

Developing Restaurant Recommendation System with Neural Collaborative Filtering Method

¹Ilham Huseyinov and ²Tolga Hamitovali

¹Professor, İstanbul Aydın University of Software Engineering

Email: ilhamhuseyinov@aydin.edu.tr

²Graduated Student Institute of Science, İstanbul Aydın University

Email: tolgahamitovali@stu.aydin.edu.tr

Abstract: Recommendation Systems are designed to provide a personalized product or service to the user. Its purpose is to predict the future actions of users based on their past behavior and make suggestions accordingly. Recent studies have proven that Deep Learning-based collaborative filtering method has a high success rate. However, there is no study on the implementation of this method in restaurant recommendation systems. The goal of this study is to fill this gap. For this purpose, different models were designed using different restaurant datasets and a deep learning-based collaborative filtering algorithm. Using the evaluation criteria of Restaurant Recommendation Systems, the models developed in this study were compared with the results of other studies. The comparison results are presented graphically. As a result, the hyperparameters of the most optimal model based on the Deep Learning-based collaborative filtering algorithm were found.

Keywords- Collaborative Filtering, Deep Learning, Neural Collaborative Filtering, Matrix Factorization, Multi-layer Perceptron, NeuMF, Layer, Inner Product

I. INTRODUCTION

Nowadays, the fact that information in the digital world is excessively large, complex, and dynamic often bores users. The main solution to overcome this problem is to guide users with recommendation systems to discover products or services in a personalized way. Recommendation Systems is the general term for systems that make recommendations for users based on their ratings, reviews, feedback, and likes, taking into account their search history. For example, the leading sites and apps in the digital world, such as Amazon, Netflix, Spotify, Apple, also provide recommendations to their users using their recommendation systems. While a high success rate in these recommendation systems increases user satisfaction, It also significantly increases the revenue generated by the applications because if they are shown the products they need at the right time and are increasingly encouraged to buy, the potential revenue per user will increase. The reason why we have turned to restaurant recommendation systems in this study is that online food applications are at the forefront of applications that have gained prominence in today's world and have become an indispensable part of our lives due to pandemic conditions. The problems with using these applications, the superficial customization of customers when ordering, and the fact that reviews that are important to customers are not transparently reflected were our motivation for this article.

Recommender systems typically use models such as 1) Content-based Filtering Methods; 2) Collaborative Filtering Methods; 3) Hybrid Recommender Systems.[1] The goal of content-based recommender systems is to create a common profile for each user and item. This created profile is considered as an example of users. The specific details of users and products are important. This model learns from the user's preferences and finds items that are similar to the user's preferences then recommends them. Various methods such as the TF-IDF method, element vector, user vector, cosine similarity are used to calculate similarities between items. The purpose of the collaborative filtering model is to create a model based on user's past behavior and use it to predict their future behavior. The basis of this method is the logic that users with similar profiles can make similar preferences. For example, if user X has the same opinion as user Y on a topic, they may both prefer a restaurant with the same type of cuisine. In the collaborative filtering method, the Pearson correlation coefficient, the Spearman correlation coefficient, the vector similarity, and the Cosine Similarity method can be used in calculating the similarities to build the user-item model. Hybrid recommendation systems, on the other hand, are the method that emerges from the need by combining content-based and collaborative recommender systems. In some studies in the literature, it has been found that the performance of this method is more accurate than the other methods under consideration.[2] One of the most used applications of this hybrid system is Spotify. There are several key factors that influence the performance of recommender systems:

1. The "cold start" problem - insufficient user information and the recommendation system does not know user preferences.
2. Limited recommendations - Recommendation systems are mostly limited to implicit user feedback.
3. Undecided users, manipulative ratings, confidentiality agreements to protect the user's personal data.
4. Incorrect selection of similarity algorithms.

In recent years, deep learning, one of the sub-branches of artificial intelligence, which is a machine learning method, has become very popular. One of the ways to use these deep neural networks in recommender systems is to model the similarity relation. [3] but this method has not been used in studies in the field of restaurant recommendation systems. For this reason, this article investigated the performance of the Neural Collaborative Filtering Method in restaurant recommendation systems. In particular, we will explain in detail the machine learning algorithms (matrix factorization, multi-layer perceptron, hybrid algorithm combining matrix factorization, and multilayer perceptron NeuMF). Among our goals in this study is to find the neural collaborative filtering algorithm that has the highest success rate in the experiments we will perform on the restaurant dataset.

The contents of the article are as follows: In the second section, studies on restaurant recommendation systems are reviewed; in the third section, the architectural design and algorithms of the neural collaborative filtering method are explained. Then, in the last section, the dataset, experiment, and results are explained.

II. LITERATURE REVIEW

In this section, we review previous work on restaurant recommendation systems and collaborative filtering methods based on deep learning. We see that articles on restaurant recommendation systems and neural collaborative filtering methods are still very limited. In this study, a music recommendation system with collaborative filtering and neural collaborative filtering method was developed. 20,000 users, 6,000 songs, and 470,000 rating records were used and separate prediction results were obtained for both methods. [4] In another study, a location-based restaurant recommendation system was developed and a model was built by combining location-based and content-based filtering methods. In the study, a Foursquare dataset was used which was run on the mobile application. [5] In the restaurant recommendation system developed in this study, more than one parameter was considered for the first time. Historical data of the user, location, preferences of the user, etc. a restaurant recommendation system was explained considering such factors.[6] In terms of deep neural networks and user comments, machine learning methods have been used to try to predict ratings from user comments.[7] The following article developed a restaurant recommendation model using a hybrid recommender system. The study used an algorithm that combines content-based and collaborative filtering in a hybrid way to recommend the most suitable restaurant.[8] In another article on a restaurant recommendation model, psychographic characteristics such as age and gender are first presented, from which the person's lifestyle, interests, and personality can be predicted based on mobile usage patterns. A new restaurant recommendation model is presented, secondly using demographic characteristics and thirdly using current location. The results are also validated with statistical metrics such as mean or standard deviation.[9] This paper explores the field of bipartite hybrid recommender systems by comparing four different recommender methods and seven different hybridization strategies. The applications of 41 hybrids, including some new combinations, were studied and compared. The study concludes that graded and driven hybrids work well, especially when two components with different strengths are combined. [10] The applicability of component-based methods in personalized nutrition counseling was explored. Variations of popular approaches used in other domains, such as Rocchio's document classification algorithm, can be adapted to provide personalized nutritional advice. In this work, the authors developed a model to evaluate a variety of Rocchio's algorithm adapted to this domain to investigate content-based methods. [11] This is a study on the use of deep learning techniques in the development of recommender systems. Very detailed information about deep learning methods such as multi-layer perceptron networks (MLP), CNN, RNN, DSSM are presented. [12] The most detailed study on the neural collaborative filtering method can be found in this article. Neural network architectures for collaborative filtering have been studied. Methods like matrix factorization, multilayer perceptron networks are explained in detail. Extensive experiments have also been conducted using real-world Pinterest and MovieLens datasets. It is intended to serve as a guide for studies that want to develop deep learning methods for recommender systems. [13] In summary, as seen in the literature review, recommender systems have been used in many different platforms, from e-commerce applications to TV applications for watching series and movies. In examining the studies, it can be seen that artificial neural networks cannot be integrated into recommender systems and that there are uncertainties regarding their performance. Starting from that approach, we intend in our study to find the highest success rated deep-learning algorithm with several trials on an actual data set and going forward to lead restaurant recommendation systems or online food order apps in particular.

III. NEURAL COLLABORATIVE FILTERING METHOD

As we mentioned in the first part of the article, the neural collaborative filtering method is used for similarity modeling. That is, the user-item inner product in collaborative filtering is replaced by neural network architecture. The most popular for this procedure is the Matrix Factorization Method, which projects users and elements into a commonly hidden space, using a vector of hidden features to represent the user or element. The user's influence on the element is then modeled as the inner product of its hidden vectors. This family of methods is widely recognized for its effectiveness, as Simon Funk reported in a 2006 blog post during the Netflix Prize competition, sharing his findings with the research community. Prediction results can be improved by assigning different processing weights to the hidden factors based on item popularity and user activity.[14] Besides this method, another artificial neural network-based method is used to model user-item-feature interaction: Multi-layer Perceptrons. It adds hidden layers on the combined user item vectors (MLP framework) to learn about user-item interactions. This gives the model a lot of flexibility and non-linearity to learn about user-item interactions. The question then becomes: how can we combine matrix factorization and multi-layer perceptrons under the framework of neural collaborative filtering so that they reinforce each other to better model complex user-element interactions? In answer to this question, we come across the combination of these two methods, matrix decomposition, referred to in the literature as NeuMF, and the hybrid combination of multilayer perceptrons.

3.1. Neural Collaborative Filtering Architecture

Below is a visualization of the neural collaborative filtering architecture. First, the user-item identification vector is binary encoded in the input layer (i.e., categorical variables are represented in binary). If item (i) is equal to 1, it means that it has interacted with user u. User (u) is equal to user.

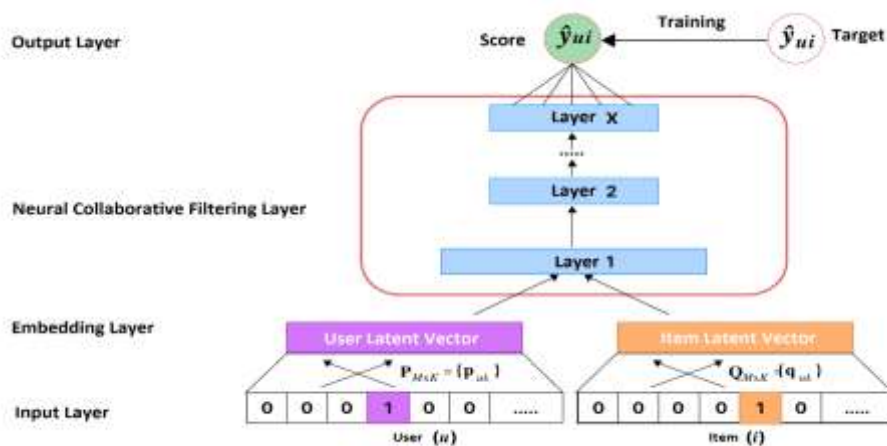


Figure 1: Neural Collaborative Filtering Architecture

The input layer is followed by the fully connected embedding layer, which projects the sparse representation into a dense vector. It is then imported into a multilayer neural network architecture, which we call the neural collaborative filtering layer, to map the prediction values of the hidden vectors of the user elements. Each layer found here is used to discover the hidden interactions/structures between the user and the item. The output layer determines the size of layer X in Fig. 1 and the capacity of the model. The last layer displays the predicted result so that the loss between the predicted result and the training data is minimal.[15] The predictive model of the neural collaborative filtering method can be mathematically formulated as follows;

$$\hat{y}_{ui} = f(P^T v_u^u, Q^T v_i^i | P, Q, \theta_f) \quad (1)$$

Where P is the latent factor matrix for users, Q is the latent factor matrix for items, and θ_f is the parameters of the model. Since F is defined as a multilayer neural network, it can be formulated as follows:

$$f(P^T v_u^u, Q^T v_i^i) = \phi_{out}(\phi_x(\dots \phi_2(\phi_1(P^T v_u^u, Q^T v_i^i))\dots)), \quad (2)$$

In the formula, ϕ_{out} denotes the output layer, and ϕ_{out} denotes the mapping task for the x-th neural collaborative filtering layer.

3.2. Matrix Factorization (MF) Method

The matrix factorization method is a class of neural collaborative filtering models. It is more effective than other methods because it helps us find the hidden properties underlying the interactions between the user and the item. It has gained popularity compared to other methods in the literature. [16] The first version of this model was proposed by Simon Funk in a famous blog post where he explained the idea of factorization of interaction matrix. In the competition organized by Netflix to develop recommender systems in 2006, the matrix factorization method won the top prize and increased its popularity.

Matrix factorization is used to discover hidden properties underlying interactions between two different types of entities (user-item). In theory, each user and item is transferred into a hidden space represented by a hidden vector. The more similar these transferred latent vectors are, the more similar the corresponding user preferences are. Here, the similarity ratio of the hidden vectors, that is, the similarity between two users, can be determined by cosine similarity or dot product. Mathematically, we can express this process with the following formula;

$$\hat{y}_{ui} = f(u, i | p_u, q_i) = p_u^T q_i = \sum_{k=1}^K p_{uk} q_{ik} \quad (3)$$

In this formula, \hat{y}_{ui} represents the prediction score, p_u represents the latent vector for user u, q_i represents the latent vector for item i, and K represents the dimension of the latent space. As can be seen in the formula, since in the matrix factorization method each dimension of the hidden space is independent, these dimensions are linearly combined with the same weights. This models the mutual interaction of hidden properties between user and item. For this reason, the matrix factorization method can also be called a linear model of the hidden factors. To better explain this with an example, let's consider the users in the first 3 rows of the matrix below.

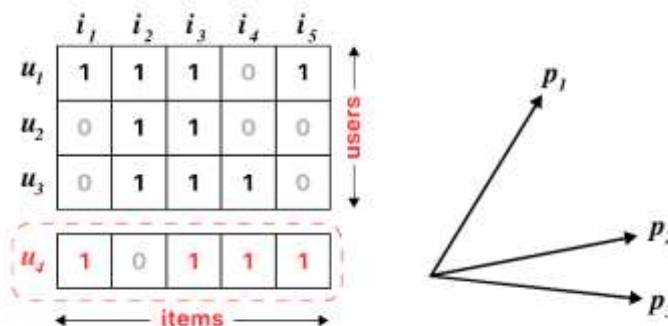


Figure 2: User-Item Matrix

$S\{x,y\}$ show the similarity between user x and user y. If the cosine similarity of the users in the first 3 rows is computed, we get the result $S\{2, 3\} > S\{1, 2\} > S\{1, 3\}$. For this reason, the geometric relations of p_1, p_2 and p_3 are in the hidden space as in the picture above. If we look at the cosine similarity of the new user u_4 in the user element matrix, we see that $S\{4, 1\} > S\{4, 3\} > S\{4, 2\}$. In other words, the newly added user u_4 is most similar to user u_1 . After that, it is most similar to user u_3 and then u_2 . With this result, the model of matrix factorization positions p_4 closest to p_1 . The final state of the hidden space is as follows.

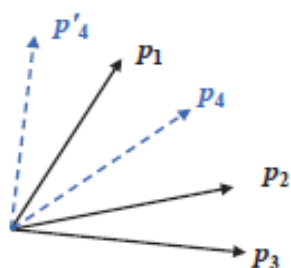


Figure 3: User Latent Space

3.3. Multi-Layer Perceptron (MLP) Method

In a simple vector concatenation, the hidden features between the user and item are usually not evaluated, which is insufficient for the method neural collaborative filtering. A standard multi-layer perceptron network is used to solve this problem and learn about hidden user-item interactions. In this way hidden layers can be added on vectors. These added layers are tightly connected so that the neuron in each layer connects to the neuron in the next layer. This process provides the model with much flexibility and nonlinearity for learning user-item interactions. Below is the mathematical formula of the multilayer perceptron network model under the neural collaborative filtering method;

$$\begin{aligned}
 z_1 &= \phi_1(p_u, q_i) = \begin{bmatrix} p_u \\ q_i \end{bmatrix}, \\
 \phi_2(z_1) &= a_2(W_2^T z_1 + b_2), \\
 &\dots \\
 \phi_L(z_{L-1}) &= a_L(W_L^T z_{L-1} + b_L), \\
 \hat{y}_{ui} &= \sigma(\mathbf{h}^T \phi_L(z_{L-1}))
 \end{aligned}
 \tag{4}$$

$W(x)$ represents the weight matrix, $b(x)$ the bias vector, $a(x)$ the activation function for the layer x detector, p the latent vector for the user, q the latent vector for the item. ReLU is well suited for sparse data and reduces the likelihood of the model being overkill. ReLU (Rectified linear activation function) is used here for the perform an activation operation between the activation functions of the layers. ReLU is well suited for sparse data and reduces the probability that the model is overloaded.

3.4. NeuMF Method

This method is a general neural network method that arises from matrix factorization and the question of how combinability of multilayer perceptron networks. The goal is for the two methods to be more mutually reinforcing in the modeling process. This method is basically a combination of matrix factorization, which implements a linear kernel to model hidden property interactions between the user and the element, and multi-layer perceptron networks, which use multiple hidden layers to model nonlinear interactions. The working principle is as follows; Embedded layers are created for both users and items, one for matrix factorization and the other for a multi-layer perceptron network.

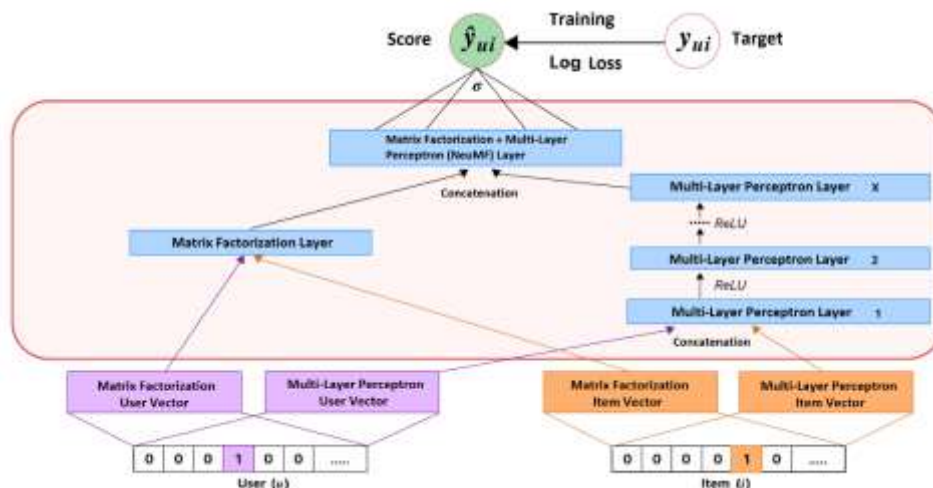


Figure 4: NeuMF Architecture

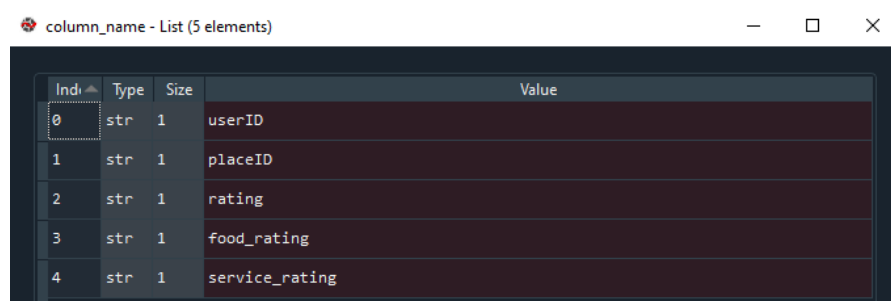
The mathematical model of the joining process in Fig. 4 is as follows.

$$\begin{aligned}
 \phi^{MF} &= p_u^G \odot q_i^G, \\
 \phi^{MLP} &= a_L(W_L^T (a_{L-1} (\dots a_2 (W_2^T \begin{bmatrix} p_u^M \\ q_i^M \end{bmatrix} + b_2) \dots) + b_L), \\
 \hat{y}_{ui} &= \sigma(\mathbf{h}^T \begin{bmatrix} \phi^{MF} \\ \phi^{MLP} \end{bmatrix})
 \end{aligned}
 \tag{5}$$

In this formula, p_u^G and p_u^M specifies the user embedding layer for matrix factorization and multi-layer perceptron networks, respectively. q_i^G and q_i^M indicates the embedding of items. As in multilayer perceptron networks here too ReLU is used for the activation procedure between activation functions.

IV. DATASET

In this study, a real-world dataset was used to conduct the experiments of the proposed models. The dataset used was taken from Kaggle (<https://www.kaggle.com/uciml/restaurant-data-with-consumer-ratings>). It was selected because the dataset was used to create a list of the best restaurants according to consumer preferences. It has also been used in several studies to gain experience with collaborative filtering methods. The dataset contains data for a total of 3,684 user-item interactions with 3 different evaluations from 1162 customers. The original state of the dataset, which consists of 5 columns including user ID, place ID, rating, food rating, and service rating, is as shown in Fig. 5. There are some operations that need to be performed to understand the data and prepare the data set for the experiment to be performed. On this dataset, operations such as loss data testing, the creation of a correlation matrix between numerical properties, outlier control, and skewness testing using the pandas, scipy, and sci-kit learn libraries in the spyder editor in the python environment were performed to further increase the success rate of the experiment.



Indx	Type	Size	Value
0	str	1	userID
1	str	1	placeID
2	str	1	rating
3	str	1	food_rating
4	str	1	service_rating

Figure 5: Data Set Column Name

V. EXPERIMENTS

In the experiment part of the study, we conduct experiments on the dataset we prepared to find answers to the following questions;

1. Does neural collaborative filtering or collaborative filtering method perform better on the same data set?
2. If we consider the neural collaborative filtering algorithms separately, which one has the highest success rate?

To evaluate the performance of the experiments in this study, we use two mathematical metrics that have been adopted by offline systems as evaluation criteria and are used in particular by machine learning models in the literature. These are the root mean squared error (RMSE) and the mean absolute error (MAE). In the RMSE method, the distance between the values predicted by the estimation algorithm and the actual values is determined. The results can range from zero to infinity. Negative or lower results mean better performance. If the result is zero, it means that the prediction model is error-free. The RMSE value can be calculated with the formula in Eq.6.

$$RMSE = \sqrt{\frac{1}{n} \sum (f_i - o_i)^2} \quad (6)$$

In the formula n stands for the number of samples, f_i for the predicted value, O_i for the actual values. The principle of operation of the method MAE is as follows, a prediction error is taken for each data, and all these errors are then converted into positives. This is done by taking the absolute value. Then the MAE value is calculated by taking the average sum of all the absolute errors calculated. The MAE value can be calculated with the formula in Eq 7.

$$MAE = \frac{\sum_{i=1}^n |f_i - o_i|}{n} \quad (7)$$

Once the operations on the dataset and the evaluation criteria were determined, the appropriate hyperparameters were determined to create the model. When designing deep learning models, hyperparameters such as the number of layers, data size (batch size), learning rate, number of training rounds (epoch size) should be selected that represent the problem well and are appropriate for the experiments to be performed. The values to be used in this model were determined according to other studies in the literature and can be seen in the following table 1;

Table 1: Hyperparameters

Data size	64
Epoch	10
Learning Rate	0.001
Optimization Algorithm	Adam

We separate the data in 2 different ways for 2 different models that will be developed later. For collaborative filtering, the highest score given by each user (i.e., 2 points of the score column) is separated as test data and the rest is the training data. In the neural collaborative filtering method, the dataset is divided into 80% training data and 20% testing data. For this reason, we have 1,037 test data and 2,947 training data.

After loading the data, importing the Python libraries and the hyperparameters we specified, latent layers of size 64 were created for the User_ID and Restaurant_ID elements. Experiments were tested with 3 and 4 hidden layers for all methods.

First, a user hidden property vector and an item hidden property vector are created to construct a user item matrix in the matrix factorization method. The neural network layer was chosen as the target matrix and defined as the output layer of a single neuron, since the goal was to update the generated matrices by the inner product and make them as similar as possible to the real user item matrix. ADAM an optimization algorithm was used to find the optimal value. Another experiment was conducted for multilayer perceptron networks with the same dataset. Since our goal is to learn the interaction function between user and item, the hidden layer is used where the user item layers are combined. The neuron in each layer is connected to the neuron in the next layer. The activation function with relu is applied between all hidden layers. After this experiment, matrix factorization and multilayer perceptron networks are combined before implementing the last output layer in parallel. The obtained results have been evaluated against the established performance criteria and compared with the results of the CF algorithm.

Details are given in the results section.

VI. CONCLUSION

In this section, the results of the model are discussed and analyzed in comparison to other benchmarking methods. Looking at each method separately, we see that the NeuMF method with 4 hidden layers gives the best results in both outcome metrics. The 3-layer tests conducted here showed that the NeuMF method had the lowest error rate. After the NeuMF method, the multi-layer perceptron method was found to outperform the matrix factorization and collaborative filtering methods in both 3-layer and 4-layer tests. Among the neural collaborative filtering algorithms, matrix factorization performed the worst in the experiments. But even mf has done better on all levels than CF. Detailed comparisons of the results of the experiments are presented in Table 2 according to the target metrics.

Table 2: Comparison of Experiments Results

Layers	Metrics	NeuMF	MLP	MF	CF
3	RMSE	12,501	17,055	24,144	46,523
	MAE	9,812	13,783	18,393	38,744
4	RMSE	10,977	15,801	19,114	30,314
	MAE	6,798	9,714	17,071	28,109

The experimental results have clearly shown us that algorithms based on artificial neural networks have a lower error rate. On this basis, it has been shown that for a restaurant recommendation system to be developed, a NeuMF-based neural collaborative filtering system is most suitable. In this study, we have shown that when a choice between NCF and CF is desired, one of the NCF algorithms should always be chosen for restaurant recommendation system.

Finally, this paper examines the architecture and algorithms of neural collaborative filtering in detail. Experiments were conducted on a real restaurant dataset to measure how artificial neural networks integrate with the collaborative filtering method and the contribution of this combination. At the same time, experiments have also compared neural collaborative filtering algorithms with different layers simultaneously. On the performance criteria we set for this study, NeuMF far outperformed MF and MLP. This study can be continued with other performance criteria to be determined from the same data set. Ultimately, a 3-sided satisfaction environment can be created for restaurants as well as users and applications.

REFERENCES

- [1] Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S., Stettinger, M. (2014). Basic Approaches in Recommendation Systems.
- [2] Çano, E., Morisio, M. (2019), Hybrid Recommender Systems: A Systematic Literature Review.
- [3] Roy, A. "Introduction To Recommender Systems- 2: Deep Neural Network Based Recommendation Systems" (2020), <<https://towardsdatascience.com/introduction-to-recommender-systems-2-deep-neural-network-based-recommendation-systems-4e4484e64746>>.
- [4] Girsang, A., Wibowo, A., Jason, Roslynlia. (2021). Comparison between Collaborative Filtering and Neural Collaborative Filtering in Music Recommendation System, ISSN: 2415-6698.
- [5] Gupta, A. and Singh, K. (2013). Location Based Personalized Restaurant Recommendation System for Mobile Environments.
- [6] Md. Habib, A., Md. Rakib, A. (2016). Location, Time and Preference Aware Restaurant Recommendation Method.
- [7] Sahu, S., Anand, A., Oruganty, K., Gattu, M. (2016). Relation extraction from clinical texts using domain invariant convolutional neural network.
- [8] Mr. N. Varatharajan, J. Guruprasad, K. Mathumitha. (2020). Restaurant Recommendation System Using Machine Learning. E-ISSN: 2456-6713.
- [9] Katarya, R. (2015). Restaurant recommender system based on psychographic and demographic factors in mobile environment. Electronic ISBN:978-1-4673-7910-6.
- [10] Burke, R. (2007). Hybrid Web Recommender Systems.
- [11] Almeida, J. (2015). Personalized Food Recommendations Exploring Content-Based Methods.
- [12] Elahi, M., Ge, M., Ricci, F., Fernández-Tobías, I., Berkovsky, S., David, M. (2015). Interaction Design in a Mobile Food Recommender System.
- [13] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T. (2017). Neural Collaborative Filtering.
- [14] Hallinan, B. and Striphas, T. (2014). Recommended for you: The Netflix Prize and the production of algorithmic culture
- [15] Bobadilla J., Ortega, F., Gutiérrez, A., Alonso, S. (2020). Classification-based Deep Neural Network Architecture for Collaborative Filtering Recommender Systems.
- [16] Koren, Y., Bell R., Volinsky, C. (2009). MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS