Product Recommendation System Using NLP Technique

Jugal Kishore Gandhesiri
Department of Computer Sciences
Texas A And M University - Corpus Christi
Corpus Christi, Texas, USA, 78414
jgandhesiri@islander.tamucc.edu

***Abstract*—Recommendation systems are information filtering tools that present items to users based on their preferences and behavior, for example, suggestions about scientific papers or music a user might like. Based on what we said and with the development of computer science that has started to take an interest in big data and how it is used to discover user interest, we have found a lot of research going on in the area of recommendation and there are powerful systems available. In the unsupervised learning domain, this paper introduces a novel method for creating a recommender framework that combines Collaborative Filtering with Content Based Approach and Self-Organizing Map neural network technique. By testing our system on a subset of the Amazon Review Dataset, we demonstrate that our method outperforms state-of-the-art methods in terms of accuracy and precision, as well as improving the efficiency of the traditional Collaborative Filtering methodology**

***Index Terms*—Recommender systems, Collaborative Filtering, Content-based filtering, Hybrid system, Clustering, Neural network.**

I. Introduction

A product recommendation system is a software application that provides recommendations for content that is user-specific and users might like to acquire. This approach uses Machine learning algorithms and a vast variety of data about particular items and users.

In the mid-1990s, recommendation systems became a famous topic of research. Awareness in recommendation systems has grown significantly in recent years, and recommendation systems now play a significant role in commercial websites and well-known businesses such as Spotify, Facebook, LinkedIn, and IMdb because these systems help them to increase the number of items sold and sell more diversified items or even increase the user satisfaction. This demand for this type of systems has given the green light to researchers to develop powerful systems and several researches have been carried out in this field. A recommendation method must demonstrate that a product needs to be recommended in order to find relevant products for the user. There are many kinds of recommendation algorithm approaches for this, the most prominent of which are collaborative filtering, content-based filtering.

We intend to build a recommendation system that can predict products that are available in the database and that are like the chosen product. We will leverage the product description, rating, and category which can be useful in recommending products. We will also keep testing various algorithms and find out the most suitable algorithm to work with. We will also use various combinations of vectors to find out the best, which can give us a good result.

II. Related Work

Akhilesh Kumar Sharma, Bhavna Bajpai, Rachit Adhvaryu, Suthar Dhruvi Pankaj Kumar, Prajapati Parth Kumar Gordhan Bhai, Atulkumar [1] proposed a recommendation system by using NLP technique.The main focus of the authors was to create a content based recommendation system and improve the efficiency by utilising NLP.Our approach share a similarity in building a recommendation system by leveraging the NLP technique but by adding collaborative filtering to the approach to get better results.In addition to collaborative filtering we also use user review and rating to improve the results.

Amy Trappey,Charles V.Trappey,Alex Hsieh [2] proposed a An intelligent patent recommender adopting machine learning approach for natural language processing in which they use tf-idf,doc2vec algorithm to train a neural network model for document vectorization using the context of the domain patents where as We move beyond the common use of tf-idf of finding similar products to predict ratings and apply doc2vec, to extract information contained in the context of the product's descriptions.

Yassine Afoudi,Mohamed Lazaar,MohammedAl Achhab [3] proposed Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network in which the matrix factorization aspect of recommendation structures maintains the same dimensionality, the key function of Singular Value Decomposition is to decompose a matrix into three other matrices where our matrix factorization is done on the user-item ratings matrix as 2 matrices whose product is the original matrix.

III. Proposed work

This section proposes a new type of recommendation model with key components: collaborative filtering, content-based filtering, SOM Neural Network, Bag Of Words,TF-IDF and Cosine Similarity.

A. Content Based Recommendation

Content-based filtering technique is based on the description of the item and a profile of the user's preferences. These methods are best suited to situations where there is known data on an item (name, location, description, etc.), but not on the user. The content-based recommendation technique treats recommendation as a user-specific classification problem and learns a classifier for the user's likes and dislikes based on the attributes. In the initial stage, the system first analyses similarities between all pairs of products then use the most similar products to a user's selected product to generate a list of recommendations may be explicitly or implicitly. As the user provides new sources of information or performs actions based on the recommendations, the system becomes more precise as shown in Fig.2.

We want to add a weight to each word in the item description, review, and rating in our content-based model to assign the value of that word in the data set. We use the TF–IDF algorithm to weight a keyword in the product and attribute value to that keyword depending on the number of times it appears in the document; a higher TF–IDF score (weight) indicates that the phrase is rarer and more significant, and vice versa. As a result, each item will be interpreted by a TF–IDF vector dependent on title features at the end of the process.

B. Collaborative Based Recommendation

Collaborative filtering is a method for providing suggestions based on correlations between users and products. In other words, it is the method of filtering items based on the opinions of other users and choosing a group of users with similar tastes to a specific user. The method analyzes their favorite products and integrates them into a categorized list of suggestions. The CF system tries to find similar items based on user feedback. User feedback can be either explicit or implicit, explicit as a numerical rating to specify how much users liked a particular item, for example, 1 means dislike or 5 if the user likes the item very much, or implicit like browsing history on the website or reading time a type of product Collaborative Filtering has two types of algorithms, Memory-Based and Model-Based, The first type saves products and user data in memory, then uses mathematical methods to make estimates based on the data. Different machine learning algorithms, such as the Bayesian network, rule-based, and clustering methods, are used to construct the model process. In our approach we will use the second category, Model-Based because it can respond to user's requests instantly as shown in Fig.3.

C. Bag of words

A bag-of-words model, or BOW for short, is a way of extracting features from the text for use in modeling, such as with machine learning algorithms.This is a flexible approach and can be used in many ways for extracting features from documents.A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:

- A vocabulary of known words.
- A measure of the presence of known words.

We will use BOW technique to find the similar type of words in the description and reviews that can relate to each other.

D. TF-IDF (Term frequency - inverse document frequency)

It is a strategy for quantifying a word in a document by assigning a weight to each word that represents the word's importance in the corpus. It's a common technique in text mining and Information Retrieval.

Term Frequency (TF) is the total number of times a word appears in the current document. This means the occurrence of the word in a document when the frequency of a word is higher, it gives higher weight, this why we should normalize the result using a division by the length of the document as shown in the formula below where N(Tx, Dy) is the total number of times term Tx appears in a document Dy and N(Py) is the total number of terms in the document.

$$TF = N(Tx, Dy)/N(Py)$$

Inverse Document Frequency (IDF) of a word is the cumulative number of records containing the word x; it indicates the scarcity of the word as the IDF decreases when the word occurs in the text. It aids in determining the significance of a word across the whole corpus.For example, if we do a search on google on "the hybrid system", automatically the TF of the word "the" will be higher than "hybrid" and "system", here the role of the IDF came to reduce the weight of the word "The" to give more weight to the important words. IDF can be calculated by the formula below where N(D) is the total number of documents and N(D,Tx) is the number of documents containing term x.

$$IDF = log(N(D)/N(D, Tx))$$

Ultimately, TF–IDF is a normalization measure used to evaluate the importance of a word for a document in a corpus of documents. The formula for calculating the TF–IDF is.

$$TF - IDF = crossprod(TF, IDF)$$

E. Cosine Similarity

Cosine similarity is a metric used to determine how similar the reviews are irrespective of their size.The cosine similarity is advantageous because even if the two similar reviews are far apart by the Euclidean distance because of the size (like, the word 'great-deal' appeared 10 times in one review and 5 times in another) they could still have a smaller angle between them. Smaller the angle, higher the similarity.

F. Neural network model

Neural networks are a group of algorithms that detect patterns and are closely modeled after the human brain. We should think of neural networks as a layer of clustering and classification on top of the data we store and handle, as they assist us in clustering and classifying data. The self-organized

map is a neural network, and it is the technique we will be using in this recommendation system.

As explained in the Collaborative Filtering module, CF is the process of filtering items based on users' historical opinions and preferences on a set of items. Here, we will use the Self-Organizing Map (SOM) method to improve the traditional collaborative filtering system in order to build our system. A self-organizing map (SOM) is a form of artificial neural network (ANN) that is trained using unsupervised learning to generate a low-dimensional, usually two-dimensional. We will use the self-organizing map in our model to solve the issue of unsupervised clustering of the dataset. Clustering technology simplifies the structure of the dataset and divides it into different clusters, so the users can easily observe and analyze the data.

The map is made up of a standard grid of "neurons", which are processing units. Each unit is linked to a function vector that represents a high-dimensional observation model. Using a limited number of models, the map tries to reflect all available findings as accurately as possible. Neighbors are map groups that are close by on the grid. The model vectors are structured such that the local map units represent a common type of data and the distant map units represent various types of data after generating a map for a specific dataset.

Our system classifies products using SOM based on the category of the product. After initialization of our map dimensions and randomly initialization SOM weights, we train the model. Once the map has been trained, it gives us weights as results, then we use those weights as input data of the K-means clustering model. Here, to find the appropriate number of clusters, we will use the Elbow method the best-known method for choosing the number of clusters, this method says that by plotting the different number of clusters according to the variance, the point of the elbow is that of the number of clusters whose variance no longer decreases. To make a recommendation, we classify the dataset of products in a specific number of clusters, we choose for an active user all ratings of the positively rated products considering only the product with rating value greater than or equal to 2.5, after that, we would like to give each product its cluster number, to do this, we calculate the distance between the items attributes and the centroids of k-means using the Euclidean distance approach as mentioned. Then we choose the cluster with the smallest distance result. We measure the number of each product's cluster in the list after listing the positively scored product with their cluster class and then use the highest result class as our user's favorite class. After getting our user's favorite class, we get their age (demographic attribute) if the age is greater than or equal to a specific age, we get all users with the same age interval and vice versa, then we build a matrix of all selected users and all selected products and we use the collaborative filtering approach to predict items ratings and sort them from best to worst to give recommendations.

G. Implementation

To take advantage of the complementary advantages of two or more recommendation methods, proposed recommendation model merge them in various ways. In our architecture, we will use two methods in one, Feature augmentation and Weighted methods as shown in Fig.1. After obtaining the results of the Collaborative Filtering model and the Content-Based model, we combine them with a linear combination of their scores, as determined in weighted method, then we classify the list and we take the first 200 recommended items, finally, we use the selected results and we classify them again according to the Self-organizing map CF scores (Feature augmentation) from best to worst and show the first N items as recommendation.

A simple keyword search recommender module is built to filter products by keywords based on a user's interest. In this case, the module will filter keywords based on category and product title. The user may enter a word or series of words in as the product title and even refine the search by entering a category, or vice versa. The module will filter by the keywords entered, then rank the results by the rating method chosen by the user and return the desired amount of results.

An adjusted rating system was introduced to adjust the rating of each product to account for extreme ratings and balance out ratings for products that had several thousand reviews to products that only had a hundred or so. Products with less reviews were more likely to be affected by a series of ratings than products with more reviews, both positively and negatively. We want a balance.

$$score_i = \frac{\sum_u r_{ui} + k * \mu}{n_i + k}$$

The purpose of this recommendation system is to show the inner workings behind a basic recommendation system.

- Takes in a dataset, user and product.
- Filters to all other users that have purchased the same product as the original user.
- Filters to all users that gave the same rating as the original user.
- Filters to all products those users purchased.
- Filters to 5 star products those users purchased.
- Takes the top 10 (or most common) products those users gave 5 stars.
- Returns the products.

IV. Dataset and Analysis

Our dataset has the data of Amazon apparel. We contain dataset of around 180,000 clothes. The dataset has 9 columns giving information about each product. Some of these attributes are not useful for us in building a recommender system. We will do data analysis and exploration on each attribute to find out which can give most information. During Initial analysis, we found out:

- The "unixReviewTime" - time of the review (unix time) column is not useful.

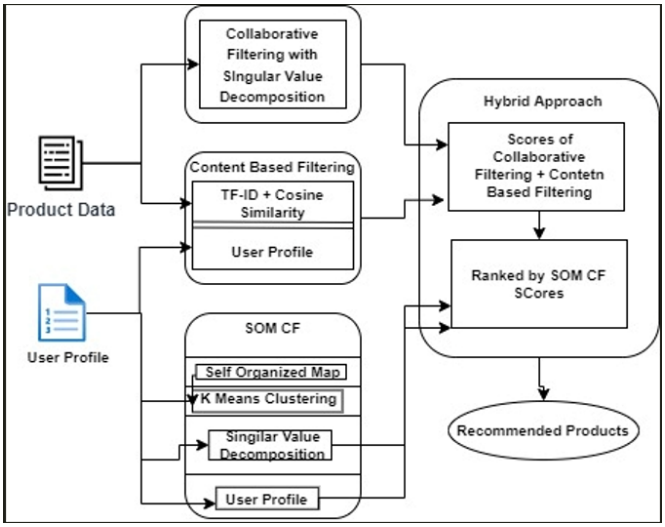

Fig. 1. Proposed Architecture

- The datatype of "helpful" column is list.Which could be complicated to analyze.

Further analysis will be proceeded in the future steps of the project.In the dataset all 7 features have been used for the recommendation.The other features were not used because they were either not useful in recommendation or they don't give information much about product.

- reviewerID - ID of the reviewer.
- asin - ID of the product, e.g. 0000013714
- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review, e.g. 2/3
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

To Sum Up Our Exploratory Data Analysis:

- Positive ratings occur over 80 Percent of the time, while negative ratings occur roughly 15 Percent of the time and neutral ratings occur 5Percent of the time. Roughly 75 Percent of all customers have only reviewed products one time, while roughly 24 Percent have reviewed up to 10 products. 1 customer has reviewed almost 700 products. Well over 50 Percent of products average a 4.5 or higher rating, while over 90 Percent have a 3.5 or higher rating as shown in Fig.4.
- Candy Crush is the most popular product reviewed on Amazon which is a mobile app. Every product in the top 10 has over 25,000 reviews with Candy Crush and The Secret Mystery having over 40,000 reviews. 6 of the top 10 categories dominate the marketplace with well over 90 Percent of total ratings and reviews.
- The range in average ratings from highest to lowest is 0.84, from 3.84 to 4.68. The categories rated low seem to be categories where customers don't know what they may be receiving, variability of quality among the same type of products, while the categories rated highly seem to have less variability in what one can receive. The biggest factor that seems to determine a product category's average rating is how many 5 stars, 4 stars and 1 stars that category receives.
- Over time, the average annual product review has stayed between 4.1 and 4.4. The Amazon marketplace started gaining popularity in 2010 and exploded exponentially from 2011 onwards.
- December has the most reviews, followed by January. This is likely because of the holiday season and the purchasing and exchange of gifts. January is likely high because money is often a popular gift during the holidays and customers make purchases after the holidays are over.

V. Difference from Existing Methodologies

In our system, a new recommendation system is proposed, which is based on four steps, namely Collaborative, Content-Based, Self-Organizing Map Collaborative Filtering, and Hybrid Model. Implicit user ratings are calculated using the singular value decomposition approach in the collaborative filtering part, we use also the item's textual features to build a content-based model. Implicit rating data from users and product features are used in Self-Organizing Map with collaborative filtering to create the Self-Organizing Map Collaborative Filtering model, then the results of these three steps are combined in the next step of the proposed system to recommend similar top-N products to the active user selected.

The proposed Recommendation system is essentially the combination of diverse algorithms. Since our proposed recommendation engine uses collaborative filtering and product-based filtering, a broader range of products can be recommended with accurate precision. This system also overcome the common issues in recommendation systems such as the cold start problem.

VI. Performance Comparison with Existing Methodologies and Evaluation Metrics

A recommendation system typically produces an ordered list of recommendations, from best to worst, for each user in the dataset. In fact, in many cases, the user does not much care about the order of recommended items, a few good suggestions are enough. But for us, we need to evaluate our system, because an evaluation technique provides insight into the relevance of the list of recommended items. Therefore, there are many evaluation approaches involved in the area of the recommendation system divided into online and offline methods. For those online, we need a real-time user to give feedback and opinions on the recommended items, in our work we will use the offline technique from a dataset of real user interactions on products.

We choose RMSE and Precision–Recall at k technique,from among several assessment approaches.The Root Mean Squared Error is a well-known technique for evaluating the accuracy of a recommender system based on ratings of data in order to look for low prediction errors. The principle of this technique is

based on the use of predicted and real ratings, then calculating the average of the errors of the test set using RMSE, where P is the predicted rating and R is the true rating, and produce a final score; we then compare our model's result to that of others; if your result is less, it indicates that your model is more accurate.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(\frac{P-R}{\#rating}\Big)^2}$$

The other offline method is the Precision–Recall at k technique, where k is a number that can be defined by the user for Top-N recommendation. Recall is the capability of the model to find all relevant items and recommend them to the user while the Precision is the ability of the model to provide the relevant items with the fewest recommendations.

$$Recall@k = Ri/Nr$$

$$Presission@k = Ri/Tr$$

Where Ri represents the number of relevant recommended items at k,Tr represents the total number of relevant items, and Nr represents the total number of recommended items at k.

- The tuned algorithm performance on the full dataset.

$$RMSE : 1.07621.0761692357254327$$
$$CPUtimes : user16min47s, sys : 13.1s, total : 17min$$
$$Walltime : 17min37s$$

- The re-tuned algorithm performs on the full dataset

$$RMSE : 1.05931.059264022518196$$
$$CPUtimes : user4min25s, sys : 4.85s, total : 4min30s$$
$$Walltime : 4min43s$$

In our Presission and Recall Evaluation, the threshold is set at 3.5 and we will first evaluate our sampled predictions from the 10000 ratings (See Recommending Products to Customers or Testing Product Recommendations for Customers). We will then evaluate the full dataset of predictions.

- Sampled 10000 Ratings

KFold Test 1: Precision: 0.8557114228456913 Recall: 0.9679358717434869
KFold Test 2: Precision: 0.845 Recall: 0.9565
KFold Test 3: Precision: 0.8223223223223223 Recall: 0.965965965965966
KFold Test 4: Precision: 0.8449224612306153 Recall: 0.9599799899949975
KFold Test 5: Precision: 0.8415915915915916 Recall: 0.9574574574574575

- Sampled Full Dataset

KFold Test 1: Precision: 0.9005001566088227 Recall: 0.8705995812909934
KFold Test 2: Precision: 0.9008590029372812 Recall: 0.8690938265449699
KFold Test 3: Precision: 0.9012094778520168 Recall: 0.8694729673910461
KFold Test 4: Precision: 0.9009671511393442 Recall: 0.8694656881732762
KFold Test 5: Precision: 0.9021510740757444 Recall: 0.8655278934356118

We can see that the sampled predictions have fairly high precision scores and recall scores only a few points lower (3 points). The full dataset of predictions have extremely high recall scores with precision scores much lower (11 points). The sampled predictions are closer to consistency in recommending unseen items than the full predictions, but the full predictions are much more consistent in recommending seen items.

TABLE I
ACCURACY COMPARISON

Dataset Used	Normal Method	Proposed Method
Amazon Product	1.076	1.0593

As we observe from the above table that the proposed method performs slightly better than the general method.

A. Sample Outputs

In this recommendation system, we not only recommend the products similar to a given product, but we display what product category the recommended product belongs to. Some are obvious, but for recommending products similar to 'Chromecast,' it's best to know what product categories the recommended products belong to. It is also worth noting that we are able to map out the full dataset to predict similar items, rather than just a sample.

Test 1 - Minecraft
Here are the result of the recommended items when a user search for Minecraft.
The 10 most similar products to Minecraft are:
0: Mobile Apps - Bloons TD 5
1: Mobile Apps - Angry Birds Epic RPG
2: Mobile Apps - Twitter
3: Mobile Apps - Farming Simulator 14
4: Mobile Apps - Goat Simulator
5: Mobile Apps - Head Soccer
6: Mobile Apps - Asphalt 8: Airborne
7: Mobile Apps - Hungry Shark Evolution
8: Mobile Apps - The Dark Knight Rises (Kindle Tablet Edition)

9: Mobile Apps - Dragon City

Test 2 - Google Chromecast HDMI Streaming Media Player.
Here are the result of the recommended items when a user search for Chromecast.
The 10 most similar products to Google Chromecast HDMI Streaming Media Player are:
0: Mobile Apps - Twitter
1: Mobile Apps - TubeMate YouTube Downloader
2: Video DVD - The Dark Knight Trilogy (Batman Begins / The Dark Knight / The Dark Knight Rises) [Blu-ray]
3: Video DVD - The Hunger Games: Catching Fire [Blu-ray + DVD + Digital HD]
4: Electronics - FiiO E6 Portable Audio Headphone Amplifier
5: Mobile Apps - Adobe Acrobat Reader- PDF Reader and more
6: PC - SanDisk Ultra 16GB UHS-I/Class 10 Micro SDHC Memory Card With Adapter- SDSDQUAN-016G-G4A [Old Version]
7: Digital eBook Purchase - 11/22/63: A Novel
8: PC - Thunderbolt to Gigabit Ethernet Adapter
9: Mobile Apps - OfficeSuite Professional

VII. Future Works

We see a variety of possible future paths for our work, in addition to the framework enhancements outlined in the results and discussion section. We could continue to refine and investigate other approaches in the field of recommendation, and machine learning and deep learning algorithms will be used to change and improve our model.
Limited time and computational power prevented from making the best predictions in selecting and tuning our model as well as making product predictions for specific customers. In the future, we can train the complete dataset instead of a very small percentage.

VIII. Conclusion

In this paper, we propose a new model of a product recommendation framework based on three models: collaborative filtering, content-based, and CF with a self-organizing map model that takes the age demographic attribute into account. The main advantages of our system is to combine all the scores of all models and benefit from the advantages of each of them. Even our system takes a lot of recommendation time speed compared to the other models but the precision and the performance improvement are very high. The experiment shows that by using Self organizing map with collaborative filtering, the RMSE was reduced in the majority of clusters against of using K-means clustering with CF, then we combine this powerful system with the other state of art approaches to create a new system.

A. Figures and Tables

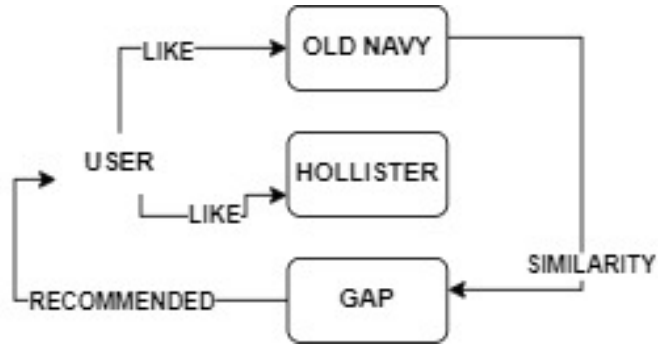

Fig. 2. Content Based Filtering

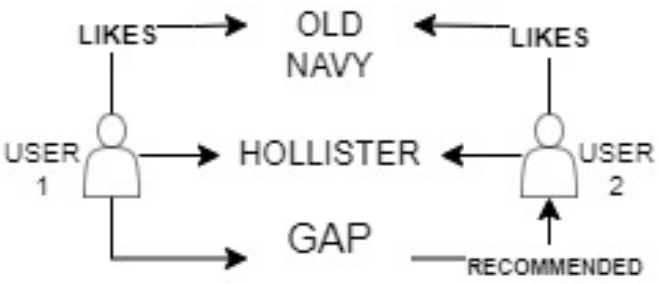

Fig. 3. Collaborative Filtering

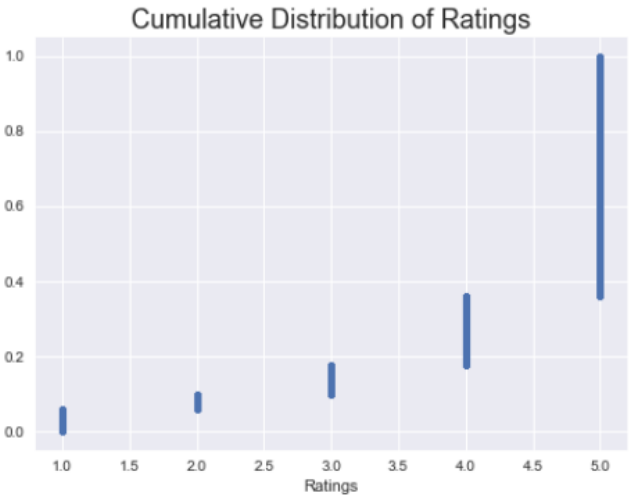

Fig. 4. Cumulative Rating Distribution

REFERENCES

[1] Akhilesh Kumar Sharma, Bhavna Bajpai, Rachit Adhvaryu, Suthar Dhruvi Pankaj Kumar, Prajapati Parth Kumar Gordhan Bhai, Atulkumar, "An Efficient Approach of Product Recommendation System using NLP Technique" July 2021 Materials Today: Proceedings,August 2021.

[2] Amy Trappey,Charles V.Trappey,Alex Hsieh, "An intelligent patent recommender adopting machine learning" Technological Forecasting and Social Change,Volume 164, March 2021, 120511.

[3] Yassine Afoudi,Mohamed Lazaar,MohammedAl Achhab "Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network".Simulation Modelling Practice and Theory,Volume 113, December 2021, 102375.

[4] Sunny Sharma, Vijay Rana, Vivek Kumar "Deep learning based semantic personalized recommendation system "July 2021.

[5] Michele Gorgoglione, Umberto Panniello, Alexander Tuzhilin "Recommendation strategies in personalization applications" Information Management Volume 56, Issue 6, September 2019, 103143.

[6] Amit Kumar KushwahaEmail,Arpan Kumar Kar,"Language Model-Driven Chatbot for Business to Address Marketing and Selection of Products".Conference paper:December 2020.

[7] M.-C. Chiu,J.-H.Huang,S.Gupta,G.Akman,"Computers in Industry" Volume 128, June 2021, 103421.

[8] M. Aamir, M. Bhusry Recommendation system: state of the art approach Int. J.Computer Appl., 120 (12) (2015), pp. 25-32

[9] R.V. Karthik, S. Ganapathy "A fuzzy recommendation system for predicting the customers interests using sentiment analysis and ontology in e-commerce".Appl. Soft Computing, 108 (2021), p. 107396, 10.1016/j.asoc.2021.107396.

[10] S. Sharda, G.S. Josan "Machine learning based recommendation system: a review Int. J. Next-Gener. Comput"., 12 (2021), p. 2

[11] K R., Kumar P., Bhasker B. DNNRec: A novel deep learning based hybrid recommender system Expert Syst. Appl., 144 (2020), Article 113054.

[12] Hernández del Olmo F., Gaudioso E. Evaluation of recommender systems: A new approach Expert Syst. Appl., 35 (2008), pp. 790-804