Modified Content-Based Filtering Method Using K-Nearest Neighbors and Percentile Concept

Nathan John J. Cordero[a*], Jermaine C. Canlas[b], Khatalyn E. Mata[c], Richard C. Regala[d], Mark Christopher R. Blanco[e], Dan Michael A. Cortez[f], Antolin J. Alipio[g]

[a] njjcordero2018@plm.edu.ph
[b] jccanlas2018@plm.edu.ph
[c] kemata@plm.edu.ph
[d] rcregala@plm.edu.ph
[e] mcrblanco@plm.edu.ph
[f] dmacortez@plm.edu.ph
[g] ajalipio@plm.edu.ph
[a]Pamantasan ng Lungsod ng Maynila, Intramuros, Manila, 1002, Philippines
[b]Pamantasan ng Lungsod ng Maynila, Intramuros, Manila, 1002, Philippines
[c]Pamantasan ng Lungsod ng Maynila, Intramuros, Manila, 1002, Philippines
[d]Pamantasan ng Lungsod ng Maynila, Intramuros, Manila, 1002, Philippines
[e]Pamantasan ng Lungsod ng Maynila, Intramuros, Manila, 1002, Philippines
[f]Pamantasan ng Lungsod ng Maynila, Intramuros, Manila, 1002, Philippines
[g]Pamantasan ng Lungsod ng Maynila, Intramuros, Manila, 1002, Philippines

Abstract

In the age of information, vast amount of data is within the grasp of everyone. The availability and amount of information is absurd that it could lead to information overload. Recommender systems exist so that it could recommend information that are relevant and appropriate based on user's preference. Content-based filtering (CBF) is a recommender system approach that focuses solely on user preference and content of an item. CBF works by recommending items that satisfies user's interest based on user's previously liked items. CBF suffers the problem of overspecialization or also called the serendipity problem. Overspecialization occurs when the items that are being recommended is very similar to the previously liked item of the user, thus, not being able to recommend unexpected recommendations. The researchers used a pure content-based approach in eliminating the overspecialization problem. The researchers' first method is to use K-Nearest Neighbors (KNN) algorithm to find the nearest neighbors of the top recommended items. The researchers' premise is to recommend similar items of similar items. The researchers' second method is to use the percentile concept in the cosine similarity matrix of all the items. This method lets the researchers prevent overspecialization by recommending items that are in the lower percentiles since overspecialization occurs in the higher percentiles. The result of this study shows that the first and second are effective in preventing overspecialization because these methods recommended unexpected yet relevant items.

Keywords: Content-based filtering; Cosine similarity matrix; K-Nearest Neighbors; Overspecialization; Percentile Method

1. Introduction

With the availability of vast amount of information in the internet, internet "information overload" is now a thing. Almost all general topics, fields, and information known to man can be easily found in the internet. A person using the internet has his/her own preferences which information or topics he/she wants to browse and does not probably want to view all the available selection existing in the internet. This is where recommendation systems come to take place. Recommendation systems are systems that help users personalize their user experience of a system or an application according to their preferences (Thorat et al., 2015). There are several approaches on how a recommender system be fit by their user's preferences and these major approaches collaborative filtering, knowledge-based recommendation, and the content-based filtering (Felfernig et al., 2014). Content-based filtering is a recommendation system approach that focuses on user's preference profile, user's interaction with a system or application, and item description (Sharma and Gera, 2013). Content-based filtering could also let users build their profile explicitly by asking users their field of interest upfront (Badriyah et al., 2018).

In creating Content Based Filtering, we will first extract the attributes of items that will be used for recommendation. Then, compare the extracted attributes with the user's preferences. User's preferences refer to items liked or consumed by the user. Lastly, items that fits the user's interest the most will be recommended and displayed (Thorat et al., 2015). An advantage of content-based filtering is it is personalized because users are the ones building their preference profile yet there is also the disadvantage of overspecialization of recommended items (Thorat et al., 2015). According to Barragáns-Martnez et al. (2010), the fundamental problem of content-based filtering algorithms is their tendency to over-specialize item selection by proposing only things that are quite similar to previous products liked by the user. Content-based filtering (CBF) is incapable of creating unexpected recommendation results, also known as serendipity problem (Badriyah et al., 2018).

The objective of the study is to remove the overspecialization problem of the Content-based filtering algorithm and to provide users a wider range of related items. In order to do so, we will utilize cosine similarity matrix, K-Nearest Neighbors Algorithm(KNN) on initial similar items recommended, and Percentile Concept as the range of how similar recommended items will be.

(10 pt) Here introduce the paper, and put a nomenclature if necessary, in a box with the same font size as the rest of the paper. The paragraphs continue from here and are only separated by headings, subheadings, images and formulae. The section headings are arranged by numbers, bold and 10 pt. Here follows further instructions for authors.

2. Related Work

According to Thorat et al. (2015), The Content-based filtering method suffers from overspecialization, which is the problem when the recommendation system suggests the same type of items, using only the active user's preference as basis for the recommendation. Content-based filtering recommendation systems works by calculating set of items that are closely related to the items that the user is already familiar to or is based on the user's item preferences (Felfernig et al., 2014). According to Sharma and Gera (2013), Content-based recommendation methods utilizes the similarities of an item to the user's item preferences. Overspecialization occurs in recommendation system when the items being recommended are already known to the user. In content-based filtering method, only the very similar items to the previous items that the target user has previously consumed are recommended to the target user and this leads to the tendency of over specialization according to Barragáns-Martínez et al. (2010). Sollenborn and Funk (2002) said that the content-based filtering

method runs the tendency of recommending items that are pretty much similar, almost alike, to the previously consumed items of the target user.

By combining content-based filtering with collaborative filtering method, it is possible to eradicate the problem with using only content-based filtering or using only collaborative filtering (Polcicova et al., 2000). Reddy et al. (2018), indicated that both content-based and collaborative-based filtering have their own disadvantages and drawbacks. To overcome those, researchers suggested a new approach, a hybrid approach which basically combines the advantageous features of both methods. Kamran et al. (2020) discussed a movie recommender system by integrating content-based and collaborative filtering algorithms. Lenhart, P., & Herzog, D. (2016) developed Personalized Sports News Recommendations (Hybrid: CBF and CF) that utilizes creation of user Profile based on user reading history (article keywords), and user's specifying her or his favorite sport and team. Vector Similarity was used for similar articles. Adjusted Cosine Similarity was used for ratings of other users (CF).

Ali et al. (2018) suggested a hybrid movie recommendation algorithm that uses Cosine Correlation to know the degree of relevance of similar movies to one another. Badriyah et al. (2017) study creates a hybrid recommendation system for e-commerce that use both Content-based and Collaborative Filtering approaches to determine the similarity between product descriptions and user profiles. Cosine Distance was used to determine the similarity of related items and profiles. Badriyah et al. (2018) conducted study on association rule mining on a property site utilizing a content-based filtering approach and an apriori algorithm. Barragáns-Martínez et al. (2010)'s study was about TV Program recommendation that uses content-based filtering and collaborative filtering. Profile Creation was utilized in this study that takes into consideration what user likes, what channels the user has access to, demographic and lifestyle information. Cosine Correlation was used to determine the correlation of the product vectors and the user model. Consideration of other user's rating history is where the collaborative filtering method takes place. De Campos et al (2010)'s study uses a Hybrid approach (uses content-based filtering and collaborative filtering) and Bayesian network model for Movie Recommendation. Kamran et al. (2020) developed a movie recommender system by integrating content-based and collaborative filtering algorithms. User Profile creation was utilized specifically User's rating for the CBF part and Pearson Correlation was used for user-user correlation as the Collaborative Filtering part.

The study by Mathew et al. (2016) describes a Book Recommendation System (BRS) that generates efficient and effective suggestions by combining features from content-based filtering (CBF), collaborative filtering (CF), and association rule mining. They suggested a hybrid algorithm for this, which combines two or more algorithms to help the recommendation system suggest the book depending on the buyer's interests. Creation of User Profile was utilized by looking into the book purchase history of the user. Eclat Algorithm was also used to mine the related frequent item sets. Ratings of Other users about a book were also taken into consideration. In Pandya et al. (2016)'s study, they clustered the rating matrix by user similarity. The clustered data is then converted to Boolean data and Efficient rules for applying the Eclat Algorithm on Boolean data generation occurs. Finally, depending on the rules generated recommendation occurs. Their research demonstrates that strategy not only reduces the amount of sparsity but also increases the precision with which a system operates.

In Shahbazi and Byun (2019)'s study, CBF extracts user metrics such as clicks, purchases, visited pages, time spent on a website, and product categories. This information is used to create a client profile, which is then used to propose goods in this category. CF extracts information about users based on their behavior and priorities and forecasts their resemblance to other users. Zhao et al. (2015) suggested a hybrid filtering approach for customized mobile search that combines content-based and collaborative filtering. The former utilizes the mobile user's feature model, which is created from the user's query history, to filter the results, whilst the latter filters the results using the user's social network, which is formed from the user's communication history.

Serendipity impacts are defined by Felfernig et al. (2014) as an accidental meeting with something beneficial despite the user not conducting a relevant search. They are mostly performed by the application of CF methods.

Such effects are impossible to achieve with content-based filtering because it does not take other users' preferences (ratings) into account. According to Sharma and Gera (2013), in CBF, users may be limited to obtaining ideas that are similar to those already known or indicated in their profiles in some scenarios, which is referred to as an overspecialization issue. It makes it more difficult for the user to discover new things and other accessible choices. On the other hand, diversification of suggestions is a desired characteristic of all recommendation systems. Lenhart and Herzog (2016) mentioned that the choice of items suggested in CBF remains limited. This is a typical limitation of pure content-based RS and can be overcome through the use of a hybrid method. They upgraded the recommendation system in this study by including a collaborative filtering component that increased the variety of options. Thorat et al. (2015) stated how CBF suffers from overspecialization as a result of advocating for the same kind of themes. According to Lops et al. (2010), there is no intrinsic technique for content-based recommenders to help users discover anything surprising. The system recommends goods that have a high score in comparison to the user profile; hence, the user will be shown items that are comparable to those previously rated. This disadvantage is also known as the serendipity problem, alluding to content-based algorithms' proclivity for generating recommendations with a low novelty.

Pereira and Varma (2018) study utilizes C4.5 algorithm for users similarity FP-Growth algorithm for similar items in a Financial Planning Recommendation System (Hybrid). Pandya et al. (2016) identify important challenges faced by recommendation systems, including "data scarcity" and "cold start". they presented a novel technique based on the combination of clustering to overcome these obstacles. They combined the approach with the Eclat Algorithm to improve rule generation. Badriyah et al. (2018) conducted study on association rule mining on a property site utilizing a content-based filtering approach and an apriori algorithm. The Apriori Algorithm was used to build a search database based on advertising content by examining the weight of data in relation to the frequency of view on the user's search. The study by Mathew et al. (2016) describes a Book Recommendation System (BRS) that generates efficient and effective suggestions by combining features from content-based filtering (CBF), collaborative filtering (CF), and association rule mining. They suggested a hybrid algorithm for this, which combines two or more algorithms to help the recommendation system suggest the book depending on the buyer's interests. Tewari and Barman (2017) present a recommendation system (RS) that uses dynamic content-based filtering, collaborative filtering, association rules, and opinion mining to create item suggestions for users.

3. The Proposed Method

Content-based filtering works by recommending users similar items based on what the user previously liked or consumed. Figure 1 shows how content-based filtering basically works according to Thorat et al. (2015).

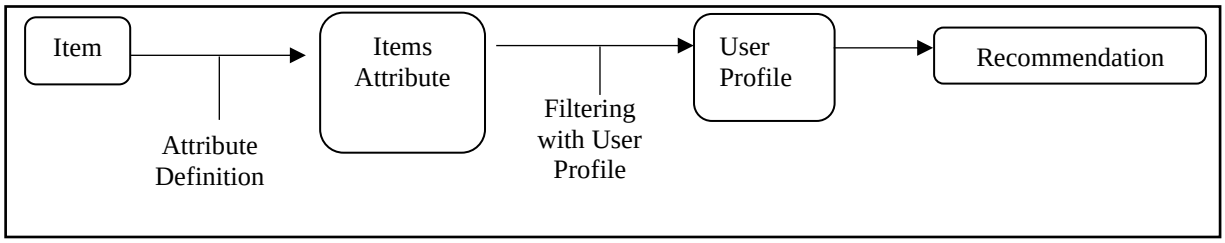

Fig. 1. Thorat et al. (2015)'s representation of content-based filtering. Note. This model was produced by Thorat et al. in 2015, describing how content-based filtering works. From "Survey on collaborative filtering, content-based filtering and hybrid recommendation system." By Thorat, P. B., Goudar, R. M., & Barve, S. 2015, International Journal of Computer Applications (0975 – 8887), 110(4), 31-36. January 2015. Adapted with permission.

In Figure 1, the first step in content-based filtering approach is to educe the attributes of the item. Attributes in this sense means the useful features that can describe an item. Next is creating a user profile. User profiles can be created by looking on what items the user liked. Finally, the algorithm will recommend similar items to what the user has liked. Overspecialization occurs at the "Recommendation" part in Figure 1 when the recommended items are very similar to user's liked item.

3.1. The Proposed Modification of the Content-Based Filtering Method

The researchers proposed a modified approach to content-based filtering that aims to eliminate the overspecialization problem. Figure 2 shows the overview of the proposed approach.

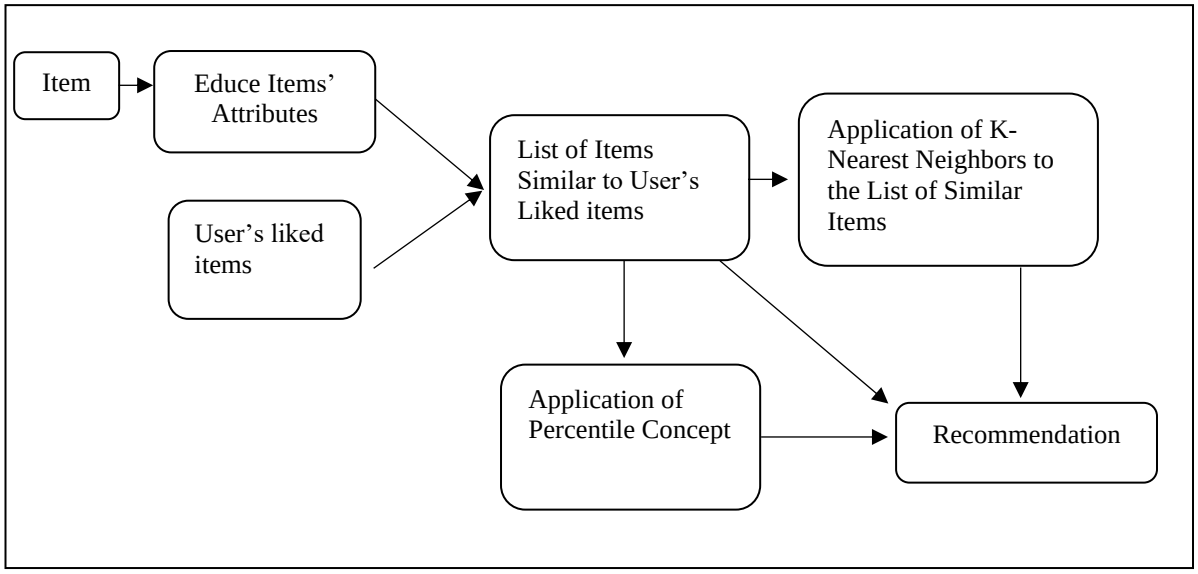

Fig. 2. The proponent's modified approach to content-based filtering

In Figure 2, the first step in the proposed method will still be the extraction of useful features of items that came from item attributes (description, title, etc.) using Term Frequency - Inverse Document Frequency(tf-idf). TF-IDF is used to extract keywords from a document. It also scores the weight or importance of each word in a document in comparison to all other documents. The TF*IDF algorithm weighs a keyword in a content and assign importance based on the number of times it appears in the content. The study of Badriyah et al. (2018) uses the Text Mining TF-IDF approach to extract tags automatically from a product's description. In general, the TF-IDF methodology is used to determine the quantity of words that are connected between texts. The formula for computing the TF-IDF is as follows:

$$TFIDF_{d,t} = FREQ_{d,t}\left(1 + \log\left(\frac{N}{DFREQ_t}\right)\right) \qquad (1)$$

where,
$FREQ_{d,t}$ = number of term t in the document d
N = Total number of document used
$DFREQ_t$ = number of documents where term t appears

The next step will be taking note of user's liked items then comparing it to other items using cosine similarity between the items' tf-idf score and the liked item's tf-idf score. Cosine similarity is the similarity measure between two vectors. The cosine computation of the angles between two vectors is used to determine their similarity. Basically, the cosine similarity score goes up when two vectors have less distance between them and similarity goes down as the distance between them grows. Thus, two vectors are said to be identical if they produce an angle of 0 ° (zero degrees) or if their cosine equals one (1).

$$sim(A, B) = \cos(\theta) = \frac{A \,.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \, B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (2)$$

where,
sim(A, B) is a similarity measure between vector A and B
$A_i \, B_i$ are components of vector A and B respectively

Basically, the tf-idf scores of items will be used to create the cosine similarity matrix. This cosine similarity matrix is a matrix that shows the cosine similarity score of an item to all other items. The matrix will then be used to produce the list of recommended items. The items will be listed in a descending order based on their scores in the matrix. This list would be called the "trunk list". The first method of the researchers to prevent overspecialization would be the application of K-Nearest Neighbors algorithm to items with the highest cosine scores in the "trunk list". According to Ali et al. (2019), the K-Nearest Neighbors algorithm is as follows:

1. Assign the number of nearest neighbors which is also called the K values.
2. Determine the distance between the sample to all other samples.
3. Arrange the distance and based on the K-th minimum distance, determine the nearest neighbors
4. Organize the categories of the nearest neighbors
5. Assign the prediction value of the new data object based on the category of the majority of its nearest neighbors

Fig. 3. K-Nearest Neighbors Algorithm according to Ali et al. (2019)

Applying K-Nearest Neighbors will produce items that are similar to the top recommended items. Basically, the premise of the researchers is recommending "similar items of similar items". In figure 4 below, the researchers shows a visualization of the application of K-Nearest Neighbors to the "trunk list" and how it will be used for recommendation.

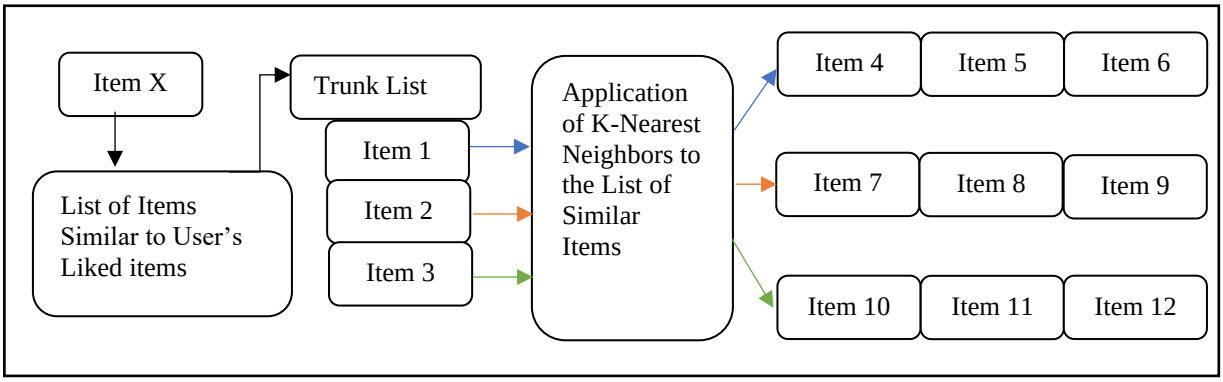

Fig. 4. Example of Recommending similar items of the "trunk list"

In figure 4, let us assume that there is item X that came from user's liked items. Using the cosine similarity matrix, it produced a "trunk list" that recommends the most similar items to item X. The researchers used a trunk list that contains items 1, 2, and 3 as an example. After applying K-Nearest Neighbors to items 1, 2, and 3, it shows their nearest neighbors. For item 1, items 4, 5, and 6 came out to be its nearest neighbors. As for item 2, its nearest neighbors are items 7, 8 and 9 and for the 3rd item, its nearest neighbors are items 10, 11, and 12. The researcher's premise in this approach is that some of the nearest neighbors of the "trunk list" may prevent the overspecialization problem in the occasion that 1.) it is not that similar to item X or 2.) it is novel to the user, thus, the probability of eliminating the serendipity/ overspecialization problem of content-based filtering is present.

The second method of the researchers in preventing the overspecialization problem will be the utilization of percentile concept. Percentile indicates the percentage of scores that a given value is higher. For an example, 75th percentile refers to items that has a greater cosine similarity score than 75% of all the items. Using percentile concept, we can set a range of values where the recommendation will be coming from.

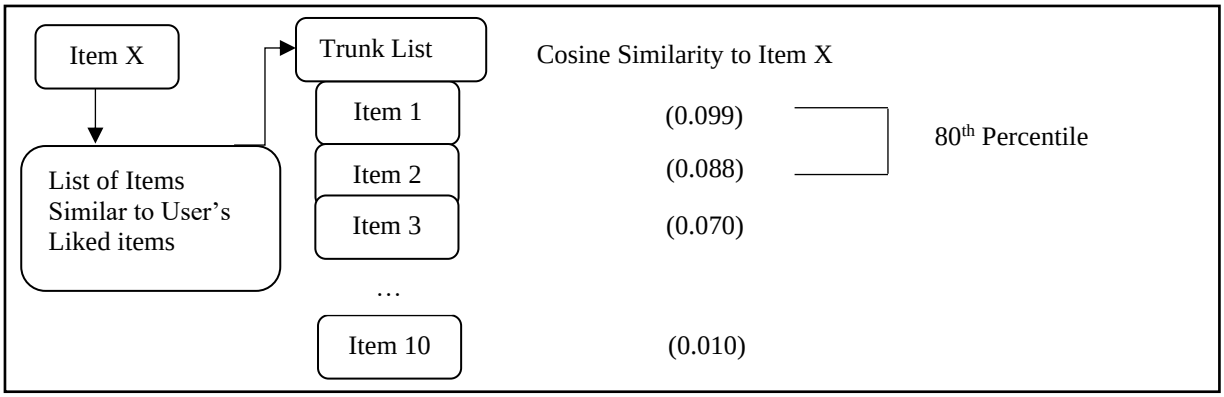

Fig. 5. Using Percentile Concept in Content-Based Filtering

In figure 5, let us assume that item X has a “trunk list” that contains items 1, 2, up to item 10,arranged in descending order based on their cosine similarity score to item X. The researcher’s premise in using percentile as the range of where recommendations will come from is that the higher the percentile, the more similar it is to item X, the higher the chance of overspecialization. Thus, recommending from lower percentiles could prevent overspecialization. Aside from preventing overspecialization, this proposed method gives transparency to the user on the degree of similarity of the recommended items to item X. In summary, the proposed modified algorithm is listed in Figure 6.

1. Educe the item attributes using tf-idf.
2. Create cosine similarity matrix of all the items.
3. Collect user’s liked item.
4. Arrange items in descending order based on their cosine similarity score in comparison to the user’s liked item to create the “trunk list”. Recommend the top 5 items in the “trunk list”
5. By applying K-Nearest Neighbors, get 3 nearest neighbors of each of the top 5 items in the “trunk list” and recommend it.
6. In the “trunk list”, determine the 60^{th} percentile in the cosine similarity matrix of item X and assign it as the start of the percentile range and 80^{th} percentile as the end. Recommend items that have a cosine similarity score between the 60^{th} and 80^{th} percentile.

Fig. 6. Framework of the Proposed Modified Algortihm

4. Results and Discussion

For the experimentation of the modified algorithm, the dataset that the researchers used is the top 250 rated movies according to IMDB.

4.1. Data Pre-processing

IMDB’s top 250 rated movies include information such as title, year, genre, ratings, etc. There is a total of 35 columns in the dataset. To extract the attributes of the 250 movies, we first need to pre-process the raw data that we have. The researchers only used the columns ‘Title’, ’Genre’, ’Director’, ’Actors’, ’Plot’ as features for the recommendation. Then, we created a “bag of words” from the features ’Genre’, ’Director’, ’Actors’, and ’Plot’. The “bag of words” is a set of words that comprises words that are used in the said features. Basically, the researchers combined the said columns to create a unified column which is the “bag of words”. In addition to that, this “bag of words” has been cleared of English stop words such as ‘a’, ‘the’, ‘is’, ‘are’, etc. so that only the relevant words remain to describe the movie. This leaves the researchers with a dataset that contains two columns namely, the title column and the “bag of words” column that the researchers referred as keywords column.

Table 1. Raw Data from the Top 250 IMDB Movies Dataset (First 3 Movies)

Title	Genre	Director	Actor	Plot
The Shawshank Redemption	Crime, Drama	Frank Darabont	Tim Robbins, Morgan Freeman, Bob Gunton, William Sadler	Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.
The Godfather	Crime, Drama	Francis Ford Coppola	Marlon Brando, Al Pacino, James Caan, Richard S. Castellano	The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son.
The Godfather: Part II	Crime, Drama	Francis Ford Coppola	Al Pacino, Robert Duvall, Diane Keaton, Robert De Niro	The early life and career of Vito Corleone in 1920s New York is portrayed while his son, Michael, expands and tightens his grip on the family crime syndicate.

Table 1 shows the raw data from the dataset. It contains the title, genre, name of director and actors, and the plot of the movie.

Table 2. Pre-processed data (First 3 Movies)

Title	Keywords
The Shawshank Redemption	crime drama frankdarabont timrobbins morganfreeman bobgunton two imprisoned men bond number years finding solace eventual redemption acts common decency
The Godfather	crime drama francisfordcoppola marlonbrando alpacino jamescaan aging patriarch organized crime dynasty transfers control clandestine empire reluctant son
The Godfather: Part II	crime drama francisfordcoppola alpacino robertduvall dianekeaton early life career vito corleone 1920s new york portrayed son michael expands tightens grip family crime syndicate

Table 2 shows the data that has undergone pre-processing.

4.2. Item Attribute Extraction

Term Frequency - Inverse Document Frequency(tf-idf) was used to calculate the weight of each word on each movie's keyword column since the keyword column is in string format. Each word on the keyword column was given a weight based on how important it is across all the document. This also converts the features of the movies into a numerical value which will be useful for creating the cosine similarity matrix.

4.3. Cosine Similarity Matrix

Using the tf-idf scores of the keywords of the movies, we can get the cosine similarity score of all the movies in comparison to all other movies.

Table 3. Cosine Similarity Matrix of the First 3 Movies in Comparison to One Another

	The Shawshank Redemption	The Godfather	The Godfather: Part II	…
The Shawshank Redemption	1	0.014	0.012	…
The Godfather	0.014	1	0.095	…
The Godfather: Part II	0.012	0.095	1	…
…	…	…	…	…

Table 3 depicts a 3x3 cosine similarity matrix. With all the movies considered, the researchers made a 250x250 matrix. These values refer to how similar the movie is to another movie. As an example in Table 3, the movie "The Godfather" in comparison to "The Shawshank Redemption" has a cosine value of 0.014. On the other hand, the movie "The Godfather" in comparison to "The Godfather: Part II" has a cosine value of 0.095. This means that "The Godfather" is closely similar to "The Godfather: Part II" compared to "The Shawshank Redemption".

4.4. *User's Liked Item and "Trunk List"*

For the recommendation to commence, the researchers used the movie "The Godfather". The researchers looked up the cosine similarity matrix of the movie and arranged the movies in descending order based on their cosine similarity score to get the highest similar movie down to the lowest, thus creating the "trunk list".

Table 4. The Godfather" trunk list (Top 5 movies)

Title	Cosine Similarity Score compared to "The Godfather"
The Godfather: Part II	0.095
Apocalypse Now	0.050
Scarface	0.032
On the Waterfront	0.031
The Night of the Hunter	0.029

4.5. K-*Nearest Neighbors of "Trunk List"*

We will apply the K-Nearest Neighbors algorithm to the top 5 movies in the trunk list of "The Godfather". We will look for 3 nearest neighbors of each of the 5 movies as a way to prevent overspecialization.

Table 5. K-Nearest Neighbors of "The Godfather: Part II"

Title	Cosine Similarity Score compared to "The Godfather"
Goodfellas	0.018
Annie Hall	0
Taxi Driver	0

Table 6. K-Nearest Neighbors of "Apocalypse Now"

Title	Cosine Similarity Score compared to "The Godfather"
The Deer Hunter	0.001
Full Metal Jacket	0.001
Platoon	0.001

Table 7. K-Nearest Neighbors of "Scarface"

Title	Cosine Similarity Score compared to "The Godfather"
Casino	0.026
No Country for Old Men	0.012
Prisoners	0.013

Table 8. K-Nearest Neighbors of "On the Waterfront"

Title	Cosine Similarity Score compared to "The Godfather"
12 Angry Men	0.015
The Exorcist	0
The Departed	0.015

Table 9. K-Nearest Neighbors of "The Night of the Hunter"

Title	Cosine Similarity Score compared to "The Godfather"
The Killing	0.016
Touch of Evil	0.015
Double Indemnity	0.014

The tables 5 up to table 9 shows the nearest neighbors of the first 5 movies in the trunk list of the movie "The Godfather". As you can see, it recommends movies such as "Annie Hall" with a cosine score of 0, "Taxi Driver" with also a cosine score of 0, and "Platoon" with a score of 0.001. This shows that those recommended movies are of low to no similarity to "The Godfather". This attempt of recommending similar movies of similar movies have solved the problem of serendipity since it recommended movies that are not too specialized for the movie "The Godfather".

4.6. Percentile Concept

The researchers used the 60th up to 80th percentile as the range where recommendations will be coming from. This attempt of the researchers to prevent overspecialization utilizes the cosine similarity scores in the matrix of the movie "The Godfather". Using the percentile concept, the researchers can determine beforehand how similar recommended items will be. 60th percentile has a cosine value of 0.0014 while the 80th percentile has a cosine value of 0.012. The recommendations in this method will be considered as "other movies you might like" since it offers a wider range of related movies.

Table 10: Movies From 60th Percentile up to 80th Percentile

Title	Cosine Similarity Score compared to "The Godfather"
Dogville	0.012
Out of the Past	0.011
…	…
The Great Escape	0.001
Judgment at Nuremberg	0.001

Table 10 shows the movies from 60th up to 80th percentile. These movies has a cosine score from 0.001 to 0.012 in comparison to "The Godfather" movie. If queried, The 10th movie in the trunk list is the movie "Fargo", with a cosine score of 0.019. This means that the movie "Dogville" which has a score of 0.012 is close to the top 10 recommended movie but not that close for the possibility of overspecialization to occur.

5. Conclusion

In this paper, the researchers proposed a modified content-based filtering method that uses K-nearest neighbors algorithm and percentile concept to prevent the overspecialization problem of content-based filtering method. Collaborative filtering method is often used as the answer to the overspecialization problem of CBF. This paper uses a pure content-based filtering approach to prevent the overspecialization of item recommendations. The first method of the researchers to prevent overspecialization by recommending similar items of similar items using the K-Nearest Neighbors Algorithm was seemed to be effective since it recommended movies with little to zero cosine scores in comparison to the user's liked item. In addition, this method recommends a wider range of items yet not totally random since the researchers utilized the most similar items to user's liked item. The second method of the researchers to prevent overspecialization, which is the percentile concept was also effective since it gives transparency and could easily be calibrated on how similar recommended items will be. We can prevent overspecialization by choosing a range of percentile that is less than 90th.

For future works, the researchers recommend to find the optimum range of percentile values to be used wherein it is not overspecialized yet not that irrelevant to the user's liked item. It is also recommended to test the study in an actual system with larger datasets and with user feedbacks to further enhance the study.

6. Acknowledgement

The researchers would like to thank Prof. Richard C. Regala for being the adviser of the study. The researchers would also like to thank the Pamantasan Ng Lungsod Ng Maynila College of Engineering and Technology Computer Science Department Faculties and staffs for the never-ending support and guidance. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

7. References

[1] Ali, N., Neagu, D., & Trundle, P. (2019). Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. SN Applied Sciences, 1(12), 1-15.

[2] Ali, S. M., Nayak, G. K., Lenka, R. K., & Barik, R. K. (2018). Movie recommendation system using genome tags and content-based filtering. In Advances in Data and Information Sciences (pp. 85-94). Springer, Singapore.

[3] Badriyah, T., Azvy, S., Yuwono, W., & Syarif, I. (2018, March). Recommendation system for property search using content based filtering method. In 2018 International conference on information and communications technology (ICOIACT) (pp. 25-29). IEEE.

[4] Badriyah, T., Wijayanto, E.T., Syarif, I., & Kristalina, P. (2017). A hybrid recommendation system for E-commerce based on product description and user profile. 2017 Seventh International Conference on Innovative Computing Technology (INTECH), 95-100.

[5] Barragáns-Martínez, A. B., Costa-Montenegro, E., Burguillo, J. C., Rey-López, M., Mikic-Fonte, F. A., & Peleteiro, A. (2010). A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. Information Sciences, 180(22), 4290–4311. doi:10.1016/j.ins.2010.07.024

[6] De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Combining content-based and collaborative

recommendations: A hybrid approach based on Bayesian networks. International Journal of Approximate Reasoning, 51(7), 785–799. doi:10.1016/j.ijar.2010.04.001

[7] Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S., & Stettinger, M. (2014). Basic approaches in recommendation systems. In Recommendation Systems in Software Engineering (pp. 15-37). Springer, Berlin, Heidelberg.

[8] Kamran, M., SHAH, S. S. A., BAIG, M. N. J., & KHAN, R. H. (2020). A MOVIE RECOMENDER SYSTEM BY COMBING BOTH CONTENT BASED AND COLLABORATIVE FILTERING ALGORITHMS.

[9] Lenhart, P., & Herzog, D. (2016, September). Combining Content-based and Collaborative Filtering for Personalized Sports News Recommendations. In CBRecSys@ RecSys (pp. 3-10).

[10] Lops, P., de Gemmis, M., & Semeraro, G. (2010). Content-based Recommender Systems: State of the Art and Trends. Recommender Systems Handbook, 73–105. doi:10.1007/978-0-387-85820-3_3

[11] Mathew, P., Kuriakose, B., & Hegde, V. (2016, March). Book Recommendation System through content based and collaborative filtering method. In 2016 International conference on data mining and advanced computing (SAPIENCE) (pp. 47-52). IEEE.

[12] Pandya, S., Shah, J., Joshi, N., Ghayvat, H., Mukhopadhyay, S. C., & Yap, M. H. (2016, November). A novel hybrid based recommendation system based on clustering and association mining. In 2016 10th international conference on sensing technology (ICST) (pp. 1-6). IEEE.

[13] Pereira, N., & Varma, S. L. (2019). Financial planning recommendation system using content-based collaborative and demographic filtering. In Smart Innovations in Communication and Computational Sciences (pp. 141-151). Springer, Singapore.

[14] Polcicova, G., & Návrat, P. (2000). Combining content-based and collaborative filtering.

[15] Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B. (2018). Content-Based Movie Recommendation System Using Genre Correlation. Smart Innovation, Systems and Technologies, 391–397. doi:10.1007/978-981-13-1927-3_42

[16] Shahbazi, Z., & Byun, Y. C. (2019). Product Recommendation Based on Content-based Filtering Using XGBoost Classifier. Int. J. Adv. Sci. Technol, 29, 6979-6988.

[17] Sharma, L., & Gera, A. (2013). A survey of recommendation system: Research challenges. International Journal of Engineering Trends and Technology (IJETT), 4(5), 1989-1992.

[18] Sollenborn, M., & Funk, P. (2002). Category-Based Filtering and User Stereotype Cases to Reduce the Latency Problem in Recommender Systems. Advances in Case-Based Reasoning, 395–405. doi:10.1007/3-540-46119-1_29

[19] Tewari, A. S., & Barman, A. G. (2017). Collaborative recommendation system using dynamic content based filtering, association rule mining and opinion mining. International Journal of Intelligent Engineering and Systems, 10(5), 57-66.

[20] Thorat, P. B., Goudar, R. M., & Barve, S. (2015). Survey on collaborative filtering, content-based filtering and hybrid recommendation system. International Journal of Computer Applications, 110(4), 31-36.

[21] Zhao, F., Yan, F., Jin, H., Yang, L. T., & Yu, C. (2015). Personalized mobile searching approach based on combining content-based filtering and collaborative filtering. IEEE Systems Journal, 11(1), 324-332.