

Article

A Reliable Prediction Algorithm Based on Genre2Vec for Item-Side Cold-Start Problems in Recommender Systems with Smart Contracts

Yong Eui Kim ^{1,†}, Sang-Min Choi ^{2,3,†}, Dongwoo Lee ⁴, Yeong Geon Seo ^{2,5} and Suwon Lee ^{2,3,5,*}

¹ Law School, Dong-A University, Busan-si 49236, Republic of Korea; yongeuk@dau.ac.kr

² Department of Computer Science, Gyeongsang National University, Jinju-si 52828, Republic of Korea; jerassi@gnu.ac.kr (S.-M.C.); young@gnu.ac.kr (Y.G.S.)

³ The Research Institute of Natural Science, Gyeongsang National University, Jinju-si 52828, Republic of Korea

⁴ Manager S/W Development Wellxecon Corp., Seoul 06168, Republic of Korea; dongwoo.lee@wellxecon.com

⁵ Department of AI Convergence Engineering, Gyeongsang National University, Jinju-si 52828, Republic of Korea

* Correspondence: leesuwon@gnu.ac.kr

† These authors contributed equally to this work.

Abstract: Personalized recommender systems are used not only in e-commerce companies but also in various web applications. These systems conventionally use collaborative filtering (CF) and content-based filtering approaches. CF operates using memory-based or model-based methods; both methods use a user-item matrix that considers user preferences as items. This matrix denotes information on user preferences, which refers to the user ratings for items. The model-based method exploits the fact that the input matrix is factorized. CF approaches can effectively provide personalized recommendation results to users; however, cold-start problems arise because both these methods depend on the users' ratings for items to predict users' preferences. We proposed an approach to alleviate the cold-start problem along with a methodology for utilizing blockchain that can enhance the reliability of the processes of the recommendations. We attempted to predict an average rating for a new item to alleviate item-side cold-start problems. First, we applied the concept of word2vec, treating each user's item-selection history as a sentence. Then, we derived genre2Vec based on the skip-gram technique and predicted an average rating for a new item by utilizing the vectors and category ratings. We experimentally demonstrated that our approach could generate more accurate results than conventional CF approaches could. We also designed the processes of the recommendation based on the concept of blockchain addressing the smart contract. Based on our approach, we proposed a system that can secure reliability as well as alleviate the cold-start problems in recommender systems.



Citation: Kim, Y.E.; Choi, S.-M.; Lee, D.; Seo, Y.G.; Lee, S. A Reliable Prediction Algorithm Based on Genre2Vec for Item-Side Cold-Start Problems in Recommender Systems with Smart Contracts. *Mathematics* **2023**, *11*, 2962. <https://doi.org/10.3390/math11132962>

Academic Editors: Pasquale De Meo and Zhao Kang

Received: 13 May 2023

Revised: 24 June 2023

Accepted: 30 June 2023

Published: 3 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the increase in the use of smart devices such as smartphones and tablets, many users are increasingly searching for various types of information through the web or mobile applications and purchasing products through e-commerce platforms. Accordingly, there is a vast amount of information not only in digital content-related services but also in various services that provide product sales, maps, and search results. Various studies on providing such large amounts of information more effectively to users by utilizing not only basic machine learning techniques but also deep neural network approaches have been conducted [1–8]. Services that require more personalized information, such as e-commerce

or digital content services, have raised the need for research on personalized recommender systems [1,5,8–12] to provide more accurate information catering to each individual user.

We tend to utilize acquaintances to obtain advice on the consumption of video content, books, and foods in our daily lives. This phenomenon appears similarly on the web. In e-commerce such as Amazon, the recommender system analyzes users' consumption patterns and provides similar items based on search results to boost product sales [13]. Video platforms, such as Netflix and YouTube, provide recommendations through a personalized home page; video content viewed by the user is analyzed, and the items recommended by the platform are organized on the home page to provide more suitable content to the user [14,15]. Additionally, when a user consumes new content, they are provided with personalized recommendation results. Like video platforms, music streaming services such as Spotify and Apple Music also provide personalized music content for each individual [16–18].

Recommender systems conventionally employ collaborative filtering (CF) and content-based filtering (CBF) approaches [19–21]. CF operates using memory-based or model-based methods [21,22]; both methods use a user-item matrix that treats user preferences as items. This matrix indicates the preference information, which indicates users' ratings for items [21,22]. CBF is a method that involves classifying and recommending users or items by analyzing the users' demographic information or item features that represent characteristics [21,23,24]. Numerous methods exist for CBF; the available information differs depending on the domain [25].

Recently, a recommender system with a variety of approaches using deep learning and MF has been proposed. Recently, a neural collaborative filtering (NCF) model wherein MF is developed into a deep neural network was proposed [26]. There are other methods based on the concepts of deep neural networks, such as using an autoencoder and item2vec [27–32].

Recommender systems that work based on CF offer the advantage of providing suitable information to users rapidly and individually according to their unique preferences. However, this method also has several disadvantages, including the cold-start problem, which decreases the reliability of the offered recommendations when there is little or no information regarding the user or item [33,34]. Moreover, the issue of the magic barrier, which means problems with noise when converting users' thoughts to a given scale, of recommender systems may arise when predicting user preferences based on numerical information, which makes it difficult to reflect 100% of user preferences [35,36]. Solving accuracy problems in recommender systems is a field of continuous research [1,22,37–39]. Problems such as cold start do not exist under certain CBF approaches because their working is not based on the user action history for items such as preferences. However, because CBF operates based on metadata, the recommendation reliability cannot be guaranteed. In other words, the extraction of significant features from various metadata requires more complexity than that offered by the CF approach since it is difficult to ensure the reliability of the predicted user preferences based on the results obtained under this approach [21,40–42].

In this study, we propose a method to predict the representative score of a new item by utilizing the metadata of the item and exploiting the structure of word2vec. It is a category vectorization method, based on the working of genre2vec, that alleviates the cold-start problem of a recommender system by considering the category information among the metadata of an item, and it can derive predictions using information on the category of new items. We treated the history of the item selected by the users as a document and the category of each item as a word. Subsequently, by applying skip-gram based on the user's item history, we could vectorize the category information like that which is done under word2vec. Under the proposed methodology, predictive scores can be derived using the vectors derived in this manner and the categories of new items. The proposed method allows us to alleviate the cold-start problem that is faced when new items are introduced.

In addition, we propose a blockchain system to enhance the reliability of the proposed method. E-commerce and media applications on the web are characterized by the fact that the users are unilaterally exposed to the contents by the information administrators. In this

process, there is no way for users to accurately know how the content is delivered to them, whether there is a specific biased situation or fabricated situation in the middle process, or whether the delivered content is caused by advertisements. To solve this problem, we propose not only a recommendation method for alleviating the cold-start problems but also a blockchain system to prevent ambiguity and data manipulation of content providers. To this end, an Ethereum-based smart contract is used [43,44]. Therefore, if we take advantage of these smart contracts for recommendation processes, we can provide users with content that exists on the web in the same form as a contract according to the automated code between the user and the information provider. Our contribution can be summarized as follows.

- We propose a method to predict the representative score of a new item by utilizing the metadata of the item: category information.
- We treat the history of the item selected by the users as a document and the category of each item as a word. Subsequently, by applying skip-gram based on the user's item history, we could vectorize the category information like done under word2vec.
- Based on Genre2vec, predictive scores can be derived using the vectors derived in this manner and the categories of new items.
- The proposed method allows us to alleviate the cold-start problem faced when new items are introduced. By applying the concept of CBF to CF, the recommendation accuracy in existing recommender systems can be improved.
- We proved that the proposed approach outperforms the conventional collaborative filtering-based approaches: the proposed approach was 33.3% more accurate than memory- and model-based CF in terms of mean absolute error (MAE) in best cases.
- We propose data processing systems based on the concept of blockchain and smart contracts for improving the reliability of recommendation processes.

The remainder of this paper is organized as follows. The related work is introduced in Section 2. In Section 3, the proposed algorithm is presented. The experiments and results are detailed in Section 4, and Section 5 provides the concluding remarks.

2. Related Work

2.1. Item-Side Cold-Start Problems in Recommender Systems

Recommender systems conventionally employ collaborative filtering (CF) and content-based filtering (CBF) approaches [19–21]. CF operates using memory-based or model-based methods [21,22]; both methods use a user-item matrix that treats user preferences as items. This matrix indicates the preference information, which indicates users' ratings for items [21,22].

Item-Side Cold-Start Problems

Recommender systems that work based on CF offer the advantage of providing suitable information to users rapidly and individually according to their unique preferences. However, this method also has several disadvantages. First, there is the cold-start problem, which decreases the reliability of the offered recommendations when there is little or no information regarding the user or item [33,34]; CF-based recommender systems operate based on information regarding the user selecting a specific item.

The item-side cold-start problem occurs for new items such as new movies or books in e-commerce services [2,8]. A new item could get excluded from the recommendation process and the results because users' reactions have not been provided. Studies wherein the cold-start problem has been alleviated using CBF, which provides recommendation results using various metadata that include the features of items or the demographic information of users [45–49], have been conducted. Regarding studies wherein the features of items are considered, the scores of new items have been predicted by performing classification or clustering in advance using metadata that include the categories of items [8,50–52].

2.2. Word Embedding for Recommender Systems

A recommender system with a variety of approaches using deep learning and MF has been proposed. There are methods based on the concepts of deep neural networks, such as using an autoencoder and item2vec [27–29]. These approaches leverage dimension reduction or embedding to produce recommendation results. Since deep learning approaches require the embedding of user preferences or item information, embedding not only the input matrix but also the item features or similarity of the user and item is recommended [30,31]. However, owing to the nonlinear structure of recommendation models, those that utilize these deep neural networks do not always provide recommendation results that are more precise in all situations than those provided by model-based approaches [32].

Research wherein text-embedding technology that can vectorize text data and perform classification or clustering through user or item features has been conducted [53,54]. There are various techniques related to text embedding, with the word2vec technique being currently used in many studies. This technique vectorizes each word by performing learning based on the frequency of the simultaneous appearance of words in a document, like that which is done under the skip-gram technique. Vectorizing words provides the advantage of enabling computations between words. Additionally, the distance and similarity between words can be derived more easily using these operations.

2.3. Blockchain and Smart Contracts

Blockchain is a decentralized, distributed ledger technology that allows for secure and transparent storage and the transfer of data. It is based on a peer-to-peer network of computers that collectively maintain a shared database, or ledger of transactions [55]. Each transaction in the blockchain network is verified by a network of nodes, or computers, and once verified, it is added to a block. This block is then added to the blockchain, forming an unalterable chain of blocks that contains a record of all transactions ever executed on the network. The security of the blockchain is achieved through cryptography and the consensus of the network. Each block in the chain contains a unique cryptographic hash that is calculated based on the transactions contained within it. This hash is then used to link the block to the previous one, forming an unbreakable chain [55]. Such blockchain technology can be applied in a variety of ways through a function called a smart contract.

A smart contract is a self-executing computer program that is stored on a blockchain and automatically enforces the terms of an agreement between two parties. It is essentially a computer program that executes the terms of a contract automatically, without the need for intermediaries [44,56]. Smart contracts are built on top of blockchain technology and are designed to be transparent, secure, and tamper-proof. Once a smart contract is created and deployed on a blockchain, it is immutable and cannot be modified, making it a powerful tool for enforcing contracts in a trustworthy and transparent way. It can also be used to create decentralized applications (dApps) that run on blockchain networks [44,56–58].

2.4. Background

There are studies that have examined recommender systems using CF and CBF. CBF has been implemented in various fields, including e-commerce, e-learning, news recommendations, and user preference analyses [59–63]. These approaches make use of item or user features, such as category information or demographic data [2,64–67]. The concept of CBF methods can alleviate the recommender systems faced various problems such as cold-start problems by utilizing item or user features [2,59,68–70].

Volkovs et al. [46] tackled the cold-start problem by combining content-based and neighbor-based models, drawing on the principles of CBF and CF. Their approach consistently produced satisfactory results during testing. Sun et al. [51] employed attribute data and preferences to cluster items. They created a decision tree that could be applied to both new and existing items, enabling them to predict preferences for new items. Moreover, studies have been conducted to improve the accuracy of various hybrid recommender

systems [10,23,71]. A Bayesian network model incorporating the user, item, and feature nodes was proposed [23]. This model was based on a combination of CF and CBF; it used various features to derive predictions through CF. A superior recommendation quality was provided using this model. User features were constructed based on the action history of the users, following which the similarities between the users and items (website content) were derived to recommend items [10]. There are also several studies that utilize deep learning techniques to improve the performance of point-of-interest (POI) recommendations [72,73].

Additionally, studies have focused on enhancing the accuracy of hybrid recommender systems [10,23,71]. One study proposed a Bayesian network model that incorporated user, item, and feature nodes [23]. This model combined CF and CBF, utilizing various features to make predictions through CF, resulting in improved recommendation quality. Another study constructed user features based on users' action history and derived similarities between users and items (website content) to provide recommendations [10]. Furthermore, deep learning techniques have been employed in several studies to enhance the performance of point-of-interest (POI) recommendations [72,73]. These studies leverage deep learning algorithms to improve the accuracy and effectiveness of recommending points of interest to users.

Chen et al. [12] proposed a hybrid recommendation algorithm that involved using Latent Dirichlet Allocation (LDA) topic modeling to reduce the dimension of user data. They generated a user-theme matrix to mitigate data sparsity in CF. Additionally, they employed the VGG16 deep learning model to extract feature vectors. These matrices and vectors were used as inputs for content-based recommender systems, from which recommendation results were derived. Duong et al. [28] developed a tag genome for movie data using natural language processing (NLP) techniques. They also proposed a three-layer autoencoder to create a more condensed representation of the tags. Subsequently, they utilized matrix factorization (MF) to provide recommendation results. Meel et al. [74] presented an approach to enhance CF accuracy by analyzing item features based on techniques such as word2vec and term frequency-inverse document frequency (tf-idf). They utilized singular value decomposition (SVD) to obtain recommendation results. The item features were analyzed based on CBF principles, and an embedding method was employed where items were analyzed using frequency-based methodologies and then applied to CF. Mehrabani et al. [75] introduced a method to extract item features as words using the NLP technique word2vec. They used the resulting vectors to calculate similarities between features. Based on these similarities, the proposed system derived recommendation results according to the principles of CBF.

Qi et al. [72] proposed a deep learning-based point-of-interest (POI) category recommendation model to alleviate the sparsity problem of collecting user location data and making recommendations. In this study, locality-sensitive hashing is utilized to classify users' personal information. In addition, the proposed method effectively manages users' long-term dependencies and interests based on the attention mechanism related to long-term short-term memory (LSTM) and the sliding window paradigm. Through these studies, the authors have shown that categories of POI can effectively elicit users' interests more than POIs based on improved recommendation performance. Liu et al. [73] conducted a deep learning-based study on mining between users and POIs using graph neural networks, which provides high-order connectivity between users and POIs but fails to consider the dynamic timeline. Therefore, this study proposes an Interaction-enhanced and Time-aware Graph Convolution Network (ITGCN) technique for more effective POI recommendation. This method learns the dynamic representation of users and POIs and embeds high-order connectivity into the node representation. The authors verify the superiority of the proposed model through comparative experiments.

There are also studies for applications that utilize blockchain and smart contracts. Ullah et al. [56] proposed a framework using blockchain smart contracts for managing real estate deals in smart cities. The authors explore literature published between 2000 and 2020, focusing on blockchain smart contracts in smart real estate, and propose a conceptual

framework for adoption in smart cities. In this study, the authors present decentralized applications and interactions with Ethereum virtual machines (EVMs) to show smart contract development that can be used for blockchain smart contracts in real estate. It also implements detailed design and interaction mechanisms for property owners and users to parties to smart contracts. The authors also propose a list of features for the initiation, creation, modification, or termination of smart contracts for real estate and step-by-step procedures for establishing and terminating smart contracts. Agrawal et al. [57] demonstrated a framework using smart contracts for supply chain collaboration. This work explored the design of a smart contract-based blockchain collaboration framework that supports operations for resource sharing across a wide range of networks or ecosystems. The authors also developed a demonstration framework for stakeholder interaction through procurement and distribution units that support blockchain technology. The proposed framework consists of a network architecture to demonstrate, rules on network work principles based on supply cooperation requirements, UML diagrams to define smart contract interaction sequences, and algorithms for verifying smart contract network. The applicability has been verified by deploying them on the Ethereum blockchain.

Besides Ethereum-based smart contract applications, there is an application system based on the concept of blockchain. Anitha et al. [58] proposed a reliable voting system using blockchain. The purpose of this research is to implement decentralized transparent voting systems based on the concept of blockchain. Based on the concept of blockchain, we can expect an efficient and highly secure method of election systems. The authors focused on secure voting systems, low costs, fast latency, and high scalability to build the system. The proposed decentralized application allows voters to vote comfortably in their homes, saving them time and reducing the number of false ballot registrations.

2.5. Analysis and Motivation

Current studies aim to predict users' item preferences using item features, alleviating the cold-start and sparsity problems [2,47,64,65]. In addition, deep learning methods are used to analyze item feature vectors and predict user evaluations [25,31,74]. Techniques such as frequency-inverse document frequency (tf-idf) or natural language processing (NLP) are employed to enhance recommendation performance [12,28,74]. These studies transform existing systems such as matrix factorization and analyze various item features to address cold-start issues in recommender systems.

Previous studies have improved recommendation performance and addressed existing issues. However, applying proposed techniques in real situations requires further analysis. Leveraging NLP techniques and test data to alleviate cold-start problems is challenging without sufficient text data [12,28]. To overcome these challenges, we propose using category information as metadata to predict ratings for new items, simplifying the process and addressing cold-start issues. The category information is generally provided with input items as metadata. Because of this reason, we suppose that this category information can be more meaningfully utilized for predicting user preferences for new items if it can be embedded in a form that can contain both user preferences and item information using NLP techniques.

Moreover, we aim to enhance the reliability of recommender systems for users by leveraging blockchain smart contracts. Smart contracts have been proven reliable in various fields like voting systems and supply chain collaboration [56–58]. By applying this structure to the content decision process of recommender systems, users can have more trust in the provided content. Managing the content decision process through smart contracts ensures that each step is executed through code, and the results are recorded in a blockchain system.

3. Proposed Approach

The basic idea of our approach is that category embedding is possible by characterizing a user's item-selection history as a sentence. This approach involved three steps. In the first step, a user category selection vector was generated by addressing the input matrix,

which was the user-item rating matrix. In the second step, genre2vec was derived based on the concept of word embedding for word2vec and the users' item-selection history. The third step involved the proposed method, wherein the average preference for a new item was calculated using genre2vec.

In addition, to improve the reliability of the recommendation, the user's preference information is stored on the blockchain, and the result of the recommendation is managed using smart contracts. Figure 1 shows the overall process of the proposed approach.

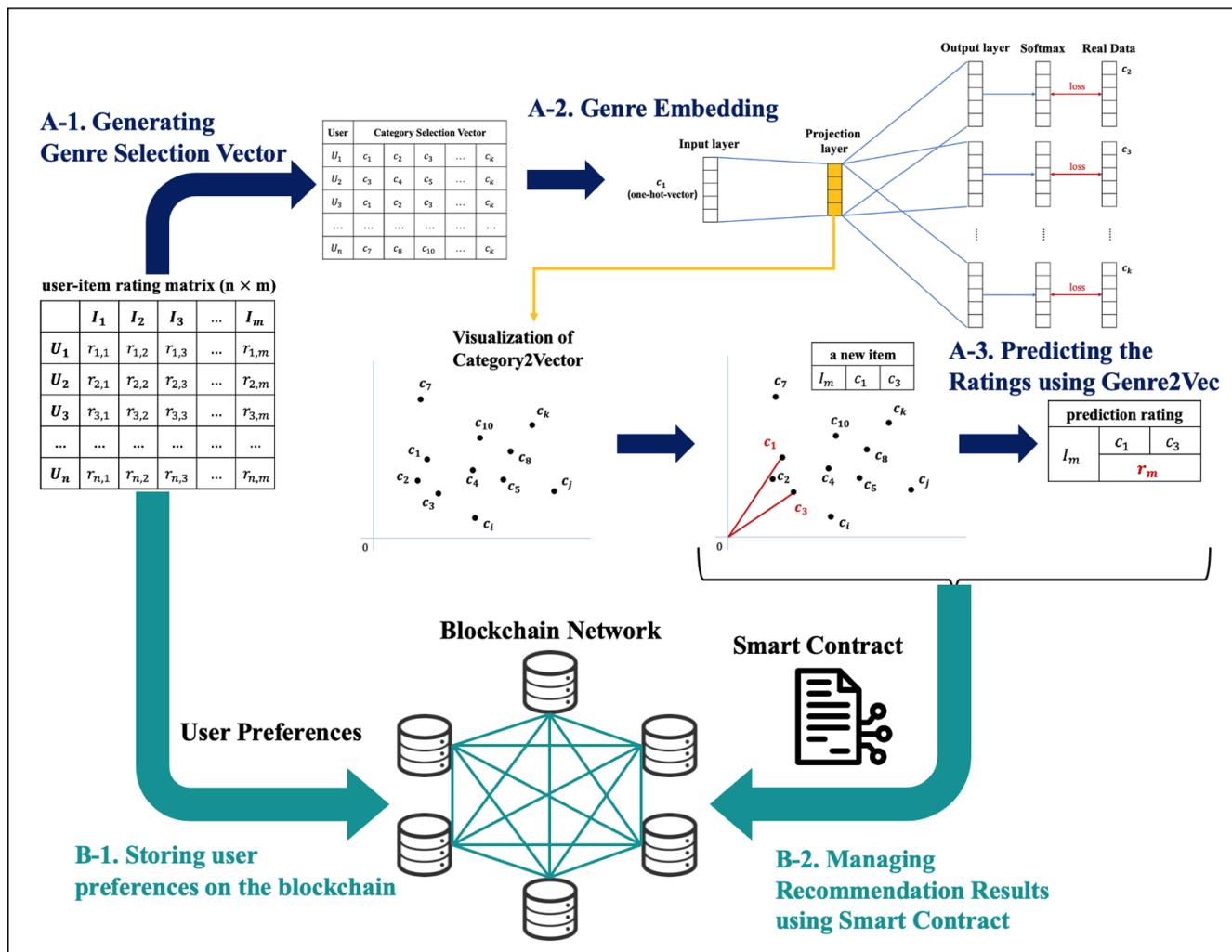


Figure 1. Overall process of our approach.

3.1. Generation of Users' Category Selection Vector

The first step involved generating a user category selection vector. The category selection vector refers to the vectorization of the categories in the items selected by the user. In this study, to apply word2vec, we considered the information of the item selected by a user as one sentence. For word embedding in word2vec, the words in each sentence were vectorized in a one-hot encoded vector. These vectors were learned using a methodology such as skip-gram to perform word embedding. To apply this approach, we first considered the category information of the items selected by a user in the input matrix (user-item rating matrix) as a sentence. Figure 2 shows an example of a category vector of an item selected by user U_1 in the input matrix, which is a user-item rating matrix.

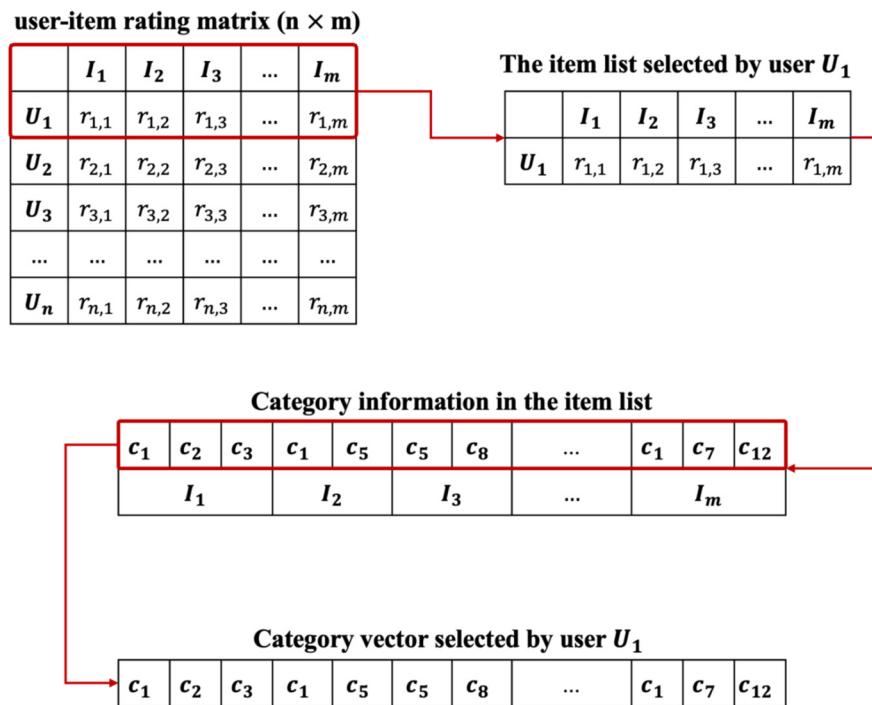


Figure 2. Example of a category vector of an item selected by user U_1 .

In Figure 2, we have considered that the user-item rating matrix represents an input. In this example, there are n users and m items. It was assumed that all the users scored all the items. In this case, there was a rating for the $n \times m$ matrix, as shown in the figure; $r_{n,m}$ denotes the rating given to the m^{th} item by the n^{th} user. In the user-item rating matrix in this example, each row and column indicated an evaluation of the user's item; the first row of the matrix indicated the rating for each item for user U_1 . Although the example in Figure 1 assumes that user U_1 has a rating for every item, in real cases, it is very rare for most users to have a rating for every item.

We could extract the item-selection history of user U_1 based on the first row, which represented the rating of each item of user U_1 . Items did not appear simultaneously in the rows for a single user. For example, in Figure 2, U_1 has $r_{1,1}$ points for item I_1 . In the rating vector for user U_1 , the U_1 row of the input matrix, the rating information for I_1 does not appear repeatedly. This may also be considered as the selection information for item I_1 for user U_1 ; the selection information for each item in the rating vector of a user needed to appear once.

We performed embedding using the category information of the item, considering each user's item-selection information as a sentence. The category information was metadata, and there were cases wherein each item had the same category information. Therefore, the frequency of simultaneous appearances for the category could be derived based on the item-selection information of a single user. If the embedding was performed based on the item-selection information of multiple users, the vectorization of the items could be conducted based on the frequency of the simultaneous appearance of the items. However, if embedding was performed this way, a methodology for predicting new item scores could not be derived; the category vector in Figure 2 represents the category information of an item selected by U_1 . We applied this method to extract the category-selection information for each user.

Figure 3 shows a user-category matrix extracted from the user category-selection information based on the user-item rating matrix. In this example, $u \times k$ was assumed to be the matrix size, where k may have different sizes for each user because the number of items selected by the users and the number of categories in each item are different.

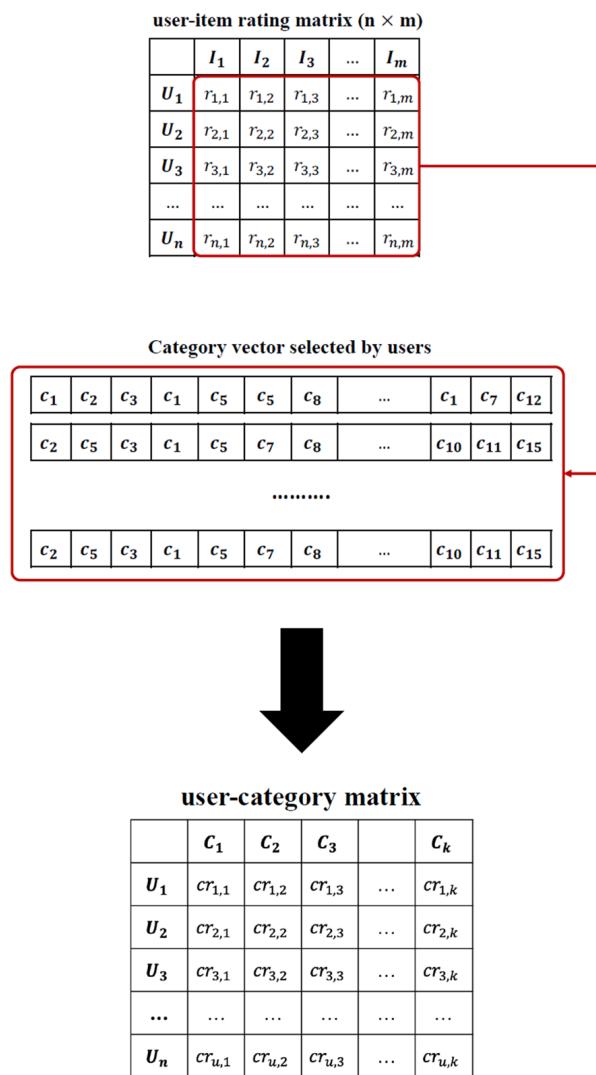


Figure 3. Example of a category matrix extracted from input matrix.

3.2. Generation of Genre2vec Using Concept of Word Embedding

Category embedding was performed based on the category-selection information, considering each row of the user-category matrix as one sentence. For this purpose, we assumed the users' category-selection information as a sentence and applied the skip-gram used in word2vec. By applying skip-gram to the category-selection information, the feature vectors of categories could be extracted based on the frequency of the simultaneous appearance of categories selected by the users. We considered the values in the projection layer updated in the learning process as the feature vector of the category.

In the user-category matrix, one-hot encoding was performed for a row, that is, for each dimension of a user's category-selection history. This implied that if one-hot encoding was performed for each dimension, a one-hot vector would be created for the categories selected by the user. Subsequently, the skip-gram method learned by predicting the surrounding words within a range based on the input word i.e., the central word. In this situation, the central word was considered a one-hot vector. Thus, in our method, the central word could be considered an input category, with the input becoming a one-hot vector for that category.

The neural network structure in a skip-gram is a multilayer neural network consisting of an input layer, a hidden layer, and an output layer. The hidden layer comprises k neurons, where k can be considered a vector dimension representing a word. In our approach, k was the dimension of a vector representing a category; words can be considered categories. The softmax function is used as the activation function for the neurons in the hidden

layer, and cross-entropy is used as the loss function. The result of the softmax function is a real number between zero and one, and the sum of the vectors is one. The resulting vector obtained through the softmax function can be considered as a score vector for multiclass classification.

A value between zero and one for the j^{th} index of this score vector indicated the probability of the j^{th} word being a surrounding word. The value of this score vector needed to also be close to that of the one-hot vector for the surrounding word, which was the vector corresponding to the label; as learning progressed, the value of the j^{th} index needed to be close to one. When the one-hot vector of a word was y , a cross-entropy function was used as a loss function to reduce the error between the two vector values. Learning was performed by placing a one-hot vector and a score vector of the surrounding words as input values in the cross-entropy function.

Figure 4 shows an example of the neural network structure of the skip-gram model for deriving genre2vec, wherein the category selection vector of user U_1 is treated as the input. The input was a one-hot vector of category c_1 , and the learning processes proceeded by considering the hidden layer of p -dimension as the projection layer. Here, the output layer was the learning result of the one-hot vector from c_2 to c_k , which surrounded the values of category c_1 , which was the input. This value was converted to a real value between zero and one using the softmax function. Based on cross-entropy, learning was conducted based on the difference from the one-hot vector of each actual category, and finally, a vector for c_1 was derived. After the learning process was completed, the p -dimensional projection layer was considered a vector of category c_1 .

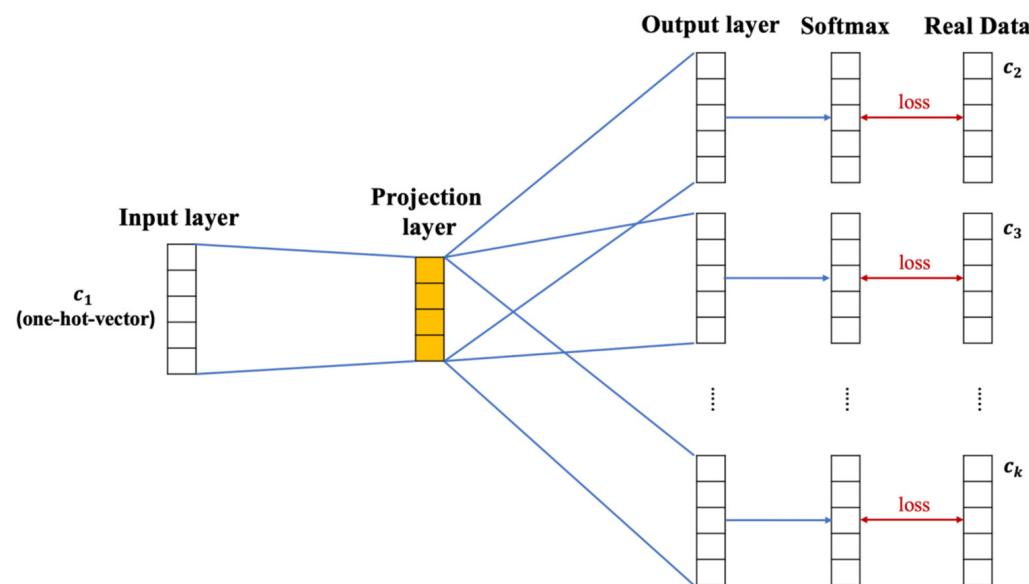


Figure 4. Example of a skip-gram model structure wherein the category selection vector c_1 is the input.

3.3. Predicting Average Preference of a New Item

We attempted to predict the average rating of a new item by utilizing the vector information from each category derived through genre2vec. We could predict the average rating for a new item and not the rating of a specific user by utilizing category vectors; skip-gram addressed the category-selection history by all the users in the processes for the vectorization of categories.

Under the skip-gram process of vectorizing categories, first, a one-hot vector was considered for each category as the input. Subsequently, these input vectors were derived from the categories extracted from each user's item-selection information. Additionally, because all the users' category-selection information was considered a single sentence, the vectorization process of each category was a learning process based on all this information.

Therefore, the vector of any category c_k derived from genre2vec resulted from the category-selection information of all the users in the database. The vector representation of category c_k could be considered as a value representing the category-selection information of all the users; the prediction using the vector derived through genre2vec could be considered as an average rating for the item. In this study, we considered the results of predictions for new items as average ratings.

Figure 5 shows an example of the process for deriving the average rating of a new item using a category vector. The first graph in Figure 5 shows an example of a two-dimensional representation of the vectors of each category derived using genre2vec. In this example, the new item is I_m , and it includes categories c_1 and c_3 as the metadata. The new items also contain category information because the categories are metadata.

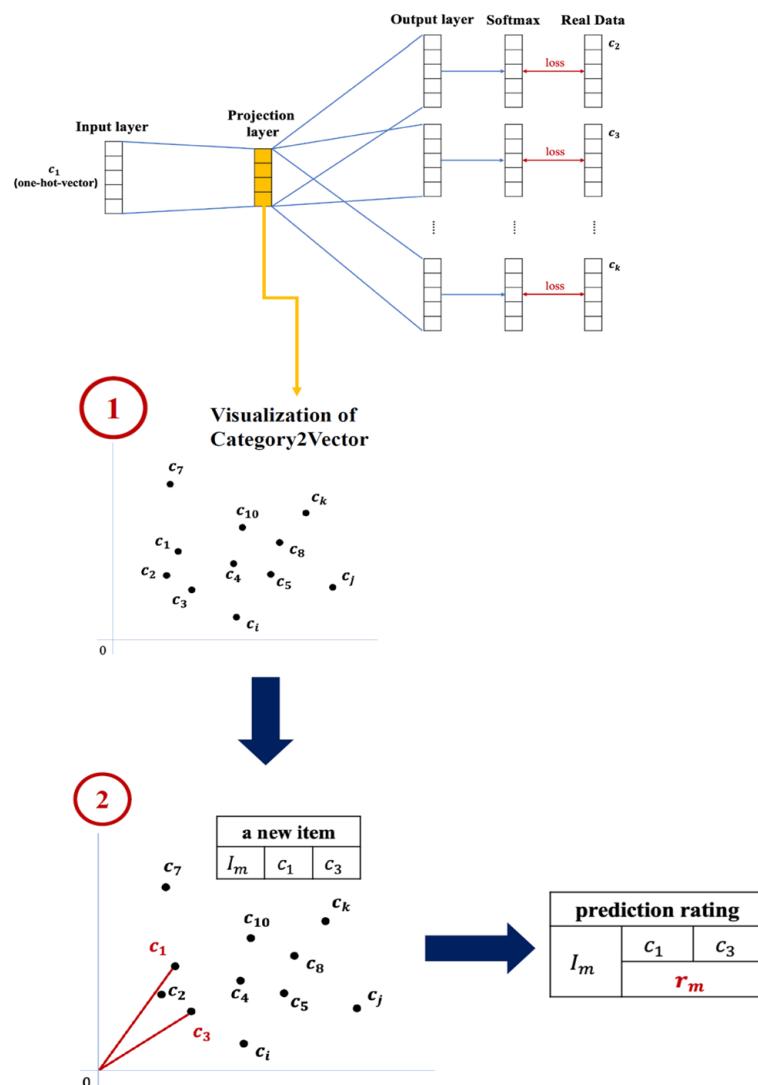


Figure 5. Example of predicting average preference of new item using genre2vec.

We derived the prediction results by utilizing the position on the coordinates of the category of the new item and the average rating of each category. Equation (1) represents the prediction process for a new item:

$$PI_k = \frac{\sum_{i \in C_k} (\sum_{j \in C_k} cr_j * sim(c_i, c_j)) / |C_k| - 1}{|C_k|}, \quad (1)$$

where C_k is the set of categories in item I_k ; cr_i is the category rating of category c_i ; and $\text{sim}(c_i, c_j)$ is the result of the similarity between categories c_i and c_j . Equation (2) shows the calculation processes for similarity:

$$\text{sim}(C_a, C_b) = \frac{\sum_{i=1}^n C_{ai} C_{bi}}{\sqrt{\sum_{i=1}^n C_{ai}^2} \sqrt{\sum_{i=1}^n C_{bi}^2}}, \quad (2)$$

where C_a and C_b are the vectors for the categories c_a and c_b , respectively; and C_{ai} and C_{bi} are the values in the i^{th} dimension of vectors C_a and C_b , respectively. The results of Equation (2) agree with cosine similarity.

Figure 6 shows an example of the process used to derive the prediction rating for a new item. In this example, the new item I_k has three categories: c_1 , c_2 , and c_3 ; each category has category ratings. Figure 6 shows an example of the process for deriving category ratings. The similarity between two categories was applied to the category rating of the criterion; in Figure 6, the first criterion is category c_1 . Then, we applied the similarities between c_1 and c_2 , and c_1 and c_3 to the category rating of cr_1 . We iterated this step in the last category.

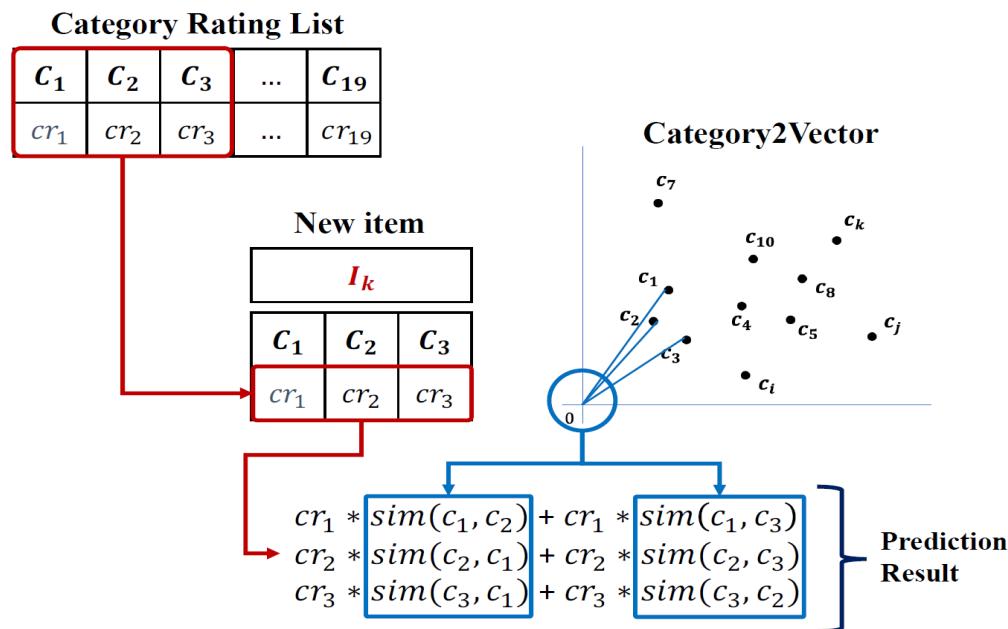


Figure 6. Example of the process for deriving a prediction rating for a new item.

In the prediction process, we utilized the category ratings derived from the input matrix. Figure 7 shows the calculation of the category ratings. In this figure, the user-item rating matrix represents the input for the approach. There was a category combination that consisted of more than one category for each item. For example, item I_1 had c_1 , c_2 , and c_3 as the category combination, which was metadata. The reason why we use category information for vectorization is that category information is metadata of items in a database. It means that the category information is provided by content or item provider without users' actions such as ratings for an item. Because of this reason, there can exist metadata such as category information for a new item even if the new item has no ratings by users in a database.

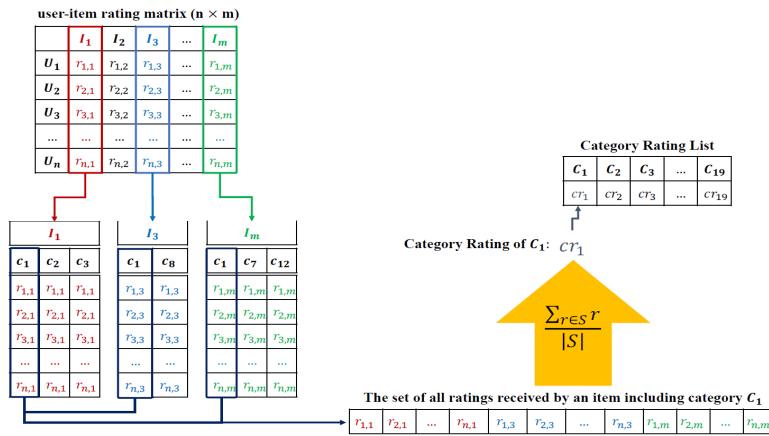


Figure 7. Example of the process for deriving category ratings.

Item I_1 had c_1 and c_8 , and Item I_m had c_1 , c_7 , and c_{12} as metadata. These three items had a common category, c_1 . We derived the rating of category c_1 by using the average of the ratings received from each user for the items containing the category. We calculated the average for the elements in the set of all the ratings received by an item, including that in category c_1 , using Equation (3):

$$cr_k = \frac{\sum_{r \in S_k} r}{|S_k|}, \quad (3)$$

where S_k is the set of all the ratings received by an item, which includes a category c_k , and r is one of the elements in the set S_k . The result of Equation (3) is the average of the elements in the set S_k . We derived category ratings from the MovieLens dataset using these steps. Algorithm 1 shows the process of predicting an average rating of a new item using genre2vec.

Algorithm 1 Predicting an average rating of a new item through Genre2vec

Require:

CRM = Category rating map

C2V = Category vector

NCL = category list (category combination) of new item

sim (c1, c2) = cosine similarity between c1 and c2

1: **function** predictAverageRating(CRM, C2V, NCL)

2: temp1 = 0

3: **for** c in NCL **do**

4: temp2 = 0

5: **for** cc in NCL **do**

6: **if** c ≠ cc **then**

7: temp2 += CRM[c] * sim(C2V[c], C2V[cc])

8: **end if**

9: **end for**

10: temp1 += temp2

11: **end for**

12: average = temp1 / size(NCL)

13: **return** average

14: **end function**

The algorithm shows the process for predicting the average rating of a new item using Genre2vec. To predict the average rating, we utilize category ratings, Genre2vec, and category information of a new item. In the algorithm, we first select the criteria category, then calculate the similarities between the criteria category and others. After that, we apply the similarity to the category rating of criteria. Lines 3 to 11 show the calculation processes.

Based on the calculation result, we derive the average rating of a new item by dividing the calculation result by the number of categories in a new item.

In the case of conventional matrix factorization, the user-item rating matrix, which means the input is factored and recombined to predict and derive recommendation results [22]. In comparison, the proposed method first performs embedding by utilizing the user's evaluation information and the metadata of the items. Therefore, our method first derives the embedding result for the genre information of each item and does not consider this result as a prediction as in the result of matrix factorization. In our method, we perform genre embedding based on existing users' item selection information and use the results to derive the prediction score of new items with the similarity of the genres in new items.

3.4. Recommendation Management Systems Based on Smart Contract

We construct a system that provides the recommendation data based on the algorithm predicting an average rating of a new item through genre2Vec to operate as a smart contract. Figure 8 shows the flow for the recommendation in general.

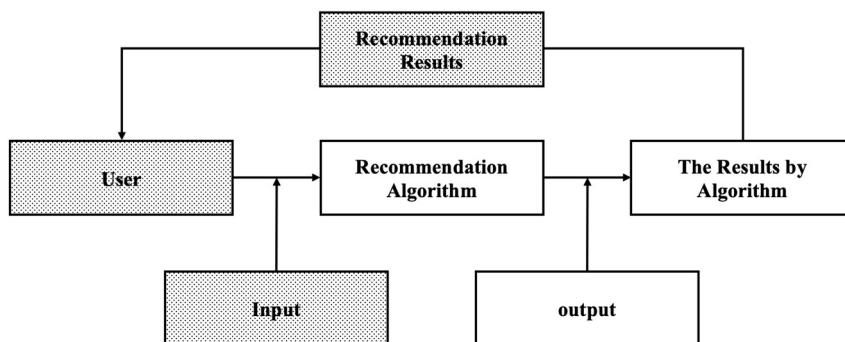


Figure 8. Recommendation flow for a user (a general case).

In Figure 8, the user provides input data such as evaluation or selection information of an item to the service provider's recommendation algorithm. The service provider generates a recommendation result through the algorithm and provides it to the user. In this case, the user does not know the processes other than input data and the recommendation result. Figure 9 shows the process of managing the same process in Figure 8 through a blockchain network.

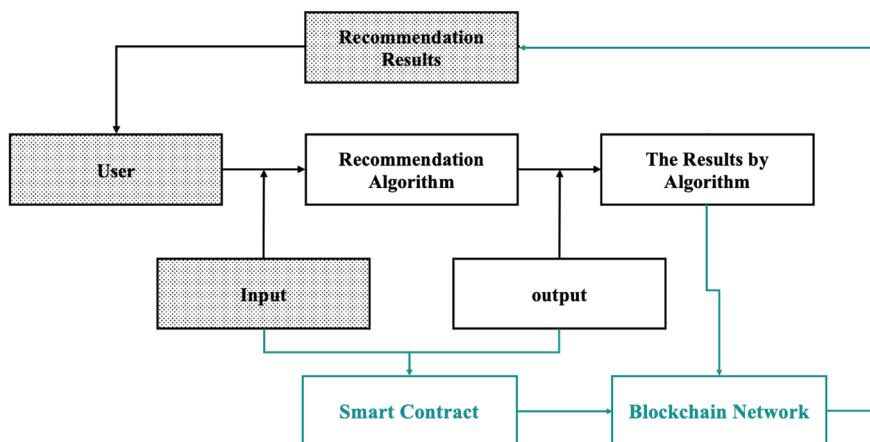


Figure 9. Recommendation flow for a user with blockchain smart contract.

In the structure of Figure 9, like the structure of Figure 8, information other than input data and recommendation results is not provided to the user. However, due to the nature of blockchain networks, it is impossible to manipulate the recommendation results.

In the conventional recommender system, the data holder and the algorithm provider are the same as the service manager. Because of this reason, it has a problem regarding

the fact that the user cannot know the processes of recommendation results. That is, it can cause a problem of social reliability different from the experimental accuracy of the recommendation results. To address these problems, we record the results of recommendations on a blockchain network and build a way to compare the results of new recommendations with existing ones. We construct a system that can ensure transparency in recommendation results and processes by providing data from recommendation systems based on smart contracts. As shown in Figure 9, the user's input and recommendation results are recorded on the blockchain network through a smart contract, and the stored recommendation results are provided to the user. We construct a smart contract using Equation (4) and derive the reliability score of the content provider.

$$rs_k = \frac{\sum_{t_k \equiv r_k} k}{\sum_{i=1}^k i}, \quad (4)$$

where t_k is the k^{th} set of items stored in the blockchain network provided to the user, and r_k is the k^{th} set of items derived through the recommendation algorithm.

We calculate the information at the point in time as a cumulative sum when the set of items derived from the recommendation algorithm is equal to the set of items stored in the blockchain network. The reliability score is derived by dividing the result of this calculation by the cumulative sum at the time. Therefore, the reliability score has a real number between 0 and 1.

4. Experiments

4.1. Database

As shown in Table 1, we employed the MovieLens dataset, which comprises 9125 movies and 671 users. A movie database provides genre information as an item feature. All the movies in the database had at least one genre, and each movie had a genre combination. For example, the genres for "Toy Story" were "Animation," "Children's," and "Comedy." Table 2 presents the 19 genres in the database.

Table 1. MovieLens dataset.

Dataset	Attribute	Explanation
Movie dataset	MovieID, Title, Genre	9125 movies
Rating dataset	UserID, MovieID, Rating, Timestamp	100,004 ratings provided by 671 users

Table 2. Genres in MovieLens dataset.

No	Genre	No	Genre	No	Genre
G_1	Action	G_8	Drama	G_{15}	Sci-Fi
G_2	Adventure	G_9	Fantasy	G_{16}	Thriller
G_3	Animation	G_{10}	Film-Noir	G_{17}	War
G_4	Children's	G_{11}	Horror	G_{18}	Western
G_5	Comedy	G_{12}	Musical	G_{19}	IMAX
G_6	Crime	G_{13}	Mystery		
G_7	Documentary	G_{14}	Romance		

4.2. Experimental Process

We first applied genre2vec based on the category of items in the MovieLens dataset to proceed with the experiments. Then, we implemented vectorization for the genre information in the MovieLens dataset. Next, we predicted the average rating of a new item

using vectorized category information. To verify the accuracy of the proposed method, these results were compared with those under existing CF approaches.

Recent studies have shown that the MF method still performs well in general situations [32]. In addition, the study analyzing the comparison results of various deep learning methods has also verified the efficiency of MF in many cases [32]. Therefore, in this paper, we verify the performance of the proposed methodology through a comparison with various forms of MF techniques.

4.2.1. Results of Genre2vec Based on MovieLens Dataset

For each category, which refers to the genre information in the MovieLens dataset, genre2vec was learned by applying skip-gram in word2vec to the user-category-selection information. Figure 10 shows the two-dimensional representation of the 19 vectorized categories. In this study, since we used movie data, we used genre as the category information.

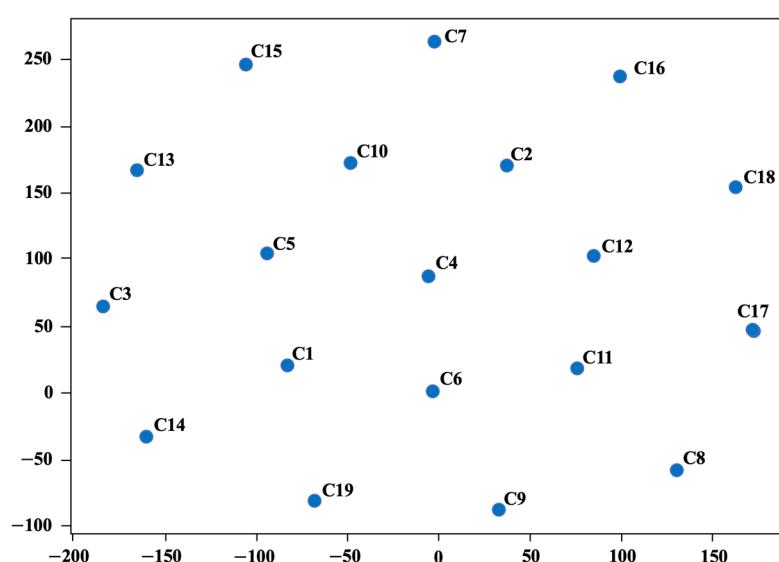


Figure 10. Results of genre2vec based on 19 genres in MovieLens dataset.

In Figure 10, each category (genre) is the result of learning; in this study, we leveraged vector information from categories for predicting new items. It can be considered as the real results of the genre2vec. We already show the example for utilizing genre2vec in Figure 5. Thus, we address the results of genre2vec in Figure 10 to derive the prediction results. We derived the prediction results from the category relationship in Figure 10 using Equation (1).

4.2.2. CF Approaches Used in Experiments

Here, we discuss the memory- and model-based CF approaches used for comparison with our approach. The memory-based approach is considered similarity-based and provides a method for identifying similar users and using them to derive recommendation results [1]. The CF approach measures the similarity between users or items in the input matrix, which is a user-item rating matrix, and selects similar users or items called neighbors. The cosine similarity [1] and Pearson correlation coefficient [76] have been used to calculate similarity. Similarity calculations can be performed on either a user or item basis; we calculated the similarities between the users for the experiments. After selecting a neighbor, namely, similar users, we calculated the prediction results based on the existing ratings of the neighbor. The model-based CF approaches were based on MF [22]. We predicted the ratings for the items in the test set and calculated the averages of the prediction results. We utilize K-nearest neighbor (KNN-Basic), KNN (means), KNN (Z-score), KNN (baseline), SVD (MF), and Non-negative matrix factorization (NMF) as CF approaches [77] to compare our approach for the experiments.

4.2.3. Experimental Design

We divided the test set using 10-fold cross-validation [76,78,79] for items in the input matrix. For each test set, we derived the mean absolute error (MAE) and root mean square error (RMSE) [78]; Equations (5) and (6) show the calculations for the MAE and RMSE, respectively:

$$MAE = \frac{1}{|T|} \sum_{n \in T} |r_n - \hat{r}_n|, \quad (5)$$

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{n \in T} (r_n - \hat{r}_n)^2}, \quad (6)$$

where T is the test set of items; n is one of the test items; and r_n and \hat{r}_n denote the real and predicted ratings of the item n , respectively.

Figure 11 shows an example of the 10-fold cross-validation in our experiments, with m items in the input matrix. Regarding these m items, $m/10$ was the test set, and $9m/10$ was the training set, which was applied to the last $m/10$ items. In our experiments, m becomes 9125 since there are 9125 movies in our database. We calculated the category ratings using the training set and predicted the items in the test set. In this case, we can consider the items in test set as new item in real situations. Namely, through the 10-fold cross-validation, we can check the accuracy of our approaches in real situations.

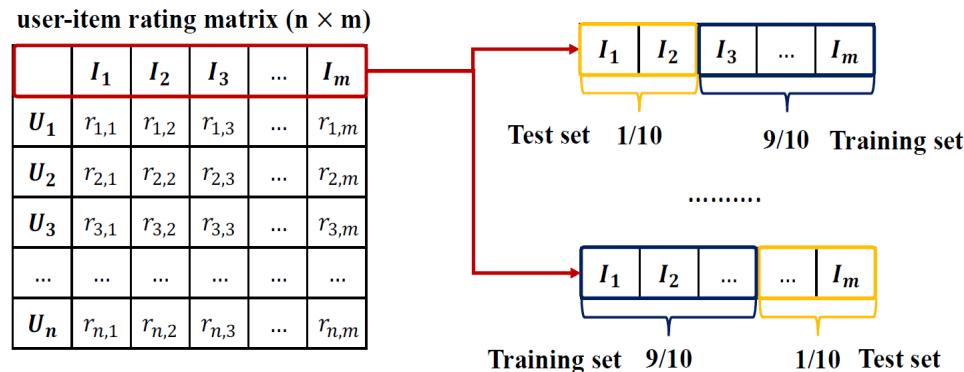


Figure 11. Example of data in 10-fold cross-validation.

4.3. Experimental Results

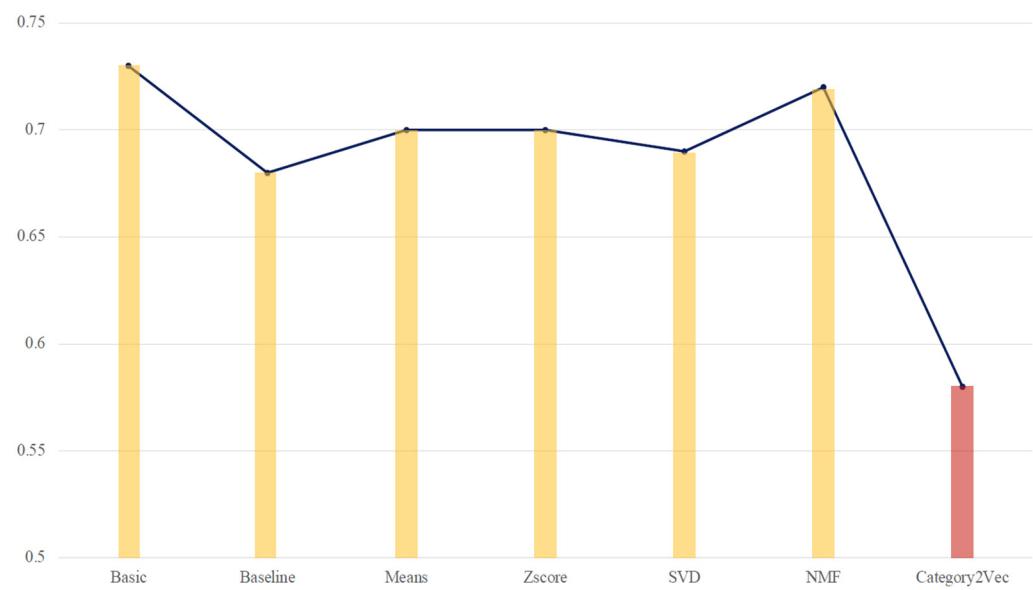
Table 3 and Figure 12 show the MAE results for an average of the 10-fold cross-validation using the CF and genre2vec approaches. These results revealed that the result under genre2vec had a minimum MAE compared with that under the other methods. In Table 3, the results for each fold based on each method show that the minimum MAE has been derived using Genre2vec. The average of the results for the 10-fold cross-validation in genre2vec was also lower than that under the other approaches. This meant that the prediction approach based on genre2vec could derive more accurate results than the existing CF methods could.

The similarities between categories learned from the item-selection information by the users in the input matrix could affect the prediction results because the vectors included the preferences of all the users for the item-selection information. Additionally, the user preferences for selecting items were applied by addressing the similarity between the vectors presented in this study with the category ratings. Therefore, we could consider that the results under genre2vec shown in Table 3 and Figure 11 included the average of the users' item-selection preferences.

Table 4 and Figure 13 show the RMSE results for an average of 10-fold cross-validation using the CF and genre2vec approaches. From these results, we verified that the RMSE under genre2vec was the least when compared to those under the other methods.

Table 3. Mean absolute error (MAE) results for 10-fold cross-validation with collaborative filtering (CF) approaches and genre2vec.

Fold \ Method	KNN (Basic)	KNN (Baseline)	KNN (Means)	KNN (Z-Score)	SVD	NMF	Genre2vec
1	0.74	0.69	0.69	0.69	0.68	0.68	0.71
2	0.74	0.69	0.69	0.69	0.70	0.69	0.73
3	0.74	0.69	0.69	0.70	0.69	0.69	0.73
4	0.74	0.68	0.68	0.70	0.70	0.68	0.71
5	0.73	0.68	0.68	0.70	0.69	0.69	0.72
6	0.74	0.68	0.68	0.70	0.70	0.68	0.73
7	0.74	0.68	0.70	0.70	0.69	0.73	0.51
8	0.73	0.68	0.70	0.71	0.69	0.71	0.61
9	0.74	0.69	0.69	0.70	0.71	0.72	0.67
10	0.74	0.68	0.70	0.69	0.68	0.73	0.70
Average	0.74	0.68	0.70	0.70	0.69	0.72	0.68
Stdev	0.003	0.005	0.007	0.005	0.007	0.006	0.09

**Figure 12.** MAE results for an average of 10-fold cross-validation with CF and genre2vec approaches.**Table 4.** RMSE results for 10-fold cross-validation with CF approaches and genre2vec.

Fold \ Method	KNN (Basic)	KNN (Baseline)	KNN (Means)	KNN (Z-Score)	SVD	NMF	Genre2vec
1	0.96	0.91	0.90	0.90	0.88	0.93	0.83
2	0.97	0.90	0.91	0.92	0.89	0.94	0.70
3	0.97	0.90	0.91	0.91	0.89	0.95	0.67
4	0.96	0.88	0.92	0.91	0.89	0.93	0.63
5	0.95	0.88	0.91	0.91	0.88	0.94	0.61
6	0.97	0.89	0.92	0.92	0.90	0.95	0.60
7	0.95	0.89	0.92	0.92	0.90	0.95	0.65
8	0.95	0.88	0.92	0.93	0.90	0.93	0.78
9	0.96	0.90	0.91	0.92	0.92	0.93	0.89
10	0.97	0.89	0.92	0.92	0.89	0.95	0.91
Average	0.96	0.89	0.91	0.91	0.89	0.94	0.72
Stdev	0.006	0.008	0.008	0.008	0.01	0.009	0.11

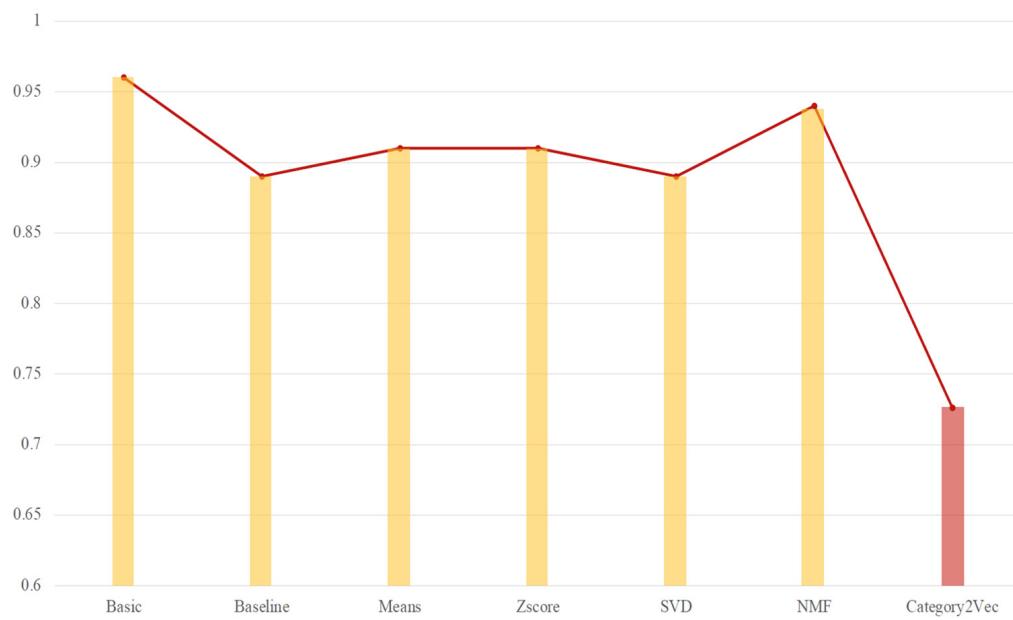


Figure 13. RMSE results for an average of 10-fold cross-validation with CF and genre2vec approaches.

In Table 4, the maximum value of the average for 10 folds is 0.96 under KNN (Basic), and the minimum value is 0.72 under genre2vec. Although the standard deviation (stdev) for the average of genre2vec compared to the other methods had a larger value, it was a smaller result than the other averages. Additionally, the smallest result at each fold was under genre2vec. Regarding the MAE, we considered that the results under genre2vec shown in Table 4 and Figure 13 included the average users' item-selection preferences. Regarding the RMSE, we could check that the users' item-selection preferences under genre2vec had been applied to the results.

4.4. Constructing a Smart Contract System

In this paper, to construct a smart contract system considering the characteristics of the experimental environment, we configure the conditions as follows:

- Private Ethereum Network Environment
- All content providers have one wallet
- If the recommendation result provided to the user matches the result stored in the blockchain network, the content provider receives 1 ETH by smart contract
- The reliability of the content provider is calculated as (the number of ETH)/(the number of recommendations)

Therefore, the reliability score of the content provider is derived based on the conditions, and the composition of the smart contract is as shown in the algorithm below. We leverage this algorithm to implement a real-world smart contract system. Algorithm 2 shows the smart contract to calculate a reliability score for a content provider.

Algorithm 2 Smart contract to calculate a reliability score for a content provider

Require:
Input = the set of input data by a user in blockchain network
Output = the set of the recommendation results for a user in blockchain network
REC_IN = the set of input data for the recommendation result
REC_OUT = the set of recommendation result by an algorithm

```
1: function calReliabilityScore(INPUT, OUTPUT, REC_IN, REC_OUT)
2:   cnt = 0
3:   eth = 0
4:   for i in range (0, size(INPUT)) do
5:     if (INPUT[i] = REC_IN[i]) and (OUTPUT[i] = REC_OUT[i]) then
6:       eth += 1
7:     end if
8:     cnt += 1
9:   end for
10:  reliability_score = eth/cnt
11:  return reliability_score
12: end function
```

5. Conclusions

The cold-start problem occurs when either new items or users are introduced in recommender systems. Regarding a new item, there is the problem of it being excluded from the recommendation process and the result because the users' reactions are absent. We propose a method to predict the representative score of a new item by utilizing the metadata of the item and exploiting the structure of word2vec.

We propose a category vectorization method to alleviate the cold-start problem of a recommender system by utilizing category information from the metadata of the item. In this study, we utilized category information as metadata. Subsequently, we derived genre2vec based on the users' item-selection preferences. Based on this derivation, our method can derive predictions using the category information of new items.

We used the MovieLens dataset to verify our approach; genre information was a category. We learned genre2vec based on genre information in the MovieLens dataset and skip-gram for word2vec. We predicted the average ratings for new items using genre2vec and the average ratings for genres in the dataset.

To compare the results, we addressed various CF approaches: KNN-basic, baseline, means, Z-score, SVD, and NMF. We also used the MAE and RMSE measures to verify the accuracy of the prediction results. We experimentally showed that our approach could derive more accurate results than other CF-based approaches could. We also found that the derived genre2vec could affect the averages for users' item-selection preferences.

Moreover, we have also proposed blockchain systems with smart contracts for the recommendation processes in cold-start situations. We have proposed a structure that builds the recommendation processes proposed in this paper through smart contracts. In addition, a smart contract system based on blockchain has been implemented by utilizing the Ethereum smart contract function. Through these systems, we can expect to provide more reliable content to users.

Our contributions can be divided into academic and industrial sides. The academic contributions are summarized as follows:

- We have proposed Genre2vec by vectorizing category information.
- We have proposed the learning model by applying the concept of Word2Vec for vectorizing category information.
- We have demonstrated that the prediction results based on Genre2vec are more precise than those of conventional CF approaches.
- Comparing the worst case (KNN-Basic) among the conventional CF techniques and the results of the proposed method, the proposed method shows about 25% better results.

- Comparing the best case (SVD) among the conventional CF techniques and the results of the proposed method, the proposed method shows about 19% better results.
- Based on our approach, we have alleviated item-side cold-start problems in recommender systems.
- In addition, we have implemented the blockchain system with smart contracts that address the recommendation processes in the item-side cold-start situation to improve content reliability.

In our approach, we have proposed a model that performs learning using only existing metadata, not using various data such as text information of items. For this reason, the proposed method is applicable to various domains with metadata and usage information. The industrial contributions of our approach are summarized as follows:

- In the case of web or mobile applications such as e-commerce or media content recommendation services, the items have metadata such as category. In this case, our approach can be applied to these domains.
- Furthermore, based on the vectorization of metadata, we can derive a higher prediction accuracy than the existing input for the aspect of the average ratings.
- Through our approach, it is possible to provide more reliable recommendation results than conventional CF approaches for a new item to online service users.
- In addition, online service providers can build more reliable recommender systems in cold-start situations for a new item.
- Through the structure for blockchain smart contracts, e-commerce and media-services on the web can provide more reliable contents.

The limitations of the proposed method are, first, that for genre2vec, the skip-gram method utilized in word2vec is applied, so it is difficult to obtain reliable results when the amount of data for training is small. In the case of the number of users utilized in this paper, genre2vec is performed based on a sufficiently large amount of evaluation data, more than a thousand. This limits the applicability of this methodology to new systems. However, genre2vec has the advantage of being able to predict the ratings of new items by using the information from the results, assuming that users' evaluation systems are similarly organized in domains that utilize similar genres. Second, the smart contract proposed in this paper considers the private chain. Therefore, if it is configured as a public chain based on Ethereum, it may be costly to derive recommendation results.

In future work, we will apply this approach to more diverse databases with various item features that include not only categories but also users' demographic information. Furthermore, we will apply word2vec to the various features in recommender systems. In this paper, we have utilized the MovieLens dataset. Because of this reason, we have only addressed genre information as categories and derived Genre2vec based on genre information in movie data. In the MovieLens dataset, there are 19 genres, and thus, we have leveraged 19 categories for the skip-gram model. If we can utilize more categories, then we can derive more various prediction values based on the results of Genre2vec. In the near future, we will apply our approaches to more various datasets with more statistical experiments.

Based on analysis by using these various results derived by a variety type of dataset, we can expect that the recommender systems can predict the presentative ratings for the times by utilizing few information of the items in database. It means that if we use Genre2vec, we can predict the reliable representative ratings for a new item than the conventional CF approaches.

Author Contributions: Conceptualization, S.-M.C. and Y.E.K.; methodology, S.-M.C. and D.L.; software, S.-M.C. and Y.G.S.; validation, Y.E.K. and S.L.; formal analysis, Y.E.K. and S.L.; investigation, S.-M.C. and D.L.; resources, Y.G.S.; data curation, Y.G.S.; writing—original draft preparation, S.-M.C.; writing—review and editing, D.L. and S.L.; visualization, S.-M.C. and Y.G.S.; supervision, S.L.; project administration, S.L.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (NRF-2022SIA5C2A03093301).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sarwar, B.M.; Karypis, G.; Konstan, J.A.; Riedl, J. Item-based Collaborative Filtering Recommendation Algorithms. In Proceedings of the 10th International World Wide Web Conference WWW '01, Hong Kong, China, 1–5 May 2001; pp. 285–295.
2. Choi, S.M.; Ko, S.K.; Han, Y.S. A movie recommendation algorithm based on genre correlations. *Expert Syst. Appl.* **2012**, *39*, 8079–8085. [[CrossRef](#)]
3. Okura, S.; Tagami, Y.; Ono, S.; Tajima, A. Embedding-based News Recommendation for Millions of Users. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '17, Halifax, NS, Canada, 13–17 August 2017; pp. 1933–1942.
4. Ratner, A.; Bach, S.H.; Ehrenberg, H.R.; Fries, J.A.; Wu, S.; Ré, C. Snorkel: Rapid Training Data Creation with Weak Supervision. *PVLDB* **2017**, *11*, 269–282. [[CrossRef](#)]
5. Peng, F.; Lu, X.; Ma, C.; Qian, Y.; Lu, J.; Yang, J. Multi-level preference regression for cold-start recommendations. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 1117–1130. [[CrossRef](#)]
6. Althbiti, A.; Alshamrani, R.; Alghamdi, T.; Lee, S.; Ma, X. Addressing Data Sparsity in Collaborative Filtering Based Recommender Systems Using Clustering and Artificial Neural Network. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 27–30 January 2021; pp. 0218–0227.
7. Jiang, B.; Yang, J.; Qin, Y.; Wang, T.; Wang, M.; Pan, W. A Service Recommendation Algorithm Based on Knowledge Graph and Collaborative Filtering. *IEEE Access* **2021**, *9*, 50880–50892. [[CrossRef](#)]
8. Zhao, W.; Tian, H.; Wu, Y.; Cui, Z.; Feng, T. A New Item-Based Collaborative Filtering Algorithm to Improve the Accuracy of Prediction in Sparse Data. *Int. J. Comput. Intell. Syst.* **2022**, *15*, 15. [[CrossRef](#)]
9. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback. *arXiv* **2009**, arXiv:1205.2618.
10. Wen, H.; Fang, L.; Guan, L. A hybrid approach for personalized recommendation of news on the Web. *Expert Syst. Appl. Int. J.* **2012**, *39*, 5806–5814. [[CrossRef](#)]
11. Chen, X.; Chen, H.; Xu, H.; Zhang, Y.; Cao, Y.; Qin, Z.; Zha, H. Personalized Fashion Recommendation with Visual Explanations based on Multimodal Attention Network: Towards Visually Explainable Recommendation. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 765–774.
12. Chen, S.; Huang, L.; Lei, Z.; Wang, S. Research on personalized recommendation hybrid algorithm for interactive experience equipment. *Comput. Intell.* **2020**, *36*, 1348–1373. [[CrossRef](#)]
13. East, R.; Hammond, K.; Lomax, W.; Robinson, H. What is the Effect of a Recommendation? *Mark. Rev.* **2005**, *5*, 145–157. [[CrossRef](#)]
14. Covington, P.; Adams, J.; Sargin, E. Deep Neural Networks for YouTube Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems RecSys 16, Boston, MA, USA, 15–19 September 2016; pp. 191–198.
15. Gomez-Uribe, C.A.; Hunt, N. The Netflix Recommender System: Algorithms, Business Value and Innovation. *ACM Trans. Manag. Inf. Syst.* **2016**, *6*, 1–19. [[CrossRef](#)]
16. Song, Y.; Dixon, S.; Pearce, M.T. A Survey of Music Recommendation Systems and Future Perspectives. In Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval (CMMR), London, UK, 19–22 June 2012; pp. 395–410.
17. van den Oord, A.; Dieleman, S.; Schrauwen, B. Deep content-based music recommendation. In Proceedings of the Advances in Neural Information Processing Systems NIPS '13, Lake Tahoe, NV, USA, 5 December 2013; Volume 26, pp. 2643–2651.
18. Li, J.; He, Z.; Cui, Y.; Wang, C.; Chen, C.; Yu, C.; Zhang, M.; Liu, Y.; Ma, S. Towards Ubiquitous Personalized Music Recommendation with Smart Bracelets. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2022**, *6*, 1–34. [[CrossRef](#)]
19. Herlocker, J.L.; Konstan, J.; Borchers, A.; Riedl, J. An Algorithm Framework for Performing Collaborative Filtering. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '99, Berkeley, CA, USA, 15–19 August 1999; pp. 230–237.
20. Tkalcic, M.; Odic, A.; Kosir, A.; Tasic, J.F. Affective Labeling in a Content-Based Recommender System for Images. *IEEE Trans. Multimed.* **2013**, *15*, 391–400. [[CrossRef](#)]
21. Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P.B. *Recommender Systems Handbook*; Springer-Verlag: Berlin/Heidelberg, Germany, 2010.
22. Koren, Y.; Bell, R.M.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *IEEE Comput.* **2009**, *42*, 30–37. [[CrossRef](#)]
23. de Campos, L.; Fernández-Luna, J.; Huete, J.; Rueda-Morales, M. Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *Int. J. Approx. Reason.* **2010**, *51*, 785–799. [[CrossRef](#)]

24. Cano, E.; Morisio, M. Hybrid Recommender Systems: A Systematic Literature Review. *Intell. Data Anal.* **2019**, *21*, 1487–1524. [[CrossRef](#)]
25. Javed, U.; Shaukat, K.; Hameed, I.A.; Iqbal, F.; Alam, T.M.; Luo, S. A Review of Content-Based and Context-Based Recommendation Systems. *Int. J. Emerg. Technol. Learn. (Ijet)* **2021**, *16*, 274–306. [[CrossRef](#)]
26. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web WWW 17, Perth, Australia, 3–7 April 2017; pp. 173–182.
27. Liang, D.; Krishnan, R.G. Variational Autoencoders for Collaborative Filtering. In Proceedings of the 2018 World Wide Web Conference WWW 18, Lyon, France, 10 April 2018; pp. 689–698.
28. Duong, T.N.; Vuong, T.A.; Nguyen, D.M.; Dang, Q.H. Utilizing an Autoencoder-Generated Item Representation in Hybrid Recommendation System. *IEEE Access* **2020**, *8*, 75094–75104. [[CrossRef](#)]
29. Barkan, O.; Koenigstein, N. Item2Vec: Neural Item Embedding for Collaborative Filtering. In Proceedings of the 26th International Workshop on Machine Learning for Signal Processing (MLSP), Salerno, Italy, 13–16 September 2016.
30. Chen, C.; Wang, C.; Tsai, M.; Yang, Y. Collaborative Similarity Embedding for Recommender Systems. In Proceedings of the World Wide Web Conference WWW 2019, San Francisco, 13–17 May 2019; pp. 637–2643.
31. Zhao, X.; Liu, H.; Liu, H.; Tang, J.; Guo, W.; Shi, J.; Wang, S.; Gao, H.; Long, B. AutoDim: Field-aware Embedding Dimension Searchin Recommender Systems. In Proceedings of the WWW '21: The Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 3015–3022.
32. Rendle, S.; Krichene, W.; Zhang, L.; Anderson, J.R. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In Proceedings of the RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Online, 22–26 September 2020; pp. 240–248.
33. Schein, A.I.; Popescul, A.; Ungar, L.H.; Pennock, D.M. Methods and metrics for cold-start recommendations. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '02, Tampere, Finland, 11–15 August 2002; pp. 253–260.
34. Ishikawa, M.; Géczy, P.; Izumi, N.; Morita, T.; Yamaguchi, T. Information Diffusion Approach to Cold-Start Problem. In Proceedings of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology-Workshops WI-IAT '07, Fremont, CA, USA, 2–5 November, 2007; pp. 129–132.
35. Said, A.; Jain, B.; Narr, S.; Plumbaum, T. Users and Noise: The Magic Barrier of Recommender Systems. In Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization, Montreal, QC, Canada, 16–20 July 2012; pp. 237–248.
36. Bellogín, A.; Said, A.; de Vries, A. The Magic Barrier of Recommender Systems—No Magic, Just Ratings. In Proceedings of the International Conference on User Modeling, Adaptation, and Personalization UMAP 2014, Aalborg, Denmark, 7–11 July 2014; pp. 25–36.
37. Sarwar, B.M.; Karypis, G.; Konstan, J.A.; Riedl, J. Analysis of recommendation algorithms for e-commerce. In Proceedings of the 2nd ACM Conference on Electronic Commerce EC '00, Minneapolis, MN, USA, 17–20 October 2000; pp. 158–167.
38. Bell, R.M.; Koren, Y. Lessons from the Netflix prize challenge. *SIGKDD Explor.* **2007**, *9*, 75–79. [[CrossRef](#)]
39. Levy, O.; Goldberg, Y. Neural Word Embedding as Implicit Matrix Factorization. In Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2177–2185.
40. Wei, K.; Huang, J.; Fu, S. A Survey of E-Commerce Recommender Systems. In Proceedings of the 2007 International Conference on Service Systems and Service Management, Chengdu, China, 9–11 June 2007; pp. 1–5.
41. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. Recommender systems survey. *Knowl. Based Syst.* **2013**, *46*, 109–113. [[CrossRef](#)]
42. Ronen, R.; Koenigstein, N.; Ziklik, E.; Nice, N. Selecting Content-Based Features for Collaborative Filtering Recommenders. In Proceedings of the 7th ACM Conference on Recommender Systems RecSys '13, Hong Kong, China, 12–16 October 2013; pp. 407–410.
43. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
44. Atzei, N.; Bartoletti, M.; Cimoli, T. A survey of attacks on Ethereum smart contracts (SoK). In Proceedings of the Principles of Security and Trust: 6th International Conference POST 2017, Uppsala, Sweden, 22–29 April 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 164–186.
45. Ahn, H.J. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Inf. Sci.* **2008**, *178*, 37–51. [[CrossRef](#)]
46. Lika, B.; Kolomvatsos, K.; Hadjiefthymiades, S. Facing the cold start problem in recommender systems. *Expert Syst. Appl.* **2014**, *41*, 2065–2073. [[CrossRef](#)]
47. Rong, Y.; Wen, X.; Cheng, H. A Monte Carlo algorithm for cold start recommendation. In Proceedings of the 23th International World Wide Web Conference WWW '14, Seoul, Republic of Korea, 7–11 April 2014; pp. 327–336.
48. Volkovs, M.; Yu, G.W.; Poutanen, T. Content-based Neighbor Models for Cold Start in Recommender Systems. In Proceedings of the Recommender Systems Challenge RecSys Challenge '17, Como, Italy, 27 August 2017; pp. 1–6.

49. Kang, S.; Hwang, J.; Lee, D.; Yu, H. Semi-Supervised Learning for Cross-Domain Recommendation to Cold-Start Users. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management CIKM 2019, Beijing, China, 3–7 November 2019; pp. 1563–1572.
50. Anand, S.S.; Griffiths, N. A Market-based Approach to Address the New Item Problem. In Proceedings of the 2011 ACM Conference on Recommender Systems RecSys ’11, Chicago, IL, USA, 23–27 October 2011; pp. 205–212.
51. Sun, D.; Luo, Z.; Zhang, F. A novel approach for collaborative filtering to alleviate the new item cold-start problem. In Proceedings of the 11th International Symposium on Communications and Information Technologies ISCIT ’11, Hangzhou, China, 12–14 October 2011; pp. 402–406.
52. Choi, S.M.; Han, Y.S. Identifying representative ratings for a new item in recommendation system. In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication ICUIMC ’13, Kota Kinabalu, Malaysia, 17–19 January 2013.
53. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems NIPS ’13, Lake Tahoe, NV, USA, 5–10 December 2013; Volume 2, pp. 3111–3119.
54. Efstathiou, V.; Spinellis, D. Semantic Source Code Models Using Identifier Embeddings. In Proceedings of the 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), Montreal, QC, Canada, 25–31 May 2019; pp. 29–33.
55. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Tech. Rep.* **2008**, 21260.
56. Ullah, F.; Al-Turjman, F. A conceptual framework for blockchain smart contract adoption to manage real estate deals in smart cities. *Neural Comput. Appl.* **2023**, 35, 5033–5054. [[CrossRef](#)]
57. Agrawal, T.K.; Angelis, J.; Khilji, W.A.; Kalaiarasan, R.; Wiktorsson, M. Demonstration of a blockchain-based framework using smart contracts for supply chain collaboration. *Int. J. Prod. Res.* **2023**, 1497–1516. [[CrossRef](#)]
58. Anitha, V.; Marquez Caro, O.J.; Sudharsan, R.; Yoganandan, S.; Vimal, M. Transparent voting system using blockchain. *Measurement: Sensors* **2023**, 25, 100620. [[CrossRef](#)]
59. Aslanian, E.; Radmanesh, M.; Jalili, M. Hybrid Recommender Systems based on Content Feature Relationship. *IEEE Trans. Ind. Inform.* **2016**, 1. [[CrossRef](#)]
60. Rojsattarat, E.; Soonthornphisaj, N. Hybrid Recommendation: Combining Content-Based Prediction and Collaborative Filtering. In Proceedings of the Intelligent Data Engineering and Automated Learning, Hong Kong, China, 21–23 March 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 337–344.
61. Lang, K. NewsWeeder: Learning to Filter Netnews. In *Machine Learning Proceedings*; Elsevier: Amsterdam, The Netherlands, 1995; pp. 331–339.
62. Krulwich, B. Learning user interests across heterogeneous document databases. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 106–110.
63. Chughtai, M.W.; Selamat, A.; Ghani, I.; Jung, J. E-Learning Recommender Systems Based on Goal-Based Hybrid Filtering. *Int. J. Distrib. Sens. Netw.* **2014**, 10, 912130. [[CrossRef](#)]
64. Pirasteh, P.; Jung, J.J.; Hwang, D. Item-Based Collaborative Filtering with Attribute Correlation: A Case Study on Movie Recommendation. In Proceedings of the Intelligent Information and Database Systems-6th Asian Conference ACIIDS ’14, Phuket, Thailand, 7–10 April 2014; pp. 245–252.
65. Zhang, J.; Peng, Q.; Sun, S.; Liu, C. Collaborative filtering recommendation algorithm based on user preference derived from item domain features. *Phys. A Stat. Mech. Its Appl.* **2014**, 396, 66–76. [[CrossRef](#)]
66. Christensen, I.; Schiaffino, S. A Hybrid Approach for Group Profiling in Recommender Systems. *J. Univers. Comput. Sci.* **2014**, 20, 507–533.
67. Lekakos, G.; Giaglis, G. A hybrid approach for improving predictive accuracy of collaborative filtering algorithms. *User Model. User Adapt. Interact.* **2007**, 17, 5–40. [[CrossRef](#)]
68. Burke, R. Hybrid Recommender Systems: Survey and Experiments. *User Model. User Adapt. Interact.* **2002**, 12, 331–370. [[CrossRef](#)]
69. Gope, J.; Jain, S.K. A survey on solving cold start problem in recommender systems. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 133–138.
70. Carrer-Neto, W.; Hernández-Alcaraz, M.L.; Valencia-García, R.; García-Sánchez, F. Social knowledge-based recommender system. Application to the movies domain. *Expert Syst. Appl.* **2012**, 39, 10990–11000. [[CrossRef](#)]
71. Deng, Y.; Wu, Z.; Tang, C.; Si, H.; Xiong, H.; Chen, Z. A Hybrid Movie Recommender Based on Ontology and Neural Networks. In Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications International Conference on Cyber Physical and Social Computing, Washington, DC, USA, 18–20 December 2010; pp. 846–851.
72. Qi, L.; Liu, Y.; Zhang, Y.; Xu, X.; Bilal, M.; Song, H. Privacy-Aware Point-of-Interest Category Recommendation in Internet of Things. *IEEE Internet Things J.* **2022**, 9, 21398–21408. [[CrossRef](#)]
73. Liu, Y.; Wu, H.; Rezaee, K.; Khosravi, M.R.; Khalaf, O.I.; Khan, A.A.; Ramesh, D.; Qi, L. Interaction-Enhanced and Time-Aware Graph Convolutional Network for Successive Point-of-Interest Recommendation in Traveling Enterprises. *IEEE Trans. Ind. Inform.* **2023**, 19, 635–643. [[CrossRef](#)]
74. Meel, P.; Bano, F.; Goswami, A.; Gupta, S. Movie Recommendation Using Content-Based and Collaborative Filtering. In Proceedings of the International Conference on Innovative Computing and Communications ICICC ’21, Ostrava, Czech Republic, 21–22 March 2021; Springer: Singapore, 2021; pp. 301–316.

75. Mehrabani, M.M.; Mohayeli, H.; Moeini, A. A Hybrid Approach to Enhance Pure Collaborative Filtering based on Content Feature Relationship. *arXiv* **2020**, arXiv:2005.08148.
76. Bulmer, M. *Principle of Statistics*; Dover Publications: Garden City, NY, USA, 1979.
77. Choi, S.M.; Lee, D.; Jang, K.; Park, C.; Lee, S. Improving Data Sparsity in Recommender Systems Using Matrix Regeneration with Item Features. *Mathematics* **2023**, *11*, 292. [[CrossRef](#)]
78. Choi, S.M.; Cha, J.W.; Han, Y.S. Identifying representative reviewers in internet social media. In Proceedings of the Second International Conference on Computational Collective Intelligence: Technologies and Applications-Volume Part II ICCCI '10, Kaohsiung, Taiwan, 10–12 November 2010; pp. 22–30.
79. Choi, S.M.; Cha, J.W.; Kim, L.; Han, Y.S. Reliability of Representative Reviewers on the Web. In Proceedings of the International Conference on Information Science and Applications ICISA 2011, Jeju Island, Republic of Korea, 26–29 April 2011; pp. 1–5.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.