CHAPTER 1

INTRODUCTION

Introduction to Data Mining

We are in an age often referred to as the information age. In this information age, because we believe that information leads to power and success, and thanks to sophisticated technologies such as computers, satellites, etc., we have been collecting tremendous amounts of information. Initially, with the advent of computers and means for mass digital storage, we started collecting and storing all sorts of data, counting on the power of computers to help sort through this amalgam of information. Unfortunately, these massive collections of data stored on disparate structures very rapidly became overwhelming.

This initial chaos has led to the creation of structured databases and database management systems (DBMS). The efficient database management systems have been very important assets for management of a large corpus of data and especially for effective and efficient retrieval of particular information from a large collection whenever needed. The proliferation of database management systems has also contributed to recent massive gathering of all sorts of information. Today, we have far more information than we can handle: from business transactions and scientific data, to satellite pictures, text reports and military intelligence. Information retrieval is simply not enough anymore for decision-making. Confronted with huge collections of data, we have now created new needs to help us make better managerial choices. These needs are automatic summarization of data, extraction of the "essence" of information stored, and the discovery of patterns in raw data.

What are Data Mining and Knowledge Discovery?

With the enormous amount of data stored in files, databases, and other repositories, it is increasingly important, if not necessary, to develop powerful means for analysis and perhaps interpretation of such data and for the extraction of interesting knowledge that could help in decision-making.

Data Mining, also popularly known as Knowledge Discovery in Databases (KDD), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. While data mining and knowledge discovery in databases (or KDD) are frequently treated as synonyms, data mining is actually part of the knowledge discovery process. The following figure shows data mining as a step in an iterative knowledge discovery process.

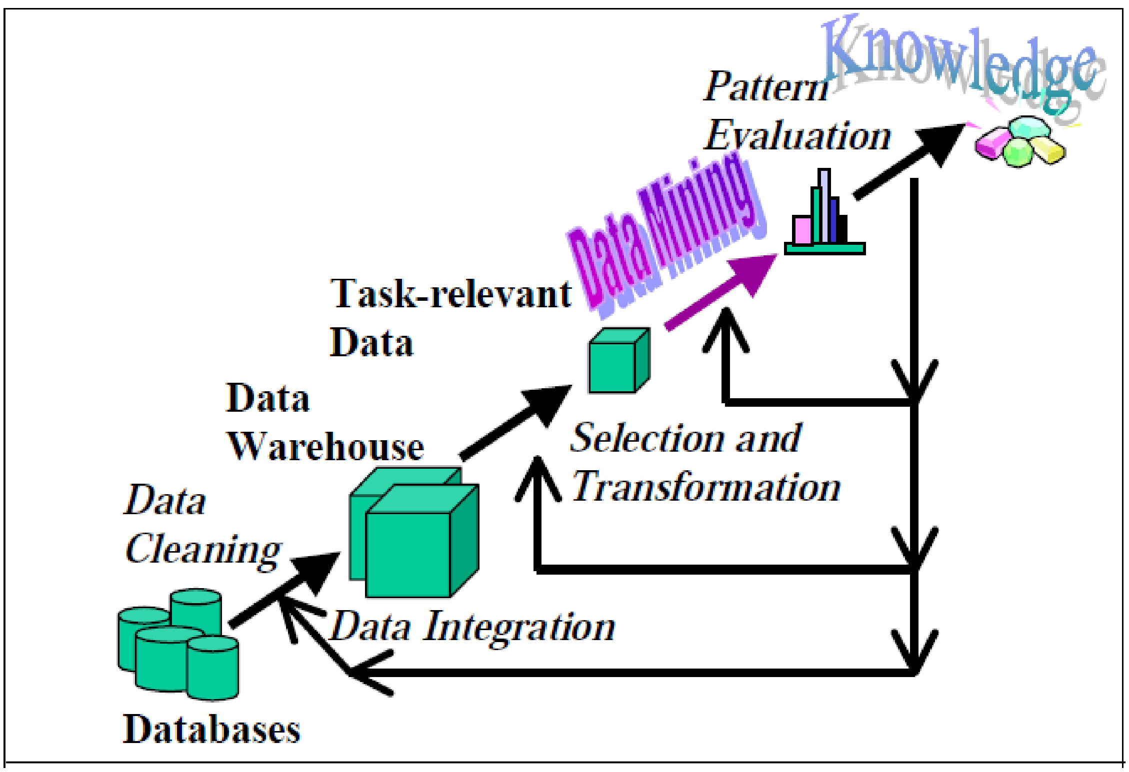


FIGURE 1.1

DATA MINING KNOWLEDGE

The Knowledge Discovery in Databases process comprises of a few steps leading from raw data collections to some form of new knowledge. The iterative process consists of the following steps:

- **Data cleaning**: also known as data cleansing, it is a phase in which noise data and irrelevant data are removed from the collection.
- **Data integration**: at this stage, multiple data sources, often heterogeneous, may be combined in a common source.
- **Data selection**: at this step, the data relevant to the analysis is decided on and retrieved from the data collection.
- **Data transformation**: also known as data consolidation, it is a phase in which the selected data is transformed into forms appropriate for the mining procedure.
- **Data mining**: it is the crucial step in which clever techniques are applied to extract patterns potentially useful.
- **Pattern evaluation**: in this step, strictly interesting patterns representing knowledge are identified based on given measures.
- **Knowledge representation**: is the final phase in which the discovered knowledge is visually represented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results.

1.1 Introduction to Opinion Mining

Internet contains large number of information which consists of online opinion such as news comments, political affair comments, product comments etc. Web users express their personal views on review websites, blogs, and discussion forum and so on. This information is publically available to users through internet. Sentiment analysis is known as opinion mining, involves in building a system to collect and examine opinions about the product made in blog posts, comments, reviews or tweets. There are several issues in Sentiment analysis. First one is opinion word that is considered to be positive or negative situation for customer product opinion. A second target is that people don't always feel opinions in a same way for product.

Several types of machine learning methods and training sets are used to perform the automatic text classification. Machine learning methods such as Support Vector machine (SVM), Artificial intelligence, Naïve Bayesian or hybrid approaches are used to improve the efficiency of classification. But all these methods are not focused on generating extractive summaries. In this paper focused on the problem of feature level opinion mining and enhance the two major tasks, recognition and classification. Recognition is the task of recognizing sentences expressing opinions; classification is the task of classifying elements in an opinion sentence into different categories such as opinion words phrases and product features. Data Mining is the solution of extracting the hidden knowledge from large volumes of raw data.

1. Training on a mixture of categorized data from other domains where such data are available.

2. Training a classified as above, but preventive the set of features to those experimental in the target domain.

3. Using groups of classifiers from domains with available labeled data.

1.2. Opinion Mining and Sentiment Analysis and Data Mining

An opinion is a subjective statement, view, attitude, emotion, or appraisal about an entity or an aspect of the entity from an opinion holder. Sentiment orientation of an opinion: positive, negative, or neutral (no opinion).It is also known as orientation, semantic orientation sentiment polarity. The term ―sentiment used in reference to the automatic analysis of evaluative text and tracking of the predictive judgments.

The term "Data mining" was introduced in the 1990s, but data mining is the evolution of a field with a long history. In the early 1960s, data mining was called statistical analysis, and the inventors were statistical software companies such as SAS and SPSS. By the late 1980s, the traditional techniques had been augmented by new methods such as fuzzy logic, heuristics and neural networks.

Opinion mining can be defined as a sub-discipline of computational syntax that concentrations on extracting people‘s opinion from the web. The current growth of the web encourages users to contribute and express themselves via blogs, videos, social networking sites, etc. All these platforms provide a vast quantity of valuable evidence that are interested to analyze.

The aspect-level sentiment analysis on the other hand assumes that a document contains opinion about multiple aspects/entities of one or more objects in a document. It is therefore necessary to identify about which entity an opinion is directed at.

1.3. Opinion Mining and Sentiment Analysis Techniques

Opining Mining is a relatively recent discipline that studies the extraction of opinions using Information Retrieval, Artificial intelligence and/or Natural Language Processing techniques. More informally, it's about extracting the opinions or sentiments given in a piece of text. There are various techniques used to extract information and knowledge are generalization, classification, clustering, association rule mining, data visualization, neural networks, fuzzy logic, Bayesian networks, and genetic algorithm, decision tree.

1.4. Information Demand on Sentiment and Opinions

Different people have different thoughts about the information presented during the process of decision making. Prior to when World Wide Web was known, people used to consider the recommendation of their friends; however, after the introduction of internet, information is readily available at any time. People can find opinions on their quest without any hustles. As people continue to live, they experience new things and ideas that require explanation. The demand for information has enabled people to depend majorly on internet. Whenever researchers find new ideas, they tend to share on the Internet their thoughts, open for critique by other authors. Opinion raised help in developing the argument of the subject.

1.5. Determining Opinion Orientation

To determine opinion orientation, we first convert the hybrid tagged sentences to basic tagged sentences, and then for each recognized product entity, we search its matching opinion entity, which is defined as the nearest opinion word/phrase identified by the tagger. The orientation of this matching opinion entity becomes the initial opinion orientation for the corresponding product entity. Next, natural language rules reflecting the sentence context are employed to deal with specific language constructs which may change the opinion orientation, such as the presence of negation words.

1.6. Opinion Search Engine

As such, people always depend on the internet for information. The creation of these systems that highlight internet information requires expertise. Some of the things to consider while developing the system include the application used in relation to the purpose of the engine. Users should consider the relevance of the searched information to the existing database. After the identification of the relevant information, the user faces the issue of selecting the target document. Thus, the system has to consider a summary of the presented query for the user to locate the target document.

1.7 OBJECTIVE

The main objective of review mining andsummarization is extracting the features on which thereviewers express their opinions and determining whether theopinions are positive or negative and identifies the reviews are positive or negative.

Our task is performed in steps:

(1)while login the customer will be verified using his/her e-mail id;

(2) mining product features that have been commented on by customers;

(3) identifying opinion sentences in each review and deciding whether each comment positive or negative;

(4)and while giving opinions if its fake then e-mail id is blocked;

(5) summarizing the results.

CHAPTER 2

LITERATURE SURVEY

2.1 Fake Review Detection from a Product Review using Modified Method of Iterative Computation Framework

The rapid growth of the Internet influenced many of our daily activities. One of the very rapid growth area is ecommerce. Generally e-commerce provide facility for customers to write reviews related with its service. The existence of these reviews can be used as a source of information. For examples, companies can use it to make design decisions of their products or services, while potential customers can use it to decide either to buy or to use a product. Unfortunately, the importance of the review is misused by certain parties who tried to create fake reviews, both aimed at raising the popularity or to discredit the product. This research aims to detect fake reviews for a product by using the text and rating property from a review. In short, the proposed system (ICF++) will measure the honesty value of a review, the trustiness value of the reviewers and the reliability value of a product. The honesty value of a review will be measured by utilizing the text mining and opinion mining techniques. The result from the experiment shows that the proposed system has a better accuracy compared with the result from iterative computation framework (ICF) method.

2.2 Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums

The Internet is frequently used as a medium for exchange of information and opinions, as well as propaganda dissemination. In this study the use of sentiment analysis methodologies is proposed for classification of Web forum opinions in multiple languages. The utility of stylistic and syntactic features is evaluated for sentiment classification of English and Arabic content. Specific feature extraction components are integrated to account for the linguistic characteristics of Arabic. The entropy weighted genetic algorithm (EWGA) is also developed, which is a hybridized genetic algorithm that incorporates the information-gain heuristic for feature selection. EWGA is designed to improve performance and get a better assessment of key features. The proposed features and techniques are evaluated on a benchmark movie review dataset and U.S. and Middle Eastern Web forum postings. The experimental results using EWGA with SVM indicate high performance levels, with accuracies of over 91% on the benchmark dataset as well as the U.S. and Middle Eastern forums. Stylistic features significantly enhanced performance across all testbeds while EWGA also outperformed other feature selection methods, indicating the utility of these features and techniques for document-level classification of sentiments.

2.3 Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification

Automatic sentiment classification has been extensively studied and applied in recent years. However, sentiment is expressed differently in different domains, and annotating corpora for every possible domain of interest is impractical. We investigate domain adaptation for sentiment classifiers, focusing on online reviews for different types of products. First, we extend to sentiment classifi- cation the recently-proposed structural correspondence learning (SCL) algorithm, reducing the relative error due to adaptation between domains by an average of 30% over the original SCL algorithm and 46% over a supervised baseline. Second, we identify a measure of domain similarity that correlates well with the potential for adaptation of a classifier from one domain to another. This measure could for instance be used to select a small set of domains to annotate whose trained classifiers would transfer well to many other domains.

2.4 Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems

The technique of collaborative filtering is especially successful in generating personalized recommendations. More than a decade of research has resulted in numerous algorithms, although no comparison of the different strategies has been made. In fact, a universally accepted way of evaluating a collaborative filtering algorithm does not exist yet. In this work, we compare different techniques found in the literature, and we study the characteristics of each one, highlighting their principal strengths and weaknesses. Several experiments have been performed, using the most popular metrics and algorithms. Moreover, two new metrics designed to measure the precision on good items have been proposed.

The results have revealed the weaknesses of many algorithms in extracting information from user profiles especially under sparsity conditions. We have also confirmed the good results of SVD-based techniques already reported by other authors. As an alternative, we present a new approach based on the interpretation of the tendencies or differences between users and items. Despite its extraordinary simplicity, in our experiments, it obtained noticeably better results than more complex algorithms. In fact, in the cases analyzed, its results are at least equivalent to those of the best approaches studied. Under sparsity conditions, there is more than a 20% improvement in accuracy over the traditional user-based algorithms, while maintaining over 90% coverage. Moreover, it is much more efficient computationally than any other algorithm, making it especially adequate for large amounts of data.

2.5 Do online reviews affect product sales? The role of reviewer characteristics and temporal effects

Online product reviews provided by consumers who previously purchased products have become a major information source for consumers and marketers regarding product quality. This study extends previous research by conducting a more compelling test of the effect of online reviews on sales. In particular, we consider both quantitative and qualitative aspects of online reviews, such as reviewer quality, reviewer exposure, product coverage, and temporal effects. Using transaction cost economics and uncertainty reduction theories, this study adopts a portfolio approach to assess the effectiveness of the online review market. We show that consumers understand the value difference between favorable news and unfavorable news and respond accordingly. Furthermore, when consumers read online reviews, they pay attention not only to review scores but to other contextual information such as a reviewer's reputation and reviewer exposure. The market responds more favorably to reviews written by reviewers with better reputation and higher exposure. Finally, we demonstrate that the impact of online reviews on sales diminishes over time. This suggests that firms need not provide incentives for customers to write reviews beyond a certain time period after products have been released.

2.6 Fake Product Review Monitoring and Removal for Genuine Online Product Reviews Using Opinion Mining

Seller selling products on the web often ask or take reviews from customers about the products that they have purchased. As e-commerce is growing and becoming popular day-by-day, the number of reviews received from customer about the product grows rapidly. For a popular product, the reviews can go upto thousands. This creates difficulty for the potential customer to read them and to make a decision whether to buy or not the product. Problems also arise for the manufacturer of the product to keep track and to manage customer opinions. And also additional difficulties are faced by the manufacturer because many other merchants sites may sell the same product at good ratings and the manufacturer normally produces many kinds of products. In this research, we aim to summarize all the customer reviews of a product and compare the products based on reviews can be done on one place. This summarization task is different from traditional text summarization, because we only mine the information of that product on which the customers have expressed their opinions and whether the opinions are positive or negative. We do not summarize the reviews by selecting a rewrite some of the original comment, from the reviews to capture the main points as in the classic text summarization. Our task is performed in steps: (1)while login the customer will be verified using his/her e-mail id; (2) mining product features that have been commented on by customers; (3) identifying opinion sentences in each review and deciding whether each comment positive or negative; (4)and while giving opinions if its fake then e-mail id is blocked; (5) summarizing the results.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

The existing framework collects product reviews from users; that is, the performance of a given product is combined with review ratings provided by consumers who have used the product in the past. It might be a single image representing an overall product rating or a vector representing icons for each feature or an attribute of a product, such as the processor, display quality, and battery quality of a Smartphone.

The existing approach is not restricted to any language and thus overcomes the language barrier often encountered in an interlingual recommendation system. Furthermore, buyers may not always provide feedback for the purchased products. However, buyers become concerned only when previous consumers are not satisfied with the product. This framework graphically represents the features of products by dynamically displaying icons, and reviewers are asked to provide their ratings in terms of stars corresponding to their experiences. With this set up, manufacturers and customers can obtain an average of all reviewers' experience.

Drawbacks:

- Buyers become concerned only when previous consumers are not satisfied with the product.
- If review ratings are genuine, they do not fluctuate considerably from the benchmarked value.
- It demonstrates that higher similarity and complete negative correlation indicate that the two users' choices are not equal, eventually enabling the deletion of users who have provided false reviews.

3.2 Proposed System

In this proposed system the opinion of people is got through search engine along with the different websites. Most of the websites provides user ratings in percentage and search engine monitors best matching web pages according to its pattern. But current search engine does not provide semantic orientation of the content in review. Sentiment classification is done by binary classification. The system will provide summary about the product reviews. The product rating depends on sentiment classification result. Feature based summarization focused on the product features on which the reviewers articulate their opinions. Identification of product features and opinion words are both essential in feature based summarization. The proposed framework collects product reviews from users. To prevent malicious review ratings, this work proposes an algorithm for preventing malicious feedbacks. As reviews are collected by the review system, the difference between each review is benchmarked and then is cumulatively summed. To identify false reviews, a real online dating service is used from Collaborative filtering dataset. The proposed framework can identify and block IP addresses that have used the review rating system to provide false reviews.

Advantages:

- The proposed system will save their efforts and time by helping the users and business organizations identify spams from different opinions quickly and also help in purchasing their valuable products from a trustworthy site.
- This automatic system can be useful to business organization as well as to customers.
- Customers can make decision whether he/she should buy or not buy the products. This can be helpful to people to purchase valuable product and spend their money on quality products.

3.3 Methodology

To determine the confidence of a rating, therefore, we have adopted three key factors of activity, objectivity, and consistencyand defined these factors in the context of onlineratings. First, the user who rates more items displays a higherlevel of activity. The above description of activity impliesthat the activity is defined by the amount of interactionsbetween an information producer and the users obtaining hisinformation. There exist, however, no interactions betweenusers in an online rating system; instead, there are actionsby users on products. Therefore, we measure user activity inan online rating system based on the amount of actions bythe user on products (i.e., the number of products he rates).

Additionally, a user whose ratings exhibit higher objectivitiesshould also have a higher level of user objectivity. Theuser objectivity is measured by the normalized average of theobjectivities of the ratings submitted by that useron the left whose ratings are similar to the reputationsof the items exhibits higher objectivity than the user.Two different states of user consistency.whose ratings are quite different from the reputations ofthe items.Third, we define the user consistency as how consistent theuser is in rating products; in other words, how consistentlyhe keeps his objectivities of ratings.

System Design and Development

The proposed framework does not require clustering orclassification, both of which necessitate considerable learningtime. Though TRUE-REP0UTATION does not require anylearning steps when solving a false reputation, extensiveexperiments show that TRUE-REPUTATION provides moretrustworthy reputations than do algorithms based on clusteringor classification.The contributions of this paper are as follows

In an online rating system, it is almost impossible to obtainthe ground-truth data because there is no way of knowingwhich users have caused a false reputation in a real-lifedatabase.

A.Dangerous Users

Based on observations of online rating systems, we identifiedtwo types of abusers who present unfair ratings regardlessof the quality of the product.

1) ***Planned Attacker:***A planned attacker is a user who“intentionally” manipulates the reputation of a targetproduct(s) by giving unfair ratings. This user may behired by a company to improve the reputation of itsproduct or to damage the reputation of competitors’products [16], [19].

2) ***Unplanned Attacker:***An unplanned attacker is eitheran extremist who evaluates the quality of a productaccording to “abnormal” standards or a don’t-career who“without planning” provides meaningless ratings. Anexample of an extremist is a user who gives an extremelyhigh rating to an author he prefers regardless of the qualityof the book

FALSE-REPUTATION SCENARIOS

Dangerous situations in which a false reputation can occur areas follows.

1) ***Product Launch Phase:***Before the release of a newproduct, there is no customer experience on which tobase an opinion. Online rating systems often allowusers to evaluate products, such as prerelease movies,before their release. **2)** ***Unpopular Products:***In online shopping malls, manyproducts are unpopular with few ratings. Because of this,the overall opinion about the unpopular product appearsto be untrustworthy.

C. False Reputation Scenarios

False reputation occurs when “dangerous users” enter“dangerous situations. We establish several experimental scenarios in our experiments.First, we generate dangerous users who behave as eithera planned or unplanned atta

CHAPTER 4

SYSTEM DESIGN

4.1 GENERAL

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

4.1.2. USE CASE DIAGRAM

A use case diagram is a type of behavioral diagram created from a Use-case analysis. The purpose of use case is to present overview of the functionality provided by the system in terms of actors, their goals and any dependencies between those use cases.

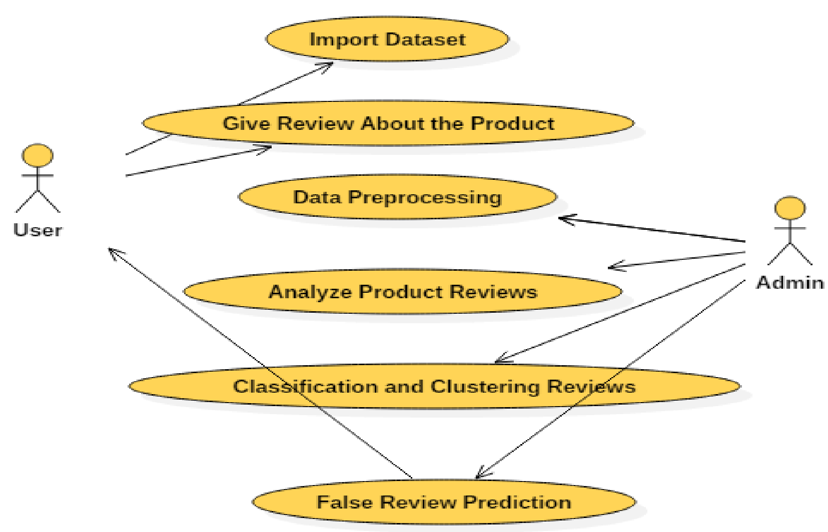


FIGURE 4.1

4.1.3. CLASS DIAGRAM

A class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class.

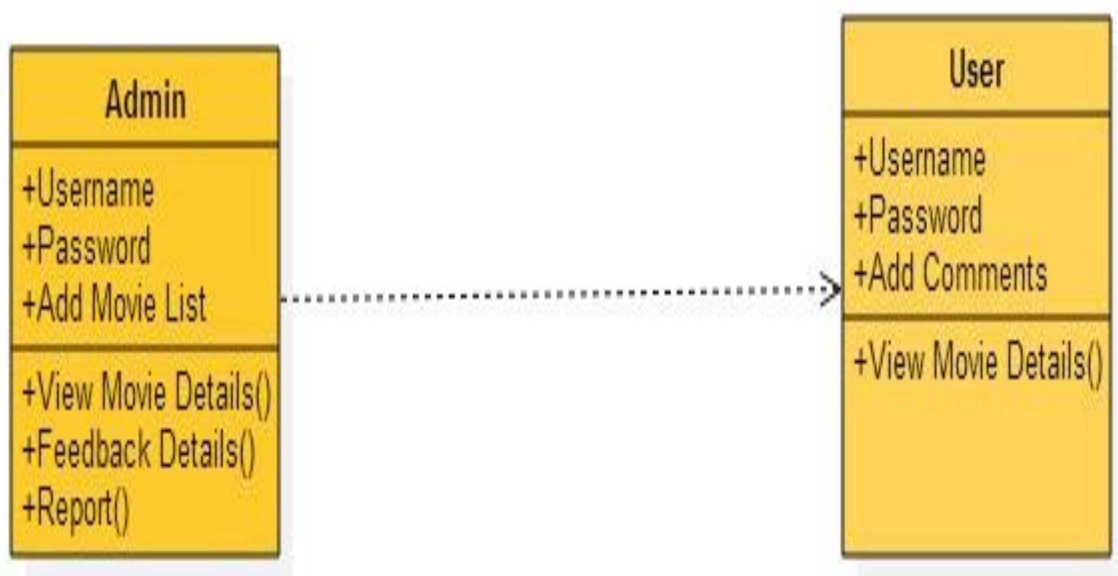


FIGURE 4.2

4.1.4. SYSTEM ARCHITECTURE

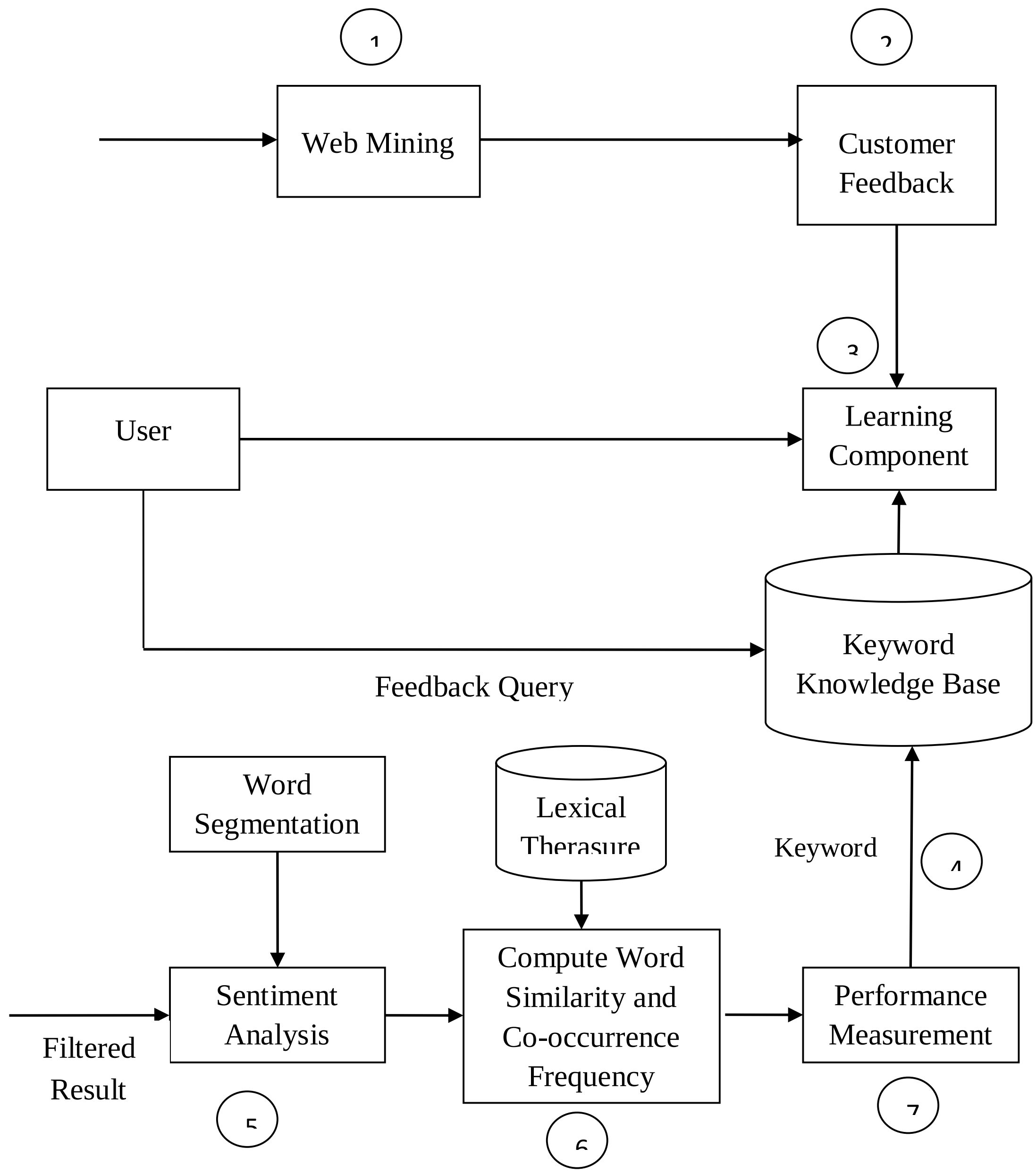


FIGURE 4.3

4.1.5. DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

Level 0:

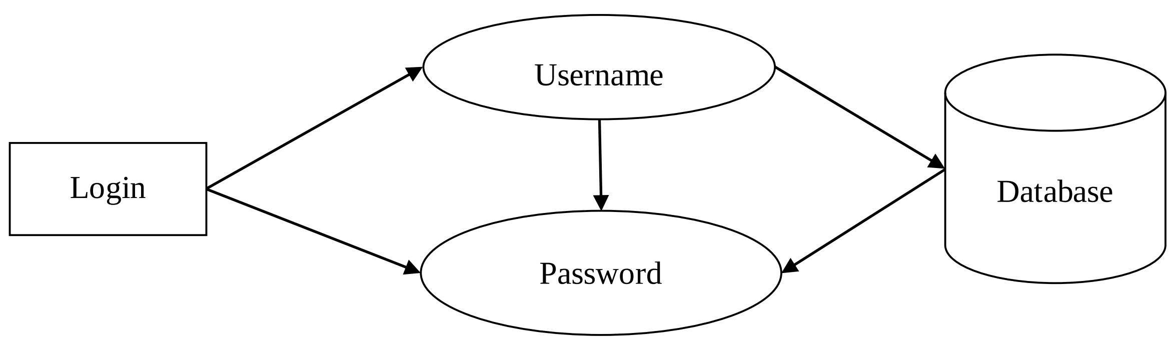
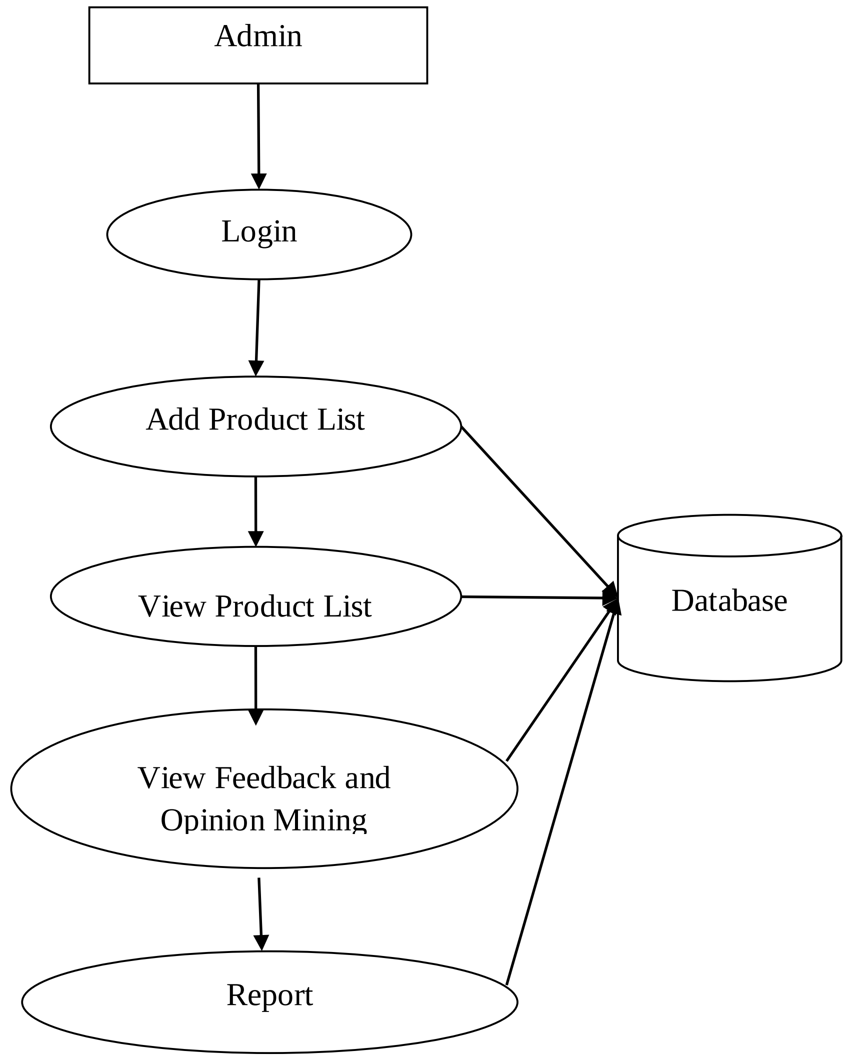
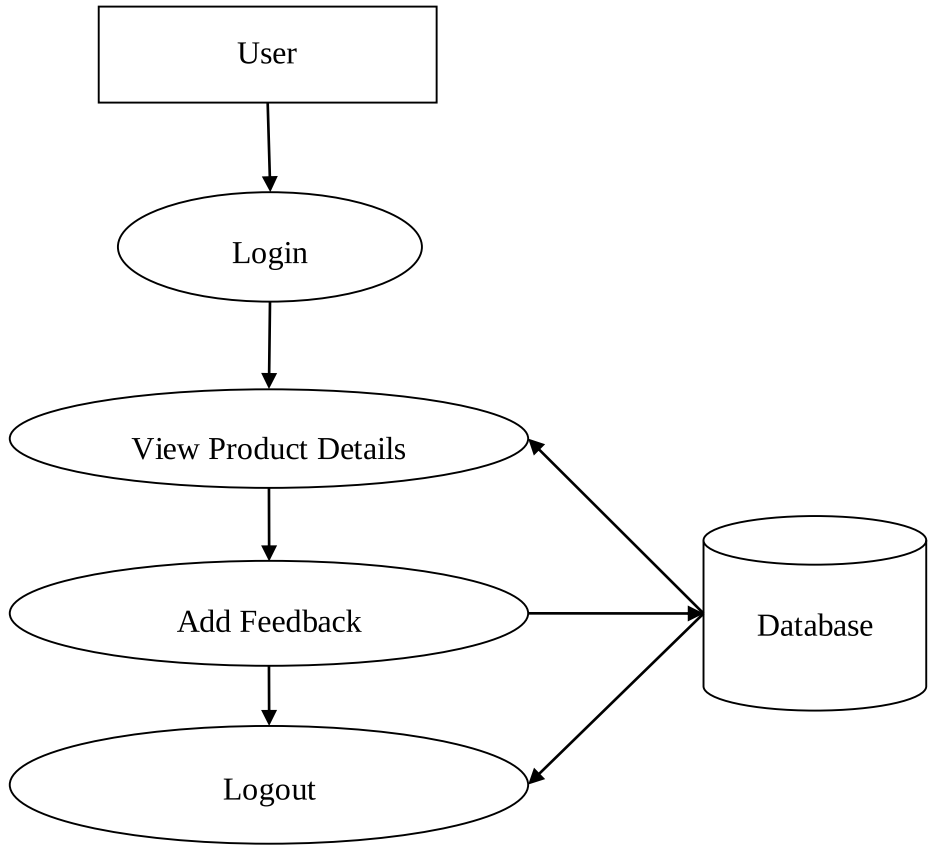


FIGURE 4.4

Level 1:


Admin
Login
Add Product List
View Product List
View Feedback and Opinion Mining
Report
Database

Level 2:



User
Login
View Product Details
Add Feedback
Logout
Database

4.1.6. SEQUENCE DIAGRAM

A sequence diagram in UML is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a message sequence chart. Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams.

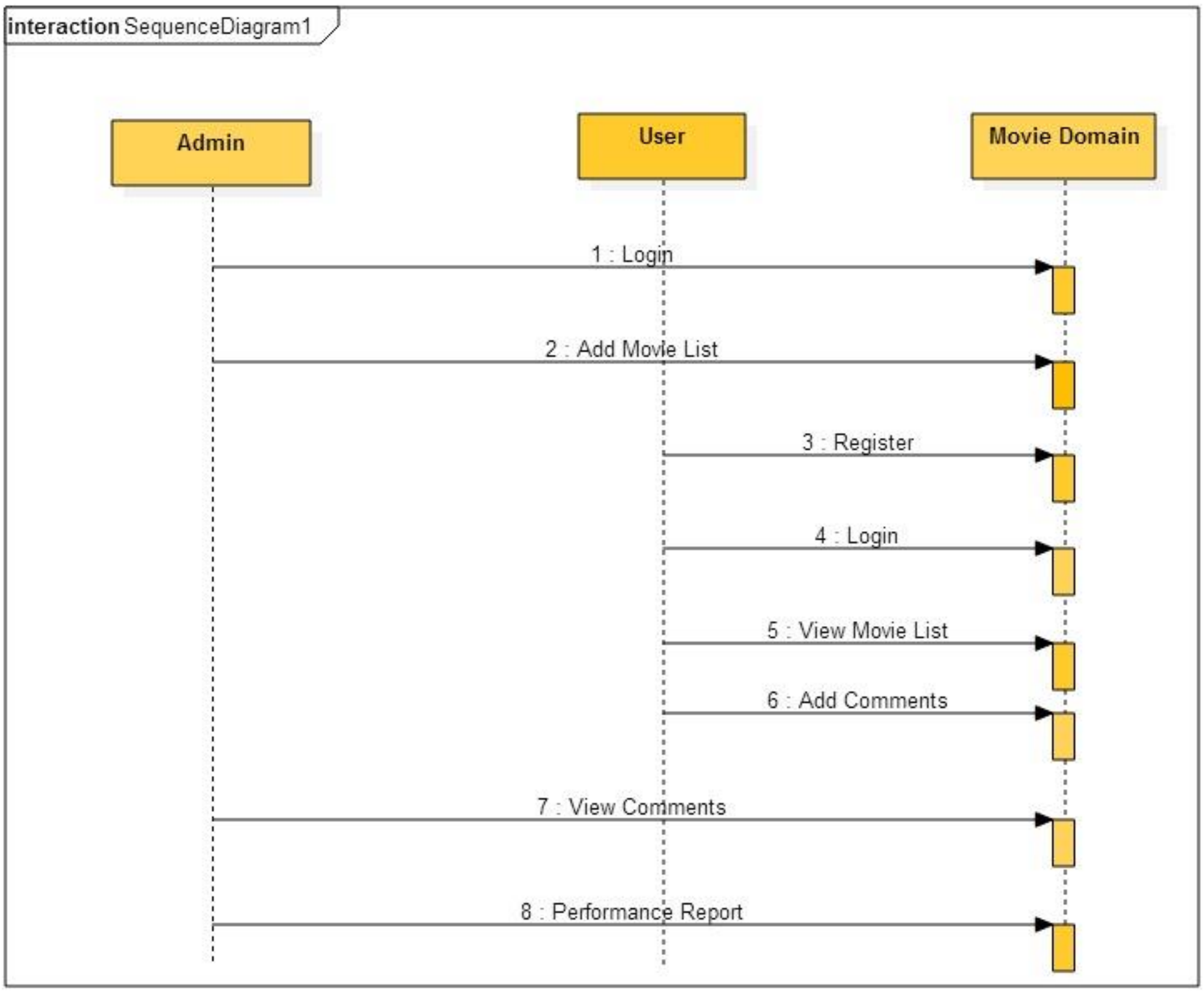


FIGURE 4.5

CHAPTER 5

SYSTEM CONFIGURATION

5.1 GENERAL

Using this requirement, the application provides high service with efficiently. Software requirements deal with defining software resource requirements and pre-requisites that need to be installed on a server that provide optimal functioning of an application.

5.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- Processor : Pentium Dual Core 2.00GHZ
- Hard disk : 40 GB
- Mouse : Logitech.
- RAM : 2GB(minimum)
- Keyboard : 110 keys enhanced.

5.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Operating system : Windows Family
- IDE : Net beans IDE
- Coding Language : Java
- Back end : MySQL

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Introduction

Comments that receive a very low score are typically hidden, while comments with a higher score are highlighted, allowing the user to easily reach quality commentary. In addition to the numerical rating posts we can also be given a rating description such as “Insightful” if it is good and “Offtopic” if it is bad, among others.

6.2 Modules Name

- ✓ **Weblogs**
- ✓ **Sentiment Analysis**
- ✓ **Product Reviews**
- ✓ **Summarization**
- ✓ **Sentiment Classification**

MODULES DESCRIPTION

Weblogs

Weblogs provide online forums for discussion that record the voice of the public. Woven into this mass of discussion is a wide range of opinion and commentary about consumer options. This presents an opportunity for companies to understand and respond to the consumer by analyzing this unsolicited feedback. For each product, we collected all relevant weblog posts appearing in the Blog pulse index. A post was considered “relevant” to a product if the following conditions hold:

• The date of the post is within a window starting a month prior to the product’s opening weekend date and ending one month after it.

• The post contained a link to the product's IMDB page, or the exact product name appeared in the post in conjunction with one of the words hproduct, watch, see, filmi (and their morphological derivatives).

Sentiment Analysis

Sentiment analysis is a type of natural language processing which tracking the mood of public about particular product or service. In earlier days, when we wanted to purchase any product from the merchants we asked those of our relatives for their opinions who had knowledge about that product. Sentiment analysis widely used in business application to determine their product quality and maintaining their reputation in the market

Product Reviews

In this module implemented support vector machine and random forest classifier for sentiment classification of product reviews into positive and negative review classes. As a result, Random forest classification technique is the best mean for mining and summarization of product reviews. It gives better accuracy than other machine learning techniques. In feature-based summarization, identification of product features and opinion words plays an essential role.

Summarization

Detection of sentiment is an important technology for applications in business intelligence, where customer reviews, customer feedback; survey responses, newsgroup postings, etc. are automatically processed in order to extract summary information

Sentiment Classification

The sentiment classification problem requires large amounts of labeled training data. Acquisition of these labeled data can be time-consuming and expensive. Sentiment classification ought to be able to address fairly sophisticated issues - identifying the object of sentiment, detecting mixed and overlapping sentiments in a text, identifying and dealing with sarcasm, etc. In practice, most work to date has been concerned with the less ambitious goal of identifying the overall polarity of sentiment in a document, i.e. whether the writer is expressing positive or negative opinions.

CHAPTER 7

SOFTWARE DESCRIPTION

Features of the language

Java is a small, simple, safe, object oriented, interpreted or dynamically optimized, byte coded, architectural, garbage collected, multithreaded programming language with a strongly typed exception-handling for writing distributed and dynamically extensible programs.

Java is an object oriented programming language. Java is a high-level, third generation language like C, FORTRAN, Small talk, Pearl and many others. You can use java to write computer applications that crunch numbers, process words, play games, store data or do any of the thousands of other things computer software can do.

Special programs called applets that can be downloaded from the internet and played safely within a web browser. Java a supports this application and the follow features make it one of the best programming language.

- It is simple and object oriented
- It helps to create user friendly interfaces.
- It is very dynamic.
- It supports multithreading.
- It is platform independent
- It is highly secure and robust.
- It supports internet programming

Java is a programming language originally developed by Sun Microsystems and released in 1995 as a core component of Sun's Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code which can run on any Java virtual machine (JVM) regardless of computer architecture.

Java Virtual Machine

The heart of the Java Platform is the concept of a "virtual machine" that executes Java bytecode programs. This bytecode is the same no matter what hardware or operating system the program is running under. There is a JIT compiler within the Java Virtual Machine, or JVM. The JIT compiler translates the Java bytecode into native processor instructions at run-time and caches the native code in memory during execution.

Class libraries

In most modern operating systems, a large body of reusable code is provided to simplify the programmer's job. This code is typically provided as a set of dynamically loadable libraries that applications can call at runtime. Because the Java Platform is not dependent on any specific operating system, applications cannot rely on any of the existing libraries

Platform independent

One characteristic, platform independence, means that programs written in the Java language must run similarly on any supported hardware/operating-system platform. One should be able to write a program once, compile it once, and run it anywhere.

Automatic memory management

One of the ideas behind Java's automatic memory management model is that programmers be spared the burden of having to perform manual memory management. In some languages the programmer allocates memory for the creation of objects stored on the heap and the responsibility of later deal locating that memory also resides with the programmer.

Performance

Java's performance has improved substantially since the early versions, and performance of JIT compilers relative to native compilers has in some tests been shown to be quite similar.

Java Runtime Environment

The Java Runtime Environment, or JRE, is the software required to run any application deployed on the Java Platform. End-users commonly use a JRE in software packages and Web browser plugins. Sun also distributes a superset of the JRE called the Java 2 SDK (more commonly known as the JDK), which includes development tools such as the Java compiler, Javadoc, Jar and debugger.

CHAPTER 8

SOFTWARE TESTING

8.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

8.3.1. Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated.

8.3.2. Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

8.3.3. System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.It ensures that the entire integrated software system meets requirements.

8.3.4. Performance Testing

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

8.3.5. Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

8.3.6. VALIDATION

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing.

1) The function or performance characteristics confirm to specification and are accepted.
2) A deviation from specification is uncovered and a deficiency lists is created.
3) Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

Require field Validation: User input the all require field for data validate.

Compare Validation: Compare the two fields of the input data.

Custom Validation: We can apply here the own validation.

Regular Expression Validation: We can use for control to validate the input class. You can use regular Expression to restrict the range of valid characters, to strip unwanted characters, and to perform length and format checks. We can constrain the input format by defining patterns the input must match.

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 Conclusion

This work defines the different approaches for finding manipulated review and detects the false reputation in online rating system. In today's world of e-commerce, there is a strong need of identifying fake reviews. More focus is given on the behavior of the reviewer and different text properties of comments. The proposed method employs decision tree algorithm to classify manipulated reviews. Decision tree is used to select the features which will give maximum accuracy. To solve the false reputation problem, a general framework is proposed that quantifies the confidence of a rating based on activity, objectivity, and consistency.

9.2 Future Work

Comparing performance of different classification methods to find the best one for our proposed opinion spam classification method could be another future research direction. However, there exist other kinds of review or reviewer related features that are likely to make a contribution to the prediction task. In the future different kinds of features will be investigated to make more accurate predictions. There is a Future scope deal with a larger dataset with more reviews attributed to each reviewer, which would make these reviewer-centric features more powerful. Moreover, many other reviewer-centric feature, including geo-locations and IP addresses, can also be incorporated in this model.

APPENDIX

A.SAMPLE CODE

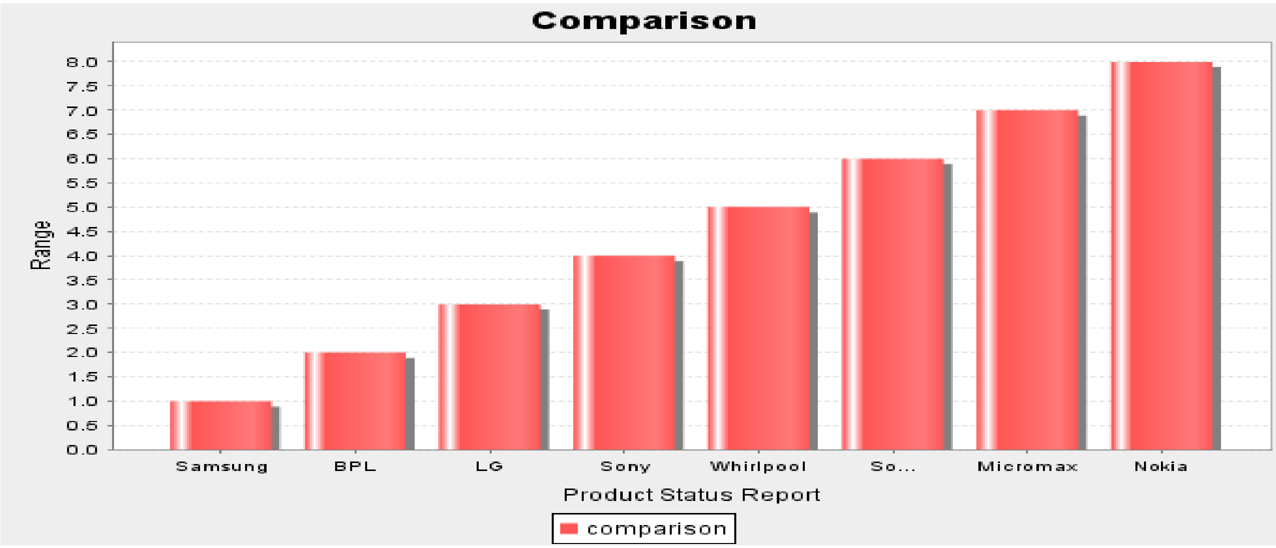
```
* To change this template, choose Tools | Template
 * and open the template in the editor.
function Register validation()
{
if(document.Register.name.value=="")
{
document.Register.name.focus();
document.getElementById("ErrorRow").style.display = "";
document.getElementById("ErrorMessage").innerHTML = "Name field is Empty..";
return false;
}
 if(document.Register.email.value=="")
{
document.Register.email.focus();
document.getElementById("ErrorRow").style.display = "";
document.getElementById("ErrorMessage").innerHTML = "Email field is Empty..";
return false;
}
 If(document.Register.contact.value=="")
{
document.Register.contact.focus();
document.getElementById("ErrorRow").style.display = "";
document.getElementById("ErrorMessage").innerHTML = "Contact field is Empty..";
document.getElementById("ErrorMessage").innerHTML = "Enter Encryption key..";
```

```
function validation()
{
if(document.EncryptionForm.FileName.value=="")
{
document.EncryptionForm.FileName.focus();
document.getElementById("ErrorRow").style.display = "";
document.getElementById("ErrorMessage").innerHTML = "Select your file..";
return false;
}
if(document.EncryptionForm.KeyString.value=="")
{
document.EncryptionForm.KeyString.focus();
document.getElementById("ErrorRow").style.display = "";
return false;
}
return true;
}
function SharePopup(ID,DocId,DocName)
{
if(document.getElementById(ID).style.display=="none")
{
document.getElementById("DocId").value = DocId;
document.getElementById("Doc_Name").value = DocName;
document.getElementById("DocName").innerHTML = DocName;
Popup.showModal(ID);
```

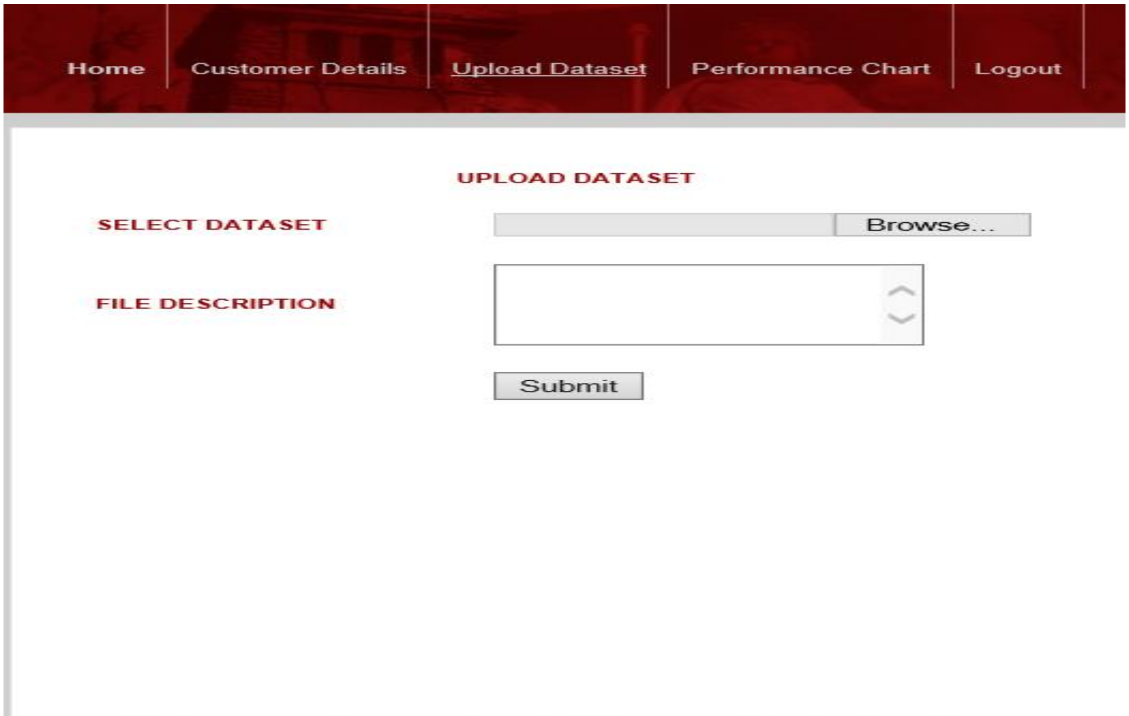
```
}
else
{
document.getElementById("DocId").value = DocId;
document.getElementById("Doc_Name").value = DocName;
document.getElementById("DocName").innerHTML = DocName;
Popup.hide(ID);
}
}
function Loginvalidation()
{
if(document.Login.UserName.value=="")
{
document.Login.UserName.focus();
document.getElementById("ErrorRow").style.display = "";
document.getElementById("ErrorMessage").innerHTML = "Enter Your UserName..";
return false;
}
if(document.Login.Password.value=="")
{
document.Login.Password.focus();
document.getElementById("ErrorRow").style.display = "";
document.getElementById("ErrorMessage").innerHTML = "Enter Your Password..";
return false;
}
return true;    }
```

B.SCREENSHOTS

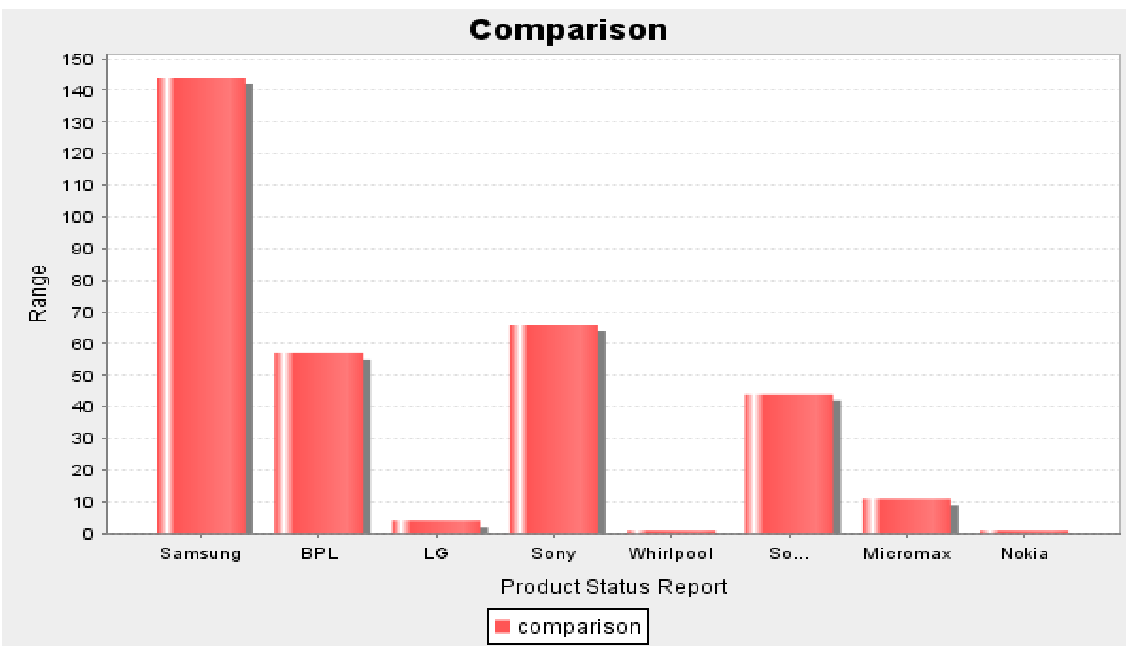
DIFFERENT RANGES OF THE PRODUCT BEFORE PREDICTION:



Name	E-Mail	Address	City	Credit Card No.
anu	anu@gmail.com	trichy	trichy	12345678910
gur	guruuu@gmail.com	asefa	trichy	5614325174
guru	guru@gmail.com	651	trichy	689416
SyedMusthafa	syed.k@oculusit.in	Chintamani	Trichy	123
veera	veera@gmail.com	TV Koil	Trichy	111
vikram	vikram@gmail.com	Palakari	Trichy	123



DIFFERENT RANGES OF THE PRODUCT AFTER PREDICTION:



REFERENCES

[1] Abbasi, A., Chen, H., & Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. ACM Transactions on Information Systems, 26(3), 12–34.

[2] Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. Association for Computational Linguistics, 7, 440–447.

[3] Cacheda, F., Carneiro, V., Ferna´ndez, D., & Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Transactions on the Web, 5(1), 2–33.

[4] El-Be`ze, M., Jackiewicz, A., & Hunston, S. (2010). Opinions, sentiments et jugements d'e´valuation. Traitement Automatique des Langues, 51(3), 7–17.

[5] Hu, N., Liu, L., & Zhang, J. (2008). Do online reviews affect product sales? The role of reviewer characteristics and temporal effects. Information Technology and Management, 9(3), 201–214.

[6] Koutrika, G., Bercovitz, B., Garcia-Molina, H. (2009). FlexRecs: Expressing and combining flexible recommendations. In ACM SIGMOD international conference on management of data (pp. 745–758).

[7] Lee, J., Park, D. H., & Han, I. (2011). The different effects of online consumer reviews on consumers' purchase intentions depending on trust in online shopping malls: An advertising perspective.
Internet Research, 21(2), 187–206.

[8] Li, J., Ott, M., Cardie, C., & Hovy, H. (2014). Towards a general rule for identifying deceptive opinion spam. In ACL (Vol. 1, pp. 1566–1576).