E-Commerce Recommender System Using Product Data

Rajesh Kumar E, Kakani Jyotsna, Keerthana Ganta, Ramya Sirisha Nori

Abstract: In previous days, before buying a product, people used to get suggestions from our close friends, family or people known. This is the basic idea of recommender systems. Recommender systems work with the same idea of predicting a product that a customer may like to buy. Recommender systems can be used in different areas to recommend products such as books, apparel, accessories, movies according to the items viewed. A recommender system is a system that helps to expect and recommend similar products to the given input of the product. In many online e-commerce websites like Amazon, Myntra and other sites, one can find sections like “recommended for you”, “products related to this item”, “customers also viewed” when a person views certain product. These are the recommended sections.

Index Terms: Content-based filtering, Convolutional neural networks, Inverse Document Frequency, One hot encoding, Recommender systems, Term Frequency, Word2Vec.

1. INTRODUCTION

The recommender system is new technology support that would help in the smart recommendation of searching for clothing which supports making shopping as easy as a conversation[1]. 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 of the user[2]. 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[3]. 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 color of the product, image features such as design to recognize similar products and recommend them to the users[9]. Examples in the e-commerce website are "product with similar color", "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[14]. They are "purchased jointly[7]. 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 e-commerce site when a person likes it. But there might be some imperfections according to the customer who wants a certain kind of product[5]. Then, the person may try to look for a similar product that may meet expectations. This can be done by content-based recommender systems[4]. Thus, in this project, the usage of content-based recommender systems that use apparel data to demonstrate how a content-based recommender system works is explained. Here, both descriptions and images of the product to give the best accurate results possible.

Thus, it has two types of recommendations:

1. text-based recommendation
2. image-based recommendation

Both the methods are combined to get more accurate results.

- *Rajesh Kumar E, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India. E-mail: rajthalopo@gmail.com*
- *[2]Kakani Jyotsna ,[3]Keerthana Kanta, [4]Ramya Sirisha Nori is currently pursuing B. Tech program in department of computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India. [2]E-mail: Jyotsna.kakani@gmail.com, [3]E-mail: Keerthana.s.ganta@gmail.com, [4]E-mail: ramyanori08@gmail.com,*

2 LITERATURE SURVEY

2.1 Content-based filtering

It is also known as cognitive filtering which differentiates the item between the content and the user profile. There are some steps to be followed as to recommending the products to the consumers[9].

1. Distinguishing and defining the products will be prominent if the user purchased the product or not.
2. Define all the products such as characteristics, descriptors, and variables.
3. Even though the user gets affected by the factor it will still contain the same amount of variables.
4. In terms of variables, recommend the closest products.

2.2 Term Frequency Score

Consider the documents as a bag of words that are agnostic. When the whole document with the term frequency 1 is not more relevant than the document with 10 occurrences of the term which is not 10 times more relevant, relevance is not proportional to frequency.

$$T, F_{i,j} = \frac{n_{i,j}}{\sum k, n_{i,j}} \quad (1)$$

TF = (Number of times term appears in a document) / (Total number of terms in the document)

It calculates the number of times each word appears in the document. The stopped words will be eliminated and it is converted to the lower cases.

2.3 Inverse Document Frequency

In inverse document frequency, it is computed as the logarithm where it assembles a collection of all the documents by the number of documents where the specific term appears[6].

$$IDF(WT) = \log \frac{N}{Df_i}$$

(2)

.4 TF-IDF

By using this TF-IDF technique it will help us find the important words and will give us the idea of the document. By using this TF-IDF it will remove all the stopped words such as “a”, “is”, “the”. TF-IDF is used because it is widely used in recommendation engines. TF-IDF is not a single method but consists of various techniques including the text classification and the summarization. The calculations and check through every document are done and count of how many times the word has appeared. Then in the next process, take the number of documents and divide by the documents which contain the word[11]. Last but not least, multiply the TF and IDF together.

2.5 Word2Vec

A Word2Vec is a two-layer neural network which builds up the language context of words[13]. In Word2Vec it will take the collection of words as input and create as vector-space with number of dimensions where each word is allocated to the vector in the space as a matrix[6].

2.6 Convolutional neural networks

CNN is a deep-learning algorithm that uses the image as an input to various objects in an image to distinguish from each other and is designed to process the pixel data. They have the ability to learn these characteristics and require lower of pre-processing techniques[8]. CNN makes it easy to process because it reduces the image into some form. By doing this, it will not lose any features which are critical for getting a good prediction. This architecture uses learning features as well as very helpful to the massive datasets. A photo is nothing but a pixel quality matrix. So for classification purposes, just flatten the image and feed it to a multi-level perceptron[6]. For extremely basic binary images, the method can show an average accuracy score when conducting class prediction but would have little to no accuracy when it comes to complex images with pixel dependencies throughout. A ConvNet can capture the spatial and temporary dependencies in an image successfully by applying appropriate filters. Because of the reduction in the number of parameters involved and the reusability of weights, the architecture provides a better fit to the image dataset.

3 PROPOSED WORK

In general, the recommendation of products is done in different ways like considering the titles, brand, color or image alone and recommending products that are similar to the considered attribute. In this project, dataset attributes like title, brand, color, and image are considered together. Then, certain weight is given to every attribute according to the requirement of the high similarity to a particular attribute. Then, by using different techniques for every attribute, Euclidian distance of product attributes is calculated. Then, according to the attribute weight and distance, products are recommended.

4 IMPLEMENTATION

4.1 Data cleaning

Data cleaning is one of the important stages of machine learning. In this stage, understanding the data and clean up some unwanted data is to be done. Though it is often overlooked, it is an extremely important stage[4]. It may take a few hours to days to perform data cleaning activity because the more one understands the data, the more they can design the modules of the system. If data cleaning is not performed well, the construction of the application will be tough[3]. While considering our data, one can observe many products that have color and price as none. At the beginning of data cleaning, a dataset has 1.83 lakhs of product data. After checking the null values of price and removing them, the products remained are 28395. And after every such process, store this data as a pickle file so that if at any certain time, the system takes a long time to run, use these pickle files with fewer data. By repeating the same step for color, the products remained are 28385. Thus after removing null values in data cleaning, the data points are reduced from 1.83 lakhs to 28K products roughly. For some products, the titles will be the same but might be of different color or size. These are title duplicated products. Thus, the products with the same titles are 2325. The next stage is to de-duplicate these products. Product titles with short descriptions are very informative and they are present in most of the products. So, products with short descriptions need to be removed. Thus the products left are 27K. As mentioned above, there will be products with the same title and different sizes. These products word difference is considered and products with fewer word differences are eliminated. Thus, in the end, product data points remained are 17K.

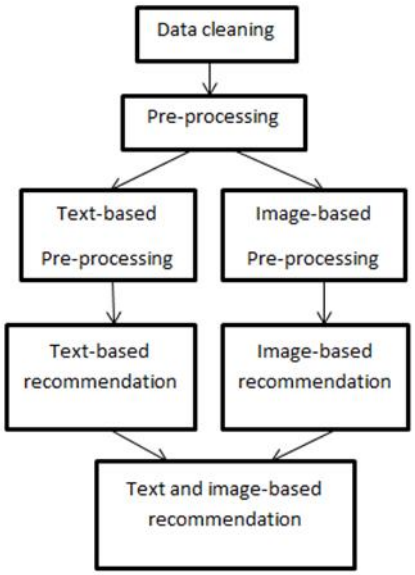


Fig1: *phases of implementation*

4.2 Text pre-processing

Previously, the data was cleaned by de-duplicating and removing those rows which have empty fields or columns. Then, have to apply some pre-processing techniques[1].

4.2.1 Stop-word removal

This technique is used for almost all text in English in machine learning. Stop words are basic words like “if”, “the”, “for”, “a”, “but”, “and”, “or”, “may”. The occurrence of such words is very frequent and usually not very informative in these scenarios. Here, these words are not useful. Stop word removal is not

useful or necessary for all algorithms. The next stage in implementation is to remove stop words. First, special characters are removed. Secondly, all the text is converted into lowercase to avoid confusion. Finally, stop words are removed.

4.3 Text-Based Product Similarity

At this point, there are 16k products. Titles are used because they are short and very informative. Next, usage of the text in the title as a primary system for product similarity takes place. So, titles will be used intensively here. The dataset is already de-duplicated and pre-processed by removing stop words[9]. Consider Ti as the title notation of titles of the products. According to linear algebra, if you can represent a data point as an n-dimensional point or as an array of any size, then methods like Euclidian distance can be used to find similar points. Now, each product has a title and it can be represented as a vector (i.e. as a d-dimensional point). For all products P1, P2, P3 and so on, there are titles T1, T2, T3 and so on. For each of those titles, if got a d-dimensional point, then, Euclidian distance method can be implemented to find similar points[2].

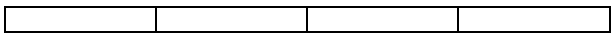
4.3.1 Weighted similarity using brand and color

Till now for the product similarity titles are used. But now let's look into the other parameters which are brand and color. For the product, the title is considered and encoded it as the vector using size 300 by using IDF-Word2Vec. This can be called as the title vector. As seen before for each of the products, there is a title Ti and created a vector. For similarly of the products by using the brand and color, construct vector for each brand (Bi) and color (Ci). Consider there are m brands such as b1, b2, b3…bm. For the product, Pi->Bi, create a vector consisting of 1 to m. This will be known as the m-dimensional vector. For the color also the same strategic procedure is used. There are k distinct colors. So, create a vector space for each color. If it had three vectors such as title, brand, and color; then concatenate all these three vectors to one single vector. If given two products Pi and Pj, then find the Euclidian distance which the final result will give us the product vector[10]. Assume that the technique prefer showing our customers products of the same brand. Simply take the weighted Euclidian distance. Firstly come up with the three weights for the title (Wt), brand (Wb) and color(Wc). Now, take all the elements of title and multiply with Wt and do the same with brand and color. After, multiply and take the Euclidian distance which is exact to the weighted Euclidian distance. Since the brand has the highest value, the technique ends up preferring the customers the products of the same brand[10].

4.4 Image pre-processing

As seen before, for title different techniques such as BOW, TF-IDF, and Word2Vec are used. Similarly, for brand and color, one-hot encoding is used. Now for the image, there is a need to convert image to n-dimensional vector[8]. In image, there are different parameters such as edges, shapes, colors, patterns (zebra stripe, leopard print).

Colors Shapes Pattern Edges



Consider the vector this and find the Euclidian distance for the similarity. A deep learning technique for the image to convert it to a d dimensional vector is used. The deep learning technique is a Convolutional Neural Network (CNN)[12]. There are many types of CNN and the most popular one is VGG-16. When 2 products are similar then the Euclidian distance is very small.

4.5 IMAGED-BASED RECOMMENDATION

Keras and Tensorflow are the two basic libraries which convert the image into a d dimensional vector. For each image we have 25088 dimensional features.

Measuring goodness of our solution: A/B testing

A solution is a mixture of pre-processing algorithms and business rules. For each image there could be different solutions. Some might say that solution one is good while other says second one. We evaluate it by using the A/B testing or the Bucket Testing.

5 CONCLUSION

By comparing all different techniques available, for text-based recommendation by using the title, IDF-Word2Vec is useful. For considering the brand and color of the products, one hot encoding gives more accurate results. The image-based recommendation is done by using VGG-16 CNN. Thus, weight is given for every attribute according to their importance. Find the weighted Euclidian distance and products are recommended according to similarity.

6 REFERENCES:

[1] Guan Shengfang, "Apparel Recommendation System Evolution, An Empirical Review", International Journal of Clothing Science and Technology, Vol. 28 Iss 6 pp, 2016.

[2] Richa Sharma and Rahul Singh, "Evolution of recommender systems", Indian Journal of Science and Technology, Vol 9(20), May 2016.

[3] Shah Khusro, Zafar Ali and Irfan Ullah, "Recommendation systems issues and challenges", Information Science and Applications (ICISA) 2016.

[4] Brent Smith and Greg Linden, "Two Decades of Recommender systems at amazon", in IEEE Computer Society, 2017.

[5] Zhang, J.-D., Chow, C.-Y "SEMAX: Multi-task Learning for Improving Recommendations" ,IEEE Access,2018.

[6] R. R. Salakhutdinov,"Probabilistic matrix factorization", in Proc. NIPS, 2008, pp. 1257–1264.

[7] S.Zhang, "enabling kernel-based attribute aware matrix factorization for rating prediction", IEEE Trans. Knowl. Data Eng., vol. 29, no. 4, pp. 798–812, Apr. 2017.

[8] P.Covington, "Deep learning based recommender system: A survey and new perspectives", [Online]. Available: https://arxiv.org/abs/1707.07435

[9] Wang, H., Wang, N., Yeung, "Collaborative Deep Learning for Recommender Systems". Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '2015.

[10] Kedar Potdar, "A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers", International Journal of Computer Applications 175(4):7-9 · October 2017.

[11] D. Kim, C. park, "Convolutional matrix factorization for document context-aware recommendation", in Proc. ACM RecSys, pp. 233–240, 2016.

[12] R. Catherine and W. Cohen, ``TransNets: Learning to transform for recommendation,'' in Proc. ACM RecSys, pp. 288–296, 2017.
[13] D. Bahdanau, K. Cho, and Y. Bengio ”Neural machine translation by jointly learning to align and translate'', in Proc. ICLR, 2015, pp. 1–15.
[14] J. Tang, H. Gao, X. Hu, and H. Liu, ``Context-aware review helpfulness rating prediction'', in Proc. ACM RecSys, 2013, pp. 1–8.