

MDPI

Article

Extreme Gradient Boosting for Recommendation System by Transforming Product Classification into Regression Based on Multi-Dimensional Word2Vec

Se-Joon Park ¹, Chul-Ung Kang ², * and Yung-Cheol Byun ¹, * and Yung-

- Department of Computer Engineering, Jeju National University, Jeju-si 63243, Korea; ppakse6@jejunu.ac.kr
- ² Department of Mechatronics Engineering, Jeju National University, Jeju-si 63243, Korea
- * Correspondence: cukang@jejunu.ac.kr (C.-U.K.); ycb@jejunu.ac.kr (Y.-C.B.)

Abstract: Now that untact services are widespread and worldwide, the number of users visiting online shopping malls has increased. For example, the recommendation systems in Netflix, Amazon, etc., have gained a lot of attention by attracting many users and have made large profit by recommending suitable products to their users. In the paper, we conduct a study to enhance recommendation accuracy using Word2Vec, widely used in natural language processing. We collect user shopping history with personal click preference information of product items as data, representing a document for natural language processing. The sequence of product item clicks is fed into the Word2Vec technology algorithm to obtain the vectors symmetrically representing all of the product items clicked by users. Training and test data have a series of vectors representing a sequence of the clicked product items as inputs and a purchased product as a target. Machine learning models recommend a product as a symmetric vector for each input and calculate the similarity among the recommended vectors and all other registered products they sell in the system to recommend multiple products as final recommendation results. We use XGBoost regressor and classifier models to recommend some products that users would like and evaluate the recommendation accuracy. A finally recommended product by the models is a vector, and the system recommends some more products by calculating the similarity as mentioned above. We evaluated the classifier model's recommendation accuracy without Word2Vec encoding first and then with the Word2Vec technique. Meanwhile, we can represent the products with single or multiple dimensional vectors. We noted that the recommendation accuracy increases when we use multiple dimensions of Word2Vec vectors from the experiments. We also evaluated the performances when the system recommends one or multiple products. For the recommendation of multiple products (five here), a regression model has higher accuracy than a classification model in all dimensions of vectors.

Keywords: machine learning; collaborative filtering; pattern recognition; word2vec; symmetric vector; multi-dimension; XGBoost; classification; regression



Citation: Park, S.-J.; Kang, C.-U.; Byun, Y.-C. Extreme Gradient Boosting for Recommendation System by Transforming Product Classification into Regression Based on Multi-Dimensional Word2Vec. Symmetry 2021, 13, 758. https://doi.org/10.3390/sym13050758

Academic Editor: Peng-Yeng Yin

Received: 20 March 2021 Accepted: 22 April 2021 Published: 27 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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

A recommendation system is an information filtering system that can recommend preferred items to a user automatically. Internet services endeavor to provide convenience to users, and as they develop, the users constantly demand convenience while using evolving services. The recommendation system plays a general and essential role in satisfying Internet service users [1].

Importance of untact service. As COVID-19 spread around the world in 2019 and 2020, governments of many countries began to recommend non-contact activities. For this reason, many users have begun to prefer online shopping using the Internet rather than visiting offline stores. Untact has been actively taking place in the lives of people, including in the area of online shopping. According to data from the Korea Statistical Office, from February to September 2020, when COVID-19 considerably spread in Korea, the total

Symmetry **2021**, 13, 758 2 of 16

amount of online shopping transactions increased by more than 15% on average compared to the same month of the year before [2]. Accordingly, the recommendation system's importance increases as the number of users visiting online shopping malls increases due to the popularity of untact [3].

Business impact of recommendation. As various Internet services, such as SNS, video streaming, and online shopping malls, emerge, the amount of data handled by the Internet has also increased due to the increased number of Internet service users [4]. As a result, Internet users spend a lot of time obtaining the information they want from numerous pieces of Internet information. A recommendation system is needed to help Internet users efficiently find the information they require. World-class video streaming services such as YouTube and Netflix have built their recommendation systems to meet service users' needs and have achieved great growth in sales [5,6].

In this study, we apply the Word2Vec technique, using both the classification and regression models of machine learning to improve the accuracy of recommendations, and we compare the results from each model with different dimensions of the Word2Vec vector.

Classification and regression model. Classification and regression models of machine learning automatically recommend items for users. However, there is a difference between the classification model and the regression model. First, the classification model classifies a given feature into one of the pre-defined classes. For example, when classifying spam mail, the classification model receives spam mail data as an input feature and classifies whether the mail is spam or not via binary classification [7]. The regression model, however, estimates the continuous target value from a given input feature. For example, when predicting stock price, it makes predictions by receiving time series data according to the trend of the previous stock as a feature. In addition, the regression model has an advantage in computation time; that is, it is faster than the classification model [8,9].

Word2Vec and vector dimension. In machine learning, it is suggested that the amount of data is one of the factors that affects performance. If the data size is small, one may fail to achieve generalization of the machine, and if the size of data is sufficient, the possibility of obtaining a better result increases accordingly. Better results can also be expected if the number of rows and columns is balanced. If the number of rows of data is less than that of columns, overfitting may occur due to excessive learning. On the contrary, underfitting may occur when the number of rows of data is much bigger than that of columns, which means less learning due to the relatively small number of columns. We adjusted the dimensions of Word2Vec to solve this problem. As the dimension increases, the vector dimension's value representing the items also increases. In one dimension, the value of x of the vector is represented in one column. In two dimension, training is conducted by dividing the vector's x and y values into two columns. As the dimension of Word2Vec increases by one dimension, the column increases by the number of dimensions. In this way, Word2Vec prevents underfitting and overfitting [10].

Related work: A Recommender System Based on Machine Learning Using Word2Vec. A previous study uses Word2Vec to improve the prediction accuracy of user-based collaborative filtering. Word2Vec finds correlations among a series of products that are listed chronologically as clicked by the user. When searching for associations between clicked product items, the more often the products appear together, the closer the vectors are expressed. Applying these characteristics of Word2Vec to a classification model of machine learning, we confirm that the recommendation accuracy increases by 2.1% compared to the accuracy before finding a correlation using Word2Vec [11].

Related work: Toward Improving the Prediction Accuracy of Product Recommendation System Using Extreme Gradient. A previous study identifies a user's interests and recommends related items by utilizing contents, such as the user's shopping profile details, visited pages, and click information. It learns each user's click pattern using machine learning models and recommends products to users. This paper is a study on combining Word2Vec technology with machine learning to increase recommendation accuracy. The model is evaluated using mean absolute error, mean squared error, and root mean

Symmetry **2021**, *13*, 758 3 of 16

squared error for the proposed methodology. Compared to other traditional approaches, the proposed model generates the smallest error rate and enhances the recommendation system's prediction accuracy [12].

2. Methods

2.1. Dataset

Table 1 shows the datasets used in this research and the information on which 10,000 users have purchased products in a Korean online shopping mall called "E-Jeju Mall" over a period of 50 days. The details of the items clicked by each user are arranged horizontally in chronological order. That is, a row represents a user's shopping history. We used 80% of the whole data for training and 20% for the test.

Table 1. Dataset information.

Dataset	Information		
Data source	E-Jeju Mall		
Data collection period	50 days		
Total number of records	10,000		
Training data	80%		
Testing data	20%		

In the data collection step, a unique ID number of a product item clicked by each user is acquired. For example, 1531787977 as a unique ID can mean a product of a pen displayed on the online shopping mall. A list of item IDs is recorded per purchase. Table 2 shows an example of the data used in this research. Each row shows a user's shopping history in the order of clicks. The maximum number of clicks by a user is 14, and if the number of clicks is more than 14, then the most recently clicked 14 items are used. The 14th item, the last clicked item, is the final item that a user purchased by putting it into their shopping cart. The reason why we set the maximum number of clicks as 14 is that 95% of users finished shopping before 14 clicks. A higher maximum click number means more columns in the data file. If a user completes shopping with less than 14 clicks, then the clicks' front part is set with null values during the preprocessing phase.

1st Click 2nd Click 3rd Click User 13th Click **Purchasing Item** ... 210.54.66.88 1531787977 1531801122 1531787977 1531801305 1531787589 95.90.201.200 469978540 365098808 481389628 445941287 397457112 157.168.67.74 1547428994 1547427506 1547430147 1547431139 1547429909 ... 15.11.67.191 256924814 442093764 315393408 557534813 384274798 ... 197.198.238.56 397457110 587617958 510163790 504844985 445940920 ...

Table 2. Example of the dataset.

2.2. Model Architecture

Figure 1 shows the overall configuration diagram of the proposed approach in the research. In the data collection process, the user's click history data are collected when they click on items. The collected click history for each user is placed in each row of data. Subsequent columns are arranged in order of the product item clicked by the user. We used Word2Vec to identify the correlation among product items. Based on the found association, Word2Vec expresses each item as a vector. We used the classification and regression model of extreme gradient boosting (XGBoost) to train the user's preferences and recommend proper products for users. The recommendation is followed by the evaluation of the results.

However, the predicted results by the classification and the regression model can be different. The classifier predicts a unique ID of a product item in the training data.

Symmetry **2021**, 13, 758 4 of 16

A regressor model differs from a classifier in that it provides continuous values of vector elements as a result. Therefore, the classification model calculates the accuracy by comparing the predicted ID with the actual target ID. In the regression model, a product closest to the predicted vector is obtained as a recommended product. Therefore, a regressor and classifier recommend products to a user. The following question then arises: how many products would be recommended by the system in total? In this research, we select five product items for final recommendations because shopping malls generally recommend multiple products on the Web. To select the final five recommendations, we use the Euclidean distance and cosine similarity to identify the items close to a vector predicted by a model.

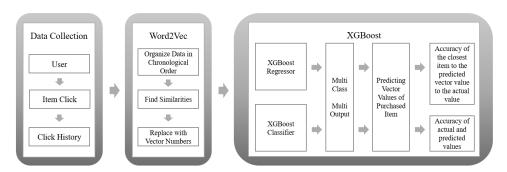


Figure 1. Recommender system using Word2Vec based on XGBoost.

2.3. Word2Vec

Word2Vec is a technique originally used for natural language processing to make up for the shortcomings of existing one-hot encoding. One-hot encoding and Word2Vec have in common that each word is expressed as a vector. One-hot encoding is a sparse representation in which only the value of the index position meaning the word is 1; all others are 0. The vectors from one-hot encoding have the same distance, which cannot represent the similarity among vectors, so a distributed representation method was devised, i.e., Word2Vec. The representation method finds similarities in the assumption that words appearing in close positions have similar meanings. A vector expressing a word can be represented in different dimensions. As the dimension of a word's vector increases, the relationship among the words can be expressed in more detail [13].

Word2Vec can be expressed in two ways, Continuous Bag of Words (CBOW) and Skip-Gram. CBOW predicts the central word based on the words around it. Conversely, Skip-Gram is a method of predicting surrounding words based on the central word [14,15]. The role of Word2Vec is to find the association among the products because similar products are close to each other in the click sequence.

As mentioned above, Word2Vec technique has been frequently used in natural language processing. The vectors of words are positioned close to each other in the vector space if they appear close together in a sentence. We can consider the clicked product ID in a user's shopping history as a word in natural language processing, and we can obtain the proper vectors representing the product relationship based on the characteristics of Word2Vec. We can adjust the window size to find the association of product items by breaking them in units of n items [16].

Figure 2 shows an example of Word2Vec Skip-Gram. It learns the central and surrounding items based on the data click on the item over time. If the clicked items appear close to each other frequently, the relationship among the items increases. Accordingly, vectors of the items are also represented close to each other in the vector space. While shopping, users generally tend to click on similar product items. Of course, there are cases of clicking on completely irrelevant items from time to time, but since such cases are rare, Word2Vec recognizes that the product items that frequently appear together are similar each other. If the vectors generated by Word2Vec are close to each other, they are likely similar items. Meanwhile, recommending items you expect to purchase with Word2Vec increases the accuracy of recommendations [17,18].

Symmetry **2021**, *13*, 758 5 of 16

As explained above, Word2Vec identifies the association among product items as vectors. The vector dimension can be set directly as a hyperparameter [19]. In the research, the vector dimension of Word2Vec is increased one by one from one to five dimension to determine the effect of dimension size on the recommendation accuracy. As the vector dimension increases, the relationship between product items in the vector space can be expressed in more detail, affecting the final recommendation accuracy. As the vector dimension increases, the size of a vector representing the product increases, and, as a consequence, the data size also increases, which affects the recommendation results and computation time.

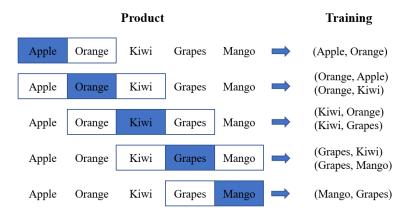


Figure 2. Examples of the Word2Vec Skip-Gram.

2.4. XGBoost

XGBoost is based on gradient boosting. XGBoost has a faster learning time and higher model performance than the models based on existing gradient boosting. The XGBoost model can solve the problem of overfitting, a big concern of gradient boosting. Overfitting happens when the learning model overtrains the training data. The error for training data becomes small, but the error becomes big when new data are input. XGBoost can prevent overfitting by setting the desired learning method directly through several hyperparameters, such as "n_estimators" and '"ax_depth", representing the width and depth of learning [20].

2.5. Training and Testing

Training. In the learning stage of the model, input and output data are used. Input data are usually expressed as *X*-data and refer to the product item click history before purchasing a product item. The output data are defined as *Y*-data and are recognized as a purchased item [21]. The model uses both input and output data for learning. Here, the user click history is recognized as a shopping pattern [22,23].

Testing. When new data come in as input during the testing phase of machine learning, the system predicts results based on the previous learning. Based on the prediction, some more products that are close to the predicted product are recommended. It is a form that recommends the purchased item with the click history that is most similar to the new click history [24]. A sequence of clicks during shopping becomes a shopping pattern, and according to the newly captured click pattern, the product with the most similar click pattern is recommended as a product for purchase [25].

3. Result and Discussion

3.1. Implementation Environment

Here, we compare the learning time according to the increase in the vector dimension of Word2Vec. Since the number of data increases as the vector dimension increases, we can obtain a better result by considering the recommendation accuracy and learning time. The computation time required for learning varies according to the research environment;

Symmetry **2021**, 13, 758 6 of 16

the information on the environment in which this study is conducted is presented below in Table 3.

Table 3. Implementation environment.

Component	Descripton	
Programming language	Python 3.7.6	
Operationg system	Window 10 pro 64 bit	
Browser	Google chrome	
Library and framework	Jupyter notebook	
CPU	Intel(R) Core(TM) i5-9600k@3.70 GHz	
Memory	16 GB	

3.2. Training Time

The most significant advantage of using a regression model on the data having categorical targets is that the learning time tends to be relatively fast. Figure 3 shows the learning time according to each dimension of the Word2Vec vector of the classification and regression model. We increased the Word2Vec vector dimension from one to five dimension to compare the learning time using a classifier and regressor. As the vector dimension of Word2Vec increases, the number of data increases, so we can expect the learning time to gradually increase. In the regressor model, the learning time increases as we increase the dimension from one to five. The learning time with the five-dimensional vector is about 1 min, which is faster than the other classifier model [26]. The learning time of the classifier model is slower than the regression model. As a result, it can be seen that the learning time increases as the Word2Vec vector dimension increases, and the regressor learns faster than the classifier [27,28].

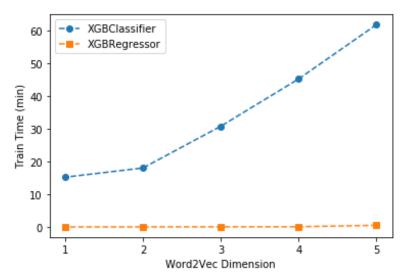


Figure 3. Training time of classification and regression models and vector dimension.

3.3. Evaluation Metrics of Classification Model

This subsection defines the evaluation matrix to evaluate the results by the classifier model, which predicts items to be purchased by learning the shopping pattern represented as a series of product click vectors. Recommended results from a model and the actual purchased item are compared to calculate the accuracy. Here, we calculate accuracy, precision, and recall to evaluate the performance of the machine learning models.

3.3.1. Accuracy

Accuracy is the case of correctly predicting true to true and false to false in all cases. TP is the case of correctly predicting an actual value of true as true, and FP is the case of

Symmetry **2021**, 13, 758 7 of 16

incorrectly predicting an actual value of false as true. FN means incorrectly predicting a true value as false, and TN means correctly predicting a true–false value as false. It is the most intuitive evaluation index of a classification model. In the case of accuracy, if the data were biased, it would not be acceptable [29]. In this study, when predicting an item to be purchased based on the click history, accuracy was obtained by dividing all cases when it matched the actual purchase item.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$
 (1)

3.3.2. Precision

Precision is the proportion of what the model classifies as true that is actually true. It is also called positive predictive value (PPV). When the click history is entered as input, the classifier is a measure to predict which item to purchase based on the click history.

$$Precision = \frac{TP}{TP + FP}$$
 (2)

3.3.3. Recall

Recall is the ratio of what the model predicts as true among what is actually true, contrary to precision. Again, taking spam mail as an example, it is the ratio of the model that predicts the actual spam mail to be true. In this case, if the item to be purchased can be predicted by certain factors, only the item to be purchased can be predicted. If the item to be purchased is not certain, there is a problem in which the precision can be extremely increased by reducing the number of cases of FP by holding it without prediction at all. Therefore, the higher the index of both precision and recall, the more accurate the prediction.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

3.4. Evaluation Metrics of Regression Model

In this subsection, we define the evaluation metrics for the regression model on the Word2Vec vector. The model results are evaluated using MAE, MSE, RMSE, and RMSLE, which are the evaluation indicators of the regression model [30].

3.4.1. MAE

Mean absolute error (MAE) is an error value representing the difference between an actual value and a predicted value as an average of absolute values.

$$MAE = \frac{1}{M} \sum_{t=1}^{M} (y_t - \hat{y}_t)$$
 (4)

3.4.2. MSE

Mean squared error (MSE) represents the difference between an actual value and a predicted value as an average of squared error.

$$MSE = \frac{1}{M} \sum_{t=1}^{M} (y_t - \hat{y}_t)^2$$
 (5)

Symmetry **2021**, 13, 758 8 of 16

3.4.3. RMSE

Root mean squared error (RMSE) means the root of the MSE. If the number handled by the data is large, a root is added to reduce the calculated errors.

$$RMSE = \sqrt{\frac{1}{M} \sum_{t=1}^{M} (y_t - \hat{y}_t)^2} \times 100$$
 (6)

3.4.4. RMSLE

Root mean squared logarithmic error (RMSLE) improves the disadvantage that the error is big when the value of the evaluation indicators is enormous.

$$RMSLE = \sqrt{\frac{1}{M} \sum_{t=1}^{M} (\log(y_t + 1) - \log(\hat{y}_t + 1))^2}$$
 (7)

3.5. Hit Ratio

When we implement a recommendation system, not only one product item is recommended; rather, multiple product items are recommended. Usually, users feel satisfied when recommending multiple items of similar relevance. Representatively, Netflix and Amazon recommend multiple things that users may prefer. In this paper, we recommend n items close to the predicted item in similarity. If there are actual purchased items in n recommended product items, then a hit can be assumed. For both classification and regression models, the recommended accuracy is obtained when n is 1 and 5. The Euclidean distance formula is used to find items close to the recommended item represented as a vector [31].

Euclidean Distance =
$$\sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$
 (8)

We also use cosine similarity to find the multiple products to recommend and compare to the results using Euclidean distance.

Cosine Similarity =
$$\frac{q \cdot p}{\|q\| \|p\|} = \frac{\sum_{i=1}^{n} (q_i \cdot p_i)}{\sqrt{\sum_{i=1}^{n} (q_i)^2} \sqrt{\sum_{i=1}^{n} (p_i)^2}}$$
 (9)

3.6. Feature Importance

Feature importance A is a quantitative value of how each column affects the final purchased product. The columns of data used are the click history before purchasing a product item, arranged according to the click order. With feature importance, we can identify how much the order of clicks influences the purchase of a target product [32]. Figure 4 shows the purchased product based on user click.

In the data, the twelfth clicked item just before purchase had the most significant impact on the results at 31.11%. Next, the first clicked item affected the purchasing by 30.28%. After that, the fifth clicked item affected the purchase by 9.72%, and the second clicked item by 5.53%. Other than the first and last clicked items, the items had less than 10% influence each. From the results, it can be seen that the first and last clicked items have the most impact on the final purchased product [33].

Symmetry **2021**, 13, 758 9 of 16

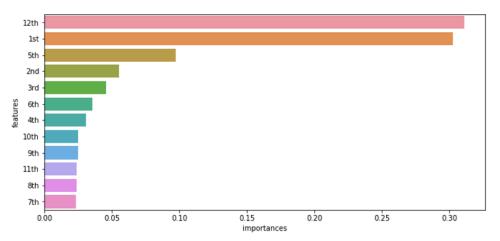


Figure 4. Effect of item click order on item purchase.

3.7. Result

We started this research from the previous study confirming the recommendation accuracy changes according to the use of Word2Vec to generate multidimensional vectors representing products.

Word2Vec has a hyperparameter named Size that controls vector dimension, allowing us to set the dimension of the vector. The recommendation accuracy was measured while increasing the vector dimension of Word2Vec from one to five. We used XGBoost classifier, a classification model of XGBoost, to train a machine learning model. The results of learning are presented below.

3.7.1. Result of Evaluation

The evaluation metrics of the classification model are indicators that evaluate the performance of the model. Classification models typically use accuracy, but accuracy is considered as a good evaluation index when the data are biased. For example, if the purchase history for all items in the data file does not exist evenly, and there is a bias for a specific item, the item can have a relatively high predictive performance, but when purchasing other items, the predictive performance can be low. Table 4 shows the results of precision, recall, and accuracy for each dimension of Word2Vec of the classification model. The precision and recall results are not biased to one side, showing that the data are not heavily biased. Thus, accuracy is used for the results of this article as presented below.

Table 4. Evalu	ıation met	rics of cla	assification	model.
-----------------------	------------	-------------	--------------	--------

Word2Vec	Precision	Recall	Accuracy
w/o Word2Vec	73.5%	71.2%	81.5%
1 Dimension	77.3%	77.3%	83.7%
2 Dimension	78.1%	76.5%	85.6%
3 Dimension	77.9%	77.1%	85.7%
4 Dimension	78.5%	78.2%	85.8%
5 Dimension	79.1%	78.5%	85.8%

Table 5 shows the evaluation metrics result of the regression model. The regressor is applicable only to the vector value of Word2Vec and shows the error between the predicted and the actual value in numerical form. MAE, MSE, RMSE, and RMSLE are used, and the results of each Word2Vec dimension are shown. In the following results, the result of accuracy is shown by the hit ratio using the Euclidean distance and cosine similarity, as mentioned earlier [34].

Symmetry **2021**, 13, 758 10 of 16

Table 5. Evaluation metrics of regression model.	Table 5.	Evaluation	metrics of	regression	model.
---	----------	------------	------------	------------	--------

Word2Vec	MAE	MSE	RMSE	RMSLE
1 Dimension	1.85	3.20	1.79	0.1165
2 Dimension	1.64	2.52	1.59	0.0844
3 Dimension	1.23	2.34	1.53	0.0750
4 Dimension	0.90	1.25	1.12	0.0485
5 Dimension	0.87	1.16	1.08	0.0425

3.7.2. Result of Classification

The classification model predicts a purchased item based on the clicked product information for each purchased product. Each product item has a unique ID, so we used a classifier for machine learning to recommend a product among the known ones. Table 6 shows the recommended accuracy before and after applying Word2Vec varying dimensions of from one to five. All training results show the average accuracy with 10 different sets of training and test data. First, without Word2Vec encoding, the recommendation accuracy of the classifier was 81.50%. Next, we used Word2Vec encoding to compare the recommended accuracy. When we encoded the products with one-dimensional vectors, the recommendation accuracy was 83.73%, which is 2.23% higher than before Word2Vec was applied. When learning with two-dimensional vector representations, the recommendation accuracy was 85.63%, which is 1.9% higher than that of one-dimensional feature representation. Afterwards, as the vector dimension increased from three to five, the recommendation accuracy also increased. The increase is not as significant as the results of one- and two-dimensional feature representations.

Table 6. Accuracy of classification model and Word2Vec dimension.

Word2Vec	Accuracy
w/o Word2Vec	81.50%
1 Dimension	83.73%
2 Dimension	85.63%
3 Dimension	85.68%
4 Dimension	85.79%
5 Dimension	85.82%

A recommendation system generally recommends multiple products rather than just one from a user's behavior. The XGBoost classifier recommended an item, and we selected four more products that were closest to the recommended one by computing the similarities. The model checked if an actual purchased item was included in the final five recommendations to calculate the accuracy, as shown in Table 7. The recommendation accuracy using the one-dimensional feature vector was 84.38%, showing an increase of 0.75% compared to recommending a single item. The accuracy of recommendation increased by 0.77% on average when the system recommended five product items.

Table 7. Hit ratio of classification model for five items.

Word2Vec	Euclidean Distance
1 Dimension ($N = 5$)	84.38%
2 Dimension ($N = 5$)	86.25%
3 Dimension ($N = 5$)	86.43%
4 Dimension ($N = 5$)	86.59%
5 Dimension ($N = 5$)	86.76%

Symmetry **2021**, 13, 758 11 of 16

3.7.3. Result of Regression

Generally, the machine learning model using a classifier works well for product recommendation problems. We can use a regressor for a recommendation because all of the products are encoded using Word2Vec vectors, which contain continuous numeric values as features. Table 8 shows the recommendation accuracy for one item closest to each dimension vector value predicted by an XGBoost regressor. We found that the accuracy increases as the vector dimension increases.

To recommend multiple products, we calculated the distance among the product items. The following is the result when we used the Euclidean distance to calculate the similarity. The Euclidean distance was used to calculate the distance between two points on the vector. The vectors of products obtained using Word2Vec are expressed in a vector space; the relationship among the products is expressed by the distance among them; and the Euclidean distance calculates the similarity among the product items. The recommendation accuracy using one-dimensional Word2Vec encoding was 67.13%, and two-dimensional vector representation had 78.75% as its recommendation accuracy, showing an increase of 11.62% in accuracy. From three-dimensional vectors, 93% accuracy was observed. The four-and five-dimensional vector representations showed 97.05% and 97.85% accuracy, showing 5.79% and 0.8% enhancement, respectively.

The following is the result obtained using cosine similarity. Cosine similarity was obtained by using the cosine value of the angle between two vectors on a vector. Cosine similarity is a method of determining the similarity of a direction rather than the size of a vector. From one- to five-dimensional vectors, it was confirmed that the recommendation accuracy increased as the dimension increased, for example, the increase in Euclidean distance from 63.43% to 69.73%, 88.31%, 93.80%, and 96.70%. Furthermore, when compared to the results using the Euclidean distance, the recommendation accuracy was relatively low.

Word2Vec	Euclidean Distance	Cosine Similarity
1 Dimension (<i>N</i> = 1)	67.13%	63.43%
2 Dimension ($N = 1$)	78.75%	69.73%
3 Dimension ($N = 1$)	91.26%	88.31%
4 Dimension ($N = 1$)	97.05%	93.80%
5 Dimension ($N = 1$)	97.85%	96.70%

Table 8. Hit ratio of regression model for the nearest item.

Table 9 shows the accuracy of recommendations for five items close to the vector predicted by an XGBoost regressor model. Table 8 shows the recommendation result using the Euclidean distance and cosine similarity. First, the Euclidean distance results showed that the recommendation accuracy with one-dimensional vectors was 87.50%, which is 20.37% higher than that of recommending one item. The recommendation accuracy with two-dimensional vectors was 85.32%, which is 6.57% higher than that of recommending one thing, but we found that it was lower than the accuracy with one-dimensional vectors. The three-dimensional vectors showed 97.49% accuracy, while the four-dimensional vectors showed an accuracy of 98.82%. The five-dimensional vectors showed 98.97% accuracy. As the dimension increased, the accuracy also increased.

When we used cosine similarity, the recommended accuracy was 85.73%, 83.67%, 95.94%, 97.94%, and 98.68% for one- to five-dimensional vectors, respectively. As with other results, the accuracy of recommendations increased as the dimensions increased. Furthermore, when recommending one item, the Euclidean distance had better results than those obtained when using cosine similarity.

Symmetry **2021**, 13, 758 12 of 16

TT 11 0	T T		c		1 1	c	c·	• •
Table 9.	Hit	ratio	Οİ	regression	model	tor	nve	items.

Word2Vec	Euclidean Distance	Cosine Similarity
1 Dimension ($N = 5$)	87.50%	85.73%
2 Dimension (N = 5)	85.32%	83.67%
3 Dimension ($N = 5$)	97.49%	95.94%
4 Dimension ($N = 5$)	98.82%	97.94%
5 Dimension ($N = 5$)	98.97%	98.68%

3.8. Validation Set

Training and test dataset were prepared for the research. After model training using the training dataset, the test followed the process using the test dataset to test and evaluate the approach. When training excessively maximizes the accuracy for the training dataset, the machine learning model may lose its generalization from overfitting. To solve this issue, we prepared a validation dataset. Table 10 shows the results of using the validation set to determine whether the results of the regression model in Tables 8 and 9 are properly trained without overfitting [35].

Table 10. Hit ratio of regression model using validation set.

Word2Vec	Euclidean Distance
1 Dimension (<i>N</i> = 1)	70.62%
1 Dimension ($N = 5$)	88.74%
2 Dimension $(N = 1)$	79.93%
2 Dimension ($N = 5$)	83.15%
3 Dimension ($N = 1$)	90.55%
3 Dimension ($N = 5$)	96.87%
4 Dimension ($N = 1$)	94.10%
4 Dimension ($N = 5$)	97.76%
5 Dimension ($N = 1$)	97.25%
5 Dimension ($N = 5$)	98.70%

3.9. Discussion

Figure 5 shows a comparison of Tables 6 and 8. Using one- and two- dimensional Word2Vec representations, the accuracy of the classification model is higher than that of the regression model. The classification model shows better results of more than 80% even when the vector dimension is small, whereas the regression model shows less than 80% accuracy when the vector dimension is small. However, we observed that the performance of the regression model exceeds that of the classification model when the dimension is three or higher. It can be observed from three-dimensional vector representations that the performance of the regression model exceeds 90%, but the classification model stays at 86%. It is confirmed that the regression model's performance increases as the relationship between the product items becomes more apparent with the increase in dimension.

Figure 6 shows a comparison of Tables 7 and 9. We noted that the recommendation accuracy when recommending five product items is higher than that of recommending one product. The accuracy with one-dimensional vector representations is 87.5%, which is about 20% higher than that of recommending one product. The regression model recommends similar product items even in the lower dimensions of Word2Vec. On the other hand, the accuracy of recommending five items by the classification model increases by about 1%. However, if a proper item is not recommended and the prediction fails, the right product items tend not to be recommended properly at the end of the recommendation. As a result, we found that the regression model recommends items with relatively close similarity, whereas the classification model does not do so.

Symmetry **2021**, 13, 758 13 of 16

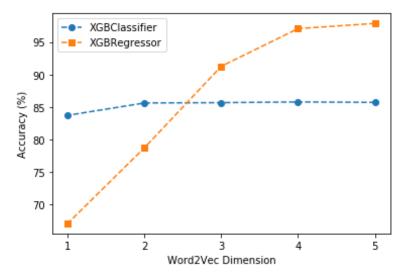


Figure 5. One-item recommendation accuracy.

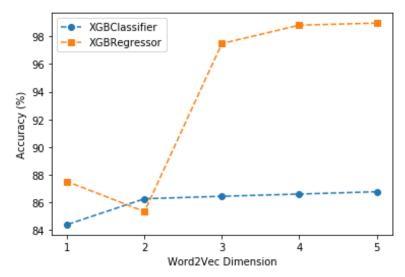


Figure 6. Five items recommendation accuracy.

Table 11 shows performance comparisons with other models. The dataset has a total of 10,000 instances. This paper used a graph convolutional neural network (GCNN)-based approach for online product recommendation. It is a method of calculating the similarity between GCNN nodes and clustering the nodes based on the interaction similarity [36]. OpGCN also uses a GCNN-based approach, but it encodes the input pattern and delivers it to the embedding layer [37]. The recommendation accuracy was 88% for OpGCN and 97% for GCNN.

Table 11. Performance comparisons.

Component	Accuracy
OpGCN	88%
GCNN	97%
Word2Vec and XGBoost	98%

4. Conclusions

In this paper, we used the Word2Vec technique to increase the accuracy of product recommendation in shopping. This technique collects and learns users' item click history as shopping patterns and recommends some product items based on the shopping

Symmetry **2021**, 13, 758 14 of 16

preferences representing their shopping experiences. Word2Vec was applied after listing the item click history of users according to the click order to identify the correlation of all products. The study was conducted under the assumption that the accuracy of the recommendation would increase when the prediction was made after determining the association among products. The feature influence of each click on the purchased items was calculated. To evaluate the proposed model, accuracy, precision, and recall were selected for a classification model. Mean absolute error, mean squared error, root mean squared error, and root mean squared logarithmic error were chosen for a regression model. Classifier machine learning models are generally suitable for a recommendation of categorical targets, but a regression model was also evaluated in this research by using the Word2Vec feature representation. We evaluated each model and compared them in terms of recommendation accuracy varying in one to five vector dimensions. From the experiment, we found that a regression model's accuracy increased as the vector dimension increased. In the case of one-product item recommendation, the classification model worked well with one and two vector dimensions, but it was revealed that the regression model with more than two-dimensional vectors surpassed the classification model. For the multiple products recommendation (five here), the regressor model's performance was higher than that of the classification model in all dimensions.

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

Funding: This research was supported by the 2021 scientific promotion program funded by Jeju National University.

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.

Abbreviations

The following abbreviations are used in this manuscript:

CBOW Continuous bag of word
XGBoost Extreme gradient boosting
MAE Mean absolute error
MSE Mean squared error
RMSE Root mean squared error

RMSLE Root mean squared logarithmic error

TP True positive
FP False positive
FN False negative
TN True negative

References

- 1. Lee, H.I.; Choi, I.Y.; Moon, H.S.; Kim, J.K. A Multi-Period Product Recommender System in Online Food Market based on Recurrent Neural Networks. *Sustainability* **2020**, *12*, 969. [CrossRef]
- 2. STATISTICS, K. Annual Report, 2020. Available online: https://kostat.go.kr (accessed on 22 April 2021).
- 3. Hao, N.; Wang, H.H.; Zhou, Q. The impact of online grocery shopping on stockpile behavior in Covid-19. *China Agric. Econ. Rev.* **2020**, 12, 459-470. [CrossRef]
- 4. Sardianos, C.; Ballas Papadatos, G.; Varlamis, I. Optimizing parallel collaborative filtering approaches for improving recommendation systems performance. *Information* **2019**, *10*, 155. [CrossRef]
- 5. Bock, J.R.; Maewal, A. Adversarial Learning for Product Recommendation. AI 2020, 1, 25. [CrossRef]

Symmetry **2021**, *13*, 758

6. Gomez-Uribe, C.A.; Hunt, N. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Trans. Manag. Inf. Syst.* **2016**, *6*. [CrossRef]

- 7. Liaw, A.; Wiener, M. Classification and regression by randomForest. R News 2002, 2, 18–22.
- 8. Loh, W.Y. Classification and regression trees. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2011, 1, 14–23. [CrossRef]
- 9. Gunn, S.R. Support vector machines for classification and regression. ISIS Technol. Rep. 1998, 14, 5–16.
- 10. Ma, L.; Zhang, Y. Using Word2Vec to process big text data. In Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 2895–2897.
- 11. Park, S.; Seong, D.; Byun, Y. A Hybrid Collaborative Filtering based on Online Shopping Patterns using XGBoost and Word2Vec. *J. Korean Inst. Inf. Technol.* **2020**, *18*, 1–8. [CrossRef]
- 12. Shahbazi, Z.; Hazra, D.; Park, S.; Byun, Y.C. Toward Improving the Prediction Accuracy of Product Recommendation System Using Extreme Gradient Boosting and Encoding Approaches. *Symmetry* **2020**, *12*, 1566. [CrossRef]
- 13. Rong, X. Word2vec parameter learning explained. arXiv 2014, arXiv:1411.2738.
- 14. McCormick, C. Word2vec Tutorial-the Skip-Gram Model. 2016. Available online: http://mccormickml.com/2016/04/19/word2 vec-tutorial-the-skip-gram-model (accessed on 3 March 2021)
- 15. Goldberg, Y.; Levy, O. word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv* **2014**, arXiv:1402.3722.
- 16. Ling, W.; Dyer, C.; Black, A.W.; Trancoso, I. Two/too simple adaptations of word2vec for syntax problems. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; pp. 1299–1304.
- 17. Lilleberg, J.; Zhu, Y.; Zhang, Y. Support vector machines and word2vec for text classification with semantic features. In Proceedings of the 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC), Beijing, China, 6–8 July 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 136–140.
- Caselles-Dupré, H.; Lesaint, F.; Royo-Letelier, J. Word2vec applied to recommendation: Hyperparameters matter. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 27 September–2 October 2018; pp. 352–356.
- 19. Tezgider, M.; Yıldız, B.; Aydın, G. Improving Word Representation by Tuning Word2Vec Parameters with Deep Learning Model. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, EA, Turkey, 28–30 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–7.
- 20. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
- 21. Khan, P.W.; Byun, Y.C.; Lee, S.J.; Park, N. Machine Learning Based Hybrid System for Imputation and Efficient Energy Demand Forecasting. *Energies* **2020**, *13*, 2681.
- 22. Bishop, C.M. Pattern Recognition and Machine Learning; Springer: Berlin, Germany, 2006.
- 23. Thakkar, P.; Varma, K.; Ukani, V.; Mankad, S.; Tanwar, S. Combining user-based and item-based collaborative filtering using machine learning. In *Information and Communication Technology for Intelligent Systems*; Springer: Berlin, Germany, 2019; pp. 173–180.
- 24. Kuhn, M. Caret: Classification and Regression Training. ASCL 2015, R Package Version 6.0-47, ascl–1505. Available online: http://CRAN.R-project.org/package=caret (accessed on 20 February 2021).
- 25. Goodfellow, I.; Bengio, Y.; Courville, A. Machine learning basics. Deep Learn. 2016, 1, 98–164.
- 26. Collobert, R.; Bengio, S. SVMTorch: Support vector machines for large-scale regression problems. *J. Mach. Learn. Res.* **2001**, 1, 143–160.
- 27. Lim, T.S.; Loh, W.Y.; Shih, Y.S. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Mach. Learn.* **2000**, *40*, 203–228. [CrossRef]
- 28. Burnaev, E.; Erofeev, P. The influence of parameter initialization on the training time and accuracy of a nonlinear regression model. *J. Commun. Technol. Electron.* **2016**, *61*, *646–660*. [CrossRef]
- 29. Hossin, M.; Sulaiman, M. A review on evaluation metrics for data classification evaluations. *Int. J. Data Min. Knowl. Manag. Process* **2015**, *5*, 1.
- 30. Botchkarev, A. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *arXiv* **2018**, arXiv:1809.03006.
- 31. Xia, S.; Xiong, Z.; Luo, Y.; Zhang, G. Effectiveness of the Euclidean distance in high dimensional spaces. *Optik* **2015**, *126*, 5614–5619. [CrossRef]
- 32. Altmann, A.; Toloşi, L.; Sander, O.; Lengauer, T. Permutation importance: A corrected feature importance measure. *Bioinformatics* **2010**, *26*, 1340–1347. [CrossRef] [PubMed]
- 33. Zien, A.; Krämer, N.; Sonnenburg, S.; Rätsch, G. The feature importance ranking measure. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Bled, Slovenia, 7–11 September 2009; Springer: Berlin, Germany, 2009; pp. 694–709.
- 34. Khan, P.W.; Byun, Y.C. Genetic Algorithm Based Optimized Feature Engineering and Hybrid Machine Learning for Effective Energy Consumption Prediction. *IEEE Access* **2020**, *8*, 196274–196286. [CrossRef]
- 35. Demšar, J. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 2006, 7, 1–30.

Symmetry **2021**, 13, 758 16 of 16

36. Shafqat, W.; Byun, Y.C. Incorporating Similarity Measures to Optimize Graph Convolutional Neural Networks for Product Recommendation. *Appl. Sci.* **2021**, *11*, 1366. [CrossRef]

37. Shafqat, W.; Byun, Y.C. Enabling "Untact" Culture via Online Product Recommendations: An Optimized Graph-CNN based Approach. *Appl. Sci.* **2020**, *10*, 5445. [CrossRef]