i-Recommender For e-Commerce: Hybrid Product Recommendation Using Microblogging and In House Information

M.S.D Dharmawardhana, W.W.G.B.P Bandara, K.M.S Bandarage and A.G.D Udayanga,
Ms D.Wijendara
Sri Lanka Institute of Information Technology, Colombo, Sri Lanka.
it14015472@my.sliit.lk

Abstract— With the evolution of Internet technology people have the trend of using internet services to get their ornaments. And the boundaries between e-commerce and social networking have become increasingly blurred in recent years. In this paper, we proposed a hybrid solution for product recommendation, which aims to recommend products from e-commerce websites to users of microblogging sites in "cold-start" situations, a problem which has rarely been explored before. The major challenges are how to extract required knowledge from the microblogging sites for cold-start product recommendation, how to do the fast data processing for generate required predictive machine learning models for collaborative product recommendation and how to combine the both collaborative and content base approaches in to one recommendation.iRecommender uses natural language processing techniques for extract the required knowledge from microblogging sites, and distributed clustered computing and graph data bases for efficient data processing. Throughout our research project iRecommender, we tried to achieve the best, efficient and highly accurate recommendation engine.

Keywords—cold-start problem; hybrid recommendation; collaborative filtering; e commerce; contet based filtering; ALS algorithm;

I. INTRODUCTION

With the evolution of Internet technology people have the trend of using internet services to get their ornaments. When it comes to the marketing, most of the market places are converted as ecommerce platforms where goods and services are offered through the internet. Even though there are number of ecommerce recommendation platforms available, the owners are struggling on the personalized product suggestions for their specific customers.

In this paper, we studied an interesting problem of recommending products from e-commerce websites to users of microblogging sites who do not have historical purchase records, i.e., in "cold-start" situations. And here we use hybrid approach to make predictions because of most studies only focus on constructing solutions within certain e-commerce websites and mainly utilized users' historical transaction records.

In our study the major challenges are how to extract required knowledge from the microblogging sites for cold-start product recommendation, how to do the fast data processing for generate required predictive machine learning models for collaborative product recommendation and how to combine the both recommendations approaches in to one recommendation. iRecommender uses natural language processing for extract the required knowledge from microblogging sites, and distributed clustered computing and graph data bases for efficient data processing.

In our problem setting here, the users' social microblogging information is available at the initial logging session of the user. Then after, time goes the system will have the user's historical data set. And it is a challenging task to transform the microblogging natural language information into latent user features which can be effectively used for product recommendation. To address this challenges, we use sentiment analysis and opinion mining concepts via using python NLTK toolkit. And for customized and fast predictive model building for every users of ecommerce we use distributed and clustered computing concepts by using Spark engine, for combine both collaborative and content based recommendations we use neo4j graph database. Apart from that many software engineering principles and information technology knowledge areas are applied to this project. Especially web development skills, data mining and machine learning techniques are applied.

II. OBJECTIVES

Our main target is to find a solution for the cold start problem and make a hybrid recommendation system for ecommerce sites. Then customers of ecommerce sites not need to have technical knowledge to find their ornaments. And our study will help ecommerce site owners to gain their income by providing consumer targeted product suggestions.

III. SYSTEM OVERVIEW

This section describes the system architecture and the design of the proposed system as shown in figure 1.

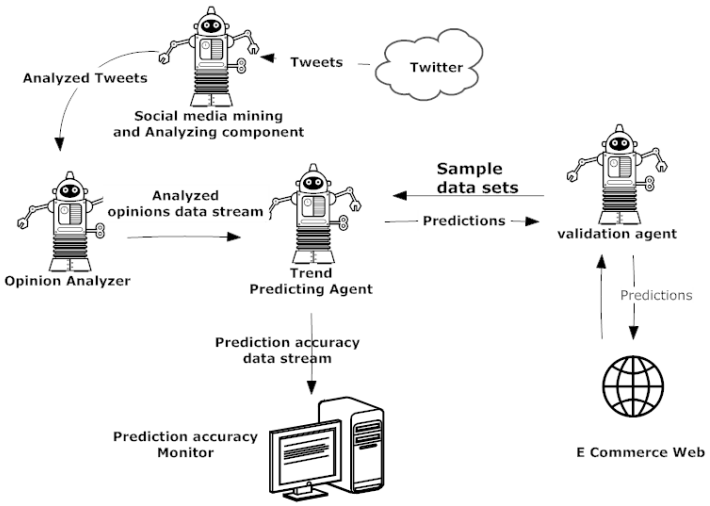

Figure 1High level view of the system

A. Social Media Mining And Analyzing Agent

This System takes texts(tweets) from twitter using twitter public API and do the following data processing tasks

- Remove noise words of tweets
- Identify the products name if customer have mentioned web site's selling products.
- Identify the emotional words which can get customer's opinion about that products.
- Finally Pass the Data to user opinion predicting agent.

B. User Opinion Predicting Agent

This system takes output data of social media mining agent and converts the raw data in to negative and positive opinions for a given product.

C. Trend Predicting Agent

This system is developed for make the personalized product recommendations. This system uses output data of opinion predicting agent to do the content based filtering and use User ratings on products to do the collaborative filtering. By combining those filtered data we make the hybrid recommendation.

D. Validation Agent And Customer Behavior Analyzer

Customer behavior analyzer is the component which tracks the user activities on the ecommerce website. This component is triggered at the point of login of the customer and important data which are required for the iRecommender's validation agent are collected through the actions performed by the user

IV. METHODOLOGY

A. Social Media Mining And Analyzing

In this component main target is extract useful data to predict the consumer openions. To do that system analyze customer's twitter account and retrieve tweets using Twitter API. Twitter is a social networking and microblogging service that allows user to post real time messages, called tweets. Tweets are short messages, restricted to 140 characters in length. Due to the nature of this microblogging services people use emoticons and other characters that express special meaning.

Sentence Analyzing is done using [1]Natural Learning Processing. NLTK is a leading platform for building Python programs to work with human languages. It provides easy-to-use interfaces to over 50 corpora and [2] lexical resources such as WordNet, along with a suite of text processing libraries for [2]classification, tokenization, stemming, tagging, parsing etc. It tokenizing each tweets as follow

Example: "I like wrist watch"
converted as [[('I', 'I', ['PRP']), ('like', 'like', ['VBP']),
('wrist', 'wrist', ['JJ']), ('watch', 'watch', ['NN'])]]

Text preprocessing is indeed for that, [3]Sentiment classification over Twitter is usually affected by the noisy nature (abbreviations, irregular forms) of tweets data. A popular procedure to reduce the noise of textual data is to remove noise words by using pre-compiled noise words lists or more sophisticated methods for dynamic noise words identification.

By using pre-define noise words list the system going to: -

•Remove single letters / punctuation marks and other symbols.
•Remove words with low inverse document frequency.

Then after removing noise words, it has to identify words which can get idea about customer taste. Words identify is done by using dictionaries [4]. Dictionary is the predefine words list and it tokenize as follow.

worthless: [ADJ]
worried: [ADJ]
abandoned: [ADJ]

Finally, words list and dictionaries are matching using tags and identify the meaning full words.

Tweet: I like wristwatch
Output: wristwatch like

Tweet: I don't like wristwatch
Output: wristwatch like don't

B. User Opinion Predicting

This is the one of major parts of the iRecommender System which is analyzing the user's feelings/emotions and extract the opinion as Positive, Negative or Neutral. These user feelings/emotions are extracted and stored in to a csv file by the Social media mining and analyzing agent. These data will be input to the User Opinion [5] predicting agent. Using Natural Language Processing (NLP) [1] techniques, develop a User Opinion predicting Algorithm. This Algorithm will be output the Negativity, Positivity or Neutrality. As well as, rating of that opinion will also extract. These set of data store inside a csv file according to each user. Each user has different csv file.

1. What Is An Opinion [6] To A Machine

It is a "quintuple [7]", an object made up of 5 different things as in equation 1

$$(o_j, f_{jk}, so_{ijkl}, h_i, t_l)$$

Equation 1

O_j - Target entity
F_{jk} - A feature/aspect of O_j
S_{oijkl} - The sentiment value of the opinion
H_i -Opinion holder
T_i -The time when Opinion is expressed

2. User Opinion Predictor Algorithm

This can be done by using Sentiment Analysis [8]. Basic Sentiment Analysis algorithms use Natural Language Processing (NLP) to classify words as positive, neutral, or negative. Keyword spotting is the simplest technique leveraged by sentiment analysis algorithms.

Keyword spotting is the simplest technique leveraged by sentiment analysis algorithms. Input data is scanned for obviously positive and negative words like 'happy', 'sad', 'terrible', and 'great'. [3]Algorithms vary in the way they score the words to decide whether they indicate overall positive or negative sentiment. Different algorithms have different libraries of words and phrases which they score as positive, negative, and neutral.

After getting extracted data set which are stored in .csv file, we are ready to execute the sentiment analysis algorithm on each Key words (feelings/emotions). Then, we will calculate an average score for all the Key words (feelings/emotions) separately.
Using this User Opinion Predictor Algorithm, we give the Key words as input to this Algorithm. Output of this will be the Negative, Positive or Neutral. Also. Such as Negative, Neutral and Positive. This Algorithm also give the level as 1 – 3. That means,

- 1 - Negative
- 2 - Neutral
- 3 - Positive

3. Expected Inputs and Outputs

Inputs are separated by a space (" ") and contain three main parts. First one is the product (Ex: car, flower, phone, vehicle). Second one is the feeling (emotion extracted by Social media mining and analyzing agent). Last and third one is the adjective. Sometimes there is no any adjectives. For this situation, we use a key word for indicate that. That key word is "no any adjective". (Ex: don't, not, etc.)
Sample input records are mention in below. These are read from text files (.txt). Each user has separate text files including these three part records. All input files are in .txt format.

- car nice don't
- flower beautiful not
- phone damn no any adjective
- vehicle awesome no any adjective

According to the number of input files, same no of output files are generated by this algorithm. Output files are in .csv files. Sample output records are mention in below in the figure 2.

car	nice	negative
flower	beautiful	negative
phone	damn	negative
vehicle	awesome	positive
plain	great	negative
table	helpful	positive

Figure 2 Sample Output of the system

C. Trend (Recommendations) Predicting.

This system is developed for make personalized product recommendations for specific users of the ecommerce. The system uses output of opinion predicting agent to do the content based filtering [9] [10]and use User ratings on products to do the collaborative [11] [12] filtering. By combining those filtered data make the hybrid [13] [14]recommendation. Figure 3 shows the work flow of the hybrid recommendation.

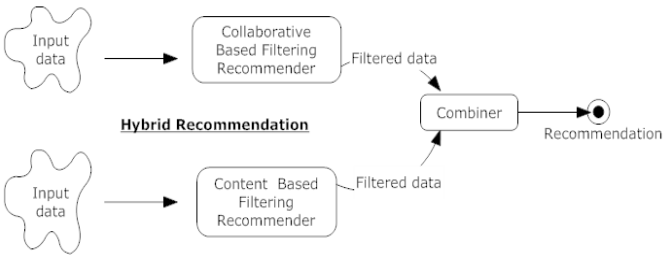

Figure 3 Hybrid recommendation overview

1. Content Based Filtering

Most of the recommendation engines use one approach to make the product recommendations the first approach is content based filtering [10] which is about description of the item and the profile of the user's preference. So in here we are using the data collected form the twitter and our main aim is to provide a solution to the "cold start" [15] problem. In [16] this paper authors discussed about to use demographic approach. we found that is a highly accurate method when it comes to content based filtering. In here we use same approach but we improved the information gathering and data processing task by using microblogging information and distributed clustered computing technologies

In here we use general Map Reduce [17] approach to analyze the collected opinions as follows

- First we get the negativity and positivity opinions of products and separate them in to two groups
- Next take the opinion positive dataset and make the grouping according to the product vise
- Next by performing counting function on those groups, filter the products which are having highest number of item counts
- Then at the very first logging of the customer above mentioned high frequency products are going to suggest.

This method will completely solve the problem of "Cold start" Problem

2. Collaborative Filtering

Next Recommendation engine approach is collaborative filtering [12] which is based on similarity of different users. So in here we need to find out similar user groups by studying their purchasing history

This task can be done by using a predictive [18] machine learning model. To build a predictive machine learning model we can use any number of historical data collected from an e commerce web site, but when it increasing the number of features, the machine learning model building time goes up. Then that will be affected to the time of prediction. Because in here we use only product ratings, which are given by the consumers on available products as historical data set. And we use highly optimized predictive algorithm for the predictive model building from user ratings

ALS - Alternating Least Square for collaborative filtering

We read that this ALS [19] formula is highly accurate and highly efficient for predictive model building using the product ratings. Other than using other formulas such as Pearson (correlation)-based [20] similarity, Cosine-based similarity [21] and Adjusted cosine [21] similarity. And most of the researchers studied about application of this algorithms in single threaded way. But in our solution we are going to use ALS algorithm in distributed and parallel way to minimize the predictive model building time and the prediction time.

Pseudo code of ALS formula shown in the figure 4

- u:- is user ID
- m:- is item ID
- f:-number of features going to consider

Algorithm 1 Pseudo-code implementation of Alternating Least Squares

$upm \leftarrow initRandomMatrix(InputMatrix.height, noLatentFeatures)$
$mcm \leftarrow initRandomMatrix(InputMatrix.width, noLatentFeatures)$
repeat
  for $u \rightarrow InputMatrix.height$ **do**
    for $m \rightarrow InputMatrix.width$ **do**
      if $InputMatrix[u][m] > 0$ **then**
        $est \leftarrow InputMatrix[u][m] - upm[u,:] \cdot mcm[:,m]$
        for $f \rightarrow noLatentFeatures$ **do**
          $upm[u][f] \leftarrow upm + \lambda(2 * est * mcm[f][m] - \alpha * upm[u][f])$
          $mcm[f][m] \leftarrow mcm + \lambda(2 * est * upm[u][f] - \alpha * mcm[f][m])$
        end for
      end if
    end for
  end for
until $RMSE \leq threshold$

Figure 4 ALS Psedo Code

3. Over Come The Scalability, Accuracy And Performance Boundaries

Recommendation engines required to process huge amount of data in order to make recommendations. Such a processing load cannot be handed by the single node of server. This research paper [22] discussed about how to gain the processing power by using clustered and distributed computing. And discussed about what are the best and accurate distributed computing systems. After analyzing huge number of factors we decided to use Apache Spark as our data processing Engine. And for all the data exchange tasks we use .CSV files to keep our system performants high and keep the high interoperability capabilities.

4. Combine Both Collaborative and Content Based Filtering Results

We found that this [23] paper discussed about graph databases for fast pattern matching and we found that this a very successful way to achieve our expected performance of our system. In order to make hybrid product recommendation we

needed to combine both collaborative and content based recommendation as a solution for this we took a graph based data base(neo4j) [24] to represent the hybrid relationships patterns.

Expected hybrid recommendation pattern show in the figure 5:

- Suggest: -This relation is generated by Predictive Model using collaborative filtering
- Likes: -This relation is generated by content base filtering algorithm

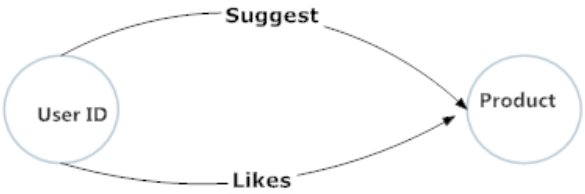

Figure 5 Expected recommendation pattern

D. Customer Behavior Analyzer

The main intension on developing the Customer Behavior analyzer and Validating agent is to gain the inputs for continuous functioning of iRecommender and to maintain the highest accuracy of the recommendations done by iRecommender. Most of the current existing recommendation engines just use the search results and the purchase history of the customers to make their suggestions. This is one of the main threat to the low accuracy of the product suggestion. To overcome this defect of existing recommendation engines, we had to introduce some new factors to the recommendation process. Customer behavior analyzer is developed to achieve the responsibility of collecting necessary information for the accurate recommendation.

Customer behavior analyzer is triggered from the point of user's registration to the ecommerce website. Once the user is registered Customer behavior analyzer collects customers Twitter ID and it is sent to the Social Media Analyzer. Social Media analyzer uses this ID to collect data from twitter feeds of the customer. After the registration, the customer is an active user on the ecommerce website. Each action customer perform is tracked by the Customer behavior analyzer. When tracking and storing the user actions, the main challenge was storing the data gathered for processing. As customer is surfing the website without a limit, lots of draft records are collected from Customer behavior analyzer. This collected information are then exposed to web mining [25] techniques to extract only the important data. As the gathered information from Customer behavior analyzer is grooving limitlessly, a better method of storing them had to be chosen. There were two types of techniques to compare when selecting a storage method.

1. Using a Database
2. Using a text based technique

When considering the above two techniques, using a database to store data is directly affecting the performance of the system. Continuously writing the collected data [26] to the database leads the system to be slower on performance and cause high traffic on the hosting. Except the performance issue, the manipulation of collected data exposing to web mining is also getting slower when a database is used to store the customer behavior analyzer's data. By considering the performance factor, option of using a database was eliminated. When a text based technique is used the process is much faster and much efficient. Once the Customer behavior analyzer is triggered, each collected data are stored in a '.csv' file under the user ID of each user. As these files are just saved on the server, a security issue is arising. To protect confidential data of customer's precautions are taken care from the host provider. Each of these collected log files from the users are then exposed to web mining techniques to extract important data for the validation agent. Figure 6 shows the code snippet used for extract the web mining data

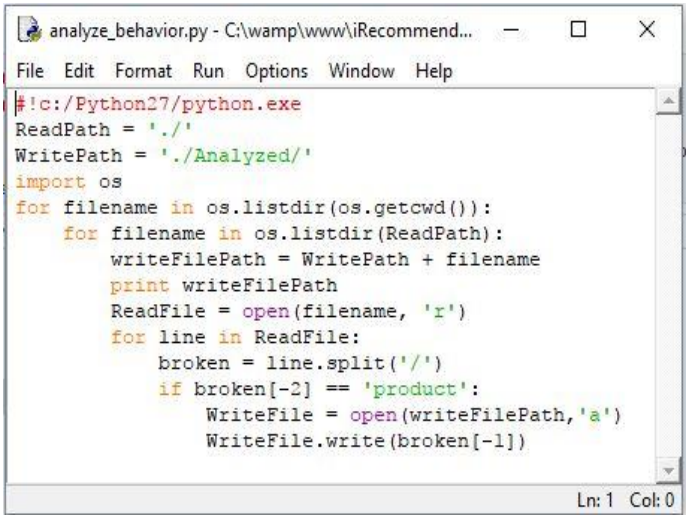

Figure 6 Web Mining Code

For each user, another '.csv' file is generated with the inputs for Validation agent.

E. Validation Agent

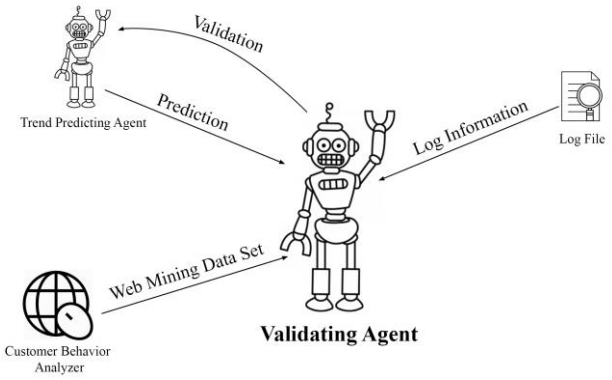

Figure 7Validation Agent processes

The main intention of developing the Validation agent is to maintain the highest accuracy of the iRecommendation suggestions. To overcome the issue of just using search results and purchase history, validation agent is improved to consider more factors on validating and suggesting. Validation agent is directly connected with the Customer behavior analyzer and the Trend predicting agent. The main output of the trend predicting agent which are the products identified from social media analyzing are taken as inputs to the Validation agent. In order to carry the validation process, the Agent uses the outputs from customer behavior analyzer as inputs for itself. By using comparison techniques, Validation agent compares the product suggestions from Trend predicting agent and Customer behavior analyzer to check whether the iRecommender suggestions are accurate. Each and every suggestion is considered as accurate if the user is purchasing an item which is suggested by the iRecommender. These accurate results are directed back to the trend predicting agent and the suggestion list is updated as suggesting a purchased item to the same customer is not efficient. Figure 7 shows the work flow of the validation agent

V.CONCLUSION

In this paper, we have studied a novel problem, product recommendation at cold start situations, i.e., recommending products from e-commerce websites for microblogging users without having historical purchase records. Our main idea is that on the how to solve the cold start problem and how to combine the both collaborative and content based filtering techniques to make the hybrid recommendations. To achieve this goal, we used NLTK toolkit for natural language processing tasks, and Apache Spark system for distributed clustered computing and neo4j for hybrid pattern matching tasks. Anyhow after completing our study we observed that when making predictions there is 1 to3 seconds delay. According to our gained knowledge though out this study that delay can be minimized by doing the Spark engine execution plan optimization we hope to do that optimization task as our future studies.

REFERENCES

[1] V. Govindasamy and H. Balaji, "Social opinion mining and concise rendition," *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pp. 641-645, 2016.

[2] S. Mandal and G. S, "A Lexicon-based text classification model to analyse and predict sentiments from online reviews," *016 International Conference on Computer, Electrical & Communication Engineering (ICCECE), Kolkata, India*, pp. 1-7, 2016.

[3] H. Cho and M. S. Yoon, "Improving sentiment classification through distinct word selection," *017 10th International Conference on Human System Interactions (HSI), Ulsan, South Korea*, pp. 202-205, 2017.

[4] T. Kawabe, Y. Yamamoto, S. Tsuruta and R. Knauf, "A dictionary-based sentiment classification method considering subject-predicate relation," *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest*, pp. 004217-004222, 2016.

[5] F. Ren and Y. Wu, "Predicting User-Topic Opinions in Twitter with Social and Topical Context," *in IEEE Transactions on Affective Computing*, Vols. vol. 4, no. 4, pp. 412-424, Oct.-Dec. 2013.

[6] S. A. A. Hridoy, "Localized twitter opinion mining using sentiment analysis," 22 Oct 2015. [Online]. Available: https://decisionanalyticsjournal.springeropen.com/articles/10.1186/s40165-015-0016-4.. [Accessed 20 Feb 2017].

[7] B. Liu, Sentiment Analysis and Subjectivity, Chicago: N. Indurkhya and F. J. Damerau, 2010.

[8] A. Goel, J. Gautam and S. Kumar, "sentiment analysis of tweets using Naive Bayes," *2016 2nd International Conference on Next Generation Computing Technologies (NGCT), Dehradun*, pp. 257-261, 2016.

[9] Weihong, He and Yi, "An E-commerce recommender system based on content-based filtering," *Wuhan University Journal of Natural Sciences*, pp. 1091-1096, 2006.

[10] P. Lops, M. d. Gemmis and G. Semeraro, "Content-based Recommender Systems: State of," *Springer Science+Business Media, LLC 2011* , p. 33, 2011.

[11] R. Sharma, D. Gopalani and Y. Meena, "Collaborative filtering-based recommender system: Approaches and research challenges," *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad,* pp. 1-6, 2017.

[12] J. Wei, J. He, K. Chen, Y. Zhou and Z. Tang, "Collaborative Filtering and Deep Learning Based," *DASC-PICom-DataCom-CyberSciTec.2016,* p. 4, 2016.

[13] A. M. Sharif and V. V. Raghavan, "Link prediction based hybrid recommendation system using user-page preference graphs," *2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK,* pp. 1147-1154, 2017.

[14] S. Liu, Y. Dong and J. Chai, "Research of personalized news recommendation system based on hybrid collaborative filtering algorithm," *016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu,* pp. 865-869, 2016.

[15] J. Yuan, W. Shalaby, M. Korayem, D. Lin, K. AlJadda and J. Luo, "Solving Cold-Start Problem in Large-scale Recommendation Engines:," *2016 IEEE,* p. 10, 2016.

[16] A. K. Pandey and D. S. Rajpoot, "Resolving Cold Start problem in recommendation system using demographic approach," *2016 International Conference on Signal Processing and Communication (ICSC), Noida,* pp. 213-218, 2016.

[17] J. Jędrzejowicz, J. Neumann, P. Synowczyk and M. Zakrzewska, "Applying Map-Reduce to imbalanced data classification," *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Gdynia, Poland,* pp. 29-33, 2017.

[18] S. Raschka, "Predictive modeling, supervised machine learning, and pattern classification," 25 Aug 2014 . [Online]. Available: http://sebastianraschka.com/Articles/2014_intro_supervised_learning.html. [Accessed 16 Feb 2017].

[19] B. W. Chen, W. S. Rho and Y. Gu, "Supervised Collaborative Filtering Based on Ridge Alternating Least Squares and Iterative Projection Pursuit," *in IEEE,* vol. 5, pp. 6600-6607, 2017.

[20] X. Su, T. M. Khoshgoftaar and R. Greiner, "Imputed Neighborhood Based Collaborative Filtering," *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, NSW,* pp. 633-639, 2008.

[21] Y. Dou, H. Yang and X. Deng, "A Survey of Collaborative Filtering Algorithms for Social Recommender Systems," *2016 12th International Conference on Semantics, Knowledge and Grids (SKG), Beijing,* pp. 40-46, 2016.

[22] A. Wijayanto and E. Winarko, "Implementation of multi-criteria collaborative filtering on cluster using Apache Spark," *016 2nd International Conference on Science and Technology-Computer (ICST), Yogyakarta,* pp. 177-181, 2016.

[23] Y. Liang and P. Zhao, "Similarity Search in Graph Databases: A Multi-Layered Indexing Approach," *2017 IEEE 33rd International Conference on Data Engineering (ICDE), San Diego, CA,* pp. 783-794, 2017.

[24] "five-signs-to-give-up-relational-database," www.neo4j.com, 27 07 2015. [Online]. Available: https://neo4j.com/blog/five-signs-to-give-up-relational-database/. [Accessed 18 03 2017].

[25] G. K. J. K. a. J. R. B. Sarwar, "Analysis of Recommendation Algorithms for E-Commerce," in *ACM (Minneapolis, Minnesota.),* 2000.

[26] V. M. a. D. N. L. J. Grace, "ANALYSIS OF WEB LOGS AND WEB USER IN WEB MINING," *International Journal of Network Security & Its Applications (IJNSA),* vol. 03, no. 1, p. 12, January 2011,.