Analyzing Reputation by Mining Feedback Comments

GRADUATE PROJECT REPORT

Submitted to the Faculty of
The School of Engineering & Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

By

Mounik Reddy Vanguri
Spring 2015

Committee Members

Dr. Dulal Kar
Committee Chairperson

Dr. Maryam Rahnemoonfar
Committee Member
ABSTRACT

Generally online (Electronic commerce or E-commerce) applications use reputation reporting system for trust evaluation where they gather overall feedback ratings from the sellers to compute the reputation score. A well-known issue with the reputation conduct system is “all good reputation” problem where over 99% of feedback ratings are positive leading to high reputation scores. This issue is hard on buyers to select accurate sellers. By analyzing buyer’s opinions on free text feedback comments, we propose an approach called the Reputation Analyzer. The main idea behind reputation analyzer is an algorithm lexical-LDA (Latent Dirichlet Allocation) topic modeling technique proposed for mining the online feedback comments by grouping aspect expressions into dimensions and compute dimension ratings. Extensive experiments on eBay and Amazon data show that the reputation analyzer can significantly solve the “all good reputation” problem and rank sellers effectively.
# TABLE OF CONTENTS

Abstract ........................................................................................................................................ ii
Table of Contents .......................................................................................................................... iii
List of Figures ............................................................................................................................... v

1. Background and Rationale ....................................................................................................... 1
   1.1 Evolution of Data Mining ................................................................................................. 1
   1.2 Existing System ............................................................................................................... 2
   1.3 Proposed System ............................................................................................................ 3
   1.4 My Eclipse IDE ............................................................................................................... 4
   1.5 Tomcat Server ................................................................................................................ 4

2. Narrative .................................................................................................................................. 6
   2.1 Problem Statement .......................................................................................................... 6
   2.2 Motivation ...................................................................................................................... 6
   2.3 Project Objective ............................................................................................................ 7
   2.4 System Requirements ..................................................................................................... 7
      2.4.1 Functional Requirements ...................................................................................... 7
      2.4.2 Software Requirements ....................................................................................... 7
      2.4.3 Hardware Requirements ...................................................................................... 8

3. System Design ........................................................................................................................ 9
   3.1 Architecture ................................................................................................................... 9

4. System Implementation .......................................................................................................... 12
   4.1 Java Server Pages ........................................................................................................... 12
   4.2 Pentium IV ..................................................................................................................... 13
6. Conclusion ................................................................................................................................. 36

7. Future Work ............................................................................................................................ 37

Bibliography and References ..................................................................................................... 38
List of Figures

Figure 3.1: Reputation analyzer framework..........................................................11
Figure 4.1: Code snippet to decide the comment is positive or negative..............16
Figure 4.2: Code snippet to store the rating of a sentence in a table......................16
Figure 4.3: Code snippet to add the negative dimension ratings............................17
Figure 4.4: Code snippet to compute final rating..................................................17
Figure 4.5: Code snippet for NLP Parser...............................................................19
Figure 4.6: Code Snippet to compute the rating to a comment..............................20
Figure 4.7: Home screen......................................................................................21
Figure 4.8: Registration Page................................................................................22
Figure 4.9: User’s Home Screen..........................................................................22
Figure 4.10: Viewcart Page....................................................................................23
Figure 4.11: Feedback Page...................................................................................24
Figure 4.12: Before Feedback................................................................................25
Figure 4.13: After feedback...................................................................................25
Figure 5.1: Initial Score and ratings for LG product..............................................27
Figure 5.2: Feedback Screen..................................................................................28
Figure 5.3: The Score and Rating for the LG product after mining the comment......28
Figure 5.4: Initial Score and ratings for Nokia product.........................................29
List of Tables

Table 5.1: Ratings computed using different algorithms…………………………….35
1. BACKGROUND AND RATIONALE

1.1 Evolution of Data Mining

The past two decades has seen a dramatic increase in the amount of information or data being stored in electronic format. It is important to extract information and useful patterns from the large amounts of data. Analyzing data can provide further knowledge about a business by going beyond the data explicitly stored to derive knowledge about the business. This is where Data Mining or Knowledge Discovery in Databases (KDD) has obvious benefits for any enterprise.

Data Mining, or Knowledge Discovery in Databases (KDD) as it is also known, is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. This encompasses a number of different technical approaches, such as clustering, data summarization, learning classification rules, finding dependency networks, analysing changes, and detecting anomalies [12].

The knowledge discovery process has the following iterative sequence of steps performed on raw data to extract useful information.

- Data Cleaning: The raw data may contain some noise or may contain inconsistent data. The noise and inconsistent data is removed in this step.
- Data Integration: The data required may be available in different source. In this step the data from different sources is integrated together i.e., the data is combined.
- Data Selection: In this step, the data relevant to the task is retrieved from the database.
- **Data Transformation:** In this step, the selected data is transformed into different forms for which suitable mining techniques can be applied.

- **Data Mining:** This is the crucial step where useful data patterns are extracted by applying intelligent methods. The extraction of patterns from the given data is done.

- **Pattern Evaluation:** The truly interesting patterns are identified in this step. The interesting patterns representing knowledge are identified based on given measures.

- **Knowledge Representation:** This is the final step where the extracted patterns are visualized and represented to the users using different visualization techniques.

By following the above steps patterns representing knowledge can be extracted which can be useful in understanding the business of the related data.

### 1.2 Existing system

The reputation analyzer is the first method to be developed for computing feedback ratings and evaluate sellers trust scores on online applications such as eBay, where by mining online feedback comments, one can significantly reduce the positive opinions towards different parts of transactions, and compute extensive reputation profiles for sellers. In the existing system, the user is required to rate / give feedback of particular service or experience in terms of stars. The more the number of stars implies higher the rating. Usually the user has to rate on scale of five, then the system averages the ratings given by the users and arrives at an overall rating.

Example- eBay, Amazon and many feedback forms on many online services use the star rating system.
The four aspects of transactions based on which the detailed seller ratings are reported in eBay:

1. Described item
2. Communication
3. Postage time
4. Postage and handling charges

One of the major drawbacks is the lack of negative ratings in online websites.

1.3 Proposed system

In our proposed system, we give the users an option of providing the feedback and rating through comments as well, not just stars. In this way, we are giving more flexibility and room for expression to the users, and also it is more natural for anyone to describe their experience through words than stars. These comments are mined, which will now be converted into ratings [23]. These comments will be tokenized into individual words to analyze them for their meaning, and thereby arriving at an appropriate rating.

The tokenized words are now compared with our database for key words like- “good”, “excellent”, “bad”, “mediocre” etc. and also with our database for key words like- “shipping”, “delivery”, “product”, ”quality” etc.

Then based on the keyword match, a rating will be decided upon. For example the keyword “excellent” has higher rating than the keyword “good”, and the keyword “worse” has lower rating than “bad”. After combining the ratings of all the keywords and phrases in the comment, an appropriate rating will be arrived at.
This rating would be an alternative to the star system. The star system has a limited range of 5 stars, in this case we can reach more precise rating based on the keywords. This proposed system is another way of arriving at an overall rating.

1.4 My Eclipse IDE

The project is implemented in java. My Eclipse is a Java Integrated Development Environment which is used in the development of this project. It is a tool that supports developers in creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn and others. It can also be used as a debugger for debugging the program. The Built-in libraries help in automatic configuration of the libraries during creation of project. The tool is very flexible and extensible for developing projects. MyEclipse 2015 CI is eclipse continuous integration system for developers who need new updates [10] [11]. MyEclipse can be used to develop Enterprise applications, Mobile apps, Cloud applications and also Web applications.

It offers the flexibility to optimize with the current Java EE technologies found in Eclipse. It is vendor-neutral, and cloud applications can be developed using REST web service technology, which can be used for developing and testing the services.

For developing mobile apps on both android and iOS, we can use PhoneGap for any target mobile. It offers easy testing for many applications at play. For testing we can deploy a range of application servers like Glasscock, tomcat, WebLogic and derby.

1.5 Tomcat Server

Apache Tomcat is a webserver and servlet container used for executing servlets and Java server pages. It is an open source web server. It provides a pure java web server environment for the code to run in. It includes tools for configuration and management. It
is cross platform software written in java language which works on various operating systems. Tomcat server can be used for running all kinds of web applications. It is a flexible and lightweight server. Apache Tomcat is an open source web server by Apache Software Foundation that executes JAVA servlets and JSP. It provides a java HTTP web server for java app to run in. We need JRE for tomcat, JRE 1.1 or higher is recommended. Since it is open-source, it can be downloaded from the site, where it is available in both source and binary versions [10]. Tomcat can be used with its own internal web server or with Apache ,Netscape enterprise server etc [11].
2. NARRATIVE

2.1 Problem Statement

As a buyer, we always look at the ratings, read the reviews posted before buying any product to make sure that it meets our requirements. The 1-5 star ratings do not always give us the exact feedback as there are cases where a buyer might leave positive feedback rating, but express some kind of negative text feedback usually towards the transactions. These kinds of ratings are biased and will confuse the buyers whether to choose a product or not. All the e-commerce websites like eBay, Amazon, etc., usually face this “all good reputation” issue.

2.2 Motivation

The traditional star rating system will give us the rating which can be biased which means even if the user gives a high star rating but in the comments if he mentions anything negative regarding the delivery of the product, then the rating becomes biased. Typically this type of rating system does not allow the buyer to rate the product in all aspects. So in order to overcome this issue, we need to consider the ratings for comments as well. Taking the comments into consideration while rating the product/service will make the buyers to trust the sellers. Comment rating serves the purpose of ranking the sellers in another way. Buyers can mislead by giving bogus feedbacks and there can be challenge of trust context change during the transactions. This kind of rating makes the buyer more convenient to trust the seller thus making the site more reliable [23]. By
analyzing content of the feedback comments, we can understand the opinion of the buyer in various points like regarding the transactions, quality of product and service provided by the sellers

2.3 Project Objective

In this project, we have model to mine the content of the comments and give rating based upon the content along with considering the normal 1-5 star rating thus providing rating for the buyers. We can analyze the opinion of the comment in both positive and negative aspects and determine the rating of the product by taking the average of ratings of all aspects. This project is an implementation of the comment rating behavior on a sample of comments from the eBay and other e-commerce sites. This model reduces the biased reviews and solves all good reputation problem in eBay and Amazon sites. Thus to overcome these challenges, we need to build a reputation rating system which ensures trustworthiness of the sellers and provide information to the buyers regarding comment ratings also.

2.4 System Requirements

2.4.1 Functional Requirements

- Should effectively mine the comments.
- Should be able to translate comments made in natural language to ratings.

2.4.2 Software requirements

- Language- JDK (1.7.0)
- Front-End- JSP,Servlets
• Back-end- Oracle 10g
• IDE- MyEclipse 2015 CI
• Operating system- Windows XP
• Server- Tomcat

2.4.3 Hardware requirements

• Processor - Pentium IV
• Hard Disk - 80GB
• RAM - 2GB
3. SYSTEM DESIGN

3.1 Architecture

To obtain an agreeable reputation score, the seller inspects the feedback comments, where the comments do not confirm explicitly on positive and negative ratings on transactions. Reputation-based trust model – it assures reliability and security in open systems by using profiles of peers whose public reputation is based on good behavior. This model is widely used in online systems, multi-agent systems, and peer-to-peer networks.

It mainly focuses classification of sentiments on feedback comments. After analyzing the feedback comments, the challenging problem in it is that the comments are noisy. Based on aspect ratings models are built to classify comments into positive and negative.

Feedback comments are usually long descriptive sentences which contain the user’s experience with the service, these are broken into meaningful phrases. Now, these phrases are given a dimension, like the word “shipping” has a dimension, the word “quality of the product” has another dimension.

These dimensions are now mined, so that they can be given a rating based on the words used, like the word “good” has higher rating than the word “average”. We can say that mining the dimensions results in getting ratings for those dimensions. After careful
evaluation of these dimensions and their ratings, we can obtain a picture on the trust evaluation of those dimensions and thereby, we can calculate their weightage.

Trust evaluation on dimension means that we are trying to extract the meaning and give a rating to the service/product used by the user for that particular dimension, for example, a user could have liked the product he bought but he might not be happy with the delivery time or he might like service but the website could not be user friendly, so obtaining trust evaluation on every dimension would give better clarity and help in getting correct overall rating as well. After this step, we obtain the exact rating given to the seller by the consumer. This will be added to the seller’s profile.

Enumerating Dimension weights is the process of adding all the weights of different dimensions and aspects to obtain an overall rating for the comment. For example – if a comment has three phrases which says “fast delivery”, “bad quality product”, “not satisfied”- the rating for the first phrase would be 5, the rating for the second phrase would be 1, and the rating for the third phrase would be 1 too. So by enumerating these dimension weights, we get an overall rating of 2.3 for this comment [23]. This step is very important because it gives us an accurate picture of what the consumer is trying to convey by their valuable feedback.

After this step, we now go to the step called Overall Trust Evaluation, where we intend to calculate the “trust” coefficient of the seller, after averaging the weights of all the comments. Up until now we have tokenized the comments, categorized them into various dimensions based the words they used, then mined those comments to obtain their weights. Now it is time to put them to use by giving an overall rating to the seller
based on all the comments because we can get the right rating only after considering all the ratings by different users.

Now we add this final rating to the seller’s profile which would be accurate. This will be of help to the prospective consumers on deciding whether or not use the services provided by the seller in question. It will also be useful to the seller for retrospection, on where he can improve his services.

![Figure 3.1: Reputation analyzer framework](image)

**Figure 3.1: Reputation analyzer framework**
4. SYSTEM IMPLEMENTATION

4.1 Java Server Pages

A Java Server page is a web technology which is used to develop webpages dynamically based on HTML and other document types. It was introduced by the Sun Microsystems in 1999. JSP is similar to PHP except that it uses Java Programming language. A compatible web server like Apache Tomcat is required to deploy and run the JSP. It is used to access Server side objects. Java server page is most likely to be the user interface in any web application based on java. The main use of JSP lies in taking inputs from users and then present from database and later generate a web page dynamically. Extracting data from database, information is shared and the control between the pages is passed using JSP tags. Instead of having a separate CGI file, JSP dynamically embeds elements in HTML pages which enhances the performance [13].

The most common delimiter used in JSP is <%-...%> in which a Java scriptlet is enclosed. A scriptlet is a small fraction of the code that is run only when the user requests the code. The in-built functionality is revoked using the additional tags that are offered and are called as JSP actions. In order to learn JSP, we need to be familiar with setting up
a development kit, setting up a web server etc [14]. Apache server is an open source developer model to implement JSP Pages.

The first web technology for Java is a Java servlet. Java servlet can also be best described as the class which upon incoming requests extends the capabilities of the server.

The main advantages of servlets are:

- Enhanced performance
- Portability
- Security

Usually a separate process is not required to handle each request of the client. The main tasks of the servlets include reading the explicit and implicit data given by the user and later process the data to generate results [15]. This process may include going through the database etc. The lifetime of a servlet depends upon its creation and destruction. The init() function is used to initialize the servlet whereas the destroy() function is used to destruct a servlet and a service() method is used when a client request is to be processed.

Servlets were also introduced by Sun Microsystems. Servlet, HttpServlet, servletrequest are examples of few classes present in it.

4.2 Pentium IV

It is an entry-level, single core CPU for laptops and PC’s by Intel. It uses NetBurst (seventh generation x86 microarchitecture), which is the first new chip
launched after p6 microarchitecture in the previous 1995 Pentium Pro CPU model. NetBurst uses deep pipeline architecture for very high clock speeds. It uses 20-stage pipeline, which increases the core frequency two times and reduces delay by executing it in half a clock cycle. Pentium IV have speeds over 2 Ghz. It has four versions – Willamette, Northwood, Prescott and Cedar with clock speed ranging from 1.3-3.8 Ghz. It has a 400 Mhz system bus which allows for a data transfer rate of 3.2 Gbps. It also makes use of advanced dynamic execution which enables faster processing for gaming and video applications. Pentium IV is also equipped with execution trace cache memory, improved multimedia units, and floating points [10] [11].

4.3 My SQL

MySQL is one of the largest open source database used in many of the web applications. SQL stands for Structured Query Language. SQL is defined by the ANSI/ISO SQL Standard. Data is stored in a structured format. It has much functionality from different database packages. MySQL connectivity, speed, access makes it highly suitable for accessing data over the internet.

Features

- It is very easy to use
- It works on different operating systems with different languages
- It works fast and works well with large data sets
- It is Reliable
- It is Scalable
- It is Cost effective

4.4 Environment
The reporting system is implemented using Java in My Eclipse IDE environment as they are more suitable for programming. The user interface is implemented using Java server pages. To run this application we are using Apache Tomcat server and in order to store the data MYSQL is used as the database. Initially in the database the words are stored. These words are used to compare with the tokenized words of the feedback comments.

4.5 Steps of Implementation

4.5.1 LDA Algorithm

In this algorithm, comments are mined, which will be converted into ratings. Initially, the comments will be divided into sentences based on identifiers such as but, and etc. Each sentence corresponds to one dimension. These sentences are stored in an array. Then they are taken one after the other to calculate the rating to the comment.

This sentence will be tokenized into individual words to analyze them. These words are stored in another array. The tokenized words are now compared with words in the database to decide the dimension whether it is shipping, quality etc. These words are then compared with words in another database to decide upon the dimension i.e., first database has words like “delivery”, “quality” “damage” which are dimensions, second database has words like “no”, ”good”, ”very good” etc.

These two databases are used to know the direction of dimension whether it is positive (ex: “good delivery”) or negative (ex: “slow shipping”).

Once the direction of dimension and dimension weights are computed, rating will be given accordingly which is stored in another database. This process continues until all the sentences are given rating.
After calculating the rating for all sentences in the comment, those values will be taken and then final rating will be given to that particular comment.

Final rating will be calculated by considering each row in the database. All the positive ratings will be added to positive score and vice-versa. Positive score and negative score will be added and then, it will be divided by numbers of rows in that table to calculate the final rating.

![Figure 4.1: Code snippet to decide the comment is positive or negative](image-url)

This code snippet uses flag bit to decide whether the dimension is positive or negative. If the flag bit (senti) is true the comment is positive and vice versa.
Figure 4.2: Code snippet to store the rating of a sentence in a table

In Figure 4.2, the code snippet is used to store the rating of a sentence which is part of a comment. In this table, the ratings of all the sentences will be computed and they will be placed.
Figure 4.3: Code snippet to add the negative dimension ratings

```java
if (rs14.next ())
{
    totalrecords++;
    negvalue=negvalue+1;
    Positive=a;
    if(ffeedback==null)
    {
        ffeedback=ffeedback+"-- "++;a;
    }
}
rs14.close();
conlv5.close();
st35.close();
catch (Exception e) {
    // TODO Auto-generated catch block
e.printStackTrace();
}
```

Figure 4.4: Code snippet to compute final rating

```java
if(totalrecords ==0)
{
    finalscore=2.5;
}
else
{
    DecimalFormat df = new DecimalFormat("#.##");
    finalscore=(posscorevalue+negvalue)/totalrecords;
    dat2= cf.format(finalscore);
}
```

From Figure 4.3 and Figure 4.4, we can say that these code snippets are used for calculating the final rating to a comment. All positive dimension ratings will be added to one variable and vice versa. Then the final rating will be computed accordingly. From
Figure 4.4, if the total records are zero, then this implies that the user comments does not match with any dimension, so a neutral rating of 2.5 is given.

**4.5.2 Stanford NLP Parser**

NLP is a mathematical tool to parse a sentence into meaningful words, categorize into required categories to analyze data for business requirements. We can use this tool to analyze various languages like English, Chinese and Spanish. We need model files specific to that language to analyze the specific language for parsing the data, we need to provide annotators as property configuration file [23]. We can specify one or two properties.

- Tokenize-It tokens the text into sentences.
- Split- It splits tokens into sentences.
- Pos- It gives index names for the tokens.
- Lemma- It generates lemmas for corresponding tokens.
- Parse- It will analyze the data syntactically, generates typed dependencies, extracts the data into tree structure.
- Sentiments- Every sentence has a sentiment. Based on the model filter, it will calculate the sentiment.
public String getSentiment(String txt1)
{
    String text = txt1;
    Properties props = new Properties();
    props.setProperty("annotators", "tokenize, ssplit, pos, lemma, parse, sentiment");
    StanfordCoreNLP pipeline1 = new StanfordCoreNLP(props);
    Annotation annotation1 = pipeline1.annotate(text);
    List<CoreMap> sentences = annotation1.get(CoreAnnotations.SentencesAnnotation.class);
    for (CoreMap sentence : sentences)
    {
        String sentiment = sentence.get(SentimentCoreAnnotations.SentimentValue.class);
        txt1=txt1 + sentiment + "\t" + sentence + "\n";
        System.out.println(sentiment + "\t" + sentence);
    }
    return txt1;
}

Figure 4.5: Code snippet for NLP Parser

From Figure 4.5, this code snippet describes how the NLP Parser calculates the rating to a comment. Stanford NLP Parser object is created and its properties are set accordingly as shown in the code snippet. Then the object created by Stanford NLP Parser is used to process the text to extract the annotations. From the annotations the sentences are obtained and on these sentences sentiment is calculated (the sentiment can be positive, negative or neutral). The sentiment value will be returned to the calling function.
double score=0;

for(String a:tempfull){

double sentencescore=0;
Scorer scorer=new Scorer();

String response=scorer.getSentiment(a);
if(response.equalsIgnoreCase("Neutral"))
{
    sentencescore=2.5;
}
else if(response.equalsIgnoreCase("positive"))
{
    sentencescore=5;
}
else if(response.equalsIgnoreCase("negative"))
{
    sentencescore=1;
}

status=status+"","+response;
score=score + sentencescore;
status=status.substring(6);
}

finalscore=score / tempfull.length;

Figure 4.6: Code Snippet to compute the rating to a comment

From Figure 4.6, the code snippet tells how the sentence is sent to the NLP Parse. getSentiment() function is used to calculate the rating of a particular sentence. This function will return either positive, negative or neutral as the value. If we get a neutral value we are assigning the sentence score to 2.5. If it is positive then the sentence score is 5 and if it is negative the sentence score is 1. The score variable will store the sum of all the sentences scores. The final score will be computed using the score variable and the number of sentences i.e., final score= score/number of sentences.
4.6 User Interface

4.6.1 Home Page

![Home Screen](image)

**Figure 4.7: Home screen**

The Figure 4.7 shows the home screen of the application. It has four links which have their own functionalities. If the user is new to the application, he/she needs to register by clicking on the register button. Then he/she can login by clicking on the login button where he/she can view the items available. Apart from normal users, there will be an administrator who have the privileges of adding new products.

4.6.2 Registration Page

If the user is new then the user need to register before buying the products. Figure 4.8 shows the registration page asking the user to provide some basic details.
4.6.3 User Page

After the user logs in, the user gets directed to his home screen where he can view all the items. There are several options available for the user. In the products option, the user can view the products and will have two options – first one is user can add product
to cart and the second one is user can provide feedback about the product. The view carts option allows the user to view the item selected to purchase. If user does not want to buy the product, he/she has the option of removing it from the cart.

4.6.4 Viewcart Page

In Figure 4.10, the user can view all the items he/she purchased for reference. The user also has option of removing the item. The user can also final price depending on the quantity of the product purchased.

![Figure 4.10: Viewcart Page](image-url)
4.6.5 Feed Backpage

In the feedback page, the user is provided with a textbox to provide his comments and the dropdown list from 1 to 5 is provided to give the rating. Once the user clicks on update feedback, in the backend the mining on the comments is done.

The sentence is read and then tokenized. If there are any conjunctions or full stops in the sentence, it considers from the start up to the conjunction as one sentence and conjunction as another sentence i.e., it divides the sentence into sub-sentences if conjunctions are available in the sentence. For example: “no delay” and “excellent product”. In this, “no delay” is considered as one sentence and “excellent product” as another sentence.

Once it is divided into sentences, each sentence is tokenized into words. E.g. consider “no delay” as one sentence, it is tokenized into two words no, delay. Once it is
tokenized, each word is compared with the keywords available in the database. If a word matches with the positive keyword, it gives a high rating to the words. In the other case, if the word matches with the negative keyword, it gives less rating to the word. If there are any words like “no”, “not” etc. in the sentence, then it gives opposite rating, i.e., a high rating will be converted into a low rating. After calculating the rating of each sentence, we will take an average of them and finally result will be obtained.

Figure 4.12: Before Feedback  Figure 4.13: After feedback

Figure 4.12 shows the score and rating before the feedback is provided.

Figure 4.13 shows the score and rating after the feedback is provided.
5. TESTING AND EVALUATION

5.1 Software Testing

Testing is the process of identifying errors in the code and to gain information about the quality of the product or service with the intent of finding whether the product or service under the test satisfies the specified requirements [9].

5.1.1 Black-Box Testing

Black-Box testing is the process of testing which involves only validating the system as a whole by providing inputs to the system and examining the corresponding outputs. It is oblivious to the internal system architecture and the testing is performed without having access to the code. It can be performed without having any knowledge about the internal functionality of the software [22].

5.1.2 White-Box Testing

White-Box testing is the detailed process of testing the logic and the control structure of the program. The tester needs to have knowledge about the functionality and the internal workings of the code. Upon White-Box testing we can find out which unit of code is having defects or behaving inappropriately.

5.2 Levels of Testing

5.2.1 Unit Testing

It is a level of software testing where individual units/components of a software are tested. It is performed by software developers themselves to make sure that there are no internal logical errors in the code. The goal of unit testing is to test whether for each function or procedure upon giving a set of inputs, it returns the correct values.
5.2.2 System Testing

System Testing is done once all the components are integrated and it tests the entire system as a whole. System Testing tests the both functional and non-functional requirements of the system. It is final phase of testing which is done before deploying the application, hence it emphasizes on meeting the business requirements as well.

5.3 Test Cases

5.3.1 Test Case 1

![Figure 5.1: Initial Score and Ratings for LG Product](image)

In Figure 5.1, we are considering LG Camera as the product. The score and rating values for the product are as follows:

- Score: 2.5 (Comment based rating)
- Rating: 2.83
Figure 5.2: Feedback Screen

In Figure 5.2, the user provides the feedback for the product. The comment given by the user is “delay in service” and the rating given is 4.

Figure 5.3: The Score and Rating for the LG product after mining the comment
In ecommerce site, there was a comment as “delay in service” and the rating as 4. The result when these are given to this system shows that the score for “delay in service” is less than the rating given by user. So, in this system we are providing scores for the comments apart from ratings which will help the end user to decide about the product.

5.3.2 Test Case 2

![Image of Nokia, Casio, and LG products]

**Figure 5.4: Initial Score and ratings for Nokia product**

In Figure 5.4, we are considering Nokia tab as the product. The score and rating values for the product are as follows:

- Score: 2.75 (Comment based rating)
- Rating: 3.75

In Figure 5.5, the user provides the feedback for the product. The comment given by the user is “product meets all the specifications but he/she is not satisfied with the delivery” and the rating given is 2.
This test case was taken based on the comment and rating given in Flipkart. The user gave the comment as “product meets all specifications” but gave the rating as 2. I have given the same comment and rating for the product Nokia. The product is missed to
check whether it is a positive or a negative comment. In this the comment appears to be a positive one. So the end result shows an increase in the score. As the rating given is very low the overall rating will decrease. The end user will have two values to decide about the product rather than reading each comment.

5.3.3 Test Case 3

![Welcome david123](image)

<table>
<thead>
<tr>
<th>Nokia</th>
<th>Canon</th>
<th>LG</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Nokia" /></td>
<td><img src="image" alt="Canon" /></td>
<td><img src="image" alt="LG" /></td>
</tr>
<tr>
<td>MRP Price : 1600 $</td>
<td>MRP Price : 4500 $</td>
<td>MRP Price : 2600 $</td>
</tr>
<tr>
<td>Discount : 15 %</td>
<td>Discount : 10 %</td>
<td>Discount : 10 %</td>
</tr>
<tr>
<td>Your Price : 1360 $</td>
<td>Your Price : 450 $</td>
<td>Your Price : 2340 $</td>
</tr>
<tr>
<td>Save : 240 $</td>
<td>Save : 450 $</td>
<td>Save : 260 $</td>
</tr>
<tr>
<td>Feedback: 🌟🌟🌟🌟</td>
<td>Feedback: 🌟🌟🌟</td>
<td>Feedback: 🌟🌟🌟🌟</td>
</tr>
<tr>
<td>Score : 3 (15 / per 5 Users)</td>
<td>Score : 2.5 (5 / per 2 Users)</td>
<td>Score : 2.43 (17 / per 7 Users)</td>
</tr>
<tr>
<td>Rating : 3.4</td>
<td>Rating : 2.5</td>
<td>Rating : 3</td>
</tr>
</tbody>
</table>

**Figure 5.7: Initial Score and ratings for Canon product**

In Figure 5.7, we are considering Canon camera as the product. The score and rating values for the product are as follows:

- Score: 2.5 (Comment based rating)
- Rating: 2.5

In Figure 5.8, the user provides the feedback for the product. The comment given by the user is “No damage and good quality” and the rating given is 5.
Figure 5.8: Feedback Screen

Figure 5.9: The Score and Rating for Nokia product after mining the comment
In this test case the comment is given as “No damage and good quality” and the rating as 5. We can say that the user is satisfied with the product. This is a positive comment so when we look at the overall score there will be an increase in score. As the end user is having a score for the comments it will help them in deciding about the product without reading all comments.

5.3.4 Test Case 4

![Figure 5.10: Initial Score and ratings for Canon product]

In Figure 5.10, we are considering Titan watch as the product. The score and rating values for the product are as follows:

- Score: 3.25 (Comment based rating)
- Rating: 3.5

In Figure 5.11, the user provides the feedback for the product. The comment given by the user is “abcdfg” and the rating given is 5.
Sometimes, a user can give a rating of 5 and can give a meaningless comment like “abcef”. In this system if we get such comments, we will take the average of the score of
the product. But with the rating given as 5, the overall rating will increase. In such cases the end user will have choice to go with either the comment score or the overall rating or sometimes the user might consider both the score and rating to decide about the product.

In the similar way, when we did this on 20 comments which are also available in other online ecommerce applications. We could conclude that 13 times my system’s score rating was higher than what was provided in other websites. Finally, from this observation, we can conclude that 65 percent of times my application provides higher score values when compared to ecommerce rating values.

In Table 5.1, the ratings to a comments are calculated using different methods and the results are shown.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Rating</th>
<th>LDA</th>
<th>NLP</th>
<th>Survey</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>3</td>
<td>2.4</td>
<td>3.4</td>
<td>3.2</td>
</tr>
<tr>
<td>2.</td>
<td>4</td>
<td>3.87</td>
<td>3</td>
<td>3.8</td>
</tr>
<tr>
<td>3.</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1.8</td>
</tr>
<tr>
<td>4.</td>
<td>1</td>
<td>2.8</td>
<td>2.3</td>
<td>1</td>
</tr>
<tr>
<td>5.</td>
<td>3</td>
<td>2.5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>2.6</td>
<td>2.914</td>
<td>2.54</td>
<td>2.76</td>
</tr>
</tbody>
</table>

Table 5.1: Ratings computed using different algorithms
6. CONCLUSION

Existing system cannot guide the buyers to select trustworthy sellers for transaction when the reputation score is high enough to effectively rank the sellers. One of the well-known example of managing good reputation in online websites is the “all good reputation” problem. To uncover dimension ratings in feedback comments we calculate the multidimensional trust profiles of the sellers comprehensively. Effective algorithms are utilized in computing the trust scores by extracting opinion expressions from feedback comments. It is a novel application of combining opinion mining along with some summarization techniques for evaluation of the E-commerce applications. This project proposes an approach to the well-known problem of “all good reputation” where over 99% of feedback ratings are positive leading to high reputation score by computing the reputation scores based on feedback comments and ratings given by the user. Based on both the comments and the ratings given by the user, ratings for the product are displayed for a product thus providing ratings through which the user can make proper decisions. This project also analyzes the multidimensional trust profiles of the sellers comprehensively. Effective algorithms are utilized in computing the dimension trust scores by extracting opinion expressions from feedback comments.
7. FUTURE WORK

The future enhancement to this project is to mine the comments with more word count by storing them in database which would improve the overall accuracy of the rating system.

In future, the end user can give the dimensional weights so that the rating to the comment can be computed accordingly.
BIBLIOGRAPHY AND REFERENCES


[22] [http://istqbexamcertification.com/what-is-system-testing/#more-256](http://istqbexamcertification.com/what-is-system-testing/#more-256)