Restaurant Recommendation System

Rutuja Shah
Khoury College of Computer Sciences
Northeastern University
Boston, United States
shah.rut@northeastern.edu

Shalvi Desai
Khoury College of Computer Sciences
Northeastern University
Boston, United States
desai.shal@northeastern.edu

I. Introduction

With cities becoming more cosmopolitan by each passing day, there have been an ample amount of options for restaurants with varied decors and relished cuisines from across the world. With dozens of restaurants available, it becomes a hassle to discover an ideal restaurant as per a user's liking since there are many factors that come into play when picking a restaurant to dine at. Hence, recommendation systems are at rescue to save time and efforts of individuals as they would help users narrow down their options from a pool of available options based on certain features of their interest. First thing checked by any individual would be reviews of a restaurant they intend to visit. As per the statistics provided by YELP, 45 percentage of customers are likely to check reviews before visiting. The businesses listed on yelp have seen an overall increase of 9 percentage of their annual revenue due to this. This implies that user reviews are a deciding factor when people are trying to choose between restaurants and this feature is exactly what recommendations make use of when providing recommendations. Using recommendation systems would help a user find restaurants similar to the restaurants they have previously liked and would also help them discover new restaurants that could be of their interest based on the preferences of similar users. The objective of this project is to create a useful recommendation system that recommends restaurants to a user based on their preferences, also known a content-based recommendation system and to recommend users restaurants based on preferences of similar users, also known as collaborative filtering.

Content-based recommendation systems provide recommendations by comparing representation of contents describing an item to the representation of the content describing the interest of the user. In early days of their discovery, these systems were used solo for applications such as predicting movie ratings for example. But over time, hybrid approaches such as content based systems with collaborative filtering and content-based systems with clustering were incorporated to make better predictions. For this project, we will be exploring content-based recommendation systems using clustering techniques that group similar restaurants together based on a set of features. We will be exploring clustering based recommendation systems using two popular models: K-means and Agglomerative clustering.

The word collaborative filtering was first coined by a group called tapestry [1] to make an effective email broadcast system, by sending email pertaining to users interests. The primary objective was to leverage the social interaction among the user-item pairs. A detailed analysis was made across multiple users and their respective interest to construct an enriched features space for user-item pairs. Initially this strategy [2] exploited the concepts of predictive analytics and correlation statistics, later on when this problem took shape of a classification problem, and after that dimensionality reduction and feature extraction was brought into the play. Collaborative filtering sorts options for the customers based on their past history, by finding restaurants having similar features or by finding users who have similar interests. In the current implementation we have used two approaches for implementing collaborative filtering namely singular value decomposition and multilayer-perceptron. For the scope of this project, we will be surveying these different content-based and collobrative filtering approaches and assessing how well each method provides recommendations.

II. Related Work

A. Content-based recommendation system

While a lot of work has been done around clustering being an integral part of the collaborative filtering process [6,7,8,12], there remains a scarcity of research surrounding content based filtering using clustering algorithms. Soma Bandyopadhyay et el [9] proposed an approach for product recommendation for E-commerce businesses in which they used K-means clustering for the effective segmentation of customers who bought apparel items and PCA for the dimensionality reduction of different features of products and customers. Dongmoon Kim et al [10] proposed using K-means clustering algorithm to recommend pieces of music which are close to a user's preference considering multiple genres. S.Renuka et al [11] proposed an unsupervised content based article recommendation system in which the articles with similar contextual meaning, i.e., the articles with similar word distributions are grouped together using K-means and agglomerative clustering. First the articles were grouped into appropriate clusters. Then, the model computed pairwise Euclidean distances between all the articles belonging to a particular cluster and the article read by the user. The 15th closest articles belonging to that cluster were then returned as recommendations for the user. Although there has been research surrounding content based recommendations using clustering in various domains such as the approaches described above, clustering data on restaurants and providing recommendations based on a user's liking on various cuisines is yet to be explored and that is what this project aims to achieve.

B. Collaborative based filtering recommendation system

The fundamental objective of the recommendation system is to predict rating of item that user has not rated yet. For this several approaches have been used in collaborative filtering. In [3] a probabilistic approach has been proposed where a bayesian network of decision tree is used to predict the rating of an item for the respective user. The primitive model makes an assumption that user will provide rating for any item asked by system, which is very unrealistic hence a concepts of active learning was introduced for user personalization which can solicit rating according to users preference. In [4] an aspect model has been introduced which is an extension over the bayesian model by adding the pool of movies which have higher probability of getting rated by the respective user. Thus a personalization term is added to maximize the likelihood of the rating that would be provided by the user. Another approach based upon item-item model have bee used by amazon [5], they build a table of similar items aggregated them with the respective user's profile and us it as recommendation. In order to scale this algorithm t millions of their users they initially segment user with simila interests using clustering algorithms and they use item-ite model for collaborative filtering. There are certain drawback of collaborative filtering that still needs to be improved upo such as the cold start problem, it is difficult to generat embeddings for the items that have not been included durin training and addressed at the time of inference. Any additiona features cannot be passed to the model apart from the query or item ID. Recommendation system are still an active area of research to overcome problems such as scalibility, sparse utility matrix and popularity bias.

III. Overview

A. K-means clustering

K-means clustering is an unsupervised machine learning algorithm that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid). This results into a partitioning of the data instances in which they are grouped by similarity (Euclidean distance).

B. Agglomerative clustering

Agglomerative clustering is a type of hierarchical clustering used to group objects in clusters based on their similarity. It follows a bottom-up approach in which it starts by treating each object as a singleton cluster. Next, pairs of clusters are successively merged until all clusters have been merged into one single cluster containing all objects. The result is a dendrogram; a tree-based representation of the objects.

C. Singular Value Decomposition

Singular value decomposition is a technique used for dimensionality reduction in machine learning models. It is used to reduce the feature space, while minimizing the reconstruction error. It retains the majority of the variance explained by the features to project the patterns of the actual data samples with the reduced dimension.

$$\mathbf{A} = USV.T$$

We constructed a utility matrix which consist of rows as users and restaurants as columns, each cell value represents the respective ratings.We decompose this matrix into three matrices i.r U with users x r dimension , S is a r x r diagonal matrix and V with r x restaurants dimension. Here r is a latent variables which projects our matrix into latent feature space. In Figure 1 r cap is the reconstructed utility matrix of the ratings after applying svd. The x and y are obtained in such a manner that expected error between actual rating and predicted rating is minimized. In order to prevent overfitting we have added the penalty term to our objective function.

$$\hat{r}_{ui} = x_i^T . y_u$$

$$Min(x,\ y) \sum_{(u,\ i)\in K} (r_{ui} - x_i^T . y_u)^2$$

$$Min(x,\ y) \sum_{(u,\ i)\in K} (r_{ui} - x_i^T . y_u)^2 + \lambda(\|x_i\|^2 + \|y_u\|^2)$$

$$Min(x,\ y,\ b_i,\ b_u) \sum_{(u,\ i)\in K} (r_{ui} - x_i^T . y_u - \mu - b_i - b_u)^2 + \lambda(\|x_i\|^2 + \|y_u\|^2 + b_i^2 + b_u^2)$$

Fig. 1. Objective function used for SVD

D. Neural network

A neural network is a collection of nodes , stacked into different layers with weighted edges. They are renowned for their ability to learn non-linear relationship amongst the data points. Matrix decomposition hinders the performance of recommendation because it accommodates only linear relationship between user-item pair. As it simple multiplication of the latent features may not be sufficient to represent the complex relation of user-item pair.

Here we have utilized neural network architecture to learn user-item feature interaction. We have used multilayer perceptron to project our feature in the latent space with high level of non-linearity . As per the approach mentioned in the paper [6], encode the user latent vector and item latent vector in single encoding space to create an embedding and further use it to predict the user ratings.

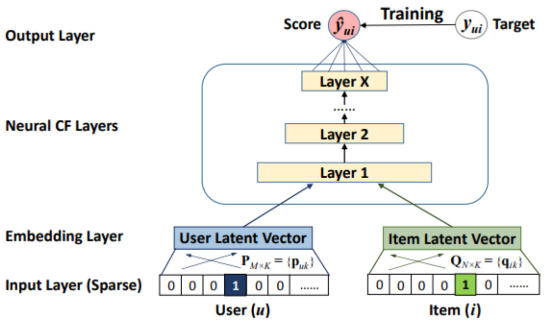

Fig. 2. Neural Network architecture

E. Cosine Similarity

Cosine similarity is one of the most popular used similarity metric. Given two vectors $\vec{a1}$ $\vec{a2}$, then its cosine similarity is computed by:

$$\cos(\mathbf{a1},\mathbf{a2}) = \frac{\mathbf{a1}\mathbf{a2}}{\|\mathbf{a1}\|\|\mathbf{a2}\|} = \frac{\sum_{i=1}^{n}\mathbf{a1}_i\mathbf{a2}_i}{\sqrt{\sum_{i=1}^{n}(\mathbf{a1}_i)^2}\sqrt{\sum_{i=1}^{n}(\mathbf{a2}_i)^2}} \tag{1}$$

Higher similarity is better. The similarity approaching 1 means they are very similar and approaching 0 means they are quite different.

F. Evaluation

1) Davis Boulder Index: Davis Boulder Index is a metric for evaluating clustering algorithms. It is an internal evaluation scheme, which is a ratio of the within cluster scatter, to the between cluster separation. A lower value means that the clustering is better.

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{i\le j\le k, j\ne i}\left(\frac{\max_{x,y\in C_i} d(x,y) + \max_{x,y\in C_j} d(x,y)}{\min_{x\in C_i, y\in C_j} d(x,y)}\right)$$

k clusters

Diameter of a cluster $C_i = \max_{x,y\in C_i} d(x,y)$

Fig. 3. Davis Boulder Index formula

2) Silhouette Coefficient: Silhouette Coefficient or silhouette score is a validation metric used to calculate the goodness of a clustering algorithm. It is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). Its value ranges from -1 to 1 where a value of 1 means clusters are well separated from each other and clearly distinguished, 0 means clusters are indifferent, or that the distance between clusters is not significant, and -1 indicates that the clusters produced are of poor quality.

3) Root mean squared error: For evaluating the restaurant recommendation system we have used Root mean square error as an evaluation metric. Here x represents the actual ratings

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

Fig. 4. Silhouette Coefficient

and $\hat{x}$ represents the predicted rating by our recommendation algorithm:

$$\textbf{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}$$

4) Precision@K: Precision@K is a renowned metric to asses the quality of recommendations. It is an external evaluation scheme that determines the quality of the predictions.

$$\textbf{P@K} = \frac{\#\text{Top K recommended items}}{\#\text{Relevant items}}$$

IV. Proposed Approach

A. Content-based recommendation system using K-means and agglomerative clustering

We want to provide restaurant recommendations to a user based on the cuisine(s) the user likes. To do this, we will use two unsupervised clustering algorithms: K-means and agglomerative clustering. We first cluster the restaurants based on their similarity in terms of the kinds of cuisines they offer. Then, given a user instance in which a user has liked a particular restaurant x, we provide recommendations of restaurants similar to the cuisine(s) of restaurant x.

1) Tables used and Preprocessing: In the yelp dataset, the business table is used. The business table contains a lot of different businesses so we filter it to only contain restaurants. Second, the business table contains businesses in various cities so we filter the business dataset to contain only the restaurants that are open in Philadelphia. Next, the "category" section in the filtered table contains a lot of non-relevant categories so we filter them out to only contain cuisines. The yelp dataset contains 121 cuisines so we further filter the table to only contain those restaurants that have at least one of the 121 cuisines. Next, we need to prepare the data so that we can feed it into the clustering algorithms. The cuisines in the categories column are textual so we need to convert them into a vector representation of 0's and 1's where the 0 indicates the absence of a cuisine for a particular restaurant and 1 indicates the presence. Doing this increases the number of columns in our table because we expand the category column into multiple columns in which we have one column per cuisine. Applying clustering to such high dimensional data won't produce good results so it is important to reduce the dimensions/columns of the data. We do this by applying Principal component analysis to our table. Using PCA, we reduce the columns to 15. Once that is done, the data would be ready for the clustering algorithms.

2) K-means clustering: The first step is to determine the optimal number of clusters needed for our data. We determine this by using the elbow method. The optimal number of clusters returned by the elbow method were 13 (Fig:5) so we trained our k-means model using 13 models. The distance metric used is euclidean distance and the kmeans++ method is used for initializing the centroids. We train the model on 1505 instances of the filtered table from the previous step and we leave out 3 instances to test our results visually (shown in code). The next step predicts the cluster label for each of the 3 test instances and returns all the instances that belong to the same cluster as the test instance as recommendations for the test instance. For example, if we predicted cluster 8 for test instance y, we would then return all the instances in cluster 8 as recommendations for test instance y. Since a lot of instances would be in the same cluster as the test instance, we return the top 10 as recommendations for y (sorted by highest similarity in cuisine to the test instance).

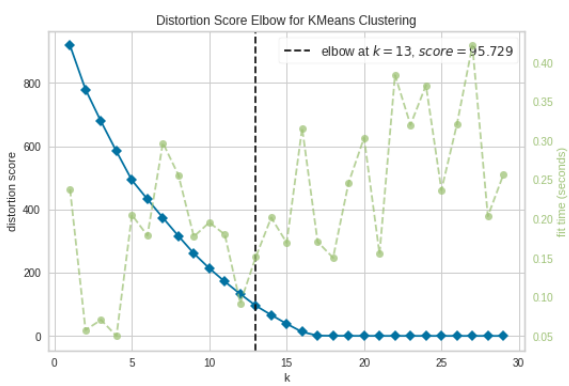

Fig. 5. Elbow method for choosing optimal number of clusters for K-means clustering.

3) Agglomerative clustering: Again, the first step is to determine the optimal number of clusters needed. To determine this, we would need to cut the dendrogram at an appropriate threshold distance. We plotted a dendrogram so that we could visually determine where the cut should be but as shown in figure 6, it is not possible to visually determine which distance threshold to make the optimal cut at. Hence, a different method was used to obtain the optimal number of clusters needed. We calculated the Davis boulder index and silhouette coefficient (table 1) on 15 models with different cluster sizes (6 to 19) and upon doing this, clusters ranging from 10 to 17 gave a good DB index value (that is not too high and not too low) and a good silhouette coefficient score (starts to drop after 17 clusters). We decided to go ahead with 13 clusters as that falls in the middle of the acceptable range. The model was then trained on 13 clusters using euclidean distance as the distance metric and average link as the linkage criteria. After training the model, the rest of the procedure is the same as K-means. We make predictions on 3 test instances.

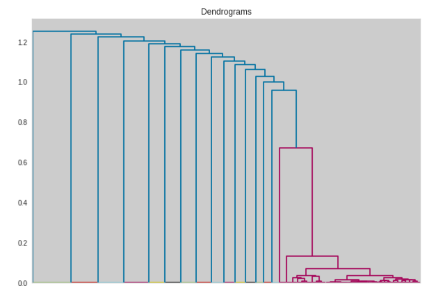

Fig. 6. Dendrogram for Agglomerative clustering.

No. of clusters	DB Index	silhouette coefficient
6	0.508	0.584
7	0.477	0.631
8	0.438	0.678
9	0.396	0.725
10	0.355	0.766
11	0.314	0.803
12	0.269	0.839
13	0.219	0.875
14	0.179	0.903
15	0.133	0.932
16	0.081	0.961
17	0.039	0.974
18	0.057	0.878
19	0.090	0.865

TABLE I
INFERENCE RESULTS

B. Collaborative filtering for recommendation system using Singular Value Decomposition and Neural Networks

Collaborative filtering addresses the shortcomings of the content based recommendation system such as overspecialization i.e recommendations are limited to users past history and none of the novel interests are explored. Another major drawback is to distinguish significant features for maintaining user's profile. Hence collaborative filtering was introduced to overcome some of these issues.

This approach mainly focuses on finding similar users , assuming that they have similar interests will help to predict unknown ratings. Another model utilized by this approach is item-item model, where a similarity is computed based on items to predict the rating for the new item. To implement these approaches we have use two main techniques namely singular value decomposition and neural network.

1) Singular Value Decomposition: Our yelp dataset consists of multiple tables, for our use case we have use two table namely business table and review table. Initially we started by data preprocessing. We filtered our review and business table for the user belonging from the city of Philadelphia as they had the maximum number of reviews pertaining to restaurant category of the business, hence our matrix would be less sparse. Next for each restaurants there were nested attributes , hence we made individual columns for them and created one hot vectors for each of them. We merged the review table and business table based upon the business id and created a utility matrix. Each cell consisted of the stars given by each user to the corresponding restaurant. Our utility matrix has dimension of (47112,623). Further we have decomposed our utility matrix using truncated SVD which factorizes our matrix into constituent arrays of users and restaurants. Here we can define K i.e number of latent components that will represent our utility matrix in the latent space. We have defined number of components equal to 12.

The transformed matrix was them used to to evaluate correlations between restaurants. We extracted a correlation matrix of (623,623) dimension using pearson correlation coefficient. We can recommend any restaurant to a user by evaluating his rating for similar restaurant by using the correlation matrix.

2) Multi-layer Perceptron: For this approach , we encoded user ids and restaurant ids and stored the values of number of unique users, number of unique restaurants,minimum rating and maximum rating. After that we have passed this encodings to our model, where we get embeddings by taking a dot product between user vector and restaurant vector. These embeddings consist of the dimension n factor to represent the weights for each user-restaurant pair. This n factor is an arbitrary value that has to be equal for user and restaurant vector. In order to avoid overfitting we have included the bias term to each embedding. Finally we add this embeddings and send it to the sigmoid layer. We scale each output using minimum rating and maximum rating and hence we get the intended ratings from the utility matrix that we passed.In the below figure we have mentioned the architecture the model we have used.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 1)]	0	[]
input_2 (InputLayer)	[(None, 1)]	0	[]
embedding (Embedding)	(None, 1, 50)	2355600	['input_1[0][0]']
embedding_2 (Embedding)	(None, 1, 50)	33350	['input_2[0][0]']
reshape (Reshape)	(None, 50)	0	['embedding[0][0]']
reshape_2 (Reshape)	(None, 50)	0	['embedding_2[0][0]']
embedding_1 (Embedding)	(None, 1, 1)	47112	['input_1[0][0]']
embedding_3 (Embedding)	(None, 1, 1)	667	['input_2[0][0]']
dot (Dot)	(None, 1)	0	['reshape[0][0]', 'reshape_2[0][0]']
reshape_1 (Reshape)	(None, 1)	0	['embedding_1[0][0]']
reshape_3 (Reshape)	(None, 1)	0	['embedding_3[0][0]']
add (Add)	(None, 1)	0	['dot[0][0]', 'reshape_1[0][0]', 'reshape_3[0][0]']
activation (Activation)	(None, 1)	0	['add[0][0]']
lambda (Lambda)	(None, 1)	0	['activation[0][0]']

Fig. 7. Neural Network architecture

V. Experiments and Evaluation

A. Dataset

https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset. The dataset consist of 6 json files, we have taken user.json and bussiness.json which are required for our use case.

There are two json files : i

1) Business.json : It consists of features like business id , business name, address,city,state,postal code,latitude , longitude,stars, review count,is open , stars , attributes and hours.
2) Review.json : It consists of features like review id, user id , business id , stars , date and text.

B. Results

Below we evaluate the results of each of the proposed methods and assess their efficiency in providing recommendations.

1) K-means and Agglomerative clustering: For clustering based recommendation systems, it is not possible to use evaluations metrics such as RMSE or precision-recall because we do not have the ground truth to compare the results with. So, we will need to assess the quality of results based on the quality of the clusters formed. Quality of a clustering algorithm depends on how cohesive and separated all of the clusters are. If clusters formed are cohesive and well-separated, it would mean that we have formed good clusters and that would mean that the recommendations returned would be of good quality. To measure the cohesiveness and well-separation of the clusters formed, we use two metrics: David boulder index and silhouette coefficient.

Method	Davis Boulder Index	silhouette coefficient
Kmeans	0.219	0.874
Agglomerative	0.219	0.875

TABLE II
Inference results

We know that the lower the DB index, the better the clustering. And we know that the closer the value of the silhouette coefficient to 1, the better the clustering. Thus, these scores prove that both clustering techniques produce good quality clusters. The DB index and the silhouette coefficient of both the algorithms is more or less the same so, we can conclude that both are just as good as each other for this application.

2) Singular Value Decomposition: For singular value decomposition we computed precision values for top 4 , top 5 and top 6 recommendations.

Method	p@4	p@5	p@6
SVD	0.75	0.6	0.666

TABLE III
Inference results

3) Neural Network: We split the dataset in 80-20 proportion i.e 80% for training data and 20% for the testing dataset.For neural networks we have used rmse metric to evaluate the quality of the predictions.

Method	RMSE
Multilayer Perceptron	1.5872

TABLE IV
INFERENCE RESULTS

Below figure is the visual representation of the actual and predicted value of the ratings.

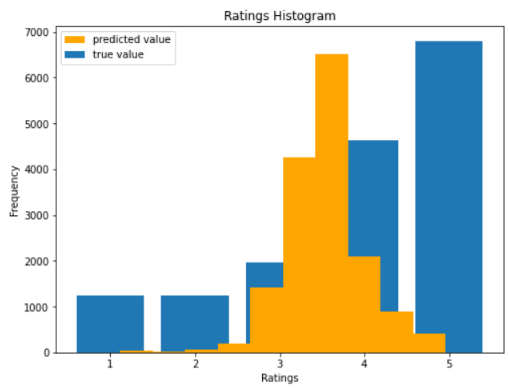

Fig. 8. Predicted vs Actual Ratings.

VI. Conclusion and Future Work

Content-based recommendation system using clustering is an effective technique for grouping similar restaurants based on the cuisines they offer and then providing recommendations to a user based on the cluster they are assigned to. The use of Principal component analysis along with clustering to reduce dimensions of the data makes clustering algorithms more efficient as they perform better with low dimensional data in comparison to high dimensional data. Both K-means and Aggomerative clustering techniques produced cohesive and well-separated clusters and as a result, the recommendations returned were similar to the types of cuisines users liked.

The clustering methods implemented can be improved by incorporating the following:
1. Incorporating the use of collaborative filtering in the current model such that clusters are formed based on similarities between users in terms of their ratings and reviews provided. Sentiment analysis can be used to interpret the reviews returned by the user as those would be more informative than using just the reviews given by users. We can then recommendation restaurants to user x based on restaurants rated by similar users (belonging to the same cluster as user x)
2. Incorporating the use of other features such as restaurant location and attributes such as: fine dining, ambience, timings etc when forming clusters so that we can provide recommendations more specific to a user's preferences.

The singular value decomposition technique shows significant results of 74% precision score .Collaborative filtering has certain drawbacks such as popularity bias , cold start problem. ignorance of crucial features. Current approach can be improved upon by taking heuristic feature into consideration along with the ratings of the user-item pairs. We need to find a way to solicit information for the new user to avoid the cold start problem. We also need to add a personalization component so that users with unique taste can be benefitted out of it. We can use hybrid approach such as global estimate models to combine the best of both the worlds and get rid of the popular bias issue.

References

[1] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. Communications of the Association of Computing Machinery, 35(12):61–70, 1992.

[2] Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98), pages 46–54, Madison, WI, 1998. Morgan Kaufmann.

[3] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI,July 1998.

[4] Abhay S. Harpale and Yiming Yang. Personalized active learning for collaborative filtering. In SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 91–98, New York, NY, USA, 2008. ACM.

[5] Xiangnan He ,Lizi Liao, Liqiang Nie, Xia Hu,Tat-Seng Chua, "Neural Collaborative Filtering",, Arxiv,2017.

[6] G Geetha et al 2018 J. Phys.: Conf. Ser. 1000 012101. A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System

[7] Qing Li and Byeong Man Kim, "Clustering approach for hybrid recommender system," Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003), 2003, pp. 33-38, doi: 10.1109/WI.2003.1241167.

[8] S. Puntheeranurak and H. Tsuji, "A Multi-clustering Hybrid Recommender System," 7th IEEE International Conference on Computer and Information Technology (CIT 2007), 2007, pp. 223-228, doi: 10.1109/CIT.2007.54.

[9] Bandyopadhyay, S., Thakur, S.S. Mandal, J.K. Product recommendation for e-commerce business by applying principal component analysis (PCA) and K-means clustering: benefit for the society. Innovations Syst Softw Eng 17, 45–52 (2021). https://doi.org/10.1007/s11334-020-00372-5

[10] D. Kim, K. Kim, K. Park, J. Lee and K. M. Lee, "A music recommendation system with a dynamic k-means clustering algorithm," Sixth International Conference on Machine Learning and Applications (ICMLA 2007), 2007, pp. 399-403, doi: 10.1109/ICMLA.2007.97.

[11] Renuka, S., Raj Kiran, G.S.S., Rohit, P. (2021). An Unsupervised Content-Based Article Recommendation System Using Natural Language Processing. In: Jeena Jacob, I., Kolandapalayam Shanmugam, S., Piramuthu, S., Falkowski-Gilski, P. (eds) Data Intelligence and Cognitive Informatics. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-15-8530-2_13.

[12] E. Kannout, "Context Clustering-based Recommender Systems," 2020 15th Conference on Computer Science and Information Systems (FedCSIS), 2020, pp. 85-91, doi: 10.15439/2020F54.

[13] Qilong Ba; Xiaoyong Li; Zhongying Bai, "Clustering collaborative filtering recommendation system based on SVD algorithm", IEEE, 2013