Modeling an e-Commerce Hybrid Recommender System Based on Machine Learning Algorithms

Antonio Panarese[a], Giuseppina Settanni[b] and Angelo Galiano[c]
Dyrecta Lab, IT Research Laboratory, Vescovo Simplicio 45, 70014 Conversano, Italy

Keywords: Recommender System (RS), Artificial Intelligence (AI), Machine Learning, XGBoost, K-Means.

Abstract: The spread of the Web and the digitalization of human society has led to the emergence of e-commerce sites. The remarkable increase in the amount of data produced by digital and automated devices forces the use of intelligent algorithms capable of processing the collected data in order to extract information. In particular, machine learning algorithms give the possibility to implement automatic models to process data and provide personalized suggestions. The advanced recommender systems are based on these models that make companies, which use the e-commerce channel, able to provide the users with suggestions on products they may be interested in. This paper proposes a model of hybrid recommender system based on the use of clustering algorithms and XGBoost, respectively, to perform a preliminary segmentation of item-customer data and predict user preference. The implemented model is discussed and preliminarily validated through a test performed using the data of a statistical sample made up of regular users of an e-commerce site.

1 INTRODUCTION

In recent years, online sales channels have established themselves overwhelmingly. This trend has been aided in part by the COVID-19 pandemic which has prompted many people worldwide to shop online through e-commerce sites, apps and social media. The multi-channel strategy improves the ability of commercial and industrial companies to collect data relating to customers and business flows. In order to cope with the need to exploit the large amount of information, which is potentially contained in the huge data collections, various methodologies have been developed whose purpose is to transform data into knowledge and subsequently into decisions. In particular, machine learning provides algorithms that allow the development of intelligent recommender systems. In fact, machine learning models are able to automatically discover hidden patterns in the data in order to make future predictions based on past events. On the other hand, users of e-commerce sites browsing the web are often subject to an information overload that disturbs their ability to make decisions (Das et al., 2019). Recommender systems are very useful in solving this problem, as they assist e-commerce users in making decisions by recommending items that match their interests or needs.

Recommender systems can be classified in the following main types (Afoudi et al., 2021), (Kiran et al., 2020), (Zhang et al., 2018):

- Content-Based approaches
  Recommendations are based on user preferences and features of the items, namely their content. These systems recommend users the items that match their profiles. The preferences are obtained from item ratings given explicitly by the users (explicit feedback) or are indirectly deduced through the consumption behaviours of the users (implicit feedback).
- Collaborative Filtering approaches
  In this case, knowledge of the characteristics of the items is not required, in fact the approach is based on the similarity relationships between users and on the score that users attribute to the various objects. The result is a flexible system, as items, that have been bought or positively

[a] https://orcid.org/0000-0001-5480-8556
[b] https://orcid.org/0000-0002-9954-5371
[c] https://orcid.org/0000-0001-6732-6695

Panarese, A., Settanni, G. and Galiano, A.
Modeling an e-Commerce Hybrid Recommender System Based on Machine Learning Algorithms.
DOI: 10.5220/0011745700003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 3*, pages 706-712
ISBN: 978-989-758-623-1; ISSN: 2184-433X

valued by users similar to the target user, are recommended to him.

- Hybrid approaches
  This solution is potentially more performing, as it combines the aforementioned approaches with the aim of mitigating their disadvantages. A hybrid approach can be implemented by combining various machine learning algorithms for processing different types of data, such as item data in a content-based perspective or user data in a collaborative filtering perspective.

Machine learning provides several Artificial Intelligence (AI) algorithms us4eful for predictive analysis. A recommender system that integrates a predictive model gives an e-commerce company the ability to suggest other products or services to customers, that might be of interest to them (Mu, 2018). The opportunity to provide personalized suggestions determines an increase in products sold by the company. In particular, the number of niche items sold increases, because the recommender system allows this type of products to gain visibility.

Compared to traditional recommender systems, AI-based recommender systems can use machine learning techniques to:

- automatically learn the latent characteristics of the user by integrating various types of heterogeneous multi-source data;
- build more suitable user models based on the user's preference requirements;
- re-process data more effectively, change different user preferences and improve the accuracy of the recommendation;
- improve user performance and satisfaction.

E-commerce users can assume different behaviors, however the deep knowledge of their needs allows a business company to predict at least part of their behavior. The study of behavior, context and interactions of users facilitates the design of the e-commerce software system. Machine learning algorithms can be used to extract important information from the data also by analyzing the behavior and attitudes of users in browsing, in order to improve the customer's user experience.

The eXtreme Gradient Boosting (XGBoost) algorithm is a boosting ensemble models that efficiently implements the gradient boosting algorithm. This type of machine learning algorithm is ideal for implementing robust and highly accurate predictive models, starting from a set of weak learners (Panarese et al., 2022). Furthermore, XGBoost is equipped with different mechanisms that allow it to control overfitting, such as regularization, pruning and sampling (Chen et al., 2016). These mechanisms are controllable through different tuning hyperparameters. The authors of (Shahbazi et al., 2019) discuss a recommender system based on a XGBoost model, which is used to process data from an online shopping mall. In detail, the dataset containing information on user profiles and clicks is processed in order to recommend items to each user.

Another machine learning tool that finds application in recommender systems is the k-means algorithm as it performs cluster analysis (Xue et al., 2005). It is an unsupervised learning technique that identifies hidden structures in data. The goal of this clustering technique is to determine a natural grouping of data according to their characteristics. Therefore, each cluster is made up of the data that most closely resemble each other (Massaro et al., 2020). K-means clusters customers into groups with similar characteristics (interests, preferences, purchases, etc.) and consequently obtains useful information to manage customers based on their behavior. The study (Xu et al., 2012) proposes the use of clustering analysis to improve a collaborative recommender system. In details, cluster algorithms are used to segment data in user-item subgroups, as user may have similar tastes only reagarding some items. This approach helps capture similar user tastes on a subset of elements and improve suggestions. The research (Bhaskaran et al., 2021) discusses an intelligent hybrid recommender system based on clustering strategy with the aim of providing suggestions in the field of e-learning. Clustering technique allows the proposed system to automatically adapt to student requirements, interests and knowledge levels. As a whole, the recommender system extracts the functional models of the students and automatically learns the varied characteristics of the students.

The authors of (Zhang et al., 2018) propose a recommender system based on a two-step method. In the first step, a quadric polynomial regression model is applied in order to obtain the latent properties of items and customers. In the next step, the properties thus obtained are processed by a Deep Neural Networks (DNN) model which provides, as output, the prediction of the rating scores of the various items by each user. During training, the DNN model processes historical data relating to the user evaluation of items and learns to generalize the relationship between items and users. The research (Gridach, 2018) presents a hybrid approach based on Convolutional Neural Network (CNN) and Probabilistic Soft Logic (PSL) in order to implement a recommender system, that reduces the problem of sparsity and non-interpretability. The framework developed using these tools provides recommendations to each user taking into consideration the user judgment of the items and reviews provided by all users on the various items. The proposed system is validated by using the Mean Square Error (MSE). Similarly, the authors of (Kiran et al., 2020) discuss a hybrid method that combines a collaborative approach to the use of content. This combination is achievable by means of deep artificial neural networks (DANNs), that process the data of items and users and integrate the extracted information. The use of DANNs also allows to reduce the problem of cold start and to improve performance both in terms of running time and precision. The method is benchmarked by means of MSE, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and R-squared.

In this paper, we discuss a hybrid recommender system based on XGBoost algorithms and k-means. The implemented XGBoost model predicts user rating coefficients for items that they have not yet judged. The predictive models based on XGBoost has already proved very accurate in the predictions in various fields (Memon et al., 2019), (Weldegebriel et al., 2020). Additionally, clustering analysis is performed on both customer and item data. This methodology allows users to be segmented into subgroups, since a user may have varied interests and belong to a different subgroup depending on the items. Clustering users into subgroups related to item groups leads to increased precision in the XGBoost model output.

2 METHODOLOGY

Companies, that offer customers the opportunity to take advantage of recommender systems based on machine learning algorithms, obtain significant benefits, including increasing customer satisfaction and the real opportunity to retain him. The choice of algorithms is fundamental, in fact, the precision of the recommendations is not only directly proportional to the number of customer interactions with the e-commerce website, but also depends greatly on the accuracy of the used predictive model.

The research discussed in this paper proposes an intelligent hybrid recommender system, that enhances the performance of a simple collaborative system due to the use of powerful machine learning tools, such as k-means and XGBoost. The XGBoost algorithm is an optimized implementation of the gradient boosting algorithm that uses a set of weak predictive models, usually decision trees, to achieve a strong learner (Chen et al., 2016). Machine learning algorithms, which are embedded in the proposed recommender system, analyze data and automatically provide intelligent recommendations to users.

Figure 1 shows the overall flowchart of the developed recommendation model. The e-commerce platform permits to collect the raw data, namely sales data, user navigation data and items data. Before being processed by machine learning algorithms, the raw data undergoes a preliminary processing.
During this data pre-processing phase, the following techniques were used:

- *data Cleaning*
  blank fields are filled with missing data, noisy data is smoothed out, and unrealistic values are removed;
- *data Integration*
  data from various sources are integrated and inconsistencies are resolved;
- *data Reduction*
  the amount of input data is reduced without compromising the validity of the analyzes;
- *data Transformation*
  the data is prepared for the processing following phases by means of one hot encoding technique that transform it into binary code.

In the following phase, the features of users and items are modelled.

First of all, the user-item utility matrix $R \in \mathbb{R}^{N \times M}$ is written, where N is the number of users and M is the number of items. R is obtained by using the historical data of the customers, such as the opinions expressed on a product, its purchase, having read or viewed information about it. In particular, the generic element R_{ij} contains information on the i-th user liking with the j-th item. If no judgment has been expressed, the R_{ij} coefficient is null.

Furthermore, two different k-means algorithm is applied to perform cluster analysis of user and item data. A K-means model clusters users with the same features, such as age, interests, frequency of purchase, products purchased, web pages visited, number of clicks, product categories and so on. The output of this analysis is the matrix $U \in \mathbb{R}^{N \times A}$, where A is the dimension of user features. The i-th row of this matrix is the vector U_i containing the latent features of the i-th user. Similarly, another k-means model is applied to cluster the items in different groups of product category, according to their features. In this case, the output is the matrix $V \in \mathbb{R}^{M \times B}$, where B is the dimension of item features. The j-th row of V is the vector V_j that represents the features of the j-th item.

This features modeling phase permit to enter normalized information about the content of customers and items into the predictive model, also including the collaborative filtering approach. The possibility of basing the predictions on the processing of information, in our model related to both the items and the users, considerably reduces the cold start problem and gives more accurate results. Furthermore, the cluster analysis simplifies and optimizes the processing by the algorithms in the next phase, resulting in greater speed and precision in obtaining the final result.

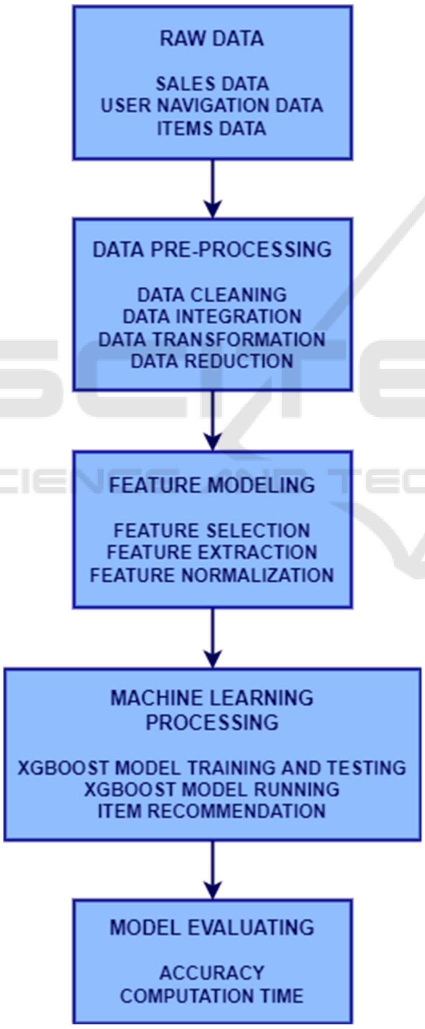

Figure 1: Flowchart of the implemented recommender system.

Machine learning processing is based on a XGBoost classifier that analyzes the training data, automatically learns the properties of the users and items and, finally, predicts the values of the rating matrix for each user-item pair. The training phase gives the predictive model the ability to predict the user's evaluation score for items that have not been evaluated by the user. In this way the recommender system is able to provide recommendations for a specific user, in fact the items with the highest score are recommended to each user.

XGBoost allows the sequential trees building process to be accomplished through a parallel implementation, which reduces the computational cost. The implementation features of XGBoost algorithm also optimize the management of hardware resources. A further optimization of the computational performance is due to the pruning technique. In addition, XGBoost very efficiently addresses the problem of data scarcity thanks to very effective training, hyperparameters can be set to their optimal value in. This feature is fundamental in the case of recommender systems, because it allows to reduce the cold start that afflicts them.

The purpose of the XGBoost algorithm is to predict the value of the coefficients of the rating matrix $S \in \mathbb{R}^{N \times M}$. Particularly important is the prediction of the coefficients S_{ij} corresponding to the missing ratings ($R_{ij} = 0$), that allows to recommend to each user an item that fits his features, but it has not yet visualized. The system suggests to i-th user the j-th item that is characterized by the maximum value of S_{ij}. The XGBoost model is capable of learning non-linear latent factors. In fact, using tunable hyperparameters, an XGBoost model can accurately capture the non-linear interactions between the features and the target variables. The coefficients of the utility matrix and the output of the clustering analysis are given as input to the XGBoost model, which predicts the value of the user-item rating matrix S. The computation of the rating coefficients can be modelled in the following form:

$$S_{ij} = f\left(\alpha R_{ij}\right) + g\left(\beta_j U_i\right) + h\left(\gamma_i V_j\right) + Z_{ij}$$

where Z_{ij} indicates the noise contribution, α is a numerical coefficient, β_j and γ_i express the data heterogeneity in relation to items and users, respectively. Finally, f, g and h are nonlinear functions.

This model extracts the functional patterns of users and items from their respective data and predicts the values of the rating matrix for each user-item pair. During the training phase, the model was trained by processing the training dataset created with 80% of the historical data available. The remaining

20% is used to obtain the testing dataset used in the subsequent model validation phase.

Machine learning algorithms are crucial as they are used in feature modeling phase and in the processing. Figure 2 shows the architecture of the machine learning overall model that takes place in these two phases.

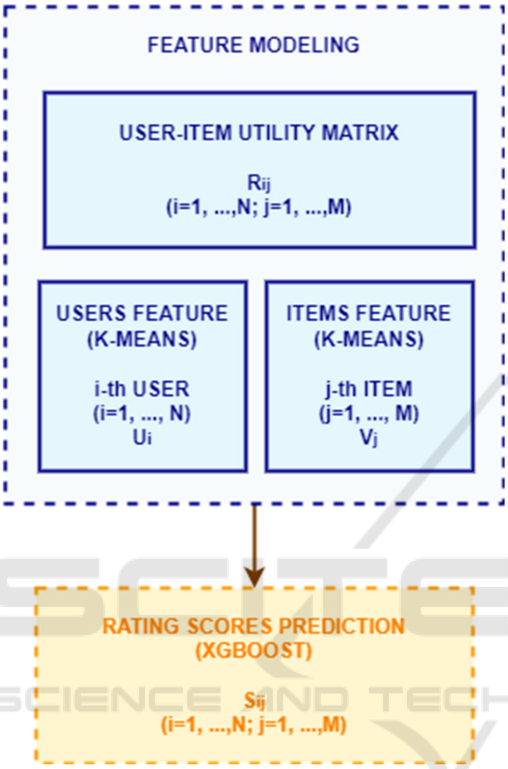

Figure 2: Architecture of the machine learning model.

The high-level language python is used for the implementation of machine learning models. This object-oriented programming language offers great flexibility and the opportunity to use a variety of development platforms. Python also has free and open-source libraries that allow the modular development of recommender systems based on machine learning algorithms. The most important tools useful for the implementation of the proposed model are tensorflow, keras, scikit-learn and XGBoost. The GridSearchCV function of scikit-learn library allows to automatically find the best values of the different parameters needed to pre-train the model.

The main XGBoost parameters that are set by GridSearchCV are the following:

- learning_rate, denoted by eta, the learning rate affects how quickly the model fits the residual error by shrinking the weights on each step;
- n_estimators, it indicates the number of decision trees of the model;
- max_depth, the maximum depth of a tree is used to control overfitting. The deep that each tree will grow in any boosting round depends on the value of this parameter;
- gamma, it states exactly the minimum loss reduction that is required to make a node split;
- min_child_weight, the value of this parameter determines the minimum sum of weights of all observations that is required in a child. This parameter allows to reduce overfitting.
- colsample_bytree, it denotes the features fraction that is sampled for constructing each tree;
- alpha, it is responsible for controlling the regularization on leaf weight by means of Lasso (Least Absolute Shrinkage and Selection Operator) technique, named L1 regularization. Alpha should be used to reduce the computational cost when there is high dimensionality.
- lambda, it controls the regularization on leaf weight by means of Ridge regularization technique, named L2 regularization. Lambda can also be used reduce overfitting.

Regarding the evaluating metrics, the default metric (logloss) has been used to validate the model prediction. During the training and testing step, RMSE and MAE have been also computed as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N_S}\sum_{i,j}\left(S_{ij} - \hat{S}_{ij}\right)^2} \tag{1}$$

$$\text{MAE} = \frac{1}{N_S}\sum_{i,j}\left|S_{ij} - \hat{S}_{ij}\right| \tag{2}$$

where $\hat{S}_{ij}$ represents the predicted score and N_S denotes the sample size, namely the number of tested ratings.

After training and validating the model, a preliminary test has been carried out. Table 1 shows the value of the main hyperparameters that were used during the test. These values were automatically tuned by using GridSearchCV. Among users with a significant number of purchases, 10 users have been randomly selected and divided into two groups each consisting of 5 users. The only condition satisfied in forming these two groups has been to consider customers who had purchased a similar number of items in the observation period equal to one month

(Month 1). The model calculated the suggestions for the 10 users, but recommendations have been provided only to the users of the first group (group 1). Table 2 shows user purchases over Month 1 and the next month (Month 2).

Table 1: Setting of XGBoost parameters during the test.

Hyperparameter	Value
learning_rate	0.11
n_estimators	125
max_depth	6
gamma	0
min_child_weight	1
colsample_bytree	0.8
alpha	0
lambda	1

Table 2: Number of items purchased during the test for group 1 (Recommendation group) and for group 2 (No Recommendation group).

	Purchased Items Number			
	Group 1 (Recommendation)		Group 2 (No Recommendation)	
	Month1	Month 2	Month 1	Month 2
User 1	12	17	8	9
User 2	24	25	17	17
User 3	19	22	25	23
User 4	9	12	11	12
User 5	13	16	14	13
Total	77	92	75	74

The number of products purchased by users of group 1 increases due to recommender system suggestions, while for the second group (group 2) the total number of purchases remains almost constant. Figure 3 shows the percentage increase for each user. In particular, we observe that using the recommender system the mean value (horizontal lines) reaches an increment of 23.6% of purchases, while without it the percentage increase is close to zero. The standard deviation of the percentage increase is equal to 14.7 for group 1 and 9.3 for group 2, which did not benefit from the suggestions. The standard deviation values are quite high because the number of users considered is low.

As regards the measurement of the error, in this preliminary test we only used the MAE, obtaining a value of 0.74 which is very promising. In fact, the dataset used in the test is not rich since it contains only 21,705 rating records relating to 959 items and 497 users. Indeed, only the 5 users selected in the test were taken into consideration when calculating MAE. Further tests are needed to more fully characterize the model performance.

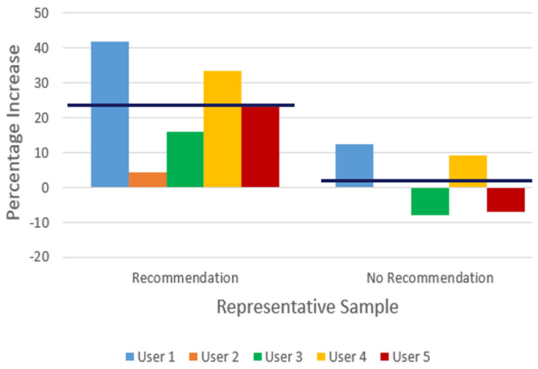

Figure 3: Results of the preliminary test.

3 CONCLUSIONS

The present paper proposes a hybrid recommender system that provides users with suggestions obtained from the processing of data relating both to the characteristics of the items and to user feedback and purchases. The original designed framework combines different machine learning technologies in order to incorporate the peculiarities of the two main approaches of the recommender systems and, therefore, exploit both the contents and the similarities between users. Novelty is also produced by the use of XGBoost algorithms combined with clustering algorithms in the context of recommendation systems. An additional advantage derives from applying clustering analysis to both users and items in order to achieve recommendations through an analysis based on user-item subgroups. In fact, the XGBoost model will carry out the predictions of S_{ij} ratings considering a subset of elements with similar characteristic and the subset of users interested in these elements. Finally, the use of XGBoost algorithms for rating prediction provides the model with great accuracy. The proposed model has been validated by means of preliminary tests, but in future work it will be deeply tested and compared with hybrid models that make use of different machine learning and deep learning models (Batmaz et al., 2019). In particular, the developed method will be tested on datasets available on the MovieLens and Epinions websites. The results obtained will be compared with those available in the state of the art.

REFERENCES

Afoudi, Y., Lazaar, M., & Al Achhab, M. (2021). Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network. *Simulation Modelling Practice and Theory*, 113, 102375.

Batmaz, Z., Yurekli, A., Bilge, A., Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. In *Artif Intell Rev* 52, 1–37. https://doi.org/10.1007/s10462-018-9654-y

Bhaskaran, S., Marappan, R., & Santhi, B. (2021). Design and analysis of a cluster-based intelligent hybrid recommendation system for e-learning applications. Mathematics, 9(2), 197.

Chen, T.; Guestrin, C. (2016). 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.

Das, J., Banerjee, M., Mali, K., & Majumder, S. (2019, December). Scalable Recommendations Using Clustering Based Collaborative Filtering. In *2019 International Conference on Information Technology (ICIT)* (pp. 279-284). IEEE.

Gridach, M. (2020). Hybrid deep neural networks for recommender systems, *Neurocomputing*, 413, 23-30, https://doi.org/10.1016/j.neucom.2020.06.025.

Kiran, R, Kumar, P., Bhasker, B. (2020). DNNRec: A novel deep learning based hybrid recommender system, *Expert Systems with Applications*, 144, 113054, https://doi.org/10.1016/j.eswa.2019.113054.

Massaro, A., Panarese, A., Galiano, A. (2020). Infrared Thermography applied on Fresh Food Monitoring in Automated Alerting Systems. In *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT,* Roma, Italy, pp. 554-558, doi: 10.1109/MetroInd4.0IoT48571.2020.9138207.

Memon, N., Patel, S.B., Patel, D.P. (2019). Comparative Analysis of Artificial Neural Network and XGBoost Algorithm for PolSAR Image Classification. In: Deka, B., Maji, P., Mitra, S., Bhattacharyya, D., Bora, P., Pal, S. (eds) Pattern Recognition and Machine Intelligence. PReMI 2019. Lecture Notes in Computer Science(), vol 11941. Springer, Cham. https://doi.org/10.1007/978-3-030-34869-4_49

Mu R. (2018) A Survey of Recommender Systems Based on Deep Learning. In *IEEE Access*, vol. 6, pp. 69009-69022, doi: 10.1109/ACCESS.2018.2880197.

Narayan, S., & Sathiyamoorthy, E. (2019). A novel recommender system based on FFT with machine learning for predicting and identifying heart diseases. *Neural Computing and Applications*, 31(1), 93-102.

Panarese, A., Settanni, G., Vitti, V., Galiano, A. (2022). Developing and Preliminary Testing of a Machine Learning-Based Platform for Sales Forecasting Using a Gradient Boosting Approach. *Applied Sciences*. 12(21):11054. https://doi.org/10.3390/app122111054

Shahbazi, Z., Byun, Y. C. (2019). Product recommendation based on content-based filtering using XGBoost classifier. *International Journal of Advanced Science and Technology*, 29, 6979-6988.

Weldegebriel, H.T., Liu, H., Haq, A.U., Bugingo, E., Zhang, D. (2020). A New Hybrid Convolutional Neural Network and eXtreme Gradient Boosting Classifier for Recognizing Handwritten Ethiopian Characters. IEEE Access. 8, 17804-17818.

Xu, B., Bu, J., Chen, C., & Cai, D. (2012, April). An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web* (pp. 21-30).

Xue, G. R., Lin, C., Yang, Q., Xi, W., Zeng, H. J., Yu, Y., & Chen, Z. (2005, August). Scalable collaborative filtering using cluster-based smoothing. *In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 114-121).

Zhang, L., Luo, T., Zhang, F., Wu, Y. (2018). A Recommendation Model Based on Deep Neural Network. *In IEEE Access*, vol. 6, pp. 9454-9463, 2018, doi: 10.1109/ACCESS.2018.2789866.