Product Recommendation in E-commerce by Using Tf-idf algorithm

Md. Rafiqul Islam

Md. Sohan Sorker

A Thesis in the Partial Fulfillment of the Requirements

for the Award of Bachelor of Computer Science and Engineering (BCSE)

Department of Computer Science and Engineering
College of Engineering and Technology
IUBAT – International University of Business Agriculture and Technology

Fall 2020

Product Recommendation in E-commerce by Using Tf-idf algorithm

Md. Rafiqul Islam

Md. Sohan Sorker

A Thesis in the Partial Fulfillment of the Requirements for the Award of Bachelor of Computer Science and Engineering (BCSE)

The thesis has been examined and approved,

_____

Prof. Dr. Md. Abdul Haque
Chairman and Professor
Dept. of Computer Science and Engineering

_____

Prof. Dr. Utpal Kanti Das
Coordinator and Professor

_____

Suhala Lamia
Lecturer

Department of Computer Science and Engineering
College of Engineering and Technology
IUBAT – International University of Business Agriculture and Technology

Fall 2020

Letter of Transmittal

12 February 2021

The Chair

Thesis Defence Committee

Department of Computer Science and Engineering

IUBAT– International University of Business Agriculture and Technology

4 Embankment Drive Road, Sector 10, Uttara Model Town

Dhaka 1230, Bangladesh

Subject: Letter of Transmittal.

Dear Sir,

It is indeed a great pleasure for us to be able to hand-over the result of our hardship of the thesis report on Product Recommendation in E-commerce by Using Tf-idf algorithm. This report is the result of the knowledge which has been acquired from the thesis research which was officially started on Spring 2020 semester and ended on Fall 2020. We tried our level best for preparing this report based on our research work which was performed using different types of RS algorithm. The datasets were collected from Kaggle.com which was an amazon dataset. We tried to do every work that is within our limit for making this report come together.

We hope that you will find this work worth reading. Please feel free for any query or classification that you would like us to explain if necessary. Hope you will appreciate our hard work and excuse minor errors if there's any. Thanking you for your cooperation.

Yours sincerely,

_____	_____
Md. Rafiqul Islam	Md. Sohan Sorker
ID. 17103318	ID.17103346

Student's Declaration

We hereby declare that this thesis is based on results obtained from our own work. The materials of work found by other researchers and sources are properly acknowledged and mentioned as references. This thesis, neither in whole nor in part, has been previously submitted to any other University or Institute for the award of any degree or diploma. We carried our research under the supervision of Suhala Lamia**,** Lecturer, Department of Computer Science and Engineering, International University of Business Agriculture and Technology.

_____	_____
Md. Rafiqul Islam	Md. Sohan Sorker
ID. 17103318	ID.17103346

Supervisor's Certification

This is to certify that the research work titled "Product Recommendation in E-commerce by Using Tf-idf algorithm" is submitted by Md. Rafiqul Islam and Md. Sohan Sorker to the Department of Computer Science & Engineering, International University of Business Agriculture and Technology in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering under my supervision from Spring 2020 to Spring 2021. They have wide knowledge in different types of recommendation System and various algorithm of data mining as well. They have performed their duties in a diligent and satisfactory manner. I wish them every success in their future endeavours.

_____

Suhala Lamia
Lecturer
Department of Computer Science and Engineering
IUBAT–International University of Business Agriculture and Technology

Abstract

In e-commerce businesses ensuring a superior customer service to shoppers is very difficult. Due to large amount information customer doesn’t find their products very easily what they looking in website. Recommender systems are a helpful way for online users to deal with loading information and have turn out to be one of the most popular and powerful tools for e-commerce. But when new user or item added in e-commerce website it can’t recommended. This problem is called cold start. Cold-start is characterized by the incapability of recommending due to the lack of enough ratings. In fact, solutions for the cold-start problem have been proposed for different contexts, but the problem is still unsolved. So, we provide a model to solve this problem using tf-idf algorithm. We designed this model in three steps-

Firstly, we calculate weighted rating form user rating. Secondly, we find out the similarity between products details using cosine similarity of tf-idfVectorizer. And finally, we multiplied cosine similarity by weighted rating to get the newly added products recommendation. This RS will get efficient result for cold start problem and make customer shopping more reliable.

Acknowledgments

We would like to offer our sincere gratitude to our thesis supervisor Suhala Lamia for guiding us on how we should approach the problems and find the most positive solutions accordingly throughout the days. Her guidance and suggestions have been a tremendous value till the last of our work. We are grateful to our parents and some of our beloved teachers who have been always inspiring. They encouraged us to make believe that we have got that potential to do the things in a better manner. We would also like to acknowledge numerous people who shared their ideas and works at discussion sessions.

Table of Contents

List of Figures

1. Introduction

With the huge growth in web e-commerce services, the issue of information search and selection has been extremely negative and users are confused about their self-assessment of these methods. Recommendation programs are a useful way for online users to deal with uploading information and have already proven to be the most popular and powerful e-commerce tools.

A major challenge for e-commerce businesses is to ensure high customer purchases from consumers. Helping them find what they want and directing their purchasing information is what makes the process challenging. By providing better services to the visitor all e-commerce sites use the recommendation system to get specific and accurate recommendations based on their preferences. A recommendation program is a support system that is used to help users find information services or product suggestions from other users. Based on past history the recommendation system provides a list of items by predicting which item is most appropriate for the user. A recommendation system is also called a data filtering system, or a search engine used to recommend information.

The collaborative filtering system differs from content-based filtering ideas. The idea of content-based filtering is that users are interested in things like the item users previously liked. On the other hand, the concept of collective filtering is that users like things that are popular with the user's peers.

The recommendation engine recommends products or items available in the data group to users according to their preferences. But it is very difficult for any marketing website to find the latest product ratings and new customer preferences. Modern system limitations include data loss, cold start issues, data switching, and user preferences. It has always been a major problem for the researcher to provide an appropriate solution to this problem. The content-based algorithm has a better solution than working with a filtering algorithm that works with the basic idea of disliking any products offered by the user. The content-based algorithm works on product content which is a great way to give recommendations to new customers and newly installed products.

1.1 Problem Statement

In E-commerce system when new users or new items arrive a problem occurred called cold start. Classic recommender systems like collaborative filtering assumes that each user or item has some ratings so that we can infer ratings of similar users/items even if those ratings are unavailable. However, for new users/items, this becomes hard because we have no browse, click or purchase data for them. As a result, we cannot “fill in the blank” using typical matrix factorization techniques. For this, purpose recommendation system can’t show recommendation for new user and newly added products.

1.2 Objective

1.2.1 General Objective

To recommend~~ed~~ newly added products to the visitor who has never visit an on-e-commerce website previously, try to make a system that will recommendation for those users.

1.2.2 Specific Objectives

- To engage the new customer
- To make the website more comfortable for buyers on large amount of data
- To ensure recommendation of new products

1.3 Research Questions

- What is the reason behind cold star?
- Which model gives best result to make the system more accurate?

- Which algorithm is most suitable to solve cold start?
- How we apply the proposed model to Recommendation System which can also recommend newly added products?

1.4 Relevance and Importance of the Research

The recommender system is a technology support that would help in the smart recommendation of searching for clothing which supports making shopping as easy as a conversation. In general, while buying a product online, the user has to view through every product until and unless the person finds the product they like. But the customer can't view every product on the website until they find a product they would like.

Thus, the recommendation of products will help in easier shopping for the user. Thus, here in this project apparel dataset to recommend products. Generally, there are two types of filtering in recommendation systems:

1. Content-based filtering
2. Collaborative filtering

So, recommendation systems are the systems that use any of the filtering mechanisms or both and recommend similar products that the user may purchase. The content-based recommender system is the mechanism that uses the data of the product such as the description of the product, the brand of the product, the colour of the product, image features such as design to recognize similar products and recommend them to the users. Examples in the e-commerce website are "product with similar colour", "products from the same brand". Collaborative recommender systems use the data of similar users and recommend the products that the user bought or viewed after this product. They are purchased jointly. Hybrid

recommender systems use both the types to recommend the products which may give more accurate results.

In general, a person views a certain product on an ecommerce site when a person likes it. But there might be some imperfections according to the customer who wants a certain type of product. Then, the person may try to look for a similar product that might fulfil their expectations. This can be done by content-based recommender systems.

In this Study We will try to solve cold start problem and make the recommendation system reliable for new user and newly added products. Due to cold start problem recommendation system doesn't recommend when new user or items are added in e-commerce site. If this problem solves then the recommendation system recommended for new user and newly added products. It will engage new customer to the e-commerce website and existing customer also get the recommendation for new products which doesn't have any like or dislike. Also, there is lack of recommendation which can complete cold start task more correctly. So, we will try to improve the recommendation system which worked wisely for cold start problem.

1.5 Scope of the Study

- Find out the solution of sparse matrix
- Recommend products in less information data.
- Products image similarity checking

2. Literature review

Various researchers have encountered problems with the onset of colds and have suggested different guessing techniques. But research has not solved the problem in a positive way. It's a big problem in this huge amount of data, in a newly recommended item that has no limitations and ratings.

Liaoliang Jiang, Yuting Cheng, Li Yang, Jing Li, Hongyang Yan, and Xiaoqin Wang et al (2018) presented a recommendation system recommending products based on reliable data using an advanced slop one algorithm. However, the predictive accuracy of the slope one algorithm is not very high. To solve these problems, they developed a slope one algorithm based on a combination of reliable data and user similarities, which can be used in various recommendation programs. This algorithm has three processes. First, they selected reliable data by dividing a set of data using useful scales. Here they consider if half the people vote yes it will be trusted otherwise it will be a fraud. User similarity measures play an important role because they are used to select neighbouring members and to weigh weight, so the method of calculating similarities between two users is a key problem for systematic filtering interactions. After that they found a similarity of users using the two models, they are Pearson's equality equals and a similarity-based on Cosine. One advanced slope algorithm that works with the same user ratings to generate recommendations. Specifically, ratings of trusted neighbours will be combined to represent the preferences of the active user. Note that the active user is considered to be his trusted neighbour. But Collaborative Filter (CF) algorithms tend to suffer from data sparsity and the problem of the onset of a cold object because the user object matrix is insufficient and too small especially when a new object can be added to the recommendation system.

Guibing Guo et al (2012) presented a program that enriches the unrestricted and where the trustworthy neighbors are not specified. They use a visual rating when someone visiting any site will take a certain amount by looking at this will recommend a new user and products. But it is not so reliable and accurate and it is difficult to get an accurate estimate. And the proposed approach, however, may require building a type of supermarket that is visible and hiring real users for a collection of ratings.

UR. Manjula & A. Chilambuchelvan et al (2016) introduced a process that can predict user preferences using past user history data and other users' historical data, and recommend things to the user. They focused on solving the first cold problem, data sparsity, distribution, and accuracy of Recommender programs. Use content-based filtering to create a recommendation system based on user behavior. To find user similarities use similarities based on Correlation. By identifying the imaginary neighbors of the active user, predictions of the popularity of new products have been made.

Rebecca A. Okaka, Waweru Mwangi, and George Okeyo et al (2016) presented a model that addresses issues related to the design and testing of a personal hybrid recommendation system. It incorporates content-based and collaborative filtering techniques to improve the accuracy of the recommendation. They use a weighty hybridization process that is probably the most straightforward approach to a hybrid system. Based on the concept of compiling the predicted ratings recommended by individual developers to create a list of predefined items from which top items (above k, k = 5) are selected and presented to the user as recommendations. There is a hybrid method that combines CBF and CF methods, while CBF is able to make predictions on any object, CF only adds an object if there are other users of the same rating, the combination of these two methods, therefore, helps eliminate the problem of the new object in CF and a new user problem in CBF. This hybrid method synchronizes the Vector Space Model (VSM) in both CBF and CF, using the algorithm Term

Frequency Inverse Document Frequency (TFIDF) and cosine similarity measurement to find interactions between. However, the recommendation system is tested on very few user numbers and resources that can ensure the effectiveness of the proposed system.

Rajesh Kumar E, Kakani Jyotsna, Keerthana Ganta, and Ramya Sirisha Nori et al (2020) have recommended products made in a variety of ways such as title thinking, product type, color or image alone and recommending products similar to those considered for quality. They take attribute attributes such as title, product, color, and image viewed together. After that, a certain weight is given to all the attributes depending on the requirement of a high uniformity with a particular attribute. After that, use the range of Euclidian brand marks. After that, in terms of weight and distance, products are recommended.

UN Muthurasu, Nandhini Rengaraj, and Kavitha Conjeevaram Mohan et al (2019) have introduced search engines that are trained to produce quick and consistent suggestions for users. They used the document frequency Frequency-Inverse and cosine similarity algorithm to recommend movies. The similarity of Cosine's method is used to measure similarity. The benefits of the program include effective recommendations, valid suggestions even if there is a small data model.

KE MA et al (2016) presented a content-based complimentary program for the movie website. Instead, similarities are calculated based on the information of users' preferences and then make appropriate recommendations. In this recommendation program, they create a user profile and products respectively. The profile was Build based on analytics of consumer purchases or visits. The system can compare user with the profile of items and then recommend the most similar products.

Alan V. Prando, Felipe G. Contratres, Solange N. A. Souza and Luiz S. de Souza et al (2017) introduced the Recommendation Program for new e-commerce users using their

communication only in social media to understand their interests. It uses a content-based approach and enhances the experience of new users by recommending specific products to the user category by analysing their data from a social network. They analyse contact details and network details. Then count the similarities between them. The difficulty of predicting a product grows exponentially with a large number of sectors and products. The proposed RS has been shown to be a reasonable alternative to cold start, that is, for users entering e-commerce for the first time.

C.P. Patidar, Yogesh Katara, and Meena Sharma et al (2020) presented a new hybrid recommendation using tf-idf and weight index for similarities. They used altered Term Frequency-Inverse Document Frequency (TF-IDF) algorithms. They should use the learning method to identify the news category and commend them accordingly. They combined the Naïve Bayes Classification-based Installation Process with the TF-IDF algorithm to make the most common and appropriate recommendation.

We've seen a lot of different approaches are used to make the recommendation system better for new users and newly installed products. On the contrary, we could not have a clear understanding of all their methods and techniques even though we tried our best to learn as much as we could and finally, we learned some important techniques that would help us throughout our research. However, to our knowledge, we have found that content-based filtering is the best way to solve a cold sore problem. Use the content of the material to find similarities between all products using cosine similarities. We are trying to combine a weighted rating with the cosine similarity to make the recommendation more accurate for the new user and for the new non-standard products. Therefore, our goal has been set to find a solution to the first cold problem using content-based filtering using a tf-idf algorithm and a rated rating.

3. Research Methodology

The present study explains and analyses naturally. The development of a research methodology helps the researcher to find a systematic approach to the research process. The current research methodology is based on the tf-idf algorithm and pre-processing methods used for data analysis, data sources, feature engineering, weight measurement, Doc2vec, NLP (Natural Language Processing), TF-IDF (term frequency-inverse document frequency), ways to check the similarity of Cosine. In this study, we created a model to remove a cold start from an e-commerce website. First, the uploaded data will be pre-processed. We then applied the feature engineering process where we extracted those features that we would use to predict to make a recommendation plan. After that, we calculate the average rating from the label review rating and the rating_reviews. We take a data set with labelled information that includes the numerical value and the field value. we needed to pre-process the database and then use those algorithms in the processed database to get product recommendations.

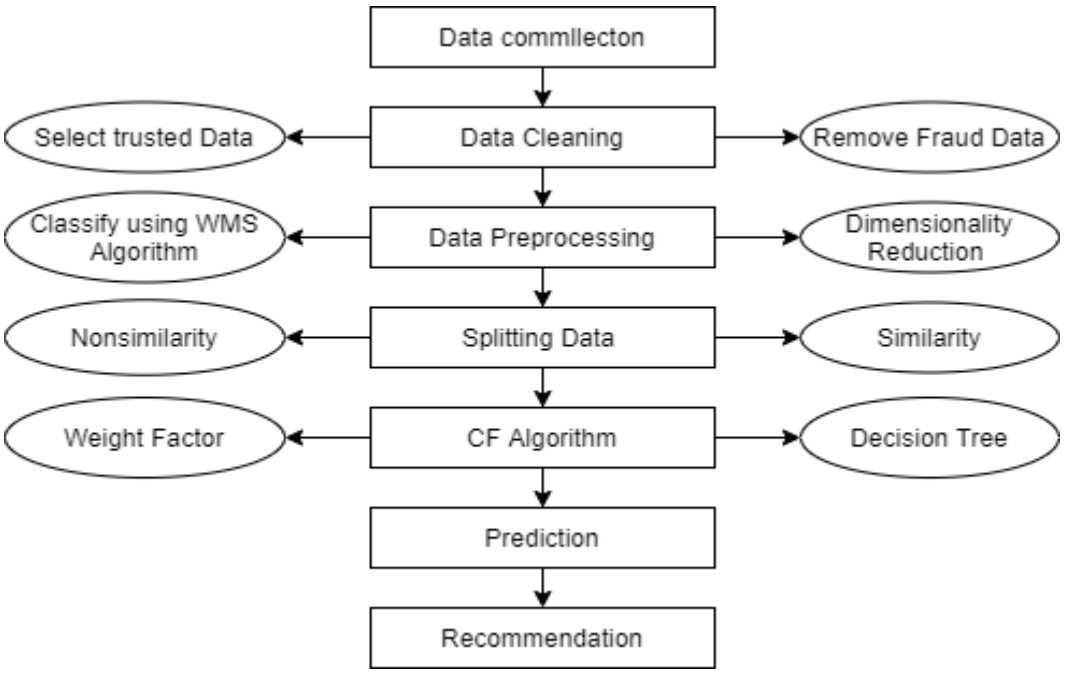

Figure:3.1 Research planning

3.1 Content-Based filtering

Content-based filtering methods are based on the description of the item and profile preferences of the user. These methods are best suited where object data is known. The separator learns from the user whether to like or dislike based on the properties of the object. In particular, the various selection items are compared to the items previously rated by the user and are highly recommended items. Basically, these methods use an object profile that includes an object within the system. Use the tf-idf algorithm presentation algorithm also called vector space representation. The program creates a user-based content profile based on vector-weighted feature elements. Here the weight means the value of each item. It can be calculated from individual content loads using a variety of methods-Weighted vector of item features. Here weight denotes the importance of each feature. It can be computed from individually rated content vectors using a variety of techniques.

3.2 Feature engineering

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. The process of creating new features from raw data to increase the predictive power of the learning algorithm. Engineered features should capture additional information that is not easily apparent in the original feature set. It is way of making new features in dataset to combine other features. As, a part of pre-processing we processed data to make two parameters for calculating the cosine similarity. In this paper we make two new features.

3.3 Weighted rating

Weighted rating is Combining the rating score of different recommendation rating components numerically. Firstly, we multiply mean vote, m with minimum number of ratings, min_n then we multiply it with average rating, avg_n. Then the result will be divided by addition of average rating, avg_n & number of ratings.

weighted_rating = (avg_n * n (m * min_n))/ (avg_n + n)

3.4 word2vec

Word embedding is one of the most popular methods used to represent a word, which can capture not only the context of the word but also its semantic and syntactic and semantic similarities and its relation to other words. Word2Vec is the most widely used process to use embedding. Represents the word as a vector using shallow neural networks divided into two parts trained to reconstruct the linguistic context of words.

As an input, it takes a large corpus of text and generates a vector space, usually with hundreds of sizes and each unique name given a vector corresponding to the space in the space. Vectors are arranged in such a way that names with common features in the corpus are located next to each other. Word2Vec uses two models to generate vector representation:

1) CBOW (common word bag): In this way, the context of each word is considered as inserting and predicting a word that matches the context. For example, consider the text: have a good day. Input is good and we try to predict the date of the target word using the single word input key context. We use a single input code for the input name and determine the error in the output compared to the single input code. of the target word (date). During this process of predicting a specific word, we learn the vector representation of the target word.

2) Skip-Gram model: We can also use the target name (the word we are supposed to produce its representation) to predict the context, this method is called the skip-gram model. In connection, it can be seen as a deviation from the bow model and to some extent, it is true. This structure gives weight to words that are much closer than words that are far away.

3.5 NLP (Natural Language Processing)

It is a process of understanding language by computer using artificial intelligence. It Separate a chunk of continuous text into separate words. For a language like English, this is fairly trivial, since words are usually separated by spaces. Sometimes this process is also used in cases like bag of words (BOW) creation in data mining.

3.6 TF-IDF algorithm

This algorithm is a mathematical method that checks how a word fits into a document set. To do this, we need to multiply 2 metrics: the number of times a word appears in a document, and the frequency of a contradictory word in a text collection. This has a number of purposes, most importantly in textual analysis, which are used for word combinations in Natural Language Processing (NLP) machine learning. It was created to search documents and to find and operate increasingly in proportion to the number of times a word appears in a text, but is removed by the number of texts containing that word. Thus, the words that are common to each text, such as these, do not mean that many of them are specially written. Interestingly, if a word appears several times in the same document, and is rare in other texts, it probably means that it is more important.

How to calculate TF-IDF:

As mentioned earlier, the TF-IDF of the text in the text is available as a product of 2 completely different metrics.

1) Normal Duration of Name in a document. There are several ways to calculate this frequency and the easiest way is to calculate the green in cases where the word appears in the document. There are also several ways to adjust the frequency with the length or quantity of a word that appears most in a text.

2) The Inverse Document Frequency of a word in a collection of documents i.e., how often or how often the word is in the whole corpus. The closer you are to 0, the more common the word is. This can be calculated by dividing the total number of documents by the number of documents containing the name and counting the algorithm.

3) Therefore, if the word is very common and appears in many documents, the number will be closer to 0, otherwise 1. Mathematics form:

tfidf (t, d, D) = tf (t, d) * idf (t, D) where $tf\ (t, d)$ = log (1 + $freq\ (t, d)$), $idf\ (t, d)$ = log ($count\ count\ (d \in D: t \in d)$)

3.7 Cosine similarity

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity. We used this method to determine inner product similarity of a product information. Products information similarity is measured by:

Sim $(dj, dk) = \vec{dj}.\vec{dk} / |\vec{dj}|.|\vec{dk}| = \sum wi, j.wi, k \; n \; i{=}1 \; \sqrt{\sum} wi, j \; n \; 2 \; i{=}1 \; \sqrt{\sum} wi, k \; n \; 2 \; i{=}1$

3.8 Data collection

We collect amazon data set from Kaggle.com.

```
df = pd. read_csv('amazon_co-ecommerce_sample.csv')
df. head ()
```

	uniq_id	product_name	manufacturer	price	number_available_in_stock	number_of_reviews	number_of_answered_questions	av
0	eac7efa5dbd3d667f26eb3d3ab504464	Hornby 2014 Catalogue	Hornby	£3.42	5 new	15	1.0	
1	b17540ef7e86e461d37f3ae58b7b72ac	FunkyBuys® Large Christmas Holiday Express Fes...	FunkyBuys	£16.99	NaN	2	1.0	
2	348f344247b0c1a935b1223072ef9d8a	CLASSIC TOY TRAIN SET TRACK CARRIAGES LIGHT EN...	ccf	£9.99	2 new	17	2.0	
3	e12b92dbb8eaee78b22965d2a9bbbd9f	HORNBY Coach R4410A BR Hawksworth Corridor 3rd	Hornby	£39.99	NaN	1	2.0	
4	e33a9adeed5f36840ccc227db4682a36	Hornby 00 Gauge 0-4-0 Gildenlow Salt Co. Steam...	Hornby	£32.19	NaN	3	2.0	

Figure: 3.2 View of Data set

3.9 Data Pre-processing

At First, we check how many features we have in dataset and is there any null value exist or not. We find there is 10000 row and 17 columns.

```
df.shape
```

```
(10000, 17)
```

```
df.isna().sum()
```

```
uniq_id                                                    0
product_name                                               0
manufacturer                                               7
price                                                   1435
number_available_in_stock                               2500
number_of_reviews                                         18
number_of_answered_questions                             765
average_review_rating                                     18
amazon_category_and_sub_category                         690
customers_who_bought_this_item_also_bought              1062
description                                              651
product_information                                       58
product_description                                      651
items_customers_buy_after_viewing_this_item             3065
customer_questions_and_answers                          9086
customer_reviews                                          21
sellers                                                 3082
dtype: int64
```

Figure: 3.3 Checking columns value

Then we drop unnecessary column from the data set.

df.dsrop(columns=['price','number_available_in_stock','number_of_answered_questions','product_description','customer_questions_and_answers','customer_reviews','customers_who_bought_this_item_also_bought','items_customers_buy_after_viewing_this_item','sellers','items_customers_buy_after_viewing_this_item'], inplace=True)

df. head ()

now we can see the output of existing column.

```
uniq_id                          0
manufacturer                     7
number_of_reviews               18
average_review_rating           18
amazon_category_and_sub_category  690
description                     651
product_information              58
dtype: int64
```

Figure: 3.4 Processed columns

To remove null value from the existing features we used lope and null value will replaced by ‘0’ for ‘number_of_reviews’, & ‘average_review_rating’ columns and other column will be replaced by ‘ ’ empty string. Output of existing column-

```
uniq_id                          0
manufacturer                     0
number_of_reviews                0
average_review_rating            0
amazon_category_and_sub_category 0
description                      0
product_information              0
dtype: int64
```

Figure: 3.5 Removed Null value

Data types of those columns are as object type. So, we continue a lope for every row to make ‘number_of_reviews’ column into integer type & ‘average_review_rating’ column into float type. Now we find out the minimum rating and mean vote using df.

	uniq_id	manufacturer	number_of_reviews	average_review_rating	amazon_category_and_sub_category	description	product_in
count	10000	10000	10000.0000	10000.000000	10000	10000	
unique	10000	2652	NaN	NaN	256	8515	
top	ae6e043c32c3e2192bdfffae87c48f27	LEGO	NaN	NaN	Die-Cast & Toy Vehicles > Toy Vehicles & Acces...		
freq	1	171	NaN	NaN	880	651	
mean	NaN	NaN	9.1235	4.698810	NaN	NaN	
std	NaN	NaN	33.7000	0.422089	NaN	NaN	
min	NaN	NaN	0.0000	0.000000	NaN	NaN	
25%	NaN	NaN	1.0000	4.500000	NaN	NaN	
50%	NaN	NaN	2.0000	5.000000	NaN	NaN	
75%	NaN	NaN	6.0000	5.000000	NaN	NaN	
max	NaN	NaN	1399.0000	5.000000	NaN	NaN	

describe(include='all').

Figure: 3.6 Description of Data set

Now we take initial value

weighted_rating = 0,

minimum_num_of_ratings = 9,

mean_vote = 4.6

Then calculate weighted rating for every row using the equation:

```
df['weighted_rating'][ind]=((df['average_review_rating'][ind]*df['number_of_reviews'
][ind])+(mean_vote*minimum_num_of_ratings))/(df['average_review_rating'][ind]+d
f['number_of_reviews'][ind])
```

We drop ‘number_of_reviews’, & ‘average_review_rating’ column’s and make a new column named weighted_rating.

	uniq_id	manufacturer	amazon_category_and_sub_category	description	product_information	weighted_rating
0	eac7efa5dbd3d667f26eb3d3ab504464	Hornby	Hobbies > Model Trains & Railway Sets > Rail V...	Product Description Hornby 2014 Catalogue Box ...	Technical Details Item Weight640 g Product Dim...	5
1	b17540ef7e86e461d37f3ae58b7b72ac	FunkyBuys	Hobbies > Model Trains & Railway Sets > Rail V...	Size Name:Large FunkyBuys® Large Christmas Hol...	Technical Details Manufacturer recommended age...	7
2	348f344247b0c1a935b1223072ef9d8a	ccf	Hobbies > Model Trains & Railway Sets > Rail V...	BIG CLASSIC TOY TRAIN SET TRACK CARRIAGE LIGHT...	Technical Details Manufacturer recommended age...	5
3	e12b92dbb8eaee78b22965d2a9bbbd9f	Hornby	Hobbies > Model Trains & Railway Sets > Rail V...	Hornby 00 Gauge BR Hawksworth 3rd Class W 2107...	Technical Details Item Weight259 g Product Dim...	7
4	e33a9adeed5f36840ccc227db4682a36	Hornby	Hobbies > Model Trains & Railway Sets > Rail V...	Product Description Hornby RailRoad 0-4-0 Gild...	Technical Details Item Weight159 g Product Dim...	7

Figure: 3.7 Create Weighted Rating

Then we remove ‘>’ sign from ‘amazon_category_and_sub_category’ and combine ‘product_information’, ‘manufacturer’, ‘amazon_category_and_sub_category’ to make a new column named products_details.

	uniq_id	weighted_rating	product_details
0	eac7efa5dbd3d667f26eb3d3ab504464	5	Technical Details Item Weight640 g Product Dim...
1	b17540ef7e86e461d37f3ae58b7b72ac	7	Technical Details Manufacturer recommended age...
2	348f344247b0c1a935b1223072ef9d8a	5	Technical Details Manufacturer recommended age...
3	e12b92dbb8eaee78b22965d2a9bbbd9f	7	Technical Details Item Weight259 g Product Dim...
4	e33a9adeed5f36840ccc227db4682a36	7	Technical Details Item Weight159 g Product Dim...

Figure: 3.8 Create Product Details

We take some library file and use a lope for taking every row of products to remove stop words and punctuation from product_details column. Now product_details are looking-

'Technical Details Item Weight640 g Product Dimensions296 x 208 x 1 cm Manufacturer recommended age6 years Item model numberR8148 Main Language English manual English Number Game Players1 Number Puzzle Pieces1 Assembly RequiredNo Scale172 Engine Typeelectric Track WidthGaugeHO Batteries Required No Batteries Included No Material Type Paper Material Care InstructionsNo Remote Control Included No Radio Control Suitabilityindoor Colorwhite Additional Information ASINB00HJ208KO Best Sellers Rank 52854 Toys Games See top 100 69 Toys Games Model Trains Railway Sets Rail Vehicles Trains Shipping Weight640 g Delivery Destinations Visit Delivery Destinations Help page see item delivered Date First Available24 Dec 2013 Feedback Would like update product info give feedback images Hornby Hobbies Model Trains Railway Sets Rail Vehicles Trains'

Figure: 3.9 Processed Information

3.10 TF-IDF Algorithm

We take some library file such as pandas, TfidfVectorizer, Linear_kernel for this algorithm. We analyse word of products_details which has a range 1 to 3 using tfidfVectorizer. Then find out the tfidf_matrix.

tf= TfidfVectorizer (analyzer='word', ngram_range=(1,3),min_df=0,stop_words='english')

tfidf_matrix = tf.fit_transform(df['product_details'])

3.11 Similarity checking

In this paper we used cosine similarity to check the similarity between products_details of every item. We used loop for every item and store similar item in an object named result.

```
cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
```

```
results = {}

for idx, row in df.iterrows():
    similar_indices = cosine_similarities[idx].argsort()[:-100:-1]
    similar_items = [(cosine_similarities[idx][i], df['uniq_id'][i]) for i in similar_indices]

    results[row['uniq_id']] = similar_items[1:]

print('done!')
```

```
done!
```

Figure: 3.10 Similarity Calculation

4. Result and Discussion

We used to take row from result variable and multiply weighted rating with cosine similarity

To get the recommended item.

```
def item(id):
    return df.loc[df['uniq_id'] == id]['product_details'].tolist()[0].split(' - ')[0]
```

```
st reads the results out of the dictionary.
recommend(item_id, num):
print("Recommending " + str(num) + " products similar to " + item(item_id)[:150] + "...")
print("-------")
recs = results[item_id][:num]
for rec in recs:
    print("Recommended: " + item(rec[1])[:150] + " (score:" + str(rec[0] * df.loc[df['uniq_id'] == item_id, 'weighted_rating'] )

mmend(item_id='eac7efa5dbd3d667f26eb3d3ab504464', num=5)
```

Figure: 4.1 Result Calculation

Recommended item is sorted in the list and show the recommendation to the user. We take 5 items from recommendation list to show. However, we can recommend more than this.

```
Recommending 5 products similar to Technical Details Item Weight640 g Product Dimensions296 x 208 x 1 cm Manufacturer recommend
ed age6 years Item model numberR8148 Main Language   Engli...
-------
Recommended: Technical Details Item Weight240 g Product Dimensions364 x 108 x 6 cm Manufacturer recommended age3 years Item mod
el numberR4526 Main Language   Engli (score:0    1.692605
Name: weighted_rating, dtype: float64)
Recommended: Technical Details Item Weight25 Kg Product Dimensions80 x 30 x 8 cm Manufacturer recommended age6 years Item model
numberR1177 Main Language   Italian (score:0    1.689282
Name: weighted_rating, dtype: float64)
Recommended: Technical Details Item Weight200 g Product Dimensions152 x 34 x 305 cm Manufacturer recommended age4 years Item mo
del number54225 Main Language   Engl (score:0    1.574841
Name: weighted_rating, dtype: float64)
Recommended: Technical Details Item Weight599 g Product Dimensions36 x 102 x 6 cm Manufacturer recommended age8 years Item mode
l numberR3274 Main Language   Englis (score:0    1.535149
Name: weighted_rating, dtype: float64)
Recommended: Technical Details Item Weight23 Kg Product Dimensions812 x 302 x 82 cm Manufacturer recommended age8 years Item mo
del numberR1176 Main Language   Engl (score:0    1.518124
Name: weighted_rating, dtype: float64)
```

Figure: 4.2 Generated Result

In the figure: 4.2 we can see after score a value is given, it is the multiplied score of cosine similarity and weighted rating. Based on the higher score of any products recommendation system recommended user. Actually, our designed system recommended newly added product and new user based on similar product details with initial review_rating 1. It will solve cold start problem by recommended newly added products and make shoppers shopping more reliable on e-commerce website.

Limitations of the Study

- Word embedding features create a dense.
- It does not capture the semantic meaning in low dimensional feature whereas tf-idf creates a sparse.
- Products information must have to well described

5. Conclusion

It is always important to give accurate recommendations to visitors or customers about their easy and comfortable purchases. Our work will provide relevant recommendations to the new user for troubleshooting a cold and data problem. This study was conducted using a database from AMAZON to generate user interest recommendations when visiting an e-commerce site. We have found that the models: Weight Loss and Doc2Vec (Paragraph Vectoring) have been very effective in generating recommendations with the help of feature extraction from training data and generating cosine similarity. Testing is best done on recommendation models where a new product can be added to an e-commerce site. Therefore, we can say that the choice of models and methodology in the construction of this program helped to find the first cold solution.

5.1 Practical Implications

Our finding will help the system more sophisticate. We designed a demo model to evaluate our research. That is working smoothly and get the solution of cold start problem. I think this solution will bring great changes in e-commerce website.

5.2 Future Work

Although our research is giving recommendation and solve the cold start problem. Still, we have to work on low dimensional data where this system can't perform well. Also, we have to solved the sparse matrix of low dense data.

6. References

Jiang, L., Cheng, Y., Yang, L., Li, J., Yan, H. and Wang, X., 2019. A trust-based collaborative filtering algorithm for E-commerce recommendation system. Journal of Ambient Intelligence and Humanized Computing, 10(8), pp.3023-3034.

Guo, G., 2012, July. Resolving data sparsity and cold start in recommender systems. In International Conference on User Modeling, Adaptation, and Personalization (pp. 361-364). Springer, Berlin, Heidelberg.

Manjula, R. and Chilambuchelvan, A., 2016. Content Based Filtering Techniques in Recommendation System using user preferences. *Int. J. Innov. Eng. Technol*, *7*(4).

Okaka, R.A., 2018. *A Hybrid Approach for Personalized Recommender System Using Weighted Term Frequency Inverse Document Frequency* (Doctoral dissertation, JKUAT-COPAS).

Ma, K., 2016. Content-based Recommender System for Movie Website.

Prando, A.V., Contratres, F.G., de Souza, S.N.A. and de Souza, L.S., 2017. Content-based Recommender System using Social Networks for Cold-start Users. In *KDIR* (pp. 181-189).

Goossen, F., IJntema, W., Frasincar, F., Hogenboom, F. and Kaymak, U., 2011, May. News personalization using the CF-IDF semantic recommender. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics* (pp. 1-12).

Philip, S., Shola, P. and Ovye, A., 2014. Application of content-based approach in research paper recommendation system for a digital library. *International Journal of Advanced Computer Science and Applications*, *5*(10).

Schafer, J.B., Konstan, J. and Riedl, J., 1999, November. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce* (pp. 158-166).

Kimble, D. and Chin, F.L., Time Warner Cable Enterprises LLC, 2015. *Recommendation engine apparatus and methods*. U.S. Patent 9,215,423.

Prando, A.V., Contratres, F.G., de Souza, S.N.A. and de Souza, L.S., 2017. Content-based Recommender System using Social Networks for Cold-start Users. In *KDIR* (pp. 181-189).

Huang, S., 2018. Word2Vec and FastText Word Embedding with Gensim.

Jin, R., Chai, J.Y. and Si, L., 2004, July. An automatic weighting scheme for collaborative filtering. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 337-344).

Jin, B.W., Cho, Y.S. and Ryu, K.H., 2010. Personalized e-commerce recommendation system using RFM method and association rules. *Journal of the Korea society of computer and information*, *15*(12), pp.227-235.

Wei, J., He, J., Chen, K., Zhou, Y. and Tang, Z., 2017. Collaborative filtering and deep learning-based recommendation system for cold start items. *Expert Systems with Applications*, *69*, pp.29-39.

Isinkaye, F.O., Folajimi, Y.O. and Ojokoh, B.A., 2015. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), pp.261-273.

Jain, S., Grover, A., Thakur, P.S. and Choudhary, S.K., 2015, May. Trends, problems and solutions of recommender system. In *International conference on computing, communication & automation* (pp. 955-958). IEEE.

Wu, Y. and Zheng, J., 2010, December. A collaborative filtering recommendation algorithm based on improved similarity measure method. In *2010 IEEE International Conference on Progress in Informatics and Computing* (Vol. 1, pp. 246-249). IEEE.