

Project Report

on

PREDICTING USEFUL PRODUCT REVIEWS ON E-COMMERCE WEBSITE USING SENTIMENTAL ANALYSIS

In partial fulfilment of requirements for the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by:

AMISHA VERMA [1710DMBIT01518]

DEVANSH KALA [1710DMBIT01528]

TARUN SANGHVI [1710DMBIT01564]

YASH JADON [1710DMBIT01574]

Under the guidance of

PROF. JITENDRA SHARMA

PROF. SUMIT NIGAM



**SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA,
INDORE**

**SHRI VAISHNAV INSTITUTE OF INFORMATIN TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY**

JULY-DEC 2020

**SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA,
INDORE**

SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

DECLARATION

We here declare that work which is being presented in the project entitled “**PREDICTING USEFUL PRODUCTS REVIEWS ON E-COMMERCE WEBSITE USING SENTIMENTAL ANALYSIS**” in partial fulfilment of degree of **Bachelor of Technology in Information Technology** is an authentic record of our work carried out under the supervision and guidance of **Prof. JITENDRA SHARMA** and **Prof. SUMIT NIGAM** Asst. Professor of Information Technology. The matter embodied in this project has not been submitted for the award of any other degree.

AMISHA VERMA
DEVANSH KALA
TARUN SANGHVI
YASH JADON

Date:

**SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA,
INDORE**

SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

PROJECT APPROVAL SHEET

Following team has done the appropriate work related to the “**PREDICTING USEFUL PRODUCTS REVIEWS ON E-COMMERCE WEBSITE USING SENTIMENTAL ANALYSIS**” in partial fulfillment for the award of **Bachelor of Technology in Information Technology** of “SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY” and is being submitted to SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE.

Team:

- 1. AMISHA VERMA**
- 2. DEVANSH KALA**
- 3. TARUN SANGHVI**
- 4. YASH JADON**

Internal Examiner

External Examiner

Date:

**SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA,
INDORE**

SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that **Ms. AMISHA VERMA, Mr. DEVANSH KALA, Mr. TARUN SANGHVI and Mr. YASH JADON** working in a team have satisfactorily completed the project entitled “**PREDICTING USEFUL PRODUCTS REVIEWS ON E-COMMERCE WEBSITE USING SENTIMENTAL ANALYSIS**” under the guidance of **Prof. JITENDRA SHARMA** and **Prof. SUMIT NIGAM** in the partial fulfilment of the degree of **Bachelor of Technology in Information Technology** awarded by SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY affiliated to SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE during the academic year **July 2020-Dec 2020**.

Prof. JITENDRA SHARMA

Prof. SUMIT NIGAM

Project Guide

Prof. Manish Kumar

Project Coordinator

Dr. Jigyasu Dubey

Head, Department of Information Technology

ACKNOWLEDGEMENT

We are grateful to a number of persons for their advice and support during the time of complete our project work. First and foremost our thanks goes to **Dr. Jigyasu Dubey** Head of the Department of Information Technology and **Prof. JITENDRA SHARMA** and **Prof. SUMIT NIGAM** the mentor of our project for providing us valuable support and necessary help whenever required and also helping us explore new technologies by the help of their technical expertise. His direction, supervision and constructive criticism were indeed the source of inspiration for us.

We would also like to express our sincere gratitude towards our Director **Dr. Anand Rajavat** for providing us valuable support.

We are really indebted to **Prof. Manish Kumar**, project coordinator for helping us in each aspect of our academics activities. We also owe our sincere thanks to all the **faculty members** of Information Technology Department who have always been helpful.

We forward our sincere thanks to all **teaching and non-teaching staff** of Information Technology department, SVVV Indore for providing necessary information and there kind co-operation.

We would like to thanks our parents and family members, our classmates and our friends for their motivation and there valuable suggestion during the project. Last, but not the least, we thank all those people, who have helped us directly or indirectly in accomplishing this work. It has been a privilege to study at SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE.

ABSTRACT

The social web has made enormous amounts of information available to users globally at just the click of a button. Consumers often tend to rely on such text, especially those in the form of opinions or experiences regarding a particular product which makes it essential that this information should be available in a systematic manner. Sentiment analysis studies these opinions. This paper explains different methods for sentiment analysis and showcases an efficient methodology. It also highlights the importance of Naïve Bayes classifier over other classification algorithms.

The product reviews from the users play a big role in the financial aspects for these big companies. We can consider the product reviews as a form of passive recommendation process or exposure of user sentiment for their purchase history. Online reviews are a category of product information created by the users based on personal handling experience.

The rise in E - commerce has brought a significant rise in the importance of customer reviews. Customers have changed their way of shopping and according to a recent survey, 70 percent of customers say that they use rating filters to filter out low rated items in their searches.

Sentiment analysis or opinion mining is nothing but classification of emotions in the reviews text into positive, negative and neutral. Opinion mining is a method of information extraction from text processing to improve or develop the business work by review analysis. Comparisons show the trained models align well with human perceptions and results on reviews with more votes are slightly better aligned. Semantic interpretation reveals that online customers prefer reviews that contain words that have positive/negative comments on the object described, that show reviewers are thoughtful and knowing when commenting on the object, and that emphasize emphasis on personal experience and positive emotions.

In this project, we aim to perform Sentiment Analysis of product-based reviews. Data used in this project are online product reviews collected from “amazon.com” and “kaggle”

List OF Figures

Sr. No.	Figure Name	Page No.
1	Fig. 4.1 Use Case Diagram	17
2	Fig. 4.2 Data Flow Diagram Level-0	20
3	Fig. 4.3 Data Flow Diagram Level-1	21
4	Fig. 4.4 Data Flow Diagram Level-2	21
5	Fig. 4.5 ER-Diagram	22
6	Fig. 5.1 Class Diagram	24
7	Fig. 5.2 Sequence Diagram	26
8	Fig. 5.3 Collaboration Diagram	28
9	Fig. 5.4 Activity Diagram	29
10	Fig. 5.5 Object Diagram	30
11	Fig. 5.6 Training data, dividing data into positive & Negative data, finding tokens, shaping data & label tensor	35
12	Fig. 5.7 Training LSTM model for Product Review	36
13	Fig. 5.8 Value of Model Performance Evaluation	36

TABLE OF CONTENTS

Declaration	I
Project Approval Sheet	II
Certificate	III
Acknowledgment	IV
Abstract	V
List of Figures	VI
CHAPTER 1 – INTRODUCTION	1-4
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Need for the proper System	2
1.4 Objective	3
1.5 Modules of the system	3
1.6 Scope	4
CHAPTER 2 - LITERATURE SURVEY	5-9
2.1 Existing System	5
2.2 Proposed System	7
2.3 Feasibility Study	8
2.3.1 Technical Feasibility	8
2.3.2 Economical Feasibility	9
2.3.3 Operational Feasibility	9
CHAPTER 3 – REQUIREMENTS ANALYSIS	10-14
3.1 Method used for Requirement analysis	10
3.2 Data Requirements	11
3.3 Functional Requirements	11
3.4 Non Functional Requirements	12
3.5 System Specification	13
3.5.1 Hardware specification	13
3.5.2 Software specification	13
CHAPTER 4 – DESIGN	14-22
4.1 Software Requirements Specification	14
4.1.1 Glossary	14
4.1.2 Use Case Model	16
4.2 Conceptual level class diagram	18
4.3 Conceptual level activity diagram	18
4.4 Data flow Diagram(Level 0,1,2)	19
4.5 Database Design (ER-Diagram)	22
CHAPTER 5 – SYSTEM MODELING	23-36
5.1 Detailed Class Diagram	23
5.2 Interaction Diagram	25
5.2.1 Sequence Diagram	25
5.2.2 Collaboration Diagram	27

5.3 Activity Diagram	28
5.4 Object Diagram	30
5.6 Test Plan & Implementations	31
5.6.1 Testing	31
5.6.2 Testing Objectives	31
5.6.3 Testing Methodologies	32
5.6.4 Test Plan	33
5.6.5 Screenshots	35
CHAPTER 6 – CONCLUSION & FUTURE WORK	37-39
6.1 Limitation of Project	37
6.2 Future Enhancement	38
CHAPTER 7 - BIBLIOGRAPHY & REFERENCES	40-41
7.1 Reference Books	40
7.2 Other Documentations & Resources	40

Chapter 1

INTRODUCTION

1. INTRODUCTION

1.1 INTRODUCTION

Opinions of others on a particular product can have influence on our decision. Online marketplaces have rapidly grown their popularity among customers over the last few years. For instance, Amazon earned sales revenue of \$107 billion in 2015. Many consumers rely on online reviews for direct information to make purchase decisions. However, a large number of reviews for just one single product have made it impractical for consumers to read all the reviews and assess the true quality of a product.

In addition, the quality and the helpfulness of each review also fluctuate. The large quantity of the reviews and their serrated quality make it tough for consumers to differentiate between useful and useless reviews.

Public opinion plays a vital role in business organization to market the products, venture new opportunities and for sales prediction.

A large amount of data can be analysed and prediction of opinion is possible using sentiment analysis techniques which helps the customers and business organisation.

The product reviews from the users play a big role in the financial aspects for these big companies. We can consider the product reviews as a form of passive recommendation process or exposure of user sentiment for their purchase history. Online reviews are a category of product information created by the users based on personal handling experience.

The rise in E —commerce has brought a significant rise in the importance of customer reviews. Customers have changed their way of shopping and according to a recent survey,

70% of customers say that they use rating filters to filter out low rated items in their searches.

Sentiment analysis or opinion mining is nothing but classification of emotions in the reviews text into positive, negative and neutral. Opinion mining is a method of information extraction from text processing to improve or develop the business work by review analysis.

In this project, we aim to perform Sentiment Analysis of product-based reviews. Data used in this project are online product reviews collected from “amazon.com” and “kaggle”. We expect to do review-level categorization of review data with promising outcomes.

1.2 PROBLEM STATEMENT

To develop back end software which can predict the usefulness of the reviews and ratings given by customers on any product? The major driving force of this recent boost in purchase behaviour is vastly influenced by the customer reviews. One has different types of products available in the market. It is impossible to go through each review manually to buy a new product so sentimental analysis should help to differentiate good product from other through analysis of the reviews that are submitted by the customers. Users tend to verify the comments from other users who already purchased the product and shared their thoughts. However, the available reviews are not all useful to take the decision. Gathering only the most helpful reviews would reduce information processing time and save efforts.

1.3 NEED OF PROPER SYSTEM

Integrating semantic characteristics into the model along with basic and stylistic characteristics greatly improve the performance of the model. Semantic characteristics play a very important role in influencing the number of helpfulness votes a review gets. Which will give direct result of sentiments? It will help users gather the best information about any particular product they desire, based on other customers reviews, and help in making a decision about any product.

By this software, we intent to provide the user with relevant and truth full reviews based on the previous customers buying experiences which will help the current user to save his/her time and get the creditability of the product he is going to purchase. This backend software acts as plug-in in e- commerce websites or web applications to enhance the user experience.

Naive Bayes classifier applies the probabilities of each attribute fit in to every class to make prediction. Python has natural language toolkit (NLTK) for text processing and classification. Our project, we have used the LSTM – Long Short Term Memory to train and test the model in order to get the maximum accuracy and precision.

1.4 OBJECTIVE OF THE PROJECT

The main objective of this project is to go about an extra mile to provide the users with an output that is the analysis of thousands and thousands of reviews.

To save time by analyzing thousands of reviews in short period and if those reviews were analyzed manually it may take up to decades.

The overall output is achieved by several steps like:

Step 1: Gathering the raw data from different repositories.

Step 2: Pre-processing the raw data in a usable format.

Step 3: Fetching the data set file using R language.

Step 4: Identifying the sentiment words in the reviews.

Step 5: Maintaining a count of different type of words.

Step 6: Data visualization in a bar graph.

1.5 MODULES OF THE SYSTEM

The hardware requirement would be a laptop with minimum 8 GB of RAM.

The software specifications include:

Software Technology and Environment

Python 3.x: Python is a powerful programming language. It has efficient high-level data structures and effective approach to object-oriented programming.

Anaconda Distribution: **Anaconda** is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.)

Tools -

NLTK: NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces, along with a suite of text processing libraries for classification, tokenization, stemming, and tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

TextBlob: TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

1.6 SCOPE

By this software, we intent to provide the user with relevant and truthful reviews based on the previous customers buying experiences which will help the current user to save his/her time and get the creditability of the product he is going to purchase. This backend software acts as plug-in in e- commerce websites or web applications to enhance the user experience.

Chapter 2

LITERATURE SURVEY

2. LITERATURE SURVEY

The purpose of a literature review is to provide foundation of knowledge on topic and identify the relationship of works in context of its contribution to the topic and to other works.

2.1 EXISTING SYSTEM

Mohan Kamal Hassan at 2017 IOP Conference presented his research paper on **Sentimental analysis of Amazon reviews using Naïve Bayes on laptop products with MongoDB and R**. In this paper, they used Naïve Bayes algorithm and semantic decision tree to classify the polarity of comments given on e-commerce websites. First, we use a web crawler to fetch comment on a particular web page. The spelling correction is done to make the most sensible comment for knowing the polarity of words using Word Net dictionary. Then stemming is performed to remove the stop words. After classifying the positive and negative words using Naïve Bayes algorithm, the overall polarity is calculated using decision tree.

Two typical approaches to sentiment analysis –lexicon look up and machine learning are used by Ji Fang and Bi Chen in. Lexicon look up starts with a lexicon of positive and negative words. Current sentiment lexicons do not capture such domain and context sensitivities of sentiment expressions. The proposed system present an alternative method that incorporates sentiment lexicons as prior knowledge with machine learning approaches such as SVM to improve the accuracy of sentiment analysis.

Sasikala P, L.Mary Immaculate Sheela in their research paper **Sentiment Analysis and Prediction of Online Reviews with Empty Ratings** presented a model which predicts the opinion from the reviews without ratings. It is observed that the logistic regression and

Naïve Bayes predictions of opinions are much similar than the multinomial and Bernouli classifiers. In few cases logistic regression performance is better than the Naïve Bayes.

Xingyou Wang, Weijie Jiang, Zhiyong Luo in their research paper published **Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts** in which they presented deep neural network architecture that takes advantage of the construction of convolutional neural network (CNN) and recurrent neural network (RNN) and joint them together for sentimental analysis of short texts. In particular, their pooling operation on adjacent words is able to retain the local features and their sequential relations in a sentence. Besides, RNN can learn the long-term dependencies and the positional relation of features as well as the global features of the whole sentence.

Sentiment analysis using product review data was published by Fang and Zhan Journal of Big Data. This paper tackles a fundamental problem of sentiment analysis, sentiment polarity categorization. Online product reviews from Amazon.com were selected as data used for this study. Software used for this study is scikit - learn an open source machine learning soft-ware package in Python. The classification models selected for categorization are: Naïve Bayesian, Random Forest, and Support Vector Machine.

Combining Review Text Content (RTC) and User Similarity Matrix to obtain more information and to improve the Review Rating Prediction published by Towards Data Science basically involves a combined approach for review rating. In an effort to obtain more information and to improve the prediction of the review rating, the researchers in the article proposed a framework combining review text content with previous user's similarity matrix analysis. They then did some experiments on two movie review datasets to examine the efficiency of their hypothesis. The results that they got showed that their framwork indeed improved prediction of the review rating. This article will describe attempt of following the work done in their research through examples from the Amazon reviews dataset.

Sentiment Analysis for Amazon Reviews presented by Wanliang Tan, Xinyu Wang and Xinyu Xu tried two types of features. For this two type of features, we tried all the algorithms we mentioned in the model part including Naive Bayes, SVM, KNN, LSTM. From the results, we can see that our accuracy on the test set is the best when we use

LSTM on the first type of feature. One of the main reasons our accuracy is not high enough is because of the data imbalance. We tried resampling and different weighting techniques that we got from the feedbacks of the audience during the poster session.

2.1.1 COMPARISON CHART

	Amazon	ebay	Alibaba	Flipkart	Walmart
Main Comparing Feature	Based on star rating and keywords and number of views per review given by user	Based on star rating and keywords	Based on star rating and keywords	Based on star rating and keywords	Based on star rating and keywords
Benefits	Enhances user experience	Enhances user experience	Enhances user experience	Enhances user experience	Enhances user experience
Tool used	Amazon Comprehend	NA due to privacy	NA due to privacy	NA due to privacy	NA due to privacy
Model	2 level CRF, Semantic	SVM, Chi-square			
Polarity	General, NLP	General, NLP	General, NLP	General, NLP	General, NLP
DB Engine Used	Amazon RDS	Neo4j Graph DB.	Alibaba Cloud Apsara DB	Flipkart Data Platform	Neo4j Graph DB.
Software Requirements	Python 3.x needed.	Python 3.x needed.	Python 3.x needed.	Python 3.x needed.	Python 3.x needed.
Scalability	yes	yes	yes	yes	yes
Disadvantages	Emoticons, spams, false negatives. User dependent.	Emoticons, spams, false negatives. User dependent.	Emoticons, spams, false negatives. User dependent.	Emoticons, spams, false negatives. User dependent.	Emoticons, spams, false negatives. User dependent.

2.2 PROPOSED SYSTEM

The existing software which works on various other algorithms are quite efficient but lacks accuracy in the terms of model training and testing.

Hence in our project, we have used the LSTM –Long Short Term Memory to train and test the model in order to get the maximum accuracy and precision.

2.3 FEASIBILITY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

- To analyse whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

Various types of feasibility that are commonly considered include:

- Technical feasibility
- Operational feasibility
- Economic feasibility.

2.3.1 TECHNICAL FEASIBILITY

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements.

This project is technically feasible as the requirement is a laptop with minimum 8GB RAM. The other requirements are as such: *Python 3.x, Anaconda Distributions, NLTK, TextBlob, etc.*

2.3.2 ECONOMICAL FEASIBILITY

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software.

This project is economically feasible as the only investment is a **laptop** or a computer system.

2.3.3 OPERATIONAL FEASIBILITY

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed.

In this project the customer feels satisfied with the system being able to successfully filter useful feedbacks and reviews of the products they are interested in and the user does not need to do much for its operation. This project developed is worthy and solution to the problem will work successfully.

Chapter 3

REQUIREMENT ANALYSIS

3. REQUIREMENT ANALYSIS

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. Requirements analysis involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore, requirements analysis means to analyse, document, validate and manage software or system requirements. High-quality requirements are documented, actionable, measurable, testable, traceable, helps to identify business opportunities, and are defined to a facilitate system design.

3.1 METHOD USED FOR REQUIREMENT ANALYSIS

Analysis and Design Process covers two major phases of the software development lifecycle. The two phases are often overlapped due to the recursive nature of the requirements. The process usually starts with getting know-how of customer business processes and existing systems, understanding customer needs, expectations, constraints, and elicitation of requirements. Requirements conflicts are removed, and issues and concerns are addressed to develop a clear understanding of customer requirements.

These requirements are then used to define the functionality of a proposed system and to determine what the systems are intended to do. System functionality and architecture is documented, and after reviewing internally, communicated to the customer for the purpose of validation. This document then gets mature as a result of customer feedback and serves as the basis for further development.

3.2 DATA REQUIREMENT

- **Data set for processing.**
- **NLTK toolkit**
- **Sentimental words for processing**

3.3 FUNCTIONAL REQUIREMENT

The functional requirements of the Sentiment Analysis System for e-commerce website product Review are as follows:

- The system can manage (add, read, update, and delete) review datasets.
- The system can manage (add, read, update, and delete) the dictionary of root words
- The system can manage (add, read, update, and delete) the dictionary of stop words.
- The system can perform the classifier training process and display the model in the form of feature sets of the term data from the training data.
- The system can display the test data result and display confusion matrix generated from the classifier testing.
- The system can display a set of review dataset terms derived from tokenizing, filtering, and stemming processes.
- The system can display sentiment analysis result derived from reviews submitted by users.

System Feature	Sentiment Classification
Priority	High
Description	Identifying the sentiment polarity (positive, negative or neutral) of reviews.
Action	This module is activated after the user provides a query (review)
Result	The system shows the results of the search of a query associated with the sentiment polarity of each item retrieved together with the percentage of Positive, Negative and Neutral sentiment of the whole result.

Functional Requirements	A focused crawler, pre-processing module, sentiment classifier module, and sentiment visualization module.
-------------------------	--

3.4 NON FUNCTIONAL REQUIREMENT

Non-functional Requirements	Priority
Performance Requirements- Detecting if the system crashed, hanged or an operating system error occurred and efficiency of integration of the different components.	High
Safety Requirements- For the safety requirements nothing but an operation of weekly backups for the data base should take place.	Very High
Security and Privacy Requirements- There are no specific security requirements; anyone can access and use the portal but only authorized persons who are allowed to use and access the database, web pages and the product engine.	Very High
Reliability- The solution should provide reliability to the user that the product will run with all the features mentioned in this document is available and executing perfectly. It should be tested and debugged completely. All exceptions should be well handled	Medium
Accuracy- The solution should be able to reach the desired level of accuracy. But also keeping in mind that this prototype version is for proving the concept of the project.	High
Extensibility - the system should be able to implement new modules without performance issues	Very High
Efficiency - the system should be able to support large intakes of data without affecting its performance	High
Speed - The system should perform sentiment analysis in seconds from the time the request was made	Very High
Robustness - the system should be able to run multiple instances of engine	Medium

Scalability - the system should scale the output size based on the size of the input	Very high
---	-----------

3.5 SYSTEM SPECIFICATION

3.5.1 HARDEWARE SPECIFICATION

- Processor : Intel i5 or more
- RAM : 8GB or more
- Cache : 512 kb
- Hard Disk : 16 GB
- Disk Drive : 1.44MB floppy disk
- Monitor : 1024*720 Display
- Speed : 2.7GHz and more

3.5.2 SOFTWARE SPECIFICATION

The software requirements are description of features and functionalities of the target system. The software specifications include:

- **Software Technology and Environment**

Operating System: Windows 7/8/8.1/10

- **Python 3.x:**

Python is a powerful programming language. It has efficient high-level data structures and effective approach to object-oriented programming.

- **Anaconda Distribution:**

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.)

Tools

- **NLTK:**

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

- **TextBlob:**

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

Chapter 4

DESIGN

4. DESIGN

The design model provides details about software data structures, architecture, interfaces and components which are required to implement the system.

4.1 SOFTWARE REQUIREMENTS SPECIFICATION

Software requirement specification (SRS) is a technical specification of requirements for the software product. SRS represents an overview of products, features and summaries the processing environments for development operation and maintenance of the product. The goal of the requirement specification phase is to produce the software specification document also called requirement document.

Requirement Specification

This requirement specification must have the system properties. Conceptually Every SRS should have the components:

- Functionality
- Performance
- Design constraints imposed on an implementation
- External interfaces

4.1.1 GLOSSARY

- Python

It is a general purpose, dynamic, high level and interpreted programming language. It supports Object Oriented programming concept for developing

various applications for users. It is very simple and easy to program and python provides most of high-level data structures.

- Python IDLE

IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI toolkit.

- TextBlob

TextBlob is a python library and offers a simple API to access its methods and perform basic NLP tasks. A good thing about TextBlob is that they are just like python strings. So, you can transform and play with it same like we did in python.

- Python libraries

Python library is a collection of functions and methods that allows you to perform many actions without writing your code. Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

4.1.2 USE CASE MODEL

In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a role (known in the Unified Modelling Language as an actor) and a system, to achieve a goal. The actor can be a human or other external system.

Use case analysis is an important and valuable requirement analysis technique that has been widely used in modern software engineering since its formal introduction by Ivar Jacobson in 1992. Use case driven development is a key characteristic of many process models and frameworks such as ICONIX, the Unified Process (UP), the IBM Rational Unified Process (RUP), and the Oracle Unified Method (OUM). With its inherent iterative, incremental and evolutionary nature, use case also fits well for V-Model development. Use cases are not only texts, but also diagrams, if needed. The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose.

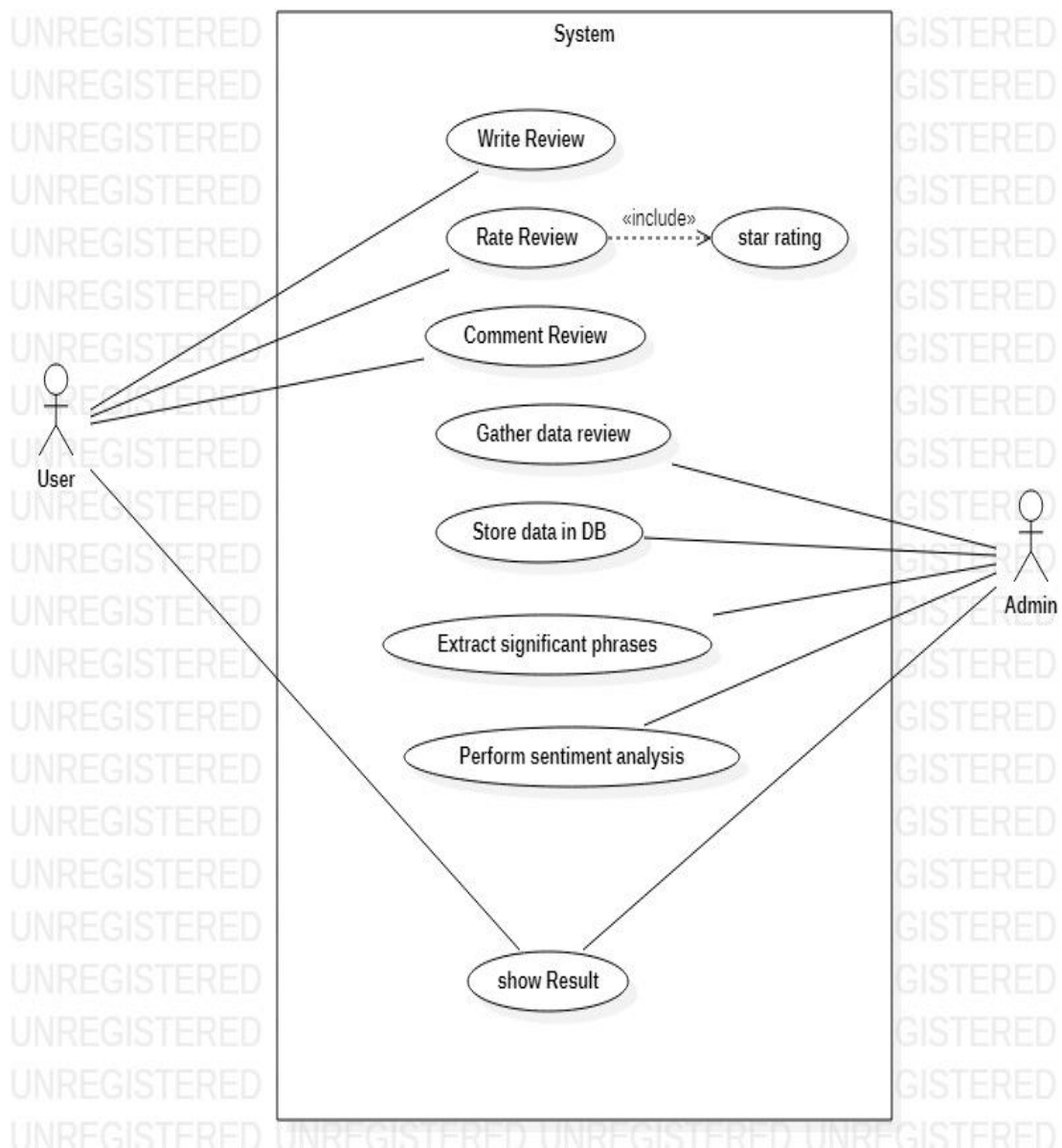


FIG 4.1 Use Case Diagram

4.2 CONCEPTUAL LEVEL CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modelling. It is used both for general conceptual modelling of the systematics of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centred, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

In the design of a system, a number of classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modelling, the classes of the conceptual design are often split into a number of subclasses.

4.3 CONCEPTUAL ACTIVIT DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Purpose of Activity Diagrams

The basic purpose of activity diagrams is similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as –

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

4.4 DATA FLOW DIAGRAM

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

Types of DFD

Data Flow Diagrams are either Logical or Physical.

Logical DFD - This type of DFD concentrates on the system process, and flow of data in the system. For example in a Banking software system, how data is moved between different entities.

Physical DFD - This type of DFD shows how the data flow is actually implemented in the system. It is more specific and close to the implementation.

DFD Components

DFD can represent Source, destination, storage and flow of data using the following set of components -

- **Entities** - Entities are source and destination of information data. Entities are represented by a rectangle with their respective names.
- **Process** - Activities and action taken on the data are represented by Circle or Round-edged rectangles.
- **Data Storage** - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.
- **Data Flow** - Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

➤ Levels of DFD

Level 0 - Highest abstraction level DFD is known as Level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details. Level 0 DFDs are also known as context level DFDs.

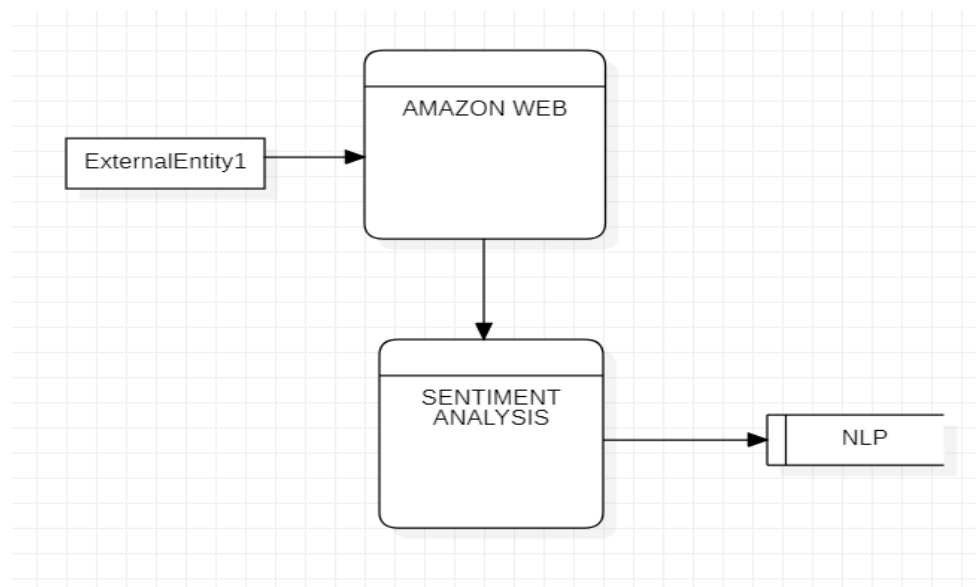


FIG 4.2 DFD LEVEL – 0

Level 1 - The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules.

Level 1 DFD also mentions basic processes and sources of information.

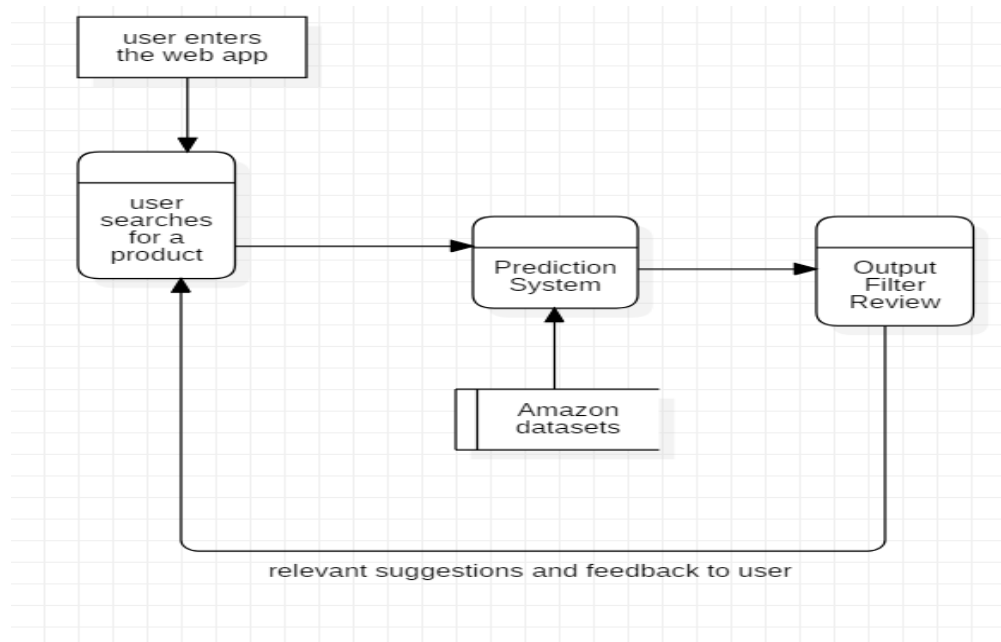


FIG 4.3 DFD LEVEL – 1

LEVEL – 2

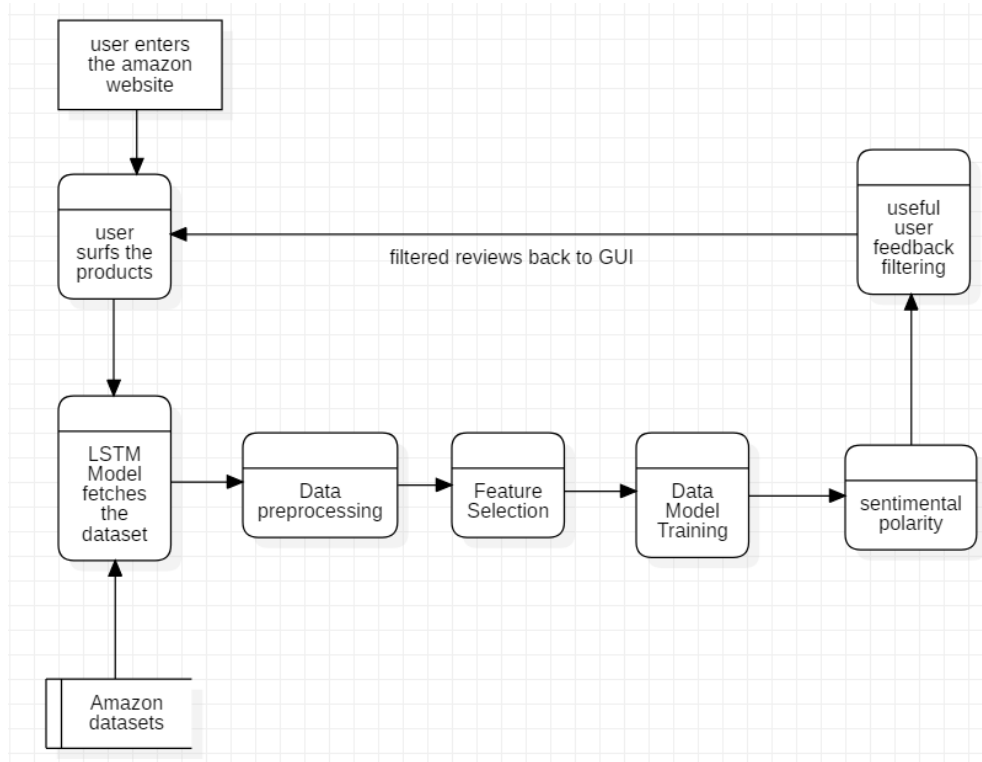


FIG 4.4 DFD LEVEL - 2

4.5 DATABASE DESIGN (ER-DIAGRAM)

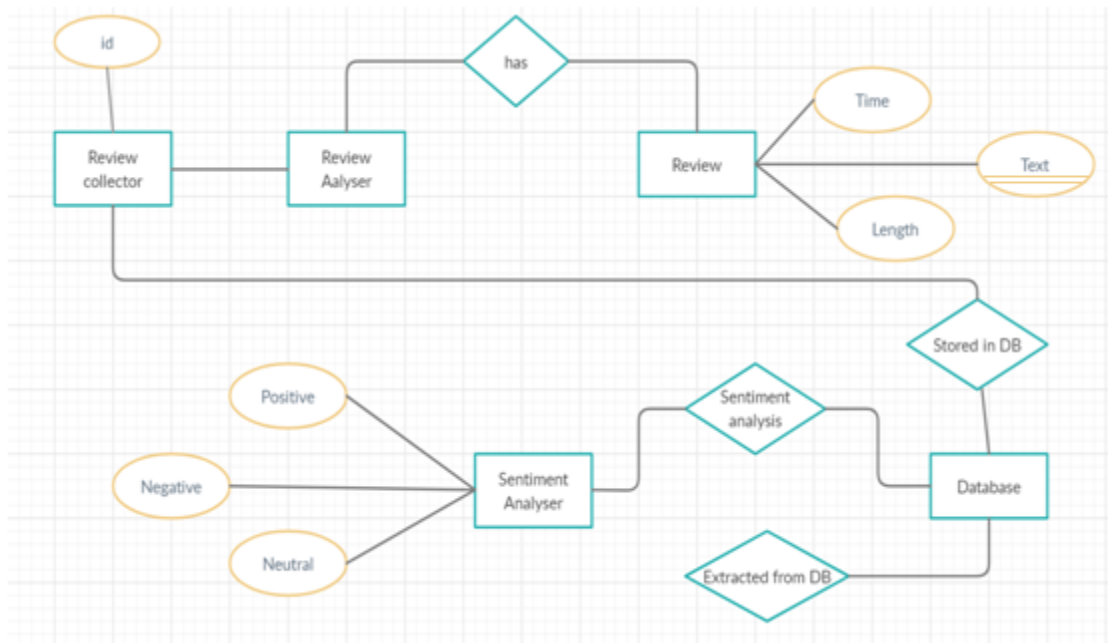


Fig. 4.5 ER DIAGRAM

Chapter 5

SYSTEM MODELING

5. SYSTEM MODELING

5.1. Detailed Class Diagram

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram. The purpose of the class diagram is to model the static view of an application.

The following points should be remembered while drawing a class diagram:

- The name of the class diagram should be meaningful to describe the aspect of the system.
- Each element and their relationships should be identified in advance.
- Responsibility (attributes and methods) of each class should be clearly identified.
- For each class minimum number of properties should be specified. Because unnecessary properties will make the diagram complicated.
- Use notes whenever required to describe some aspect of the diagram. Because at the end of the drawing it should be understandable to the developer/coder.
- Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

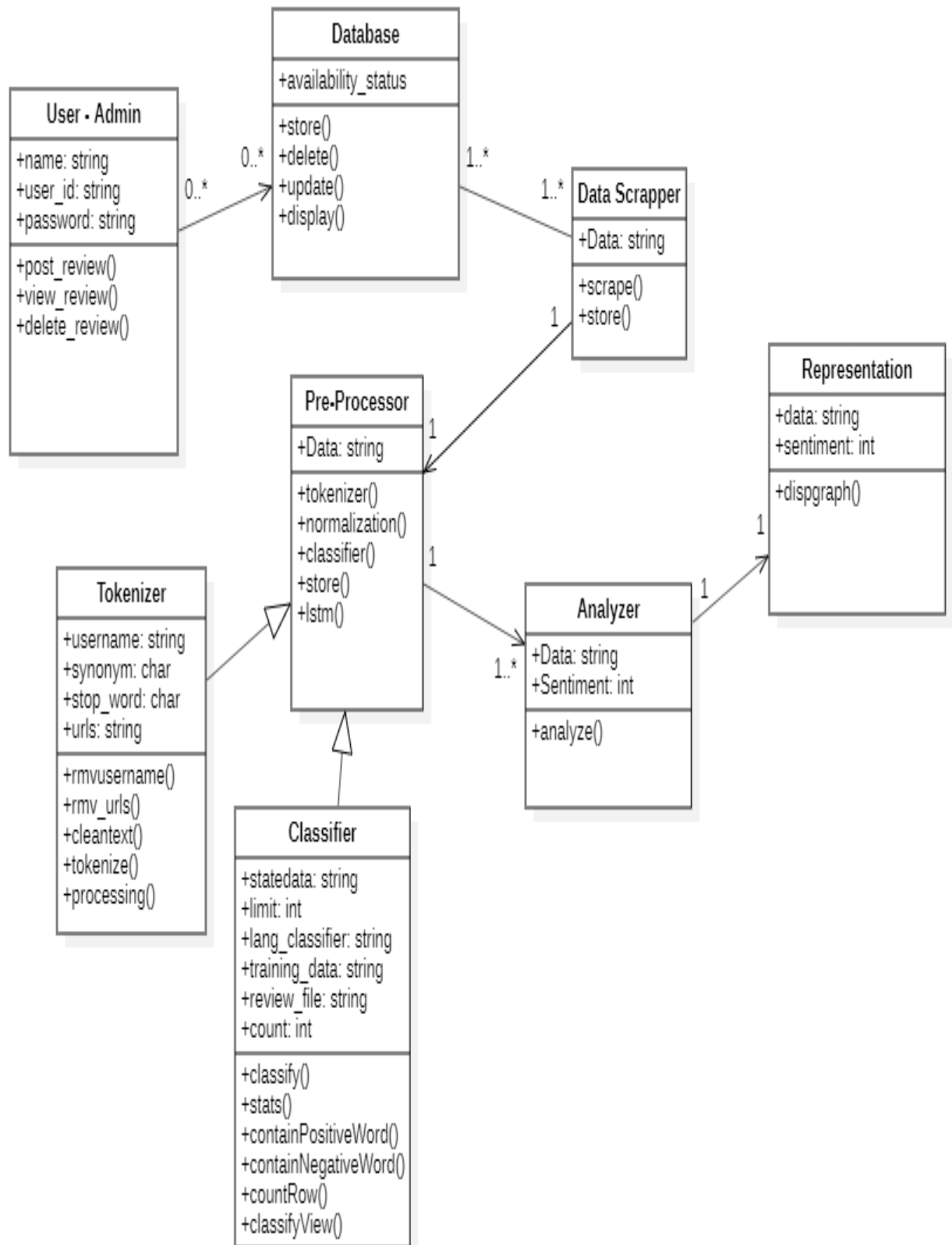


FIG 5.1 Class Diagram

5.2. INTERACTION DIAGRAM

Interaction, it is clear that the diagram is used to describe some type of interactions among the different elements in the model. This interaction is a part of dynamic behavior of the system.

This interactive behavior is represented in UML by two diagrams known as **Sequence diagram** and **Collaboration diagram**. The basic purposes of both the diagrams are similar.

Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

Purpose of Interaction Diagrams

The purpose of interaction diagrams is to visualize the interactive behavior of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction.

Sequence and collaboration diagrams are used to capture the dynamic nature but from a different angle.

The purpose of interaction diagram is –

- To capture the dynamic behavior of a system.
- To describe the message flow in the system.
- To describe the structural organization of the objects.
- To describe the interaction among objects.

5.2.1. SEQUENCE DIAGRAM

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the

Predicting useful product reviews on e-commerce website using sentimental analysis

messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams.

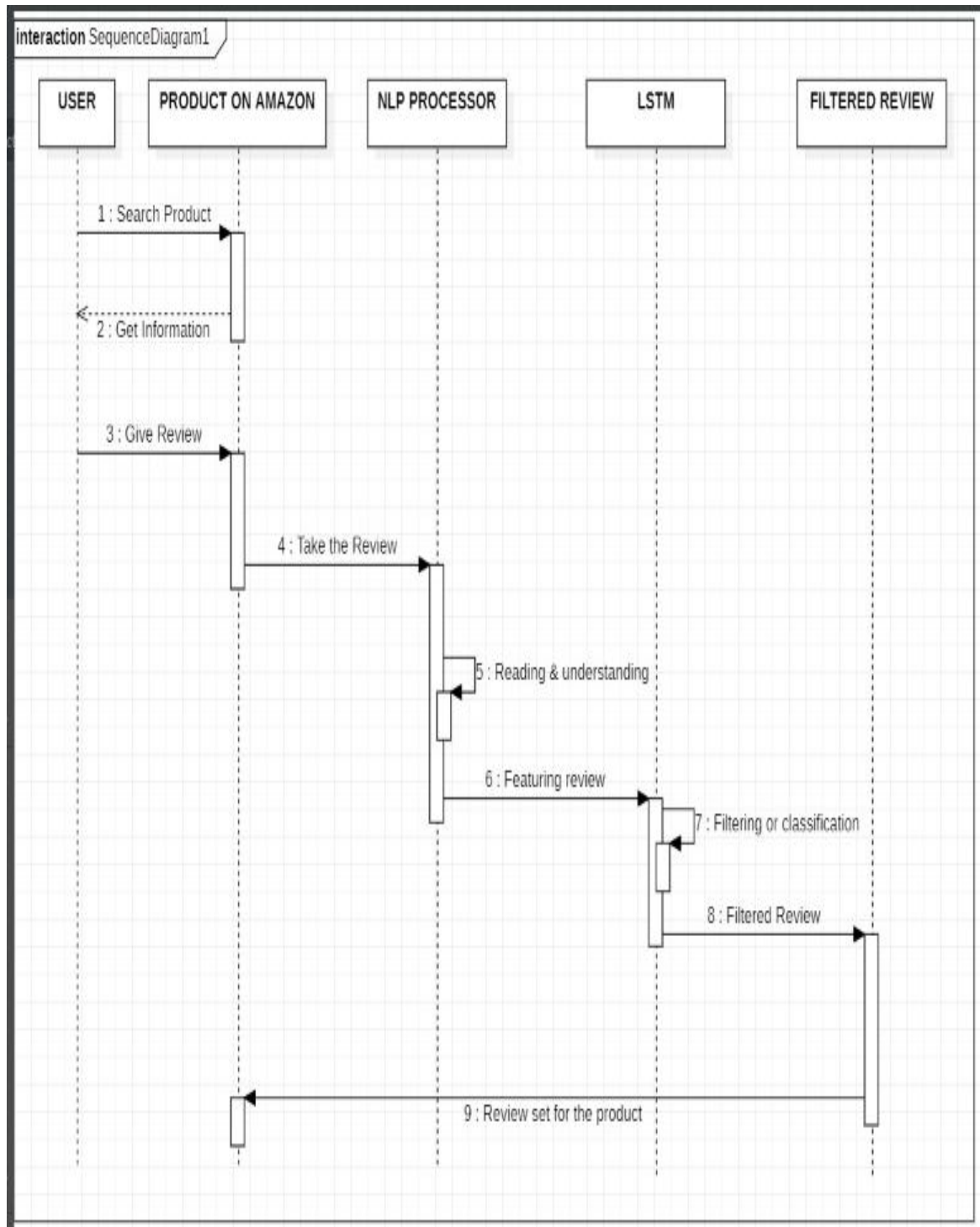


Fig 5.2 Sequence Diagram

5.2.2. COLLABORATION DIAGRAM

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming.

An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

Notations of a Collaboration Diagram

Following are the components of a component diagram that are enlisted below:

➤ **Objects:**

The representation of an object is done by an object symbol with its name and class underlined, separated by a colon.

In the collaboration diagram, objects are utilized in the following ways:

- The object is represented by specifying their name and class.
- It is not mandatory for every class to appear.
- A class may constitute more than one object.
- In the collaboration diagram, firstly, the object is created, and then its class is specified.
- To differentiate one object from another object, it is necessary to name them.

➤ **Actors:**

In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.

➤ **Links:**

The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.

➤ **Messages:**

It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labelled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.

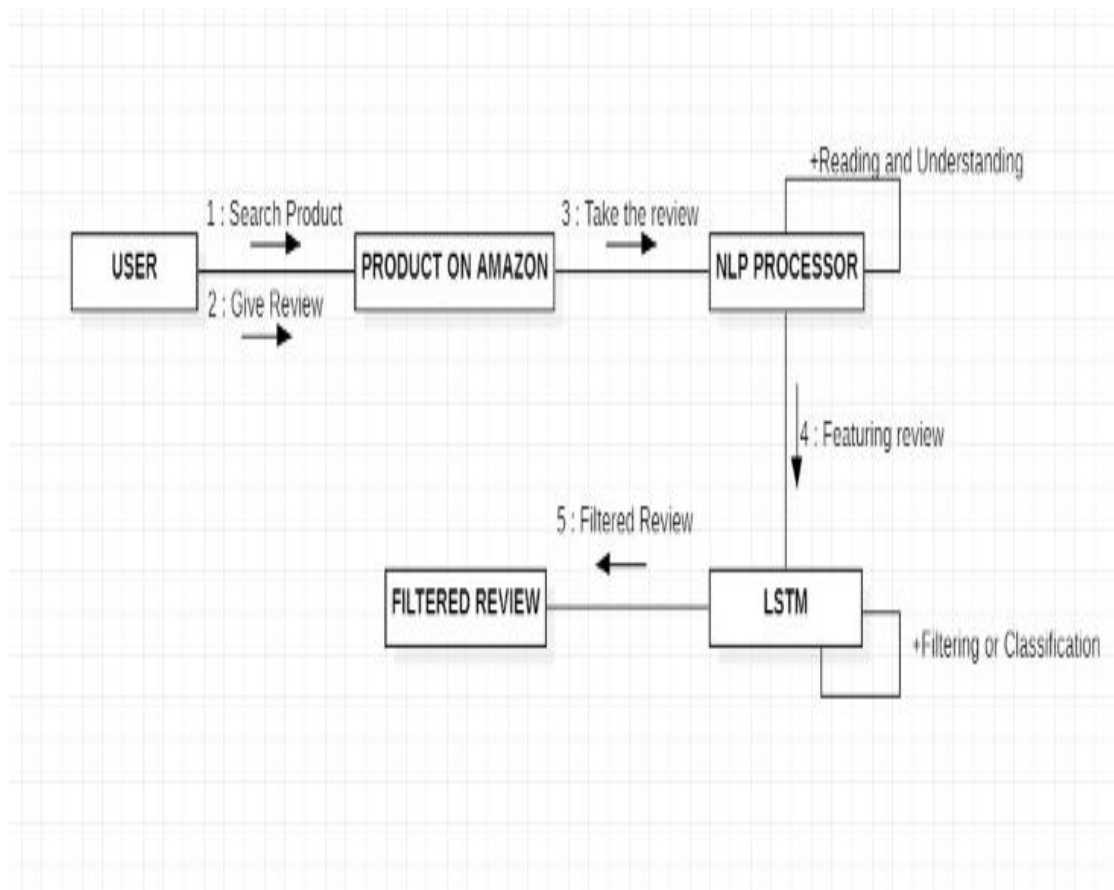


Fig 5.3 Collaboration Diagram

5.3. ACTIVITY DIAGRAM

Activity diagrams illustrate the dynamic nature of a system by modelling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation.

Activity diagrams are used to model workflow or business processes and internal operation.

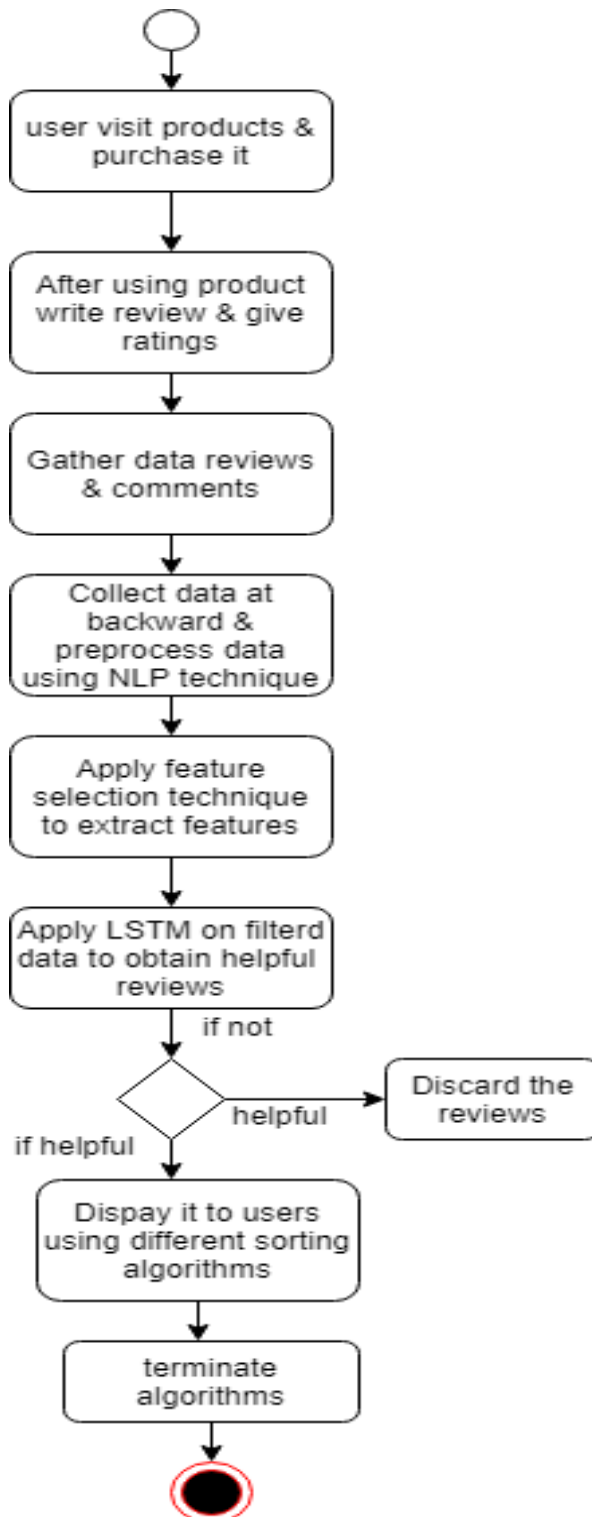


Fig 5.4 Activity Diagram

5.4. OBJECT DIAGRAM

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Object diagrams are used to render a set of objects and their relationships as an instance.

Purpose of Object Diagrams

The purpose of a diagram should be understood clearly to implement it practically. The purposes of object diagrams are similar to class diagrams.

The difference is that a class diagram represents an abstract model consisting of classes and their relationships. However, an object diagram represents an instance at a particular moment, which is concrete in nature.

It means the object diagram is closer to the actual system behavior. The purpose is to capture the static view of a system at a particular moment.

The purpose of the object diagram can be summarized as –

- Forward and reverse engineering.
- Object relationships of a system
- Static view of an interaction.
- Understand object behavior and their relationship from practical perspective

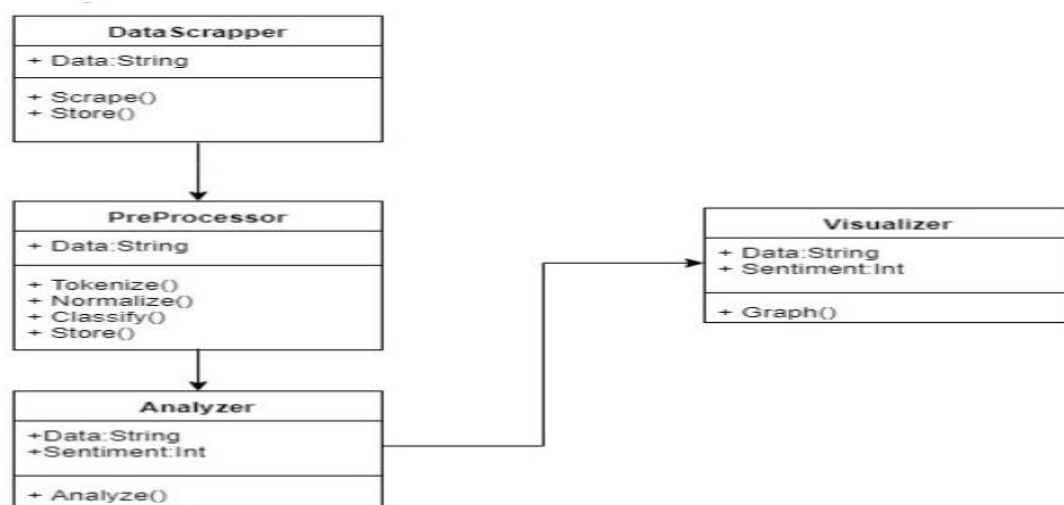


Fig. 5.5 OBJECT DIAGRAM

5.6. TEST PLAN & IMPLEMENTATION IMAGES

5.6.1. TESTING

Testing is the major quality control that can be used during software development. Its basic function is to detect the errors in the software. During requirement analysis and design, the output is a document that is usually textual and non-executable. After the coding phase, computer program is available that can be executed for testing purposes. This implies that testing not only has to uncover errors introduced during coding, but also errors introduced during previous phases. Thus the goal of the testing is to uncover requirement, design and coding errors in the program.

An elaborate testing of data is prepared and the system is tested using that test data. Errors noted and corrections made during the testing. The corrections are also noted for future use. The users are trained to operate the developed system. Both hardware and software securities are made to run the developed system successfully in future. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately before live operation commences. Testing is vital to the success of any system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

5.6.2. TESTING OBJECTIVES

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.
- A successful test is one that uncovers an as-yet undiscovered error.

TESTING PRINCIPLES

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- Testing should begin “in the small” and progress toward testing “in the large”.
- Exhaustive testing is not completely possible.

To be most effective, testing should be conducted by an independent third party.

5.6.3. TESTING METHODOLOGIES

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. As important, a software testing strategy provides a road map. Testing is a set of activities that can be planned in advance and conducted systematically.

Various strategies are given below:

- Unit Testing
- Integration Testing
- Validation Testing
- User Acceptance Testing
- System Testing

Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design of module. This is also known as “Module Testing”. Acceptance of package is used for computerization of module. Machine Utilization was prepared and approved by the project leader.

In this testing step, each module is found to be working satisfactory as regards to the expected output from the module. The suggested changes were incorporated into the system. Here each module in the Machine Utilization has been tested.

Integration Testing

After the package is integrated, the user test version of the software was released. This testing consists of testing with live data and various stress tests and result were noted down.

Then the corrections were made based on the user’s feedback. Integration testing is systematic testing for constructing the program structure, while at the same time conducting tests to uncover errors associated within the interface.

The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the vast expenses of the entire program complicate the isolation of causes. Thus the integration testing step, all the errors uncovered are corrected for the next steps.

Validation Testing

At the culmination of integration testing, software is completely assembled as a package; interfacing errors have been uncovered and corrected, and a final series of software tests - Validation testing - may begin.

User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system users at time of development and making changes wherever required.

After performing all the tests the system was found to be running successfully according to the user requirements i.e., (constraints).

System Testing

Software is only one element of a larger computer-based system. Ultimately, software is incorporated with other system elements and a series of system integration and validation tests are conducted. The various types of system testing are:

- Recovery Testing: Many computer-based systems must recover from faults and resume processing within a pre specified time.
- Security Testing: Security testing attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration.
- Stress Testing: Stress tests are designed to confront programs with abnormal situations.
- Performance Testing: Performance testing is designed to test run-time performance of software within the context of an integrated system.

5.6.4. TEST PLAN

To evaluate the impact of linguistic rules to the sentiment classification quality, the following scheme was used. We implement a certain set of rules and test its performance on training set. The set of linguistic rules, which was tested on available quotation sets, can be divided in two groups:

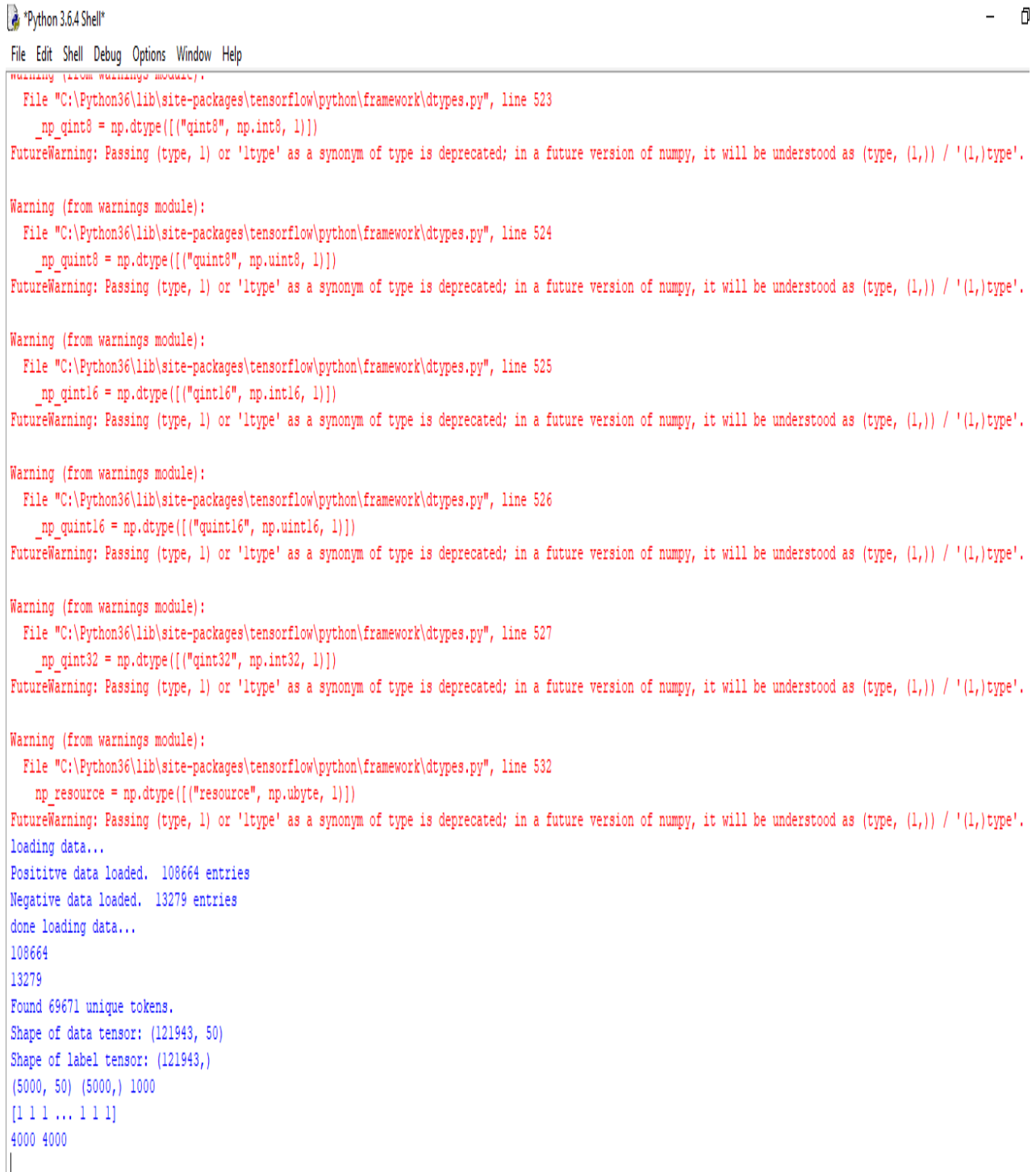
The first group contains the following set of rules:

- If an operator word is a part of a longer stop-word or sentiment expression, it does not act as an operator.
- If a group of operators appears together, their scores are multiplied.
- If there is unknown hyphenated word appeared in a text fragment, it is divided in two words and their scores are considered separately.
- If there is a sentiment word sequence, and a negative word appears among them then the score of the whole sequence becomes negative, otherwise positive.
- An operator is applied to the resulting score of a group of sentiment words.

The second group contains the following set of rules:

- If there is a question mark in a sentence, and the sentence does not begin with the words (why, for what), its sentiment score should be reduced.
- If there is (if) in a clause, the sentiment scores of the words in this fragment that go after should be reduced.
- If there is {"whether"} particle in a clause then the sentiment score of the words in this clause, which go after ("whether"), should be reduced.

5.6.5. SCREENSHOT



```

Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
WARNING (from warnings module):
  File "C:\Python36\lib\site-packages\tensorflow\python\framework\dtypes.py", line 523
    _np_qint8 = np.dtype [("qint8", np.int8, 1)]
FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

Warning (from warnings module):
  File "C:\Python36\lib\site-packages\tensorflow\python\framework\dtypes.py", line 524
    _np_quint8 = np.dtype [("quint8", np.uint8, 1)]
FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

Warning (from warnings module):
  File "C:\Python36\lib\site-packages\tensorflow\python\framework\dtypes.py", line 525
    _np_qint16 = np.dtype [("qint16", np.int16, 1)]
FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

Warning (from warnings module):
  File "C:\Python36\lib\site-packages\tensorflow\python\framework\dtypes.py", line 526
    _np_quint16 = np.dtype [("quint16", np.uint16, 1)]
FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

Warning (from warnings module):
  File "C:\Python36\lib\site-packages\tensorflow\python\framework\dtypes.py", line 527
    _np_qint32 = np.dtype [("qint32", np.int32, 1)]
FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

Warning (from warnings module):
  File "C:\Python36\lib\site-packages\tensorflow\python\framework\dtypes.py", line 532
    np_resource = np.dtype [("resource", np.ubyte, 1)]
FutureWarning: Passing (type, 1) or 'ltype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
loading data...
Positive data loaded. 108664 entries
Negative data loaded. 13279 entries
done loading data...
108664
13279
Found 69671 unique tokens.
Shape of data tensor: (121943, 50)
Shape of label tensor: (121943,)
(5000, 50) (5000,) 1000
[1 1 1 ... 1 1 1]
4000 4000
|

```

Fig 5.6: Training data, dividing data into positive & negative data, finding tokens, shaping data & label tensor

Predicting useful product reviews on e-commerce website using sentimental analysis

```
.
128/4000 [.....] - ETA: 4s - loss: 0.1748 - acc: 0.9297 - precision: 0.9478 - recall: 0.9732
256/4000 [>.....] - ETA: 4s - loss: 0.2030 - acc: 0.9219 - precision: 0
.9481 - recall: 0.9649
384/4000 [=.....]
.....] - ETA: 4s - loss: 0.2075 - acc: 0.9167 - precision: 0.9380 - recall: 0.9707
512/4000 [==>.....] - ETA: 4s - loss: 0.1985 - acc: 0.9219 - precision: 0.9393 - recall: 0.9759
640/4000 [===>.....] - ETA
: 4s
- loss: 0.1912 - acc: 0.9250 - precision: 0.9401 - recall: 0.9790
768/4000 [====>.....] - ETA: 4s - loss: 0.2083 - acc: 0.9180 - precision: 0.9301 - recall: 0.9825
896/4000 [=====>.....] - ETA: 4s - loss: 0.2071 - acc: 0.917
4 - precision: 0.9268 - recall: 0.9850
1024/4000
[=====>.....] - ETA: 4s - loss: 0.2045 - acc: 0.9170 - precision: 0.9269 - recall: 0.9848
1152/4000 [=====>.....] - ETA: 4s - loss: 0.2025 - acc: 0.9158 - precision: 0.9258 - rec
all: 0.9845
1280/4000 [=====>.....]
....] - ETA: 4s - loss: 0.1994 - acc: 0.9180 - precision: 0.9275 - recall: 0.9852
1408/4000 [=====>.....] - ETA: 3s - loss: 0.2004 - acc: 0.9176 - precision: 0.9259 - recall: 0.9865
1536/4000 [=====>.....] - ETA: 3s - loss: 0.1
977 - acc: 0.9186 - precision: 0.9279 - recall: 0.9855
1664/4000 [=====>.....] - ETA: 3s - loss: 0.2025 - acc: 0.9165 - precision: 0.9254 - recall: 0.9859
1792/4000 [=====>.....] - ETA: 3s - loss: 0.1982 - acc: 0.9185 - precisi
on: 0.9277 - recall: 0.9857
1920/4000 [=====>.....]
====>.....] - ETA: 3s - loss: 0.1956 - acc: 0.9219 - precision: 0.9309 - recall: 0.9861
2048/4000 [=====>.....] - ETA: 3s - loss: 0.1950 - acc: 0.9238 - precision: 0.9331 - recall: 0.9859
2176/4000 [=====>.....]
- ETA
: 3s - loss: 0.1953 - acc: 0.9246 - precision: 0.9346 - recall: 0.9852
```

Fig 5.7: Training LSTM model for product reviews

```
4480/5000 [=====>.....] - ETA: 4s
4512/5000 [=====>.....]
=====>... - ETA: 3s
4544/5000 [=====>.....] - ETA: 3s
4576/5000 [=====>.....] - ETA: 3s
4608/5000 [=====>.....] -
ETA: 3s
4640/5000 [=====>.....] - ETA: 3s
4672/5000 [=====>.....] - ETA: 2s
4704/5000 [=====>.....] - ETA: 2s
4736/5000 [=====>.....] - ETA: 2s
4768/5000 [=====>.....]
=====>... - ETA: 2s
4800/5000 [=====>.....] - ETA: 1s
4832/5000 [=====>.....] - ETA: 1s
4864/5000 [=====>.....]
.] - ETA: 1s
4896/5000 [=====>.....] - ETA: 0s
4928/5000 [=====>.....] - ETA: 0s
4960/5000 [=====>.....] - ETA: 0s
4992/5000 [=====>.....] - ETA: 0s
5000/5000 [====
=====] - 49s 10ms/step

Loss: 0.21556123789548873
Accuracy: 0.9186
Precision: 0.935780889415741
Recall: 0.9762321227073669
>>>
```

Fig 5.8: Value of model performance evaluation

Chapter 6

CONCLUSION & FUTURE WORK

6. CONCLUSION & FUTURE WORK

With the variety of products increasing day by day the decision to opt for a particular product is becoming difficult. So, the need for sentimental analysis is increasing gradually. Although sentimental analysis tasks are challenging due to their natural language processing origins, much progress has been made over the last few years due to the high demand for it. Not only the consumer want to know about the product but also the companies want to know about the condition of their product in the market.

The growing need for product insights – and the technical challenges currently facing the field –will keep sentiment analysis and opinion mining relevant for the foreseeable future. Next-generation opinion mining systems need a deeper bind between complete knowledge bases with reasoning methods inspired by human thought and psychology. This will lead to a better understanding of natural language opinions and will more efficiently bridge the gap between unstructured information in the form of human thoughts and structured data that can be analyzed and processed by a machine.

The intelligent opinion mining systems are capable of handling semantic knowledge, making analogies, continuous learning and detecting emotions that are leading to the highly efficient sentiment analysis.

6.1. LIMITATIONS

Sentimental analysis does not always give the right solutions; sometimes it may be even wrong. There are several issues related to sentiment analysis that could lead to degrade of the popularity of its technique.

Some of them are:

- Opinion spam
- Result measure
- Lack of complete information

- **Opinion spam:**

It can be used by competitors to portray and influence negative image on company. If sentiment analysis gains popularity as a medium to gain popularity and image of a company or organization then that malpractices will become more common that leads to decrease in the popularity of sentiment analysis technique. If a company want to establish a good image for its product then it could use this technique in a wrong way and can earn fame which is not an advantage. So, it can even cause a negative influence even on a good product.

- **Result measure:**

Since the output of sentiment analysis is based on the emotions, it is just a reactive measure and cannot be used to predict the performance of a company or other metrics. In some cases, sentiment analysis is just a reporting measure after the damage has occurred. It can be considered as a measure of scale rather than a prediction.

- **Lack of complete information:**

If analysis is made by taking incomplete sources of information then that will be absolutely incorrect. Since we don't get the complete information or data in most cases it becomes difficult to make analysis in most of the times. But sources like Facebook and twitter can be mined to get complete information whereas from sources like blogs, forums etc. we cannot get complete information.

6.2. FUTURE ENHANCEMENT

Opinion examination is as of now advancing quickly from an exceptionally straightforward (positive, negative, unbiased) to more granular and profound comprehension. At Revealed Context (the innovation arm of Converse on), we have power and certainty scores, and in addition feeling and that's only the tip of the iceberg.

These new classifier truly dimensionalize the subtleties of human expression in significant ways. There is likewise a move far from report/record level examination of the content towards substance/aspect level – meaning each statement of feeling is caught with the goal that we can truly comprehend the underlying driver drivers of suppositions. This requires machine learning approaches that are superseding more conventional principles based methodologies.

Exhibit Sentiments hold a key to the future occasions. To make it sound somewhat specialized, you can state that the slants speak to the “present estimation of future occasions”. Presently this esteem can have profound social, political and money related criticalness. It can be “Articulation of supposition about an open figure”, “assessments communicated through tweets before races”, or “the buzz before a motion picture discharge”, all these can be extraordinary signals for things to come.

In this way when individuals remark about present news stories, the conclusion examination can really offer a key to foresee the future results or atleast suspect them better!

Chapter 7

BIBLIOGRAPHY & REFERENCES

7. BIBLIOGRAPHY & REFERENCES

7.1. REFERENCE BOOKS

- Dipanjan Sarkar (2019). "Text Analytics with Python: A Practitioner's Guide to Natural Language Processing 2nd ed. Edition".
- Steven Bird, Ewan Klein, Edward Loper (2009). "Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit 1st Edition".

7.2. OTHER REFERENCES & DOCUMENTATION

➤ **Amazon Product Reviews Dataset link**

- <https://www.kaggle.com/c/caltech-cs-155-2018/data>
- <http://jmcauley.ucsd.edu/data/amazon/>

➤ **Research papers and Documentation**

- *Sentimental analysis of Amazon reviews using naïve bayes on laptop products with MongoDB and R*
- *Mohan Kamal Hassan et al 2017 IOP Conf. Ser.: Mater. Sci. Eng.263 042090*

➤ **Sentiment Analysis On Online Product Review**

- *International Research Journal of Engineering and Technology(IRJET)*

➤ **Sentiment Analysis And Prediction Of Online Reviews With Empty Ratings**

- *International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 14 (2018) pp. 11525-11531 © Research India Publications.*
<http://www.ripublication.com>

➤ **Sentiment Analysis for Amazon Reviews**

- <http://cs229.stanford.edu/proj2018/report/122.pdf>

➤ **Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts**

- <https://www.aclweb.org/anthology/C16-1229.pdf>
- **Sentiment analysis using product review data Xing Fang and Justin Zhan**
- <https://journalofbigdata.springeropen.com/track/pdf/10.1186/s40537-015-0015-2>
- **Review Rating Prediction: A Combined Approach**
- **Combining Review Text Content (RTC) and User Similarity Matrix to obtain more information and to improve the Review Rating Prediction (RRP)**
- <https://towardsdatascience.com/review-rating-prediction-a-combined-approach-538c617c495c>
- <https://towardsdatascience.com/review-rating-prediction-a-combined-approach-538c617c495c>
- <https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/>
- **Sentiment Analysis and Opinion Mining by Bing Liu**
- https://en.wikipedia.org/wiki/Long_short-term_memory
- http://en.wikipedia.org/wiki/Bayesian_probability
- http://en.wikipedia.org/wiki/Naive_Bayes_classifier
- http://www.let.rug.nl/~tiedeman/ml05/03_bayesian_handout.pdf
- <http://www.statsoft.com/textbook/stnaiveb.html>
- <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/mlbook/ch6.pdf>
- <http://www.scoop.it/t/social-media-monitoring-tools-and-solutions>
- <http://www.saama.com/sentiment-analytics-the-gold-mine-which-you-didn-t-mine/>