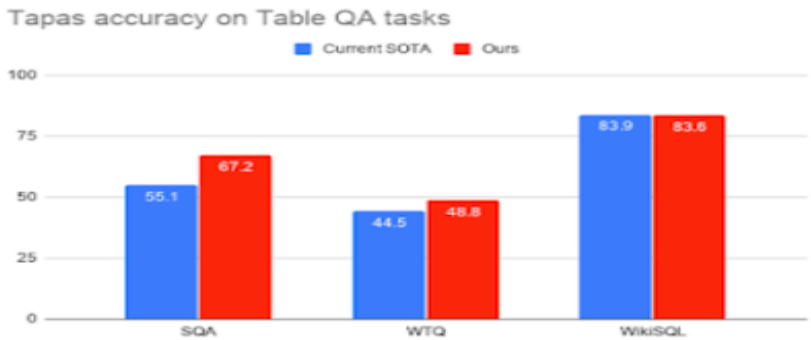
count(*) where column("Number of titles)==1", which can then be executed on the appropriate table. This approach has many advantages, including allowing users to formulate queries naturally and increasing the system's usability.

However, semantic parsing requires advanced engineering to generate syntactically and semantically correct queries. Understanding the query context and mapping natural language elements to appropriate SQL constructs is necessary. Additionally, it can be difficult to scale this approach to handle arbitrary queries, especially when the questions are more complex or involve different tables in the database.

Despite these challenges, semantic parsing remains a valuable tool in many fields, especially where users want to formulate queries naturally, and the system must understand their intentions and provide appropriate answers. Semantic parsing can be particularly useful for questions about concrete tables, such as sports scores, allowing users to get the information they need quickly and intuitively.

Instead of creating a model dedicated to a single table style, this approach will enable you to create a model that works effectively for data from various domains. This model has been pre-trained on millions of tables from Wikipedia, allowing it to understand the structure of tabular data. The model is tested on SQA, WikiTableQuestions (WTQ), and WikiSQL. The model's performance is compared to the three best tabular data parsing models currently considered to be at the highest level of performance (Current SOTA) (Figure 4).

Figure 4. *Comparison of Tapas with the best SOTA models*



Source: *Own creation.*

This approach has several significant advantages. First, the model is trained on a wide range of data, allowing it to better understand the overall structure of tabular data. Second, testing on different datasets will enable you to evaluate the model's performance in various contexts and usage scenarios. Third, comparison with other best-of-breed models gives a clear picture of whether the new model can meet or exceed existing achievements in tabular data parsing.

This approach can provide significant benefits, such as improved model versatility and scalability and improved results for various tabular data types. It is also a step towards building more general and comprehensive models for processing tabular data that can be used in multiple fields and applications. Tapas is an effective tool for analyzing individual tables that fit entirely in memory.

However, when dealing with tables that are too large for available memory or databases consisting of multiple tables, it is necessary to process the data by aggregating or filtering the table. The goal is to extract only the essential data from the table before using Tapas. It is important to note that Tapas has limitations on the type of aggregations it uses. Structures with multiple aggregations, such as the question "Enter the number of actors with an average rating above 4", may not be interpreted correctly by this model. Nevertheless, Tapas performed well on three sets of questions and answers from Wikipedia without generating these errors.

This suggests that most questions did not require a complex form that could require multiple aggregation. Despite these limitations, Tapas remains an effective tool for analyzing tabular data, especially for single tables containing data available in memory. For more complex queries or data that require repeated aggregation, you may need to use other data analysis or preprocessing methods to obtain the desired results. TapasForQuestionAnswering is an artificial intelligence model designed to analyze and answer questions based on data tables.

This model can be used in different ways depending on the data set it was trained on, which affects its capabilities and limitations. The first method, SQA, allows the model to converse with the user by asking subsequent questions about the table. This is especially useful when, after answering one question such as "What is the name of the first actor?", we want to continue questions such as "What is his age?". In this mode, responses do not involve aggregation, and the model focuses on selecting appropriate cells from the table.

The second method, WTQ, allows you to ask more complex questions involving aggregation operations such as summing values or calculating averages. For example, the question might be, "What is the total number of goals that Cristiano Ronaldo has scored in his career?" In this case, the model operates in a weakly supervised regime, meaning it must learn to apply appropriate aggregation operators without direct guidance.

The third approach, based on the WikiSQL-supervised dataset, provides the model with strong supervision during training. It is fed specific aggregation operators, making it easier to learn and answer aggregation queries.

Each of these modes has its specific uses and works best in different contexts depending on the user's needs and the nature of the data.

4. Defining the Context of Customer Purchases

In a reinforcement learning recommendation system, a state represents a comprehensive information set that an agent must understand to interact with its environment and make optimal choices effectively. An example of this in the case of a customer making an online purchase is the combination of many variables that define the purchasing context.

In this scenario, the status may include customer segment information, which describes a customer's demographic or purchasing behavior, as well as the season or specific time of year, which is essential because different seasons can influence purchasing preferences. For example, the pre-holiday period usually generates different shopping behavior than the summer period.

Another condition element is the specific day of the purchase, which may also influence the type and quantity of products purchased, such as more excellent shopping activity on weekends or special promotional days such as Black Friday.

All this information, collected and analyzed together, allows the recommendation system to understand the current situation better, anticipate potential customer needs, and respond accordingly, offering products that are most appropriate at a given moment.

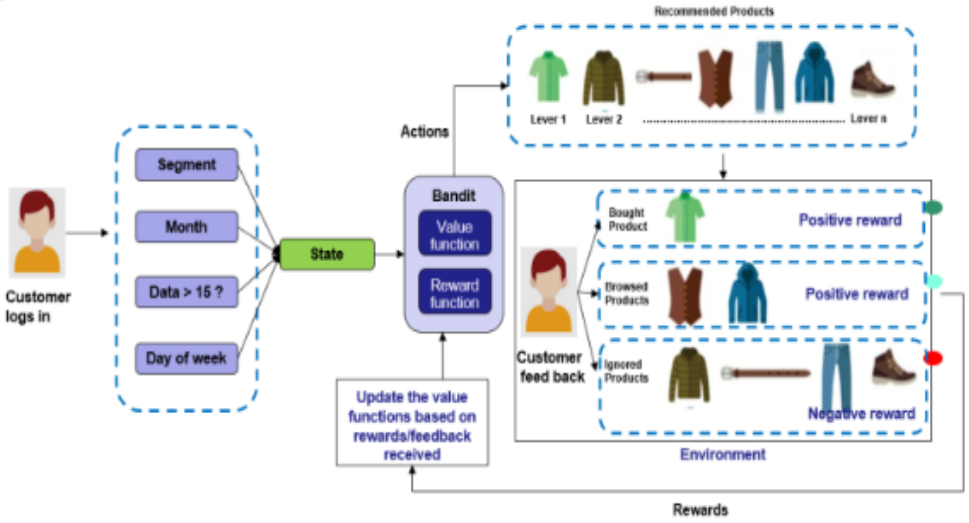
Exploratory data analysis is crucial in defining and updating what constitutes state in a reinforcement learning model because it allows you to identify and integrate new patterns and variables that may influence customer purchasing decisions.

Implementing a recommendation system based on reinforcement learning (Fig. 5) begins with defining the customer context, including the customer segment, month, half of the month, and day of the week. This detailed context allows you to create a defined state where the system can operate. Without direct access to customer data, you use simulations to generate this context, a solution to perform initial tests and train the model.

Once the status is established, the system moves to the product recommendation phase. In reinforcement learning, each recommendation is treated as the agent's action. These initial recommendations are based on a value function previously learned from historical data. This feature helps predict the most suitable products for a given customer in a specific context.

Then, the key element is customer feedback, which is simulated. The customer "decides" whether to buy the product, browse it, or ignore the recommended products. These interactions are interpreted as rewards that inform the system about the actions' effectiveness.

***Figure 5**. View statistics of the selected collection and sales leader*



***Source:** Own creation.*

This feedback allows you to assess how well the products have been tailored to the customer's needs and preferences. The final stage is the value function update, which is done by averaging the rewards received for specifically recommended products.

By iteratively adapting and improving this value function, the system gradually learns to understand better and predict what products will best meet customer needs in a given context. In this way, the agent becomes more and more advanced in selecting appropriate recommendations, which, over time, should lead to a more effective and accurate recommendation system.

4.1 Customer Segmentation

The beginning of the whole process is cleaning the data set. This is a key stage during which any errors, anomalies, or missing values that could affect further analysis and modeling quality are removed. Data cleansing ensures that subsequent stages of analysis will be based on a solid and reliable foundation. Then, customer segmentation takes place using the RFM method, i.e., analysis based on the time of the last purchase, frequency of purchases, and the value of funds spent.

This segmentation allows you to understand different types of customers better and adapt your recommendation strategy to them. The next stage is creating states for recommendations. In this phase, the system's operation context is defined, for example, by combining customer segment data with information about current market trends and purchase history. Creating states allows you to adapt recommendations to the client's current situation precisely. After defining the states, the system develops the reward system and value distribution.

This part of the process is crucial for the learning mechanism because the rewards are the signal for assessing how well the system copes with matching products to customer needs. The reward system must reflect the benefits of making accurate recommendations. The implementation of the self-learning process is the phase in which the system, using previously collected data and the developed value function, begins to test various recommendation strategies, learn from interactions with simulated customers, and gradually optimize its decisions.

The simple averaging method used in this context allows the value function to be updated based on the rewards received, which is crucial for the adaptation and evolution of the system. The last stage, the simulation of customer actions, is necessary because, in real conditions, the system does not receive direct feedback from users. These simulations allow for testing the system in controlled conditions and are crucial for assessing the effectiveness of recommendations and further improving learning mechanisms.

Applying RFM analysis to customer segmentation effectively prepares data to create different states in the recommendation system. RFM analysis focuses on three key indicators: Recency (R) – the time that has passed since the customer's last purchase; Frequency (F) – purchase frequency; and Monetary (M) – the total value of money spent. These three metrics allow you to identify and segment your customers in a way that can significantly impact the effectiveness of your product recommendations.

After defining and applying RFM analysis to customer segmentation, you can prepare a model to better predict and tailor offers to the needs of individual segments. Customers from different segments may have different purchasing preferences, meaning their state when interacting with the system will vary depending on the segment. Including information about the day of purchase and the part of the month in which the purchase occurred will allow for even more detailed customization of recommendations.

Shopping days can indicate specific behavioral patterns, such as more shopping activity on weekends or end-of-month promotions. Understanding these patterns is critical to optimizing your referral process and can lead to better effectiveness in meeting customer needs.

Consequently, each state in the recommendation system will be a unique combination of customer segment, purchase day, and part of the month. This structure of states allows for a dynamic and flexible approach to recommendations that can adapt to each customer's current circumstances and individual preferences. Therefore, preparing data from these three sources is a critical step in building an effective recommendation system.

5. Conclusions

This study presents an approach to developing a recommendation system using a reinforcement learning methodology that focuses on applying customer segmentation using RFM analysis and considering contextual factors such as the day of purchase and part of the month. Research results indicate that such a system can significantly improve the accuracy and relevance of product recommendations, adapting offers to individual customer needs and preferences.

The use of RFM segmentation enabled a precise understanding of the differences in the purchasing behavior of individual customer groups, which, combined with the analysis of the purchase context, such as the specific time and circumstances of the transaction, allowed for more effective and targeted recommendations. Essential was the use of information about the last purchase (Recency), purchase frequency (Frequency), and the total value of money spent (Monetary), which together defined customer profiles influencing their future purchasing decisions.

Additionally, integrating information about the day of purchase and part of the month allowed for the development of more nuanced recommendation strategies that effectively respond to changing consumption patterns in different periods. Simulations of customer actions used to initiate the self-learning process have shown that even without direct feedback from customers, the system can adapt and evolve, offering increasingly better product adjustments to user expectations.

As a result, implementing the recommendation system discussed can benefit both companies and customers by increasing shopping satisfaction and optimizing marketing and sales strategies. This study confirms that a complex approach to customer data analysis and reinforcement learning in product recommendation systems can significantly increase business efficiency and customer satisfaction.

References:

He, X., Zhang, H., Kan, M.Y., Chua, T.S. 2016. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 549-558.

Lakshmanan, L.V., Ng, R.T., Ling, C.X. 2008. A System for Behavioral Targeting. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 410-419.

Nermend, K., Łatuszyńska, M., Thalassinos, E.I. (Eds.). 2021. Decision-Making in Management: Methods and Behavioral Tools. Springer Nature.

Norena-Chavez, D., Thalassinos, E.I. 2023. Impact of big data analytics in project success: Mediating role of intellectual capital and knowledge sharing. Journal of Infrastructure, Policy and Development, 7(3).

Rymarczyk, T., Kłosowski, G. 2018. Application of neural reconstruction of tomographic images in the problem of reliability of flood protection facilities. Eksploatacja i Niezawodnosc - Maintenance and Reliability 20, 425-434.

Rymarczyk, T., Tchórzewski, P., Adamkiewicz, P., Duda, K., Szumowski, J., Sikora, J. 2017. Practical Implementation of Electrical Tomography in a Distributed System to Examine the Condition of Objects. IEEE Sensors Journal, 17, 8166-8186.

Tyagi, P., Grima, S., Sood, K., Balamurugan, B., Özen, E., Thalassinos, E.I. (Eds.). 2023. Smart analytics, artificial intelligence and sustainable performance management in a global digitalised economy. Emerald Publishing Limited.

Wang, Z., Du, T., Turk, G., Liu, B. 2018. Deep Reinforcement Learning for Sponsored Search Real-time Bidding. ACM Transactions on Intelligent Systems and Technology 9(4), 1-21.

Zampeta, V., Chondrokoukis, G. 2023a. Maritime transportation accidents: A bibliometric analysis. International Journal of Business and Economic Sciences Applied Research (IJBESAR), 16(1), 19-26.

Zampeta, V., Chondrokoukis, G. 2023b. A Comprehensive Approach through Robust Regression and Gaussian/Mixed-Markov Graphical Models on the Example of Maritime Transportation Accidents: Evidence from a Listed-in-NYSE Shipping Company. Journal of Risk and Financial Management, 16(3), 183.