Scientific Journal Impact Factor (SJIF): 3.518
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

PC ASSEMBLY BASED ON USER BENCHMARK

***Prof.Deepali Patil*[1]*, Arihant Chhajed*[2]*, Azhar Kanorwala*[3] *, Abitya Kondappan*[4] *, Nikhil Anand*[5]**

[1] *Assistant Professor, Computer, SLRTCE*

[2,3,4,5]*Computer, SLRTCE*

Abstract— Now days, purchasing branded Desktop PC from a single Vendor is getting expensive day by day. Also consumers are required to rely on single Vendor for their whole Desktop PC and thereby compromising on the hardware so as to reduce cost. To overcome this problem a new approach was developed i.e. to assemble the PC by purchasing its essential component from different vendors. A large number of websites allow users to post reviews about product they bought. People who find it difficult to understand the technical specification of a product often tends to read reviews. There are thousands of reviews of customers available on different e-commerce portal related to a product. So it is difficult for the customers to get an idea about the product from wide range of reviews. In this project we aim to summarize the customer's reviews in factual form using part-of-speech tagger (POS), Lexicon algorithm, and Naive Bayesian classifier which are all based on Natural language processing (NLP).We have developed a rating system which rates each computer peripherals based on its performance & cost and thereby calculating the final rating of assembled system virtually so that consumers can judge whether the assembled product meets all the requirement as expected.

Keywords- PC Assembly, Reviews, POS, Opinion Mining, Extraction, Java, Perl, XML, Products, PHP.

I. INTRODUCTION

The breakout of the e-commerce retailers has led to an increase in the reviews given by the customers for the products sold on these websites. Before buying a product a person thinks "What people think?". He will have various questions regarding the product like "Which one to buy?" etc. The answers for these questions can be found in the reviews which are posted by other buyers on the respective websites but the number of reviews found is overwhelming and difficult to analyze each and every review which are expressed in different ways by different customers. The Project 'PC Assembly based on user Benchmark helps to assemble a PC by selecting the components and provide individual rating to each peripheral and based on these the final rating is given by the system with the help of the reviews on various parameters like cost, performance, speed, throughput, etc.

II. MOTIVATION

The overwhelming demand of assembled PC and to explore the significant domain called "Opinion Mining" has motivated us to develop this project. In addition to bring the usability of wide range of data available on the web in the form of opinion and reviews into operational working model has made us to think to develop this project so as to simplify the process of assembly of PC peripheral to make the efficient Desktop Computer for the end user.

III. PROBLEM DEFINITION

The breakout of the e-commerce retailers has led to an increase in the reviews given by the customers for the products sold on these websites. Before buying a product a person thinks "What people think?". He will have various questions regarding the product like "Which one to buy?" etc. The answers for these questions can be found in the reviews which are posted by other buyers on the respective websites but the number of reviews found is overwhelming and difficult to analyse each and every reviews which are expressed in different ways by different customers. Some of the reviews are biased and fake, which may confuse the customer. In short it is difficult for a person to manually read all the reviews and decide whether the product is worth buying or not. People who find it difficult to understand the technical specification of a product often tends to read reviews.

The Project helps to assemble a PC by selecting the components and provide individual rating to each peripheral and based on these the final rating is given by the system with the help of the reviews on various parameters like cost, performance, speed, throughput, etc.

IV. EXISTING SYSTEM

Currently, people used to believe on paper specification and buy the product. Many similar websites provide similar kind of rating but based on the product specification. Actual scenario is always different i.e. until we buy that product and use it we cannot say, it is good or bad .Many times it happens that the actual performance is contrast with its specification. We believe in actual user experience hence we had used actual user's opinion to compute our rating rather than going on its paper specification.

V. PROPOSED SYSTEM

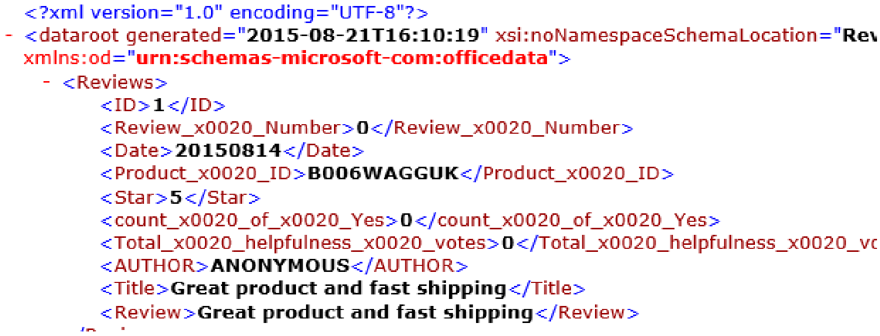
The solution infers with the extraction of customer reviews from e-commerce web sites and by using certain algorithms which falls into domain of Opinion Mining such as part-of-speech tagger (POS), Lexicon algorithm, and Naive Bayesian, etc. To compute the final rating of a system to help end users to judge about the product. It focuses on the company, the stakeholders and applications, which allow to provide appropriate consultancy to user so as to assemble his PC which is within budget yet performance efficient using smart rating computed from the reviews and opinion available on the web.

VI. Basic Concepts

Our system considers three issues while calculating product scores: 1) product reviews, 2) product popularity, and 3) product release month. Eventually, our system would prompt users to specify pc component budget in a query, and will give the results of products matching the budget. There are various techniques and approaches closely related to our work. We would briefly review the basic concepts of these techniques and approaches.

A. XML and X-Path Language

We have used XML[1] Database to store review information since xml is portable and system independent, also it's compatible with web as well as application programming language.



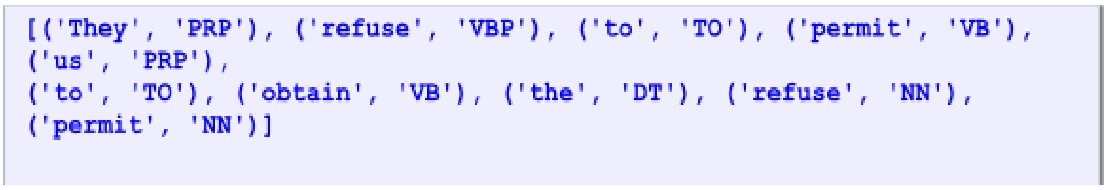
```
  <?xml version="1.0" encoding="UTF-8"?>
- <dataroot generated="2015-08-21T16:10:19" xsi:noNamespaceSchemaLocation="Rev
  xmlns:od="urn:schemas-microsoft-com:officedata">
  - <Reviews>
      <ID>1</ID>
      <Review_x0020_Number>0</Review_x0020_Number>
      <Date>20150814</Date>
      <Product_x0020_ID>B006WAGGUK</Product_x0020_ID>
      <Star>5</Star>
      <count_x0020_of_x0020_Yes>0</count_x0020_of_x0020_Yes>
      <Total_x0020_helpfulness_x0020_votes>0</Total_x0020_helpfulness_x0020_vc
      <AUTHOR>ANONYMOUS</AUTHOR>
      <Title>Great product and fast shipping</Title>
      <Review>Great product and fast shipping</Review>
```

Figure 1 XML Data

B. Part of Speech Tagging

Part of Speech (POS) [13] tagging is the problem of assigning each word its grammatical tag i.e. it assign parts of speech to each word

The POS tagging plays is a basic essential component in various Natural Language Processing (NLP) applications. Most of text analysis and data mining systems require the POS tagging, such as information extraction, information retrieval, word sense disambiguation, machine translation, and higher-level syntactic processing, etc. In the information extraction, patterns (manually defined linguistic patterns) used for extracting information from texts have references to the POS tags. For example, we might extract phrases which match the pattern "adverb + adjective" from texts.



```
[('They', 'PRP'), ('refuse', 'VBP'), ('to', 'TO'), ('permit', 'VB'),
('us', 'PRP'),
('to', 'TO'), ('obtain', 'VB'), ('the', 'DT'), ('refuse', 'NN'),
('permit', 'NN')]
```

Figure 2 POS Tagging

VII. System Framework

In this section, we propose the framework of a product ranking system as shown in Fig. 3, which consists of the following 7 components: *Downloading Product HTML Pages, Extracting Product Reviews and Information, Splitting Product Review Texts into Sentences, Identifying Sentence Polarity, Building Product XML Files, Product Ranking, and Design and Final Output*. In general, even for the same type of products (e.g. Graphic cards), users have their own different needs. Therefore, our system is designed to provide relevant product ranking according to different user needs. First, through the Internet, the system downloads data from Amazon product review pages. Then, it would extract the required product reviews and information from downloaded data, and split them into sentences. Next, it identifies the polarity of the opinion words in each sentence. Finally, the product information and sentence polarity would be integrated into an XML file. Afterwards, users can specify product features in a query, and get back product ranking results.

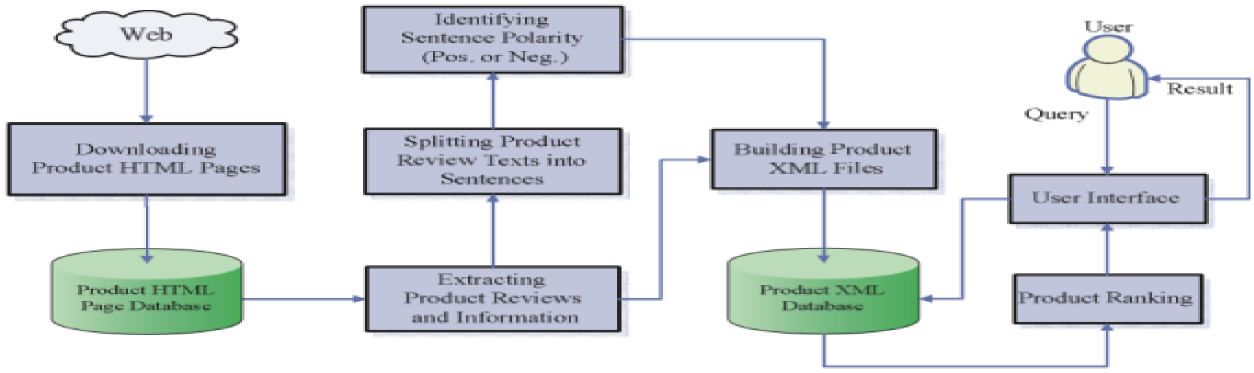

Web
Downloading Product HTML Pages
Product HTML Page Database
Identifying Sentence Polarity (Pos. or Neg.)
Splitting Product Review Texts into Sentences
Extracting Product Reviews and Information
Building Product XML Files
Product XML Database
User
Query
Result
User Interface
Product Ranking

Figure 3. System framework.
Figure 3 System Framework

A. Downloading Product HTML Pages

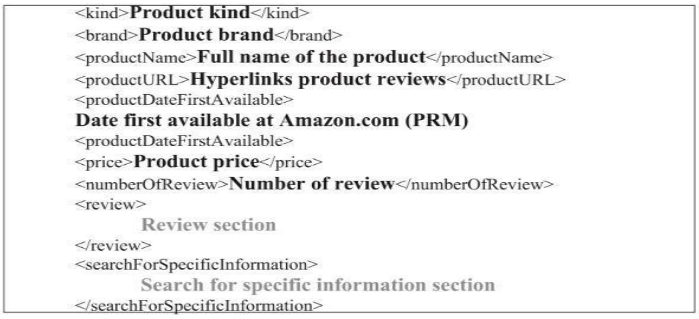
We will be extracting products reviews from Amazon.com in form of HTML Pages and using a Perl Script [12] which will be formatted as:

- A counter of the extracted reviews so far (can be used as a unique ID for the dataset).
- Date of the review in YYYYMMDD format (note: on non-English speaking domains this feature won't work, edit the script to set the name of months in the desired language).
- ID of the reviewed product.
- Star rating.
- Date of the review in human readable format (will be in the language used by the specified domain).
- ID of the author of the review.
- Title of the review

- Content of the review

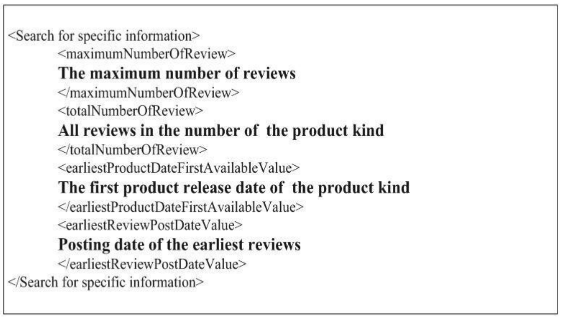
B. Building Product XML Files

In this section, the product information and sentence polarity would be integrated into an XML file. As shown in Fig. 6, this file consists of three parts: 1) product information (or features), 2) review section describing review information and sentence polarity as illustrated in Fig. 7)



```
<kind>Product kind</kind>
<brand>Product brand</brand>
<productName>Full name of the product</productName>
<productURL>Hyperlinks product reviews</productURL>
<productDateFirstAvailable>
Date first available at Amazon.com (PRM)
<productDateFirstAvailable>
<price>Product price</price>
<numberOfReview>Number of review</numberOfReview>
<review>
         Review section
</review>
<searchForSpecificInformation>
         Search for specific information section
</searchForSpecificInformation>
```

Figure 6 File Structure



```
<Search for specific information>
        <maximumNumberOfReview>
        The maximum number of reviews
        </maximumNumberOfReview>
        <totalNumberOfReview>
        All reviews in the number of  the product kind
        </totalNumberOfReview>
        <earliestProductDateFirstAvailableValue>
        The first product release date of  the product kind
        </earliestProductDateFirstAvailableValue>
        <earliestReviewPostDateValue>
        Posting date of the earliest reviews
        </earliestReviewPostDateValue>
</Search for specific information>
```

Figure 7 Specific Section

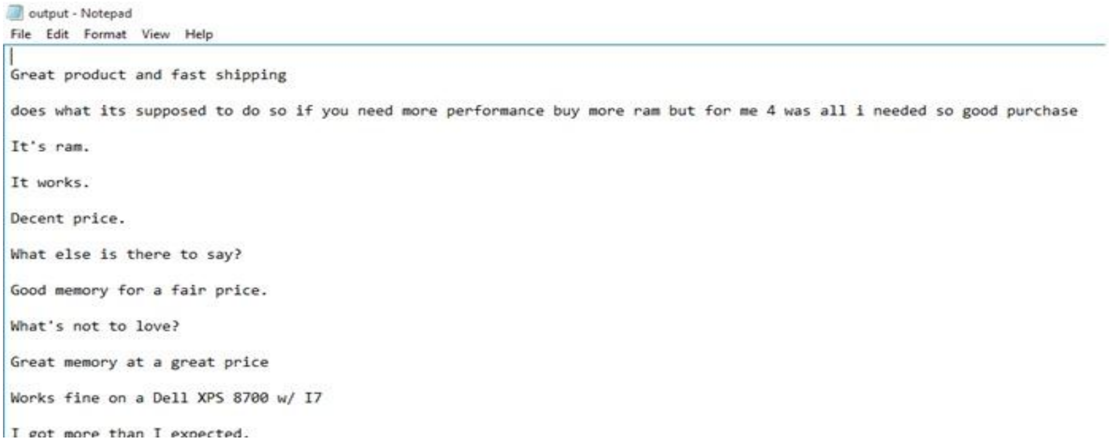
C. Extracting Product Reviews and Information

From the Graphic cards “NVidia Ge-Force GTX 980” HTML page, we have used PERL [12] script to extract different kinds of information which a counter of the extracted reviews so far (can be used as a unique ID for the dataset).

- Date of the review in YYYYMMDD format (note: on non-English speaking domains this feature won’t work, edit the script to set the name of months in the desired language).
- ID of the reviewed product.
- Star rating assigned by the reviewer.
- Count of "yes" helpfulness votes.
- Count of total helpfulness votes (yes+ no).
- Date of the review in human readable format (will be in the language used by the specified domain).
- ID of the author of the review.
- Title of the review
- Content of the review

D. Splitting Product Review Texts into Sentences

Then, we parse a product review to split texts into sentences, and produce POS (part of speech) tags, such as noun, verb, adjective, etc., for each word. The POS tagger (or called Tree-Tagger) developed at the University of Stanford [13] in annotating words with POS tags is employed.



```
Great product and fast shipping

does what its supposed to do so if you need more performance buy more ram but for me 4 was all i needed so good purchase

It's ram.

It works.

Decent price.

What else is there to say?

Good memory for a fair price.

What's not to love?

Great memory at a great price

Works fine on a Dell XPS 8700 w/ I7

I got more than I expected.
```

Figure 9 Splitting Reviews into Sentences

E. Parts of Speech Tagging: -

The Part of Speech Tagging also called grammatical tagging is the process for assigning the correct part of speeches (e.g. noun, adverb, verb, adverb etc.) to each word in a text based on both its definition and context as follows. We have used POS tagger which is developed at University of Stanford. [13]

Word	This	Article	Is	about	the	Sport	
POST	DT	NN	VBZ	RB	DT	NN	SENT

Example:

Sentence: They refuse us to obtain the permit

[('They', 'PRP'), ('refuse', 'VBP'), ('to', 'TO'), ('permit', 'VB'), ('us', 'PRP'), ('to', 'TO'), ('obtain', 'VB'), ('the', 'DT'), ('refuse', 'NN'), ('permit', 'NN')]

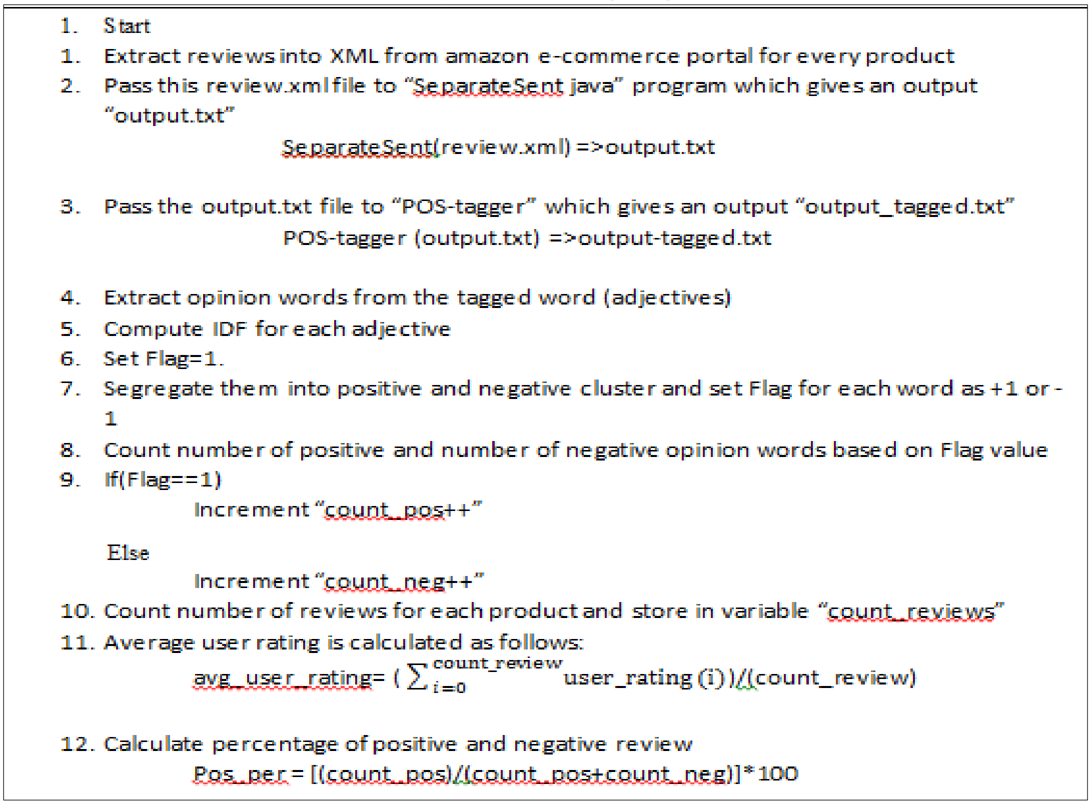
F. Extracting Opinion Words

We extract opinion word using a POS tagged output we got earlier. Here Opinion words are nothing but adverbs and adjectives which decides the polarity of sentence. Here we collect the positive words set and negative words set. First we define 30 common adjectives and another 15 adjectives are negative. Then the synonyms and antonyms of the words in seed list are found using WordNet; this step iterates until no new synonyms or antonyms are found. Finally, the closed set containing the words with positive and negative polarity is divided into positive and negative set. After the WordNet database is created we filer the non-adjective and non-adverbs from the tagged output and then polarity is found out using the WordNet database developed earlier.

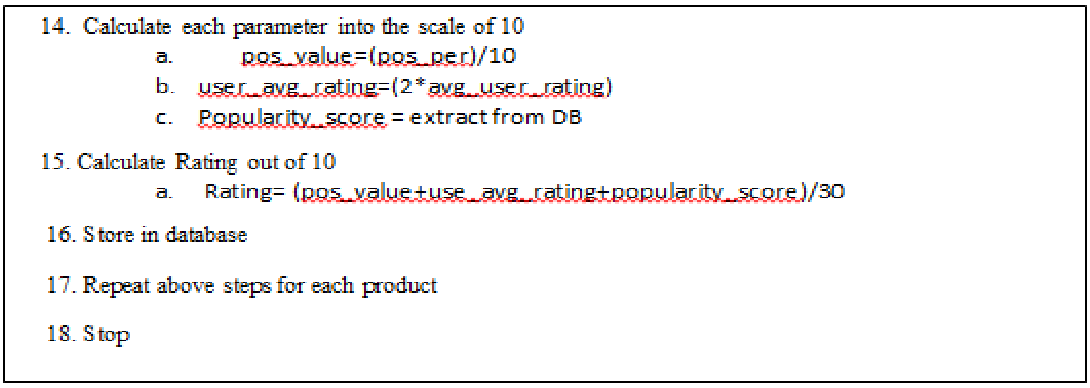
G. Rating Computation

We have designed an algorithm to calculate rating for each PC component. Before computing final rating we also check for popularity score of PC component which plays important role in shifting the orientation of computed rating.

PC Component Rating Algorithm



```
1.  Start
1.  Extract reviews into XML from amazon e-commerce portal for every product
2.  Pass this review.xml file to "SeparateSent java" program which gives an output
    "output.txt"
                SeparateSent(review.xml) =>output.txt

3.  Pass the output.txt file to "POS-tagger" which gives an output "output_tagged.txt"
                POS-tagger (output.txt) =>output-tagged.txt

4.  Extract opinion words from the tagged word (adjectives)
5.  Compute IDF for each adjective
6.  Set Flag=1.
7.  Segregate them into positive and negative cluster and set Flag for each word as +1 or -
    1
8.  Count number of positive and number of negative opinion words based on Flag value
9.  If(Flag==1)
            Increment "count_pos++"

    Else
            Increment "count_neg++"
10. Count number of reviews for each product and store in variable "count_reviews"
11. Average user rating is calculated as follows:
            avg_user_rating= ( $\sum_{i=0}^{count\_review}$ user_rating (i))/(count_review)

12. Calculate percentage of positive and negative review
            Pos_per = [(count_pos)/(count_pos+count_neg)]*100
```

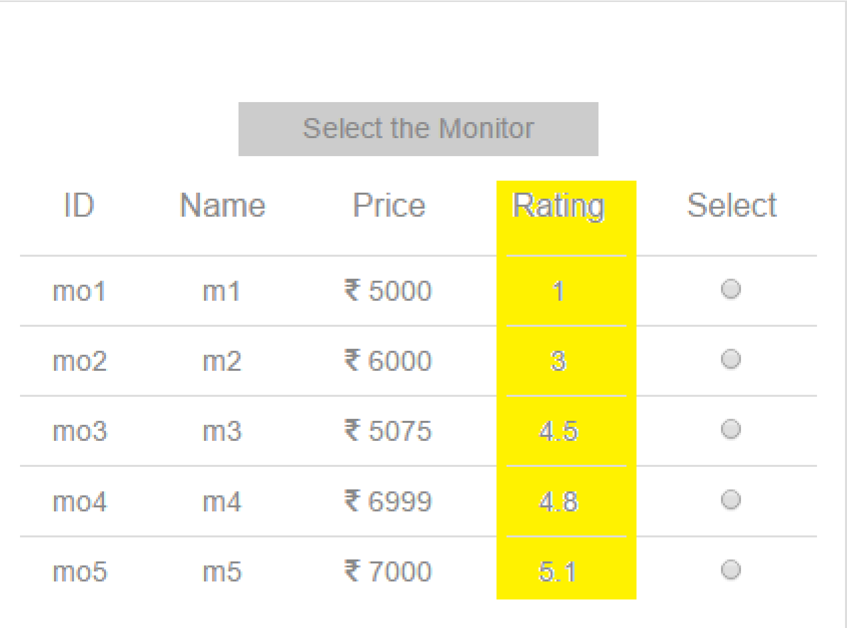


```
14. Calculate each parameter into the scale of 10
        a.    pos_value=(pos_per)/10
        b.    user_avg_rating=(2*avg_user_rating)
        c.    Popularity_score = extract from DB
15. Calculate Rating out of 10
        a.    Rating= (pos_value+use_avg_rating+popularity_score)/30
16. Store in database
17. Repeat above steps for each product
18. Stop
```

Popularity Algorithm

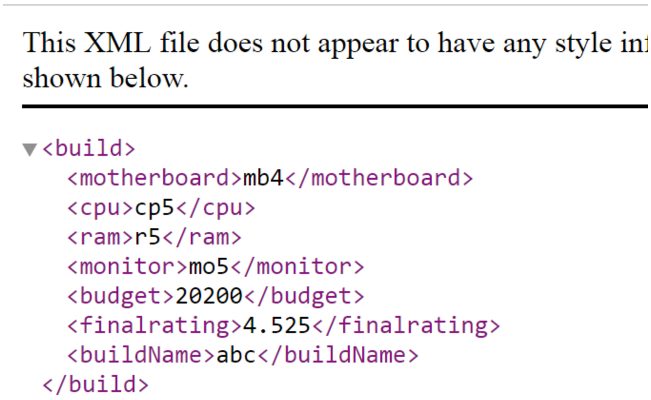
```
Input: Category No.
Output: Popularity Score

    1. Start
    2. Review []: extract no of review of each product for a particular component and
       store against given category no.
    3. max = max(review)
    4. popularity = (Review/max) *10
    5. repeat step 3 for each product
    6. store the popularity of each product in the database
    7. stop
```

VIII. INAL OUTPUT



ID	Name	Price	Rating	Select
mo1	m1	₹ 5000	1	○
mo2	m2	₹ 6000	3	○
mo3	m3	₹ 5075	4.5	○
mo4	m4	₹ 6999	4.8	○
mo5	m5	₹ 7000	5.1	○

Figure 15-Selection Menu to select component with rating



This XML file does not appear to have any style inf
shown below.

```
▼<build>
   <motherboard>mb4</motherboard>
   <cpu>cp5</cpu>
   <ram>r5</ram>
   <monitor>mo5</monitor>
   <budget>20200</budget>
   <finalrating>4.525</finalrating>
   <buildName>abc</buildName>
 </build>
```

Figure 17-Final Build XML File

IX. CONCLUSION

Online shopping will become increasingly important as more and more manufactures sell products on the Internet, and many users are using the Internet to express and share their opinions. Thus our Goal is to find the favorable or interesting products for an individual user among a huge amount of products. In this project we propose a product ranking system where users can specify product features to get back the ranking results of all matched products. We use opinion mining techniques to identify the sentence polarity in product reviews, and then calculate the scores of all matched products using the defined formulas. The experimental results show that the system is practical and the ranking results are interesting.

This Web based application will ultimately help the user to assemble PC without hassles of going and researching the whole market thereby wasting a lot of money and time. Overall our application will give a final rating based on the components or peripheral selected and the user will be easily being able to make their PC with getting all the required parameter like cost, performance, throughput etc.

REFERENCES

[1] https://www.w3.org/XML/ (Accessed: July, 2015)

[2] Yin-Fu Huang and Hung Lin," Web Product Ranking Using Opinion Mining", IEEE, 2013, pp.184-190

[3] Zhongwu Zhai,Bing Liu,Hua Xu,Peifa Jia," Clustering Product Features For Opinion Mining", WSDM Fourth ACM International Conference on Web Search and Data Mining,2011, pp.2-7

[4] Nigam K, McCallum A, Thrun S, and Mitchell T, Text classification from labeled and unlabeled documents using EM. Machine Learning, 2000. 39(2): 103-134.

[5] P. Tian, Y.Liu, M.Liu, and S. Zhu, "Research of product ranking technology based on opinion mini". Proc.c. the 2nd International Conference on Intelligent Computation Technology and Automation, pp. 239-243, 2009

[6] M. Hu and B. Liu, "Mining and summarizing customer reviews," Proc. The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages. 168-177, 2004.

[7] WordNet for Projects: (windows Platform)https://wordnet.princeton.edu/

[8] https://github.com/aesuli/Amazon-downloader (Accessed: August 2015)

[9] http://nlp.stanford.edu/software/tagger.shtml (Accessed: October 2015)