International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 06 | June -2017 www.irjet.net p-ISSN: 2395-0072

Association Rule based Recommendation system using Big Data

Sumanth Reddy M[1], Ranjith[1], Divya K V[2]

[1]Student, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India
[2]Assistant Professor, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India

---***---

ABSTRACT: Nowadays, providing recommendation to users plays an important role in retaining users and improving the individual shopping experience. The recommendation system has to analyse a large amount of transaction data to provide recommendation and this can be addressed using Hadoop ecosystem. In this paper, a recommendation system on Hadoop framework is proposed, which is scalable and also solves the cold start problem. The proposed system recommends products to the user depending on the products present in the user's cart.

Keywords: Recommendation system, Apriori, Frequent itemsets, Association rule

I. INTRODUCTION

Recommender systems are a broad area of research. It is implemented in a large number of applications to provide various kinds of recommendations. The efficiency of the product recommendations is very important in today's world. In many ways, the profits of a large e-commerce company can rise and fall on the efficacy of their product recommendation algorithms, which is why such companies often put much of their time and money into these algorithms.[1] Hence, the algorithms used to process the continuously growing data should support scalability in large amounts.

The large volumes of data can be processed with the help of various cloud computing platforms which facilitates parallel computing.[2] Hadoop, MapReduce, Mahout, etc are some of the available cloud computing tools. Hadoop is a well-known open source cloud computing platform, which supports storage for large volumes of data with satisfactory fault tolerance. The MapReduce program runs on Hadoop, and it is a parallel programming model for writing distributed applications for efficient processing of large amounts of data on large clusters of commodity hardware in a reliable, fault-tolerant manner.

Our proposed system aims at implementing a product recommendation system, which appropriately recommends products to the active user. When a user logs in through an ecommerce web portal and begins shopping, depending on the products present in the cart, the system analyses and recommends the products accordingly. To make our system scalable, we use the Apriori algorithm implemented as a MapReduce program for processing the Big Data on HDFS. We also show that the proposed recommendation system solves the cold-start problem, which is a drawback of all the existing recommendation systems mentioned above.

II. METHODS AND MATERIAL

A. Related Work

In this section, we briefly overview some relevant work in recommendation systems. There are five major recommendation algorithms that are commonly used: content-based recommendation, collaborative-filtering recommendation, association rule-based recommendation, utility-based recommendation and knowledge-based recommendation. In content based recommendation system a user profile is generated first. This profile is built based on the information provided by the user. Content-based recommendation system recommends the item based on what the user had previously purchased or liked.[6] Collaborative-filtering recommendation system gives the recommendation to the people based on the opinion of the other people who have the same type of interest. Here, the customer gets the recommendations based on the items that are rated by his/her neighbourhood. Knowledge based recommender system is case based. In this similarity function estimates how much the user's needs matches the recommendations.[8] Context aware recommendation system uses the context information such as time, temperature, personal history etc. [7] For example a travel recommender system must give recommendations to their customers based on their current location, current season, etc. Hybrid recommendation system combines one or more of the above described recommender systems. There have been a number of implementations of Apriori algorithm over the years, but, these are highly inefficient.[3]

Challenges in above mentioned recommendation systems are,[4][5]

Cold-start: Recommending items to the first time customer is difficult as he/she has never rated any items or has never purchased any items before. Since there is no enough data about the customer, the recommendations can't be made. This is called cold start problem.

Scalability: The system has large amounts of data and hence requires a large amount of resources to make efficient recommendations.

B. Problem Formulation

In this section, we formally define our research problem. First, we present an overview of the problem to be addressed by our proposed product recommendation system. Next, we define and explain the data inputs. Finally, the problem statement is presented.

1) Problem Overview: An online e-commerce website wishes to display recommendations to the active customer to increase the chances of the customer making an extra purchase during the current transaction. This requires an interaction between the active customer, the ecommerce website and the data of all the previous transactions that happened on the ecommerce website. Most often this transaction dataset of an ecommerce store is huge. Therefore, the product recommendation system should display appropriate recommendations to the active customer in a very short time and also the recommendation system should be scalable.

2) Data Inputs: In this section, we give an overview of the input data. We use an open source database that belongs to an e-commerce store. This database contains 2 tables. The products table consists of the data about all the products available and the transactions table contains the data of all the previous transactions at this ecommerce website. We extract only the columns of interest to our algorithm from the transactions table and import this data into HDFS file. The columns of interest are the transaction ids and the products that were purchased in that transaction. Snapshot of the collected data of interest is shown in Table 1 and Table 2.

Table 1. Snapshot of Product Table

ProductID	ProductName	Price
10000001	Tooth Brush	10
10000002	Bread	20
10000003	Jam	11
10000004	Sauce	10
10000005	Tooth Paste	9
10000006	Butter	15

3) Problem Statement: The objective is to design a product recommendation system where the user adds products to the cart and the recommendations are to be displayed based on these products that are currently present in the user's cart. The entire data set should be considered to generate the best possible recommendations that will increase the chances of the user purchasing an extra product and also the recommendations should be displayed with least possible delay in time.

C. Proposed Solution

In this section, we present our proposed product recommendation system.

Table 2. Snapshot of Transaction Table

TransactionID	ProductID	Quantity
1	10000001	1
1	10000005	3
2	10000001	1
2	10000002	3
2	10000005	1
3	10000005	2
3	10000001	4

Table 3. Snapshot of transformed data that is imported to HDFS file

TransactionID	Products
10000001	toothbrush, toothpaste
10000002	toothbrush, bread, toothpaste
10000003	toothbrush, toothpaste
10000004	pasta, shampoo, bath Soap
10000005	sandwich, cheese
10000006	scrub, bath soap, towel

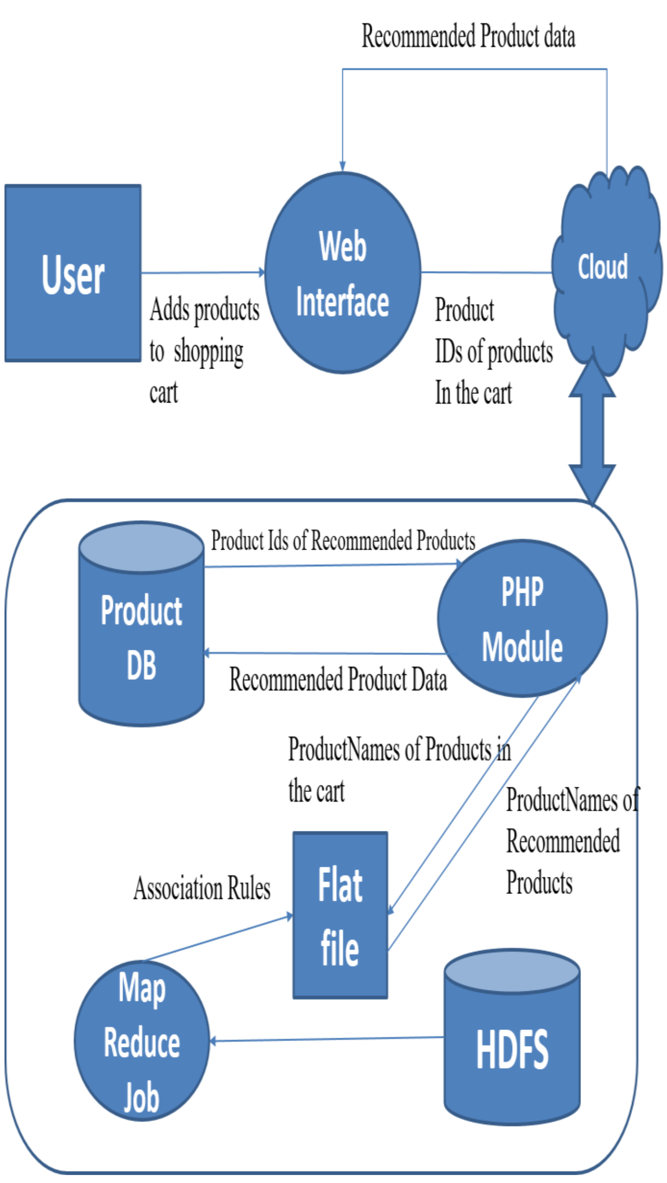

Recommended Product data
User
Adds products to shopping cart
Web Interface
Product IDs of products In the cart
Cloud
Product DB
Product Ids of Recommended Products
PHP Module
Recommended Product Data
ProductNames of Products in the cart
ProductNames of Recommended Products
Association Rules
Flat file
Map Reduce Job
HDFS

Figure 1: System Architecture

1) Association Rules Extraction: Selection, Pre-processing and transformation of the data from the database of the e-commerce store is performed and the transformed data is imported into a HDFS file.

This HDFS file is given as the input to the FrequentItemsetMapper function which is a mapper function written in Java that produces [key,value] pairs where, key is the product name or product names and value is set to 1.

After this step, shuffling of the [key, value] pairs is done thereby, rearranging the [key ,value] pairs according to the increasing order of the keys.

The output of the shuffling process is given as the input to the Frequent Item set Reducer function which is a reducer function that calculates and updates the value attribute of the [key, value] pairs based on the number of times [key, value] pairs are repeated. It further eliminates the [key, value] pairs whose value attribute is below the mins up threshold value of 0.5.

If {X,Y} is one of the frequent item sets, then, support(s) is given by,

$$s = \frac{\text{Number of transactions containing both X and Y}}{\text{Total number of transactions}}$$

Eliminate all the [key,value] pairs where $s < 0.5$

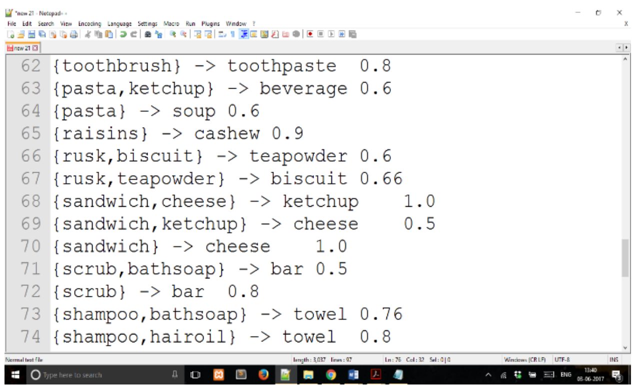
Association Rule Mapper and Association Rule Reducer functions compare the frequent item sets to the transactions and builds rules which have a confidence of more than confidence threshold of 0.5.

If $X => Y$ is one of the rules generated, then, confidence(c) is given by,

$$c = \frac{\text{Number of transactions containing both X and Y}}{\text{Number of transactions with X}}$$

Eliminate all the rules where $c < 0.5$

The output of the above process is stored in a flat file.



```
{toothbrush} -> toothpaste  0.8
{pasta,ketchup} -> beverage 0.6
{pasta} -> soup 0.6
{raisins} -> cashew 0.9
{rusk,biscuit} -> teapowder 0.6
{rusk,teapowder} -> biscuit 0.66
{sandwich,cheese} -> ketchup    1.0
{sandwich,ketchup} -> cheese    0.5
{sandwich} -> cheese    1.0
{scrub,bathsoap} -> bar 0.5
{scrub} -> bar   0.8
{shampoo,bathsoap} -> towel 0.76
{shampoo,hairoil} -> towel  0.8
```

Figure 2: Snapshot of the flat file that contains the output of the MapReduce Jobs

Figure 3 illustrates the association rules extraction on Big Data. Here, ProductIDs have been used instead of product names.

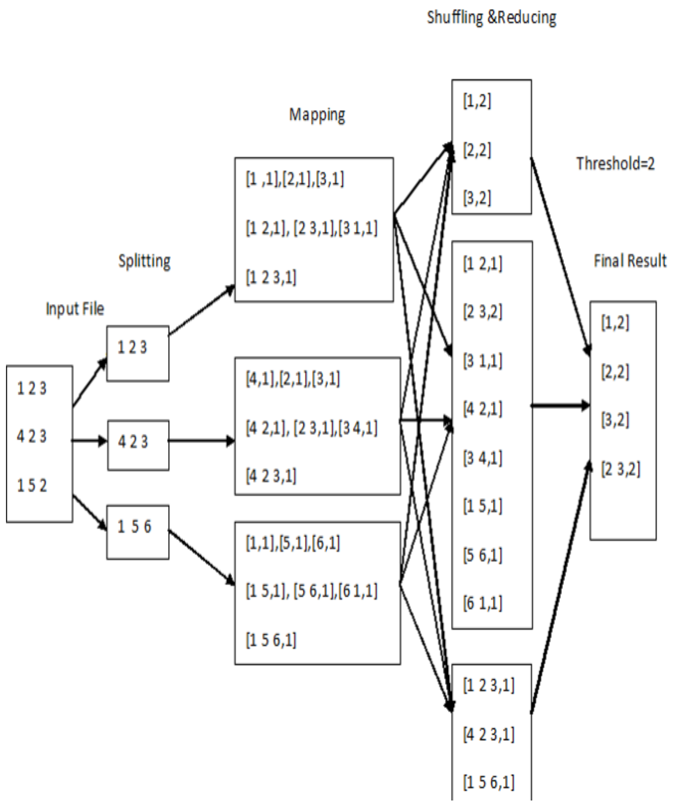


Figure 3: Diagram illustrates the Association rules extraction on Big Data

2) Executing a recommendation: The customer adds products to the shopping cart on the e-commerce website. The ProductIDs of these products in the cart are passed to the PHP module. The product names are retrieved from the Products table of the database by the PHP module by using the ProductIDs that were passed to the PHP module. The flat file where the association rules are stored, is searched for the corresponding rule whose antecedent matches the product names of the products present in the cart. If a match is found, the product names from the respective consequent part of the rule is returned. These returned product names are queried from the product database to fetch the product details of these products which are then displayed as recommendations to the active user.

3) Caching Association Rules: A Cron job that runs periodically calls a PHP function which invokes the Java MapReduce jobs that compute association rules. The output from these MapReduce jobs are the association rules which are stored in a flat file. This caching of association rules at different intervals of time rather than executing the MapReduce jobs to generate association rules every time a user adds products to the cart improves the performance of the recommendation system. Further, the MapReduce jobs can be run to generate the updated association rules when any user places an order on the e-commerce website. This ensures that all the transactions from the transactions table are considered to generate the association rules.

III. RESULTS AND DISCUSSION

In this section, we discuss the results of our proposed product recommendation system. The MapReduce jobs were run on a cluster of nodes running Cloudera's Distribution of Apache Hadoop (CDH5.3). Each cluster node is a 2.67 GHz six-core Intel(R) Xeon(R) X5650 machine with 8 GB of main memory running Ubuntu 12.04 server with the 3.5.0-23-generic kernel.

First time user of the e-commerce website registers on the registration page.

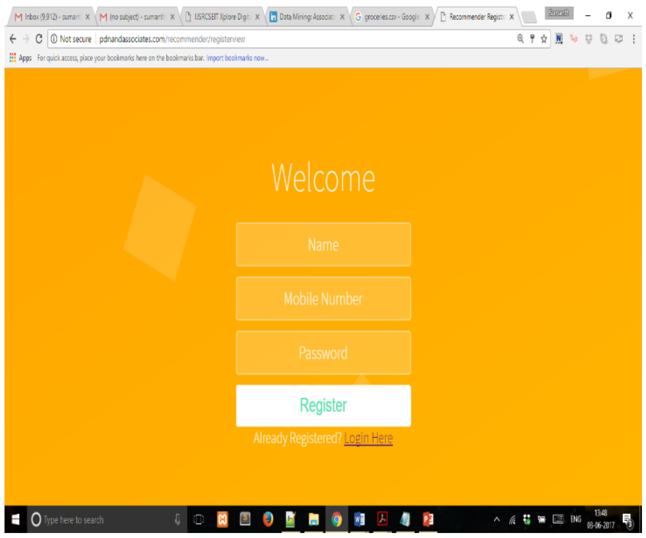


Figure 4: Screenshot of the Registration page

Users who have already registered to the e-commerce website can login directly using the login page.

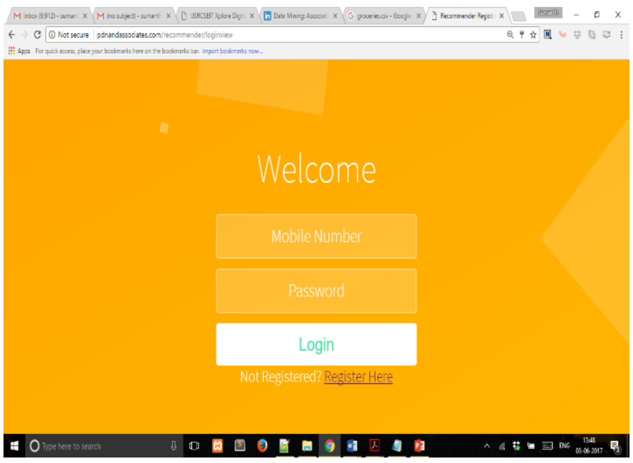


Figure 5: Screenshot of the Login page

On successful registration or login, the user is redirected to the Home Page of the e-commerce website. The page is divided into three sections. There is a section where the list of products that are available are displayed. The two other sections display a cart and the recommendations.

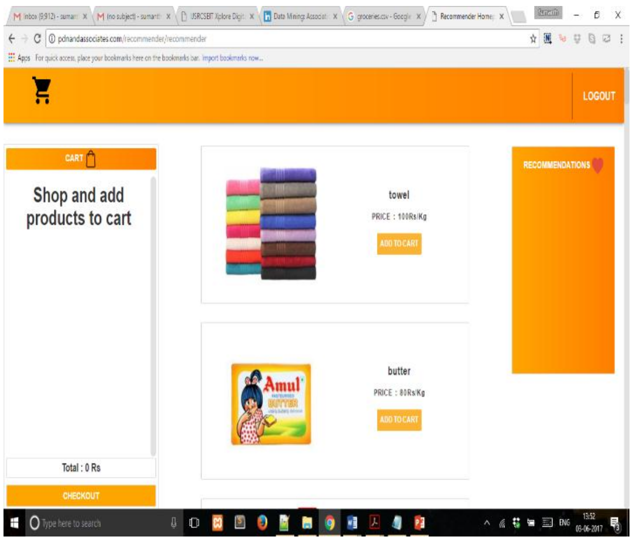


Figure 6: Screenshot of the Home page

User can shop by adding the products to the cart. When the user adds products to the cart, the recommendations are displayed in the recommendation section of the Home page. In the screenshot shown in figure 7, the user has added toothbrush to his cart. Hence, toothpaste is shown as recommendation to the user which is the appropriate recommendation as per the association rules generated using the previous transaction data.

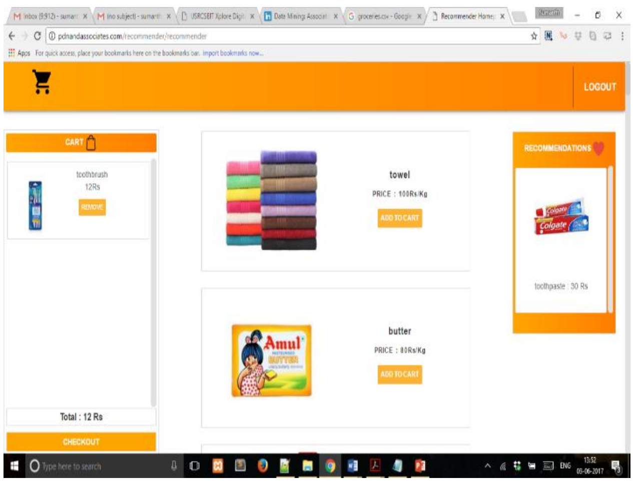


Figure 7: Screenshot of the Home page with recommendation.

When user clicks on the checkout button, he is redirected to the checkout page where the user has to enter the shipping address to complete the transaction.

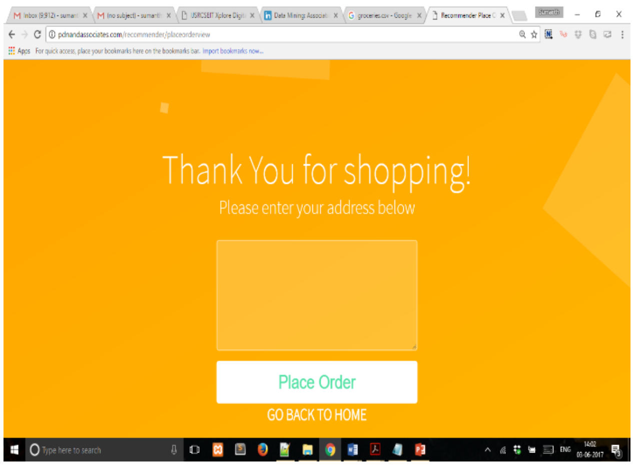


Figure 8: Screenshot of the Checkout page

After the user clicks on place order, the order is confirmed and the confirmation page displays the order confirmation message. A new record is added to the Transaction table of the database and the MapReduce Job is run to generate the updated Association rules.

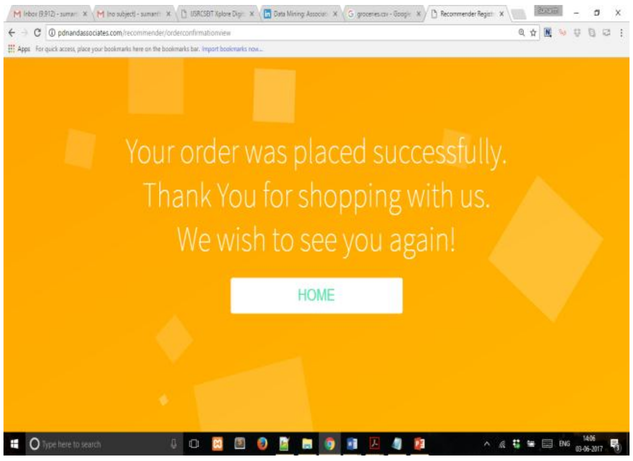

Your order was placed successfully.
Thank You for shopping with us.
We wish to see you again!
HOME

Figure 9: Screenshot of order confirmation page

We can observe from the above figures that accurate product recommendations were made even to a first time user. Therefore, the proposed recommendation system solves the cold-start problem. The proposed recommendation system is scalable and also performs better than the existing recommendation systems. However, it does not provide personalised recommendations as per the user's preferences. Due to this, accuracy of recommendations is low when compared to collaborative recommendation system. The proposed recommendation system can hence be combined with Collaborative recommendation system to form a Hybrid recommendation system. This will solve the cold-start problem which is a major drawback of collaborative recommendation system. The proposed recommendation system can be used to display the recommendations to the user until a model of the user's preferences is built. Once the model of user's preferences is built, Collaborative recommendation system can be used to display the recommendations.

IV. CONCLUSION

In this paper, we proposed a product recommendation system on Big Data. We developed an ecommerce website for shopping where a user could add products to his cart and recommendations were displayed to the user by the PHP module. To improve the scalability, we have implemented it on a MapReduce framework in Hadoop platform. As a future work, the same concept can be applied for different types of real-world applications where recommender systems are necessary. Also, the proposed recommender system can be combined with the other recommender systems to solve the cold start problem.

V. REFERENCES

[1] Junnan Chen, Courtney Miller, Gaby G. Dagher, "Product Recommendation system for small online retailers using association rules mining", IEEE International conference on Innovative design and manufacturing, August 13-15 2014, Montreal, Quebec, Canada.

[2] Baralis, Elena; Cagliero, Luca; Garza, Paolo; Grimaudo, Luigi (2015). "PaWI: Parallel Weighted Itemset Mining by means of MapReduce", 2015 IEEE International Congress on Big Data, New York (USA), 26-30 giugno 2015. pp. 25-32

[3] Kiran chavan, S.N Patil, "Frequent Itemset Mining for Big Data", IEEE 2015

[4] Mukta Kohar and Chhavi Rana, "Survey Paper on Recommendation System", International Journal of Computer Science and Information Technologies (IJCSIT), Vol.3(2), pp.3460-3462, 2012.

[5] Atisha Sachan and Vineet Richariya, "A Survey on Recommendation System Based on Collaborative Filtering Technique", International Journal of Innovations in Engineering and Technology(IJIET), Vol.2, Issue 2, pp.8-14, 2013.

[6] Shuvayan Das, "Beginners Guide To Learn About Content Based Recommender Engines", August 2015.

[7] Ebunoluwa Ashley-Dejo, Seleman Ngwira, and Tranos Zuva, "A Survey of Context-Aware Recommendation System and Services", IEEE, 2015

[8] In-Gook Chun and In-Sik Hong, "The Implementation of Knowledge Based Recommender System for Electronic Commerce Using Java Expert Library System", IEEE, 2001.