Content based Item-Item Recommendation System based on User Interactions in E-Commerce Applications

Aninda Ghosh **Ayushman Gupta** **Mrs. M. Harini**

(Assistant Professor SRM IST)

SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu-603203

Abstract - Goal of this project is to recommend products of any category to a user using an E-Commerce application from a cold start without any prior usage and after minimal interactions using Item-Item Content based filtering on these said interactions. The existing systems are mainly for large scale E-Commerce applications with larger user base and emphasize less on cold starts that can lead to a lower retention rate for small scale E-Commerce applications. So, a solution is required to provide a similar experience to customers using small scale E-Commerce applications.

I. INTRODUCTION

A. OVERVIEW

Big Data is a terminology that refers to very large or huge amounts of data, that cannot be managed or processed by a single individual manually or by existing database systems. If they manage to do so, it will require a large amount of time and will also involve huge computational power. Characteristics of, and associated with Big Data are Velocity, Variety, and Volume. Multiple day-to-day applications today involve Big Data generation such as on Social Media, Log Files, Aircraft Systems and more. This kind of data is generated in real time and on a large scale.

Big data is analyzed to allow ease of business decisions and to provide researchers with some useful information that was not available to them before. Many new things can be uncovered by this analysis such as hidden arrangements, patterns, progression, customer appeal and correlations between sets of data within the big data. Various techniques such as language processing and machine learning can be applied to the analyzed big data to provide further and new insights on any previously uninformed or unknown data. Some very important business-related outcomes can be achieved positively with the help of Big Data Analytics such as effective marketing of products, bigger revenue opportunities, improvements in departments of operational efficiency and an inevitable competitive advantage over business rivals.

B. OBJECTIVE

The objective of this project is to build a Content based Item-Item Recommender system from a dataset of PC Parts obtained from online E-Commerce websites to recommend the most closely related products to query to the user depending on the user interactions within the application or website.

For example: If a user opens a product page of a RAM module with price “X”, brand “Y” and rating

"Z", then the recommender system would immediately recommend the user with other RAM modules close to the price range "X" and different models sold by the same brand "Y" or an entirely different brand with ratings similar to "Z".

The goal is to build this system for small scale, up and coming, E-Commerce platforms that do not have the resources to spend on a computationally expensive and heavy recommender system. It also built by keeping cold starts in mind. Cold start here means that a brand-new user on the website or application would receive a recommendation immediately after the first product page visit.

C. CHALLENGES

Capturing user-interactions is of utmost importance when building a recommendation system. There are various challenges faced with such a process. The amount of data collected might often be overwhelming and every interaction might not have to be captured and recorded.When dealing with so much data, it cannot be manually accessed and analyzed all the time and hence it must be collected, and meaningful insights must be generated automatically. When data is collected across multiple, disjoint sources, they cannot be combined manually and can limit the insights required as well. Poor quality and inaccurate data can lead to significant negative consequences. Finally, scaling the data analysis system is a crucial challenge as the amount of data being captured grows.

Although Collaborative Filtering is a widely used and predominantly successful recommendation method, there still exist some major problems with this method. The first difficulty is due to "Sparseness". Since collaborative filtering utilizesspecific"non-binary" user interactions and ratings for predicting homogeneous products, the amount of user ratings already collected is very small compared to the number of ratings that need to be predicted.Therefore, recommendations based on collaborative filtering cannot accurately identify products.The quality of the recommendation plays a very important role in identifying the future buying behavior of the customer. Hence it is important that the recommender system avoids two major characteristic errors, false positives, and false negatives.The other challenge faced with the Collaborative Filtering method is "Scalability". Since the method utilizes neighbor algorithm whose computation increases proportionally to both number of products as well as customers. This can slow down the rate at which the customer receives recommendations thereby alienating them from the platform itself.

D. MOTIVATION

Analyzing the statistics of user interactions on mobile e-commerce apps can help give an important insight to the behavior of users and how often they return to the application or website to browse for products, what products they usually look at and where they usually end up within the website or application. These statistics can also give an insight of an overall trend and what are the most looked at products or categories or even price ranges allowing the business to create or build a growth trajectory in a direction that aligns with the said statistics can help the businesses implement a recommendation system that would suggest the user with products closely related to what they are looking for resulting in a longer engagement time of the user, higher sales probability, more options to choose from as well as higher retention rate on the application or website. Large scale E-Commerce platforms have already been benefitting by the implementation of a recommender system and having the same implemented for a small-scale E-Commerce platform can possibly boost their business growth.

II. RELATED WORK

A. *USER INTERACTION ON E-COMMERCE WEBSITES*

The recommendation system to be implemented in this project will be created with the help of completely non-intrusive user interaction data. This data will give an insight about what a particular user isinterested in, the kind of product the user wants and the price range of the same. Using this data, the recommendation system will recommend the user products related to these attributes to help the user in their shopping. There are various ways to obtain this data.

"Konstantinos F. X., Panagiotis K, and Reda Alhajj" [1] used the method of clickstream analysis to understand what a user does when they are on a specific E-Commerce website. Their model involved various algorithms and worked well for items that were most frequently bought. Using the resultant data from the clickstream analysis they were able to discover hidden knowledge that would allow a boost in the overall business.

"Gautam Pal, Gangmin Li, and Katie Atkinson" [2], created a data ingestion paradigm for E-Commerce analytics in order to better understand consumer interactions and, as a result, improve the company in a variety of ways. Their solution was a Real-Time Clickstream data ingestion model built on top of popular Big Data tools such as Kafka and Flume. This model was fault tolerant and was setup in a distributed manner hence boosting performance with real-time results.

"Sahana Raj, and Dr. B Satish Babu" [10] created a model that would give results more than enough to convert an e-commerce prospect to a customer. This was done by again analyzing user interaction data with the help of Data/Event Extraction and Apache Spark and it produced results in a smaller time window which in turn accelerated decision making.

B. *CONTENT-BASEDRECOMMENDATION SYSTEMS*

Recommendation system models [13] seek to predict the products or things a user may be interested in based on details about the user's profile. Collaborative filtering methods, content-based methods, or hybrid filtering methods that combine the two techniques are the most popular recommender systems. The suggestions of collaborative filtering methods are based on a comparison of a user's previous choices with those of other users who have made similar choices. Content-based systems on the other hand, utilize information about a product itself to provide recommendations. It compares the similarity of the product's key parameters to find the products most similar to it.

"Xiaofeng Li and Dong Li" [21] proposed a group detection-based collaborative recommendation algorithm that integrates social network-related technologies into collaborative recommendation technology. This paper proposes a collaborative recommendation approach based on community detection based on community discovery technology and collaborative recommendation technology in social networks to address the current problems in conventional recommendation techniques. The algorithm uses a coarse selection method of neighboring user sets to effectively reduce device calculation time, while still using a

scoring pre-processing method to avoid the problem of data scarcity.

A good recommendation algorithm is scalable across very large product catalogues and consumer bases for large retailers like Amazon.com [14], it only takes less than one second to produce recommendations online, it can respond quickly to changes in a user's data, and it provides convincing recommendations for all users regardless of the number of transactions and reviews.Collaborative item-to-item filtering, unlike other algorithms, will meet this challenge.

"Daniel Lemire and Anna Maclachlan"[20] implemented a Collaborative Filtering based on average rating differential called Slope One that can compete against more expensive memory-based schemes. The Slope One schemes are simple to implement, dynamically updateable very efficient during query time, and can deal well with cold-starts on e-commerce platforms for a user.

C. INFERENCE OF THE LITERATURE SURVEY

"Konstantinos F. Xylogiannopoulos, Panagiotis Karampelas and Reda Alhajj" [1] utilized the technique of Clickstream data analysis that paved the way to the analysis problem being transformed to a Sequential mining problem. The advantage is that a variety of algorithms can be used to solve the problem. This works well when predicting most frequent items viewed or purchased. However, it cannot analyse for the least frequent items bought or viewed.

"Dheeraj Malhotra, O.P. Rishi" [5] utilized the HDFS (Hadoop Distributed File System) – Map Reduce ranking algorithmwhich proved to be efficient to use on a fixed set of e-commerce websites and pages. However, the lack of real-time response and dynamic initialization of multiple analytic engines, as well as a high processing overhead, were challenges.

"Blanca Hernández, Julio Jiménez, M. José Martín" [8] used a Technology Acceptance Model (TAM) to research user behavior by defining two main variables: perceived ease of use (PEOU) and perceived usefulness (PU).Through these key variables they found the different types of customers that have purchased or are yet to purchase an e-commerce product. However, since it analyses buying behavior without defining the type of product being purchased, this method is obsolete.

"Shahab SaquibSohail, Jamshed Siddiqui, Rashid Ali" [12] discussed and compared the various recommendation techniques (collaborative filtering, content-based filtering, hybrid filtering) and explained the phenomena of recommendation from a native user's point of view. They also came to the conclusion that overcoming the limitations of current methods would take a lot of time.

"Silvana Aciar, Debbie Zhang, Simeon Simoff, and John Debenham" [13] propose "Domain Ontology", "Opinion Quality (OQ)", "Function Quality (FQ)", "Overall Feature Quality (OFQ)", and "Overall Assessment (OA)" parameters as a ranking mechanism for prioritizing product quality based on the level of expertise of the customer and the rating provided to certain product features. However, the shortfall of this approach was that the mapping of review comments into ontologies was not automated.

A Collaborative Filtering-based recommender scheme was used by *"M Viswa Murali, T G*

Vishnu, Nancy Victor" [16].The cosine similarity technique was used to make the similarity measurements between the consumer and research paper profile more accurate.With the aid of a prediction rating mechanism, the consumer is presented with the top-rated research papers based on his interests. However, the framework could be made more successful if research paper database platforms could include real datasets with scores.

III. PROPOSED WORK

We propose to build a recommender system using an item-item content-based filtering approach where the system will be trainedon a PC components dataset that will be obtained from data mining online PC component selling websites. This dataset would contain various information about products such as Name, Price Range, Category, Specifications, Ratings.

A. ARCHITECTURE FLOWCHART AND MODULES

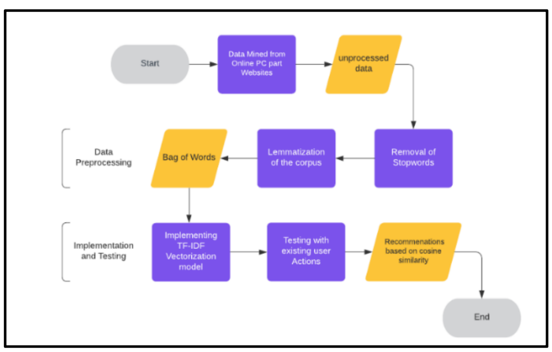

Fig. 1. Architecture Diagram

B. Data Mined from Online PC PartWebsites

First, the data will be mined from various online PC components e-commerce platforms which will provide us with key points of information such as the Name, SKU, Price Range, Category, Specifications, Ratings.

This will be achieved using the BeautifulSoup library in Python that will allow us to extract the information we require seamlessly and without much manual effort thereby essentially automating the entire process.

This information will be stored in a collection in a MongoDB database. A collection is like a table except that it can be unstructured owing to the BSON document format. An index will be generated to ensure optimum performance during query executions.We further use Pandas library in Python to generate a dataset from our collection so that it can be used to train the recommender system.

C. Removal of Stopwords

The most common, and words that might not add any meaning to an English sentence are referred to as stopwords. These words are filtered out for natural processing of the data. Examples of such words include, "their", "herself", "it", "if" etc.This method is performed in this project at the time of data preprocessing, here the data refers to the dataset generated from the Pandas library using thecollection of data obtained from online PC Component E-Commerce platforms. All stop-words are discarded from the obtained dataset and the resultant is then lemmatized.

D. Lemmatization of Corpus

Some words are of different modulated forms and but have the same meaning, this can cause many results in the final bag of words but with duplicates. To process these duplicated words as a single entity, a process called lemmatization is performed on thedataset obtained after the removal of stop-words. This way all words that meanthe same but are of different forms are linked to a single word across the entire dataset making it easy to sort the data accordingly.

E. Implementing TF-IDF Vectorization Model

Using these data pre-processing steps, we will obtain a bag of words within the dataset on which the implementation will be performed to assign importance of keywords depending on the frequency of its occurrence within the dataset.

"Term Frequency (TF)" is calculated by "dividing the number of times a word appears in a document by the total number of words in the document." Each and every document has its own term

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

frequency.

Fig. 2. Term Frequency Formula

The "Inverse Document Frequency (IDF)" is "the log of the number of terms or documents divided by the number of terms or documents that contain a name, w."Inverse Document Frequency is used to calculate the weight of all the individual words across all terms or documents in a corpus of terms or words.

$$idf(w) = log(\frac{N}{df_t})$$

Fig. 3. Inverse Document Frequency Formula

The TF-IDF algorithm is used to weigh a keyword in any of the documents and assign a value to it depending on how many times it appears in the text. It essentially means that the higher the TF-IDF score, the more uncommon and valuable the keyword is.

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF
Term x within document y

$tf_{x,y}$ = frequency of x in y
df_x = number of documents containing x
N = total number of documents

Fig. 4. TF-IDF Formula

A TF and an IDF score will be assigned to each keyword or phrase. The TF-IDF weight of a keyword or term is the product of the TF and IDF scores of that keyword or term. A matrix will be generated that contains each word or term and its TF-IDF score in relation to each document, or item in this case.Any word or object in the Vector Space Model is stored in an n-dimensional space as a vector of its attributes (which are also called vectors), with the angles between the given vectors calculated to determine the degree of similarity between the aforementioned vectors.

The procedure of calculation of a user's interest towards a particular product is calculated by using the cosine of the angle between any two document or item vectors. We utilize the cosine value of the angle as the angle between the vectors decreases, the similarity signified with increase.

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Fig. 5. Cosine Similarity Formula

F. Testing with User Actions

To properly evaluate the performance of the recommender system, we will have to test with real-time analytical inference data from Google Analytics which is regularly occurring and capturing all important key-parameters about a user.We will be capturing the user's previous history as to which kind of products they usually like viewing or add to their Wishlist. Feeding this product information to the recommender system, the model should predict 5-10 products that would be similar to the user's interests.

IV. IMPLEMENTATION

A. COLLECTION OF DATA

The data for this project will be obtained from various popular PC Parts selling websites in India such as PrimeABGB, Vedant Computers and MD Computers. This data is collected using a custom scraping utility built in Python. This utility additionally cleans the extracted data before storing it into a collection in the Mongo NoSQL Database.The data is then exported from the MongoDB collection storing the product information. A utility process filters the required features from the collection and saves it as a JSON file.

This dataset will contain a total of 6 features or columns, namely [Name, SKU, Category, Brand, Price, Description]. The entire dataset of all the PC Parts from the websites is of 4,438 rows containing a wide range of products of different and all categories available on the website such as, RAM, Processors, Graphic Cards etc. The dataset is generated via a custom scraping tool to ensure finer control on the pre-processing and analytics of the product data.

	sku	brand	category	name	description	best_price_int
0	i3-9100F	intel	CPU	Intel Core i3-9100F Processor (6M Cache, up to...	Intel Core i39100F Processor6M Cache up to 420...	6650
1	YD3200C5FHBOX	amd	CPU	AMD Ryzen 3 3200G With Radeon RX Vega 8 Graphi...	AMD RYZEN 3 3200GAMDRyzen 3000Series Processor...	9200
2	i7-10700K	intel	CPU	Intel 10th Gen Comet Lake Core i7-10700K Proce...	INTEL CORE I710700K PROCESSOR 16MCACHE UP TO 5...	30450
3	Ryzen 3 3100	amd	CPU	AMD Ryzen 3 3100 3rd Gen Quad-Core Processor	AMD Ryzen 3 3100 3rd Gen QuadCore ProcessorOve...	9050
4	100-100000031BOX	amd	CPU	AMD Ryzen 5 3600 (6 Cores 12 Threads With Max ...	AMD RYZEN 5 3600AMDRyzen 3000SeriesProcessorsA...	17200
...	...	...	...	...	...	...
4433	Z590 A PRO	msi	MB	MSI Z590 A PRO Intel Z590 LGA 1200 Intel Mothe...	No description available	18160
4434	MZ-77E250BW	samsung	SSD	Samsung 870 EVO 250GB Internal SSD (MZ-77E250BW)	870 EVO SATA III 635cm 25 SSD 250GBMZ77E250BWS...	3450
4435	NX410-WHITE	antec	CASES	Antec NX410 Cabinet (White)	Item specificsSERIESNX SERIESTYPEMID TOWER35 8...	4915
4436	N166TK-06D6	inno3d	GPU	INNO3D GeForce GTX 1660 Ti Twin X2 6GB GDDR6	INNO3D GeForce GTX 1660 Ti Twin X2N166TK06D6 S...	37500
4437	VSK10	antec	CASES	Antec VSK10 Cabinet (Black)	Item specificsSERIESVALUE SOLUTION SERIESTYPEM...	3085

4438 rows × 6 columns

Fig. 6. Collected Data

B. DATA PREPROCESSING

The data collated in the dataset, generated in the previous section cannot be used as is, and needs to be cleaned and processed before it is used by the recommender system. To do so, the dataset will undergo the processes of Noise Removal, Tokenization, Normalization and Lemmatization.

The process of Noise Removal takes place before the Tokenization. Under this process text file headers, footers, HTML and/or XML markup and metadata are removed, and the contractions are replaced with their expansions. Now we are left with words and/or sentences without the noise and unnecessary data that could cause unnecessary bias in the recommendation model.

Following Noise Removal, the process of Tokenization is performed on the data that is produced after Noise Removal of the Dataset generated. Larger text strings are broken down into smaller tokens in this method. Larger sections of text can be tokenized into sentences, and sentences into words, and so on. After a piece of text has been properly tokenized, the processing continues to the next stage.

In the process of Normalization, all the text is converted to the same case. All kinds of punctuation that can hinder the models process is also removed and numbers are converted to their word equivalents.

Documents use various forms of a word, such as arrange, arranges and arranging, for grammatical purposes. There are also word families with common meanings that are related. In certain cases, searching for one of these terms will return results that contain another word from the list. Theend aim is to reduce a word's inflectional and derivationally related forms to a simple base form. This is achieved by Lemmatization.Keywords are generated for each product by identifying the unique words from the description while omitting stop words. Stop words are words that are commonly used in the English language and do not lend much meaning to the sentence. They can hence be ignored whilst not sacrificing meaning in the sentence.

Upon completion of this data preprocessing, a word representation is created by combining column attributes to form a bag of words. In this case the columns: Brand, Category and Keywords werecombined. White spaces from the front and back and multiple whitespaces in between are also removed,if any.

```
df[['sku','brand','category','name','bag_of_words']]
```

	sku	brand	category	name	bag_of_words
0	I3-9100F	[intel]	[cpu]	Intel Core I3-9100F Processor (6M Cache, up to 4.20 GHz)	intel cpu intel64 yes instruction set 64bit extensions i...
1	YD3200C5FHBOX	[amd]	[cpu]	AMD Ryzen 3 3200G With Radeon RX Vega 8 Graphics (4 Core...	amd cpu advanced desktopprocessors amd ryzen 3 3200gamdr...
2	I7-10700K	[intel]	[cpu]	Intel 10th Gen Comet Lake Core I7-10700K Processor 16M C...	intel cpu fastest4 extended page tableseptintel 64intel ...
3	Ryzen 3 3100	[amd]	[cpu]	AMD Ryzen 3 3100 3rd Gen Quad-Core Processor	amd cpu ever beforeamd ryzen master utilitythe simple 39...
4	100-100000031BOX	[amd]	[cpu]	AMD Ryzen 5 3600 (6 Cores 12 Threads With Max Boost Cloc...	amd cpu changewithout notice seenote price feature separ...
...	...	...	...	...	...
4433	Z590 A PRO	[msi]	[mb]	MSI Z590 A PRO Intel Z590 LGA 1200 Intel Motherboard	msi mb description available
4434	MZ-77E250BW	[samsung]	[ssd]	Samsung 870 EVO 250GB Internal SSD (MZ-77E250BW)	samsung ssd 560 mbs sequential read performance may vary...
4435	NX410-WHITE	[antec]	[cases]	Antec NX410 Cabinet (White)	antec cases item specificsseriesnx seriestypemid tower35...
4436	N166TK-06D6	[inno3d]	[gpu]	INNO3D GeForce GTX 1660 Ti Twin X2 6GB GDDR6	inno3d gpu specificationsare subject virtuallink nagpu e...
4437	VSK10	[antec]	[cases]	Antec VSK10 Cabinet (Black)	antec cases cabinets brings sturdy design 55mm thickness...

4438 rows × 5 columns

Fig. 7. Generated Bag of Words

C. MODEL IMPLEMENTATION

The model we develop can read and compare two vectors (matrices). Using a TF-IDF Vectorizer, we transform the bag of words into a vector representation to achieve this model. “Term Frequency (TF)” is calculated by “dividing the number of times a word appears in a document by the total number of words in the document.”

The “Inverse Document Frequency (IDF)” is “the log of the number of terms or documents divided by the number of terms or documents that contain a name, w.

This TF-IDF Vectorizer generates the TF-IDF score of each word in the bag of words. The higher this score, the more relevant that word is amongst the entire dataset.”Everyproduct has 79305 number of features/words.

After generating the TF-IDF score we receive a matrix with the dimensions 4438x79305.To find the similarity between products we use theconcept of Cosine Similarity. This measure is thecosine of the angle between them. It is calculated with the dot product between two vectors. In this case that will be the dot product between the TF-IDF score matrix.The higher the value, the more similar the two products will be.

```
[[1.         0.0240977  0.12839379 ... 0.00955515 0.02310445 0.00574856]
 [0.0240977  1.         0.01717076 ... 0.0055486  0.01127021 0.01153811]
 [0.12839379 0.01717076 1.         ... 0.00394967 0.00875065 0.00255025]
 ...
 [0.00955515 0.0055486  0.00394967 ... 1.         0.02058192 0.11675123]
```

```
res = recommend('B450M-HDV') # Name: ASRock B450M-HDV AM4 AMD Promontory Micro ATX AMD Motherboard
res1 = res[['sku','name','brand','category','best_price_int']]
```

	sku	name	brand	category	best_price_int
120	B450 PRO4	ASRock B450 PRO4 AM4 AMD Promontory ATX AMD Motherboard	[asrock]	[mb]	7425
344	B450M-PRO4	ASRock B450M Pro4	[asrock]	[mb]	7199
2576	B450M-HDV R4.0	Asrock B450M-HDV R4.0 AMD B450 Micro ATX Motherboard	[asrock]	[mb]	5557
4357	B365 Pro4	Asrock B365 PRO4 Intel B365 ATX Motherboard	[asrock]	[mb]	7950
4020	B365 PRO4	Asrock B365 PRO4 Intel B365 ATX Motherboard	[asrock]	[mb]	7950
203	H310CM-ITX/ac	ASRock H310CM-ITX/ac LGA 1151 (300 Series) Intel H310 SA...	[asrock]	[mb]	6950
333	B450M-AC	ASRock B450M/ac (Wi-Fi) Motherboard	[asrock]	[mb]	8099
190	H310CM-DVS	ASRock H310CM-DVS LGA 1151 (300 Series) Micro ATX Mother...	[asrock]	[mb]	3700
207	H310M-HDV	ASRock H310M-HDV Micro ATX LGA1151 Motherboard	[asrock]	[mb]	3900
3164	Fatal1ty B450 Gaming-ITX/ac	ASRock Fatal1ty B450 Gaming-ITX/ac Mini ITX AM4 Motherboard	[asrock]	[mb]	9850

Fig. 8. Cosine Similarity

Fig. 9. Recommendations for SKU B450M-HDV

V. CONCLUSION

We generated a dataset of PC components as our training data for the recommender system. The description feature of thedataset was pre-processed by undergoing noise removal, further tokenization, normalization and finally lemmatization. We created a bag of words for each product by merging the category, brand and description features together.

To create the recommender system, we utilized a “TF-IDF Vectorizer”. To identify the closest products from our dataset, we use “Cosine Similarity”. The bigger the value of the cosine, the closer the two products will be.

Upon initial review, the recommendations of the system are very close to the products that wereprovided as input.The recommendation system is able to provide recommendations based on the bag of words that is generated from the brand, category and descriptions of the products.

VI. REFERENCES

[1]. “*Konstantinos F. Xylogiannopoulos, Panagiotis Karampelas, and Reda Alhajj”*, “Simplifying E-Commerce Analytics by Discovering Hidden Knowledge in Big Data

Clickstreams, Putting Social Media and Networking Data in Practice for Education, Planning, Prediction and Recommendation" (pp.51-74), Issue-2020

[2]. "*Gautam Pal, Gangmin Li, Katie Atkinson*". "Big Data Real-Time Clickstream Data Ingestion Paradigm for E-Commerce Analytics, IEEE 4th International Conference for Convergence in Technology", 2018.

[3]. "*BorianaRukanova, Yao-Hua Tan, Micha Slegt, Marcel Molenhuis and Ben van Rijnsoever*". "Value of Big Data Analytics for Customs Supervision in e-Commerce", Electronic Government (pp.288-300), Issue-2019.

[4]. "*Pei-Ju Wu, Kun-Chen Lin*", "Unstructured big data analytics for retrieving e-commerce logistics knowledge, Telematics and Informatics (2017),doi:https://doi.org/10.1016/j.tele.2017.11.004, 2017"

[5]. "*Dheeraj Malhotra, O.P. Rishi*", "An intelligent approach to design of E-Commerce and ranking system using next-generation big data analytics. Journal of King Saud University - Computer and Information Sciences, Issue-2018"

[6]. "*Olufunke Rebecca Vincent, Ayodeji Samuel Makinde, AdioTaofikAkinwale*", "A Cognitive Buying Decision-Making Process in B2B E-Commerce Using Analytic-MLP, Electronic Commerce Research and Applications" (pp.59-69), Issue-2017

[7]. "*Maria del Mar Roldan, Jose Garcia-Nieto, Jose F. Aldana-Montes*", "An Ontology-Based Data Integration Approach for Web Analytics in E-Commerce", Expert Systems with Applications (pp.20-34), Issue-2016

[8]. "*Blanca Hernández, Julio Jiménez, M. José Martín*", "Customer behavior in electronic commerce: The moderating effect of e-purchasing experience", Journal of Business Research (pp.964-971), Issue-2010

[9]. "*Sungjoo Lee, Seunghoon Lee, Yongtae Park*", "A prediction model for success of services in e-commerce using decision tree: E-customer's attitude towards online service", Expert Systems with Applications (pp.572-581), Issue-2007

[10]. "*Sahana Raj G, Dr. B Satish Babu*", "Converting an E-commerce Prospect into a Customer using Streaming Analytics, 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)", Issue-2016

[11]. "*Monika R. Astonkar, Prof. Amar Buchade*", "Exploring Mobile Analytics for Business Intelligence", IJCSMC, Vol. 3, Issue. 12, December 2014

[12].*"Sohail, Shahab &siddiqui, jamshed& Ali, Rashid"*. (2012). "Product Recommendation Techniques for Ecommerce - past, present and future."ijarcet. 1. 219-225.

[13]. "*Aciar, Silvana & Zhang, Debbie &Simoff, Simeon & Debenham, John*". (2006). "Recommender System Based on Consumer Product Reviews." 10.1109/WI.2006.144.

[14]. "*G. Linden, B. Smith and J. York*", "Amazon.com recommendations: item-to-item collaborative filtering," in IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan.-Feb. 2003, doi: 10.1109/MIC.2003.1167344.

[15]. "*Schafer, Ben &Konstan, Joseph &Riedl, John. (1999)*"."Recommender Systems in E-Commerce". 1st ACM Conference on Electronic Commerce, Denver, Colorado, United States. 10.1145/336992.337035.

[16]. "*M. V. Murali, T. G. Vishnu and N. Victor*", "A Collaborative Filtering based Recommender System for Suggesting New Trends in Any Domain of Research," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 550-553, doi: 10.1109/ICACCS.2019.8728409.

[17]. “*A. Elkhelifi, F. B. Kharrat and R. Faiz”*, "Recommendation Systems Based on Online User's Action," 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, 2015, pp. 485-490, doi:10.1109/CIT/IUCC/DASC/PICOM.2015.69.

[18]. “*H. Zarzour, Z. A. Sharif, M. A. Ayyoub, and Y. Jararweh”*, “A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques, in Proceedings of the 2018 9th International Conference on Information and Communication Systems (ICICS)”, pp. 102–106, Lille, France, April 2018.

[19]. “*H. Zarzour, F. M. Mohamed, S. Chaouki, and C. Chemam”*, “An improved collaborative filtering recommendation algorithm for big data,” Computational Intelligence and Its Applications, pp. 660–668, 2018.

[20]. “*Lemire, Daniel & Maclachlan, Anna”*. (2007). “Slope One Predictors for Online Rating-Based Collaborative Filtering”. Proceedings of the 2005 SIAM International Conference on Data Mining, SDM 2005. 5. 10.1137/1.9781611972757.43.

[21]. “*Xiaofeng Li, Dong Li”*, "An Improved Collaborative Filtering Recommendation Algorithm and Recommendation Strategy", *Mobile Information Systems*, vol. 2019, Article ID 3560968, 11 pages, 2019. https://doi.org/10.1155/2019/3560968.