A

Major Project Report

on

SENTIMENT ANALYSIS OF TEXT FEEDBACK

Submitted in Partial Fulfillment of
the Requirements for the Degree
of

Bachelor of Engineering

in

Information Technology Engineering

to

Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon

Submitted by

Harshal Pandharinath Patil
Prajwal Arun Parekh
Tejaswini Rajendra Patil
Punam Santosh Gangatire

Under the Guidance of

Mr.Pravin Keshav Patil

DEPARTMENT OF INFORMATION TECHNOLOGY ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)

2021 - 2022

SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY, BAMBHORI, JALGAON - 425 001 (MS)

DEPARTMENT OF INFORMATION TECHNOLOGY ENGINEERING

CERTIFICATE

This is to certify that the major project entitled *SENTIMENT ANALYSIS OF TEXT FEEDBACK*, submitted by

Harshal Pandharinath Patil
Prajwal Arun Parekh
Tejaswini Rajendra Patil
Punam Santosh Gangatire

in partial fulfillment of the degree of *Bachelor of Engineering* in *Information Technology Engineering* has been satisfactorily carried out under my guidance as per the requirement of Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon.

Date: April 15, 2022
Place: Jalgaon

Mr.Pravin Keshav Patil
Guide

D. Manoj E. Patil
Head

Prof. Dr. Girish K. Patnaik
Principal

Contents

Acknowledgements

We would like to express our deep gratitude and sincere thanks to all who helped us in completing this project report successfully. Many thanks to almighty God who gave us the strength to do this. Our sincere thanks to Principal Prof. Dr. Girish. K. Patnaik for providing the facilities to complete this Project report. We would like to express our gratitude and appreciation to all those who gave usthe possibility to complete this report. A special thanks to Prof. Dr. Manoj E.Patil, Head of the Department, whose help, stimulating suggestions and encouragement, helped us in writing this report. We would also like to thank (Mr. Pravin Keshav Patil), Project Guide, who has given his/her full effort in guiding us and achieving the goal as well as his encouragement to maintain the progress in track. I am also sincerely thankful to Mr. Akash D. Waghmare, Incharge of Project, for his valuable suggestions and guidance. We would also like to appreciate the guidance given by other supervisor that has improved our presentation skills by their comments and tips. Last but not the least, we are extremely thankful to our parents and friends without whom it could not reach its successful completion.

Harshal Pandharinath Patil
Prajwal Arun Parekh
Tejaswini Rajendra Patil
Punam Santosh Gangatire

Abstract

Sentiment Analysis also known as Opinion Mining refers to the use of natural language processing , text analysis to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine. 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". We expect to do review-level categorization of review data with promising outcomes

Chapter 1

Introduction

The chapter describes all the details regarding introduction of Sentiment analysis of text feedback.The introduction part of the report includes all the basic information required to elaborate the overall idea of Sentiment analysis of text feedback. The chapter also includes the different techniques which are used while developing the project. Section 1.1 describes the Background of the topic. The purpose of Section 1.2 is to explain Motivation behind the project.Section 1.3 contains the Problem Definition of project. Scope of the project is included Section 1.4. Section 1.5 contains Objective of project. Section1.6 describes identification of software development Process Model. Section 1.5 contains Organization of Report. Section 1.6 includes the summary of project.

1.1 Background

Sentiment analysis is a uniquely powerful tool for businesses that are looking to measure attitudes, feelings and emotions regarding their brand. To date, the majority of sentiment analysis projects have been conducted almost exclusively by companies and brands through the use of social media data, survey responses and other hubs of user-generated content. By investigating and analyzing customer sentiments, these brands are able to get an inside look at consumer behaviors and, ultimately, better serve their audiences with the products, services and experiences they offer.

1.2 Motivation

Since customers express their thoughts and feelings more openly than ever before, sentiment analysis is becoming an essential tool to monitor and understand that sentiment .Automatically analyzing customer feedback, such as opinions in survey responses and social media conversations, allows brands to learn what makes customers happy or frustrated, so that they can tailor products and services to meet their customers' needs.

1.3 Problem Defination

From a user's perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites. From a researcher's perspective, many social media sites release their application programming interfaces (APIs), prompting data collection and analysis by researchers and developers. However, those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. The second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral.So sentiment analysis using different machine learning algorithms will solve this issue.

1.4 Scope

The future of sentiment analysis is going to continue to dig deeper, and truly understand, the significance of social media interactions and what they tell us about the consumers behind the screens. This forecast also predicts broader applications for sentiment analysis – brands will continue to leverage this tool, but so will individuals in the public eye, governments, nonprofits, education centers and many other organizations.

1.5 Objective

1. Scrapping product reviews on various websites featuring various products specifically amazon.com.
2. Analyze and categorize review data. Analyze sentiment on data set from document level (review level).
3. Categorization or classification of opinion sentiment into- Positive Negative

1.6 Identification of software development Process Model

The four basic process activities of specification, development, validation and evolution are organized differently in different development processes. In the waterfall model, they are organized in sequence, while in incremental development they are interleaved. Software Development Life Cycle(SDLC) is a process used by the software industry to

design, develop and test high quality software. The SDLC aims to produce a high quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates

1.6.1 Waterfall Model - Design

Waterfall approach was the first Model to be used widely in Software Engineering to ensure success of the project. In The Waterfall approach, the whole process of software development is divided into separate phases. In the Waterfall model, typically the outcome of one phase acts as the input for the next phase sequentially. The following illustration is a representation of the different phases of the Waterfall Model[4].

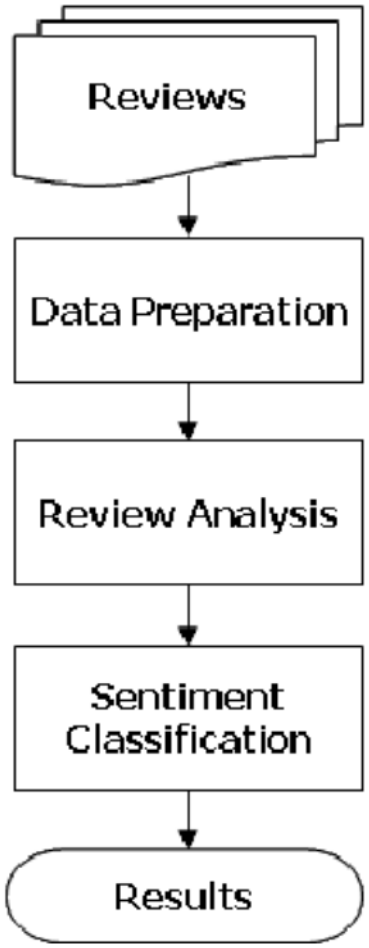


Figure 1.1: Waterfall Model

1.7 Organization of Report

Chapter 1 entitled as Introduction describes the details about Background, Problem Definition, Scope and Objective of the project, Identification of Software Development

Process Model and Organization of report. Chapter 2 entitled as Project Planning and Management consists of details about the Feasibility Study, Risk Analysis, Project Scheduling, Effort Allocation and Cost Estimation of the project. Chapter 3 entitled as Analysis describes in detail, the Requirement Collection and Identification, H/w and S/w Requirements, and a Software Requirements Specification(SRS). Chapter 4 includes design about System Architecture, Data Flow Diagram and various UML Diagrams. Chapter 5 consists of Conclusion

1.8 summary

In the chapter all the details about problem definition, motivation, scope, objective of the project and selection of software development model are included.

Chapter 2

Project Planning and Management

The chapter includes details regarding Feasibility study, Risk Analysis, Project Scheduling, Effort Allocation and Cost Estimation as well as a short summary. Section 2.1 includes details regarding feasibility that includes Operating, Economical and Technical feasibility. Details regarding Risk Analysis like Commercial risks, Design risks and other risks as well are discussed in Section 2.2. Section 2.3 provides explanation about Project Scheduling while section 2.4 describes the effort allocation i.e effort taken by each group member. Section 2.5 provides information about cost estimation of the project

2.1 Feasibility Study

Once scope has been identified ,it is reasonable to ask:Can the software be build to meet the scope? Is the Project feasible? all too often,software engineers rush past the questions(or are pushed past them by impatient managers or customers),only to become mired in a project that is doomed from the onset[6]. Feasibility is the analysis of risks,costs and benefits relating to economics,technology and user operation. There are several types of feasibility depending on the aspects they cover. Some important feasibility is as follows:

- Economical Feasibility
- Operational Feasibility
- Technical Feasibility

2.1.1 Economical Feasbility

More commonly known as cost/benefit analysis the procedure is to determine the benefits and savings that are expected from system and compare them with costs, decisions is made to design and implement the system.The part of feasibility study gives the top management the economic justification for the new system. The is an important input to the management the management,because very often the top management does not like to get confounded by the various technicalities that bound to be associated with a project of the kind. A simple economic analysis that gives the actual comparison of costs and benefits is much more meaningful in such cases. In the system, the organization is most satisfied by economic feasibility. Because, if the organization implements the system, it need not require any additional hardware resources as well as it will be saving lot of time. The proposed system i.e sentiment analysis of text feedback uses machine learning platform for development as well as a separate point of interest and approaches that requires less amount of money and produces efficient results at a low price. When the project will be a complete software functionality, people will have to pay lesser money, though it will be a proprietary one, still the money paid in comparison to the yields will be negligible. Therefore, the project will prove to be economically feasible.

2.1.2 Operational Feasibility

People are inherently resistant to change and computer has been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the organization. Propose project is beneficial only if it can be turned into information systems that will be meet operating requirements simply stated, this test of feasibility ask if the system will work when it is developed and installed. Are the major barriers of implementation? The proposed was to make a simplified application that analyses given text. It is simpler to operate and can be used in any python platform. It is free and not costly to operate.

2.1.3 Technical Feasibility

Technical feasibility centres on the existing manual system of the test management process and to what extend it can support the system.According to feasibility analysis

procedure the technical feasibility of the system is analysed and the technical requirement such as software facilities,procedure,inputs are identified. It is also one of the important phases of the system development activities. System covers greater level of a user friendliness combined with greater processing speed. Therefore the cost of maintenance can be reduced. Since processing speed is very high and the work is reduced in maintenance point of view management convince that the project is operationally feasible. Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at the point in time there is no any detailed designed of the system, making it difficult to access issues like performance, costs (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis; understand the different technologies involved in the proposed system. Before commencing the project, we have to be very clear about what are the technologies that are to be required for the development of the new system. Is the required technology available? Our system "Sentiment Analysis" is technically feasible since all the required tools are easily available. Python can be easily handled. Although all tools seem to be easily available there are challenges too

2.2 Risk Analysis

The aim of the work is to adapt, combine, implement and evaluate engineering techniques in a new application aiming at analyzing the risks that are present. The risks evaluated in the production process are directly associated to the quality and safety of the environment, as well as to the procedures involved. The risks are analyzed. The risks evaluated in the production process are directly associated to the quality and safety of the environment, as well as to the procedures involved.

2.2.1 Commercial Risks

Following are some commercial risks identified in the project1. Risk:Privacy Security also pose significant challenges that the ML industry. Due to inconsistencies in python programming, oversight, and negligence, there is a legitimate chance of getting into trouble without meaning to do so. 2. Risk: At the moment, the majority of ML apps are one trick ponies that can do one thing which can be done without their assistance with the same level of effectiveness.

2.2.2 Design/Engineering Risks

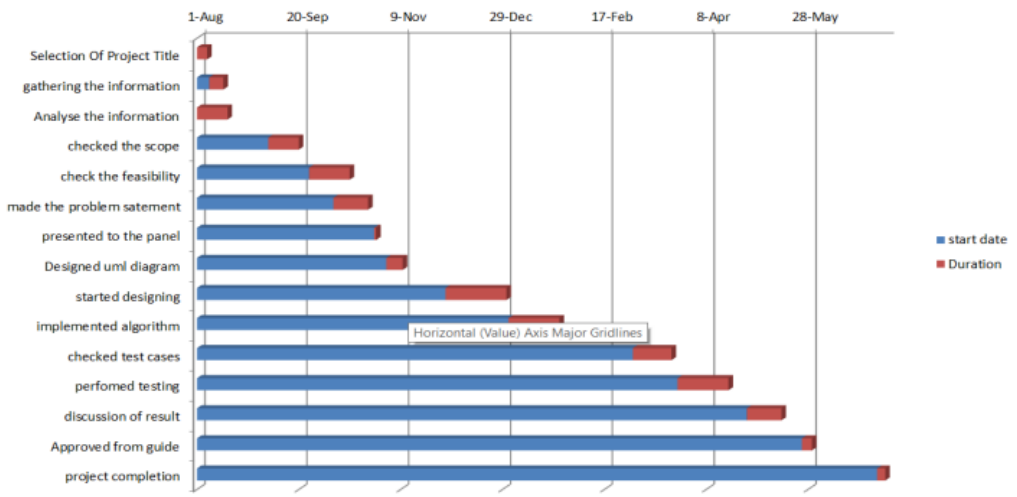
Risk:Errors and design assumptions. Action to be taken:Use the expertise by project coordinators, developers

2.2.3 Other Risks

There are some other risks other than the above discussed they are as follows- • Risk:Developers stability. Action to be taken:Legal agreement by management. • Risk:Co-ordination. Action to be taken:Give hundred percent by project coordinators.

2.3 Project Scheduling

The section specifies the project scheduling of the project. Software project scheduling is an activity that distributes estimated effort acrossthe planned project duration by allocating the effort to specific software engineering task. In the phase, we are identifying all the major software engineering activities and the product function to which they are applicable. As the linearsequential model have been selected for developing the project, the work is divided according to the phases of the model. As a group of four members working on the project, the project is scheduled accordingly. If the project development goes as per the planned schedule, the project schedule defines the task and that must be tracked and controlled as progress.



2.4 Effort Allocation

Each of the software project estimation techniquesleadsto an estimate of work unitsrequired for completion of the software development. The characteristics of each project task dictate the distribution of efforts. There are four members in the project and there are four phases such as Project Planning, Requirement Gathering, Analysis, Design, so the below table shows efforts of each member in the project[4].

	Harshal	Prajwal	Tejaswini	Punam
Project Planning	40%	20%	20%	20%
Analysis	30%	30%	20%	20%
Design	30%	30%	20%	20%
Documentation	45%	25%	20%	20%

Figure 2.1: Effort Allocation

2.5 Cost Estimation

Cost estimation is a set of techniques and procedures used to arrive at a cost estimate. These techniques are utilized by the process of cost estimation to compute the output from the given set of inputs. (Constructive Cost Model) is a regression model based on LOC,i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which makes it one of the best documented models[2]. types of Models: These are types of COCOMO model:

- Basic COCOMO Model
- Intermediate COCOMO Model
- Detailed COCOMO Mode

Basic Model The basic COCOMO model provide an accurate size of the project parameters.

The following expressions give the basic COCOMO estimation model.The effort is measured in Person-Months and as evident from the formula is dependent on Kilo-Lines of code. E = a(KLOC)b

where,

- ab are constant
- KLOC is kilo number of Lines of Code
- E is effort
- a¯2.4
- b¯1.05

Estimated Cost For Development:

Total number of persons working on project	4
Time taken in months	8 months
Total time allocated per day in terms of hour 2hrs	2 hrs
Actual working hours(2*240)	480 hrs
Cost per hours	25
Total estimated project cost is:	12,000

Table 2.1: shows the estimated cost for development

2.6 Summary

In the chapter, all the details related to Project Planning and Management are mentioned. In the next chapter, all the details regarding the requirements gathering and analysis are presented.

Chapter 3

Analysis

The chapter describes everything related to the requirement gathering and further analysis such as hardware, software, functional and non-functional requirements. Also it has the software requirements specification that provides complete description of the requirements of the system. Section 3.1 describes requirement collection and identification. All the hardware and software requirements are discussed in Section 3.2. Section 3.3 describes the Software Requirements Specification(SRS)

3.1 Requirment Collection and Identification

In system engineering and software engineering, requirements analysis focuses on the tasks that determine the needs or conditions to meet the new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, doc- umenting, validating and managing software or system requirements. The main objective here is to understand the existing system thoroughly and check the feasibility requirements. Requirements Analysis includes three types of activities:

- Eliciting Requirements:(E.g. the project charter or definition), Business process documentation, and stake holder interviews. Sometimes it is also called requirements gathering or requirements discovery
- Analyzing Requirements:Determining whether the stated requirements are clear, complete, consistent and unambiguous, and resolving any apparent conflicts
- Recoding Requirements:Requirements may be documented in various forms, usuallyincludinga summarylist andmayincludenatural-languagedocuments,use cases,user stories, process specifications and a variety of s models including data models

3.2 Hardware and Software Requirements

Hardware and software requirements of the projects such as type of processor, memory, storage, graphics and OS are:

3.2.1 Hardware Requirement

The following things are required:

- At least 60GB of usable hard Disk space
- core i5/i7 processor
- 8GB RAM

3.2.2 Software Reqirements

Software Requirement deal with denting software resource requirement and prerequisites that need to be installed on a computer to provide optimal functioning of an application. TheseRequirements or pre-requisites are generally included in the software installation package and need to be installed separately before the software isinstalled.

- Operating System:Window 7 or above
- python 3.x
- NLTK Toolkit

3.3 Software Requirement Specification (SRS)

A software requirementsspecification(SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase. The quality characteristics of SRS must be according to the given guidelines:

- Correctness
- Completeness
- Consistency
- Unambiguousness
- Ranking for importance and stability

3.4 Summary

In the chapter, Analysis was presented which included the hardware and software requirements, functional and non-functional requirements and the software requirements specification(SRS) as well. In the next chapter, Design is described along with various UML diagrams.

Chapter 4

Design

System design providesthe understanding and procedural details necessary forimplementing the system.

4.1 System Architecture

A System Architecture is the conceptual model that defines the structure,behaviour,and more views of a system and architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and Assistant architecture can comprise system components, the extremely visible properties of those components, the relationship i.e the behaviour between them. It can provide a platform in which system can be procured,and systems developed, that will work together to implement the overall system

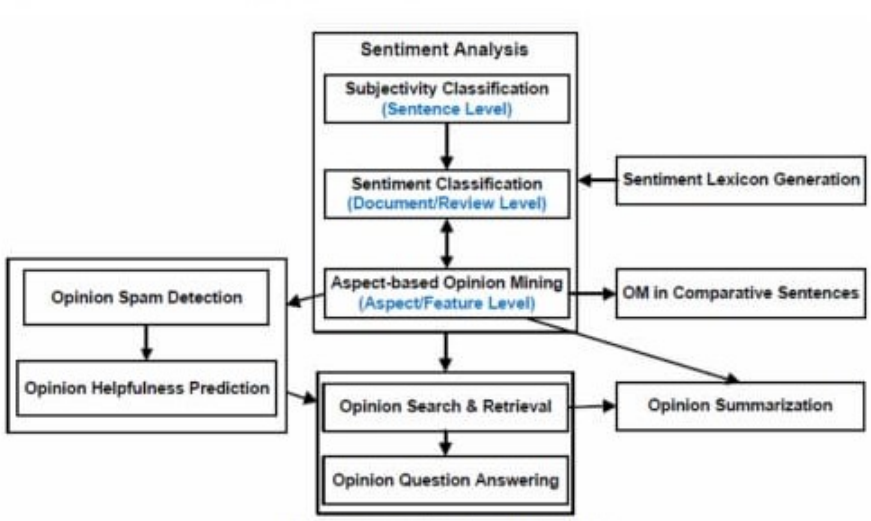


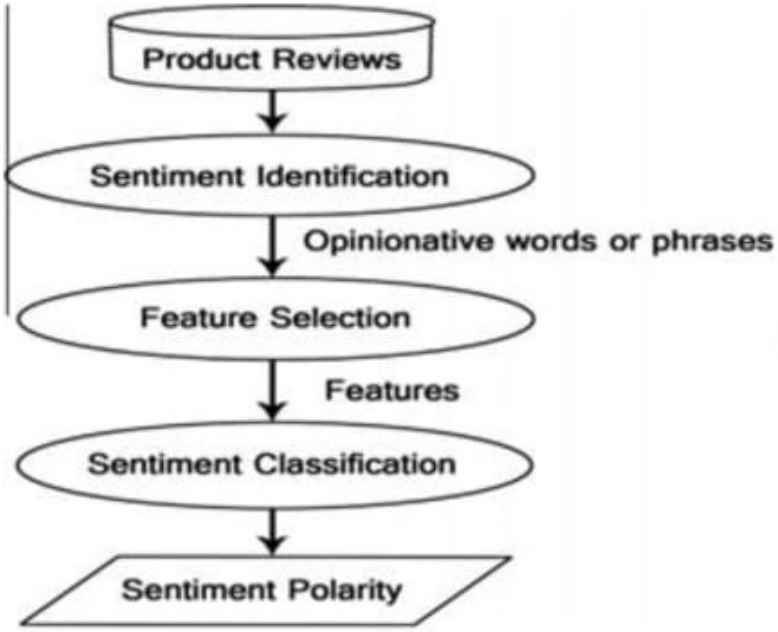
Figure 1 Sentiment analysis tasks (source: http://iccke2014.um.ac.ir/uploading/iccke2014.um.ac.ir/images/ICCKE2014-eWorkshops-OpinionMining.pdf)

4.2 Data Flow Diagram

A data flow diagram is a flowchart can help to visualize the data pipeline of a system user can trace happens to the data as it moves between components. It is a great to find redundancies and optimize the speed and responsiveness of software. A DFD is often used as a preliminary step to create an overview of the system going into great detail, it can later be elaborated. DFDs are used for the visualization of data processing (structured design). A DFD show kind of information input to and output from the system, the data will advance through the system, and it the data will be stored. It represented information of processtiming processes will operate in sequence or in parallel, unlike a traditionalstructured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data, flows as a unified model.

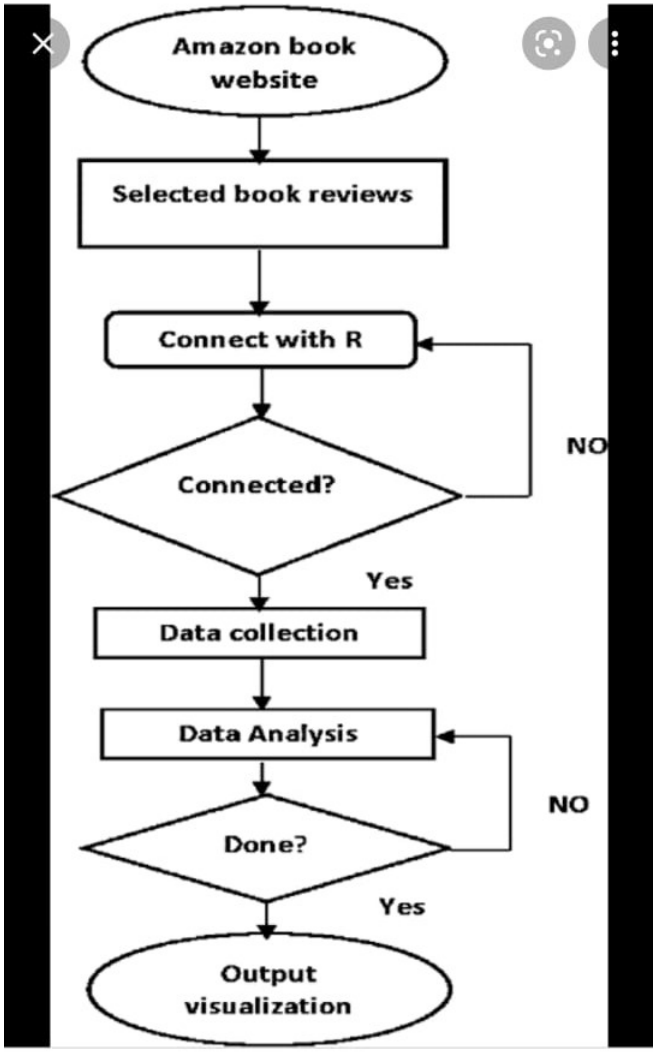
- DFD Level 0

DFD Level 0 is also called a Context Diagram. It is a basic overview of the whole system or process being analyzed or modeled. It is designed to be an at a glance view, showing the system as a single high-level process, with its relationship to external entities. It easily understood a wide audience, including stakeholders, business analysts, data analysts and developers. In the Figure 4.2 the data flow diagram level 0 is described



- DFD Level 1

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. It highlight the main functions carried out by the system, it break down the high-level process of the Context Diagram into its sub processes. In the Figure 4.3 the data flow diagram level 1 is described

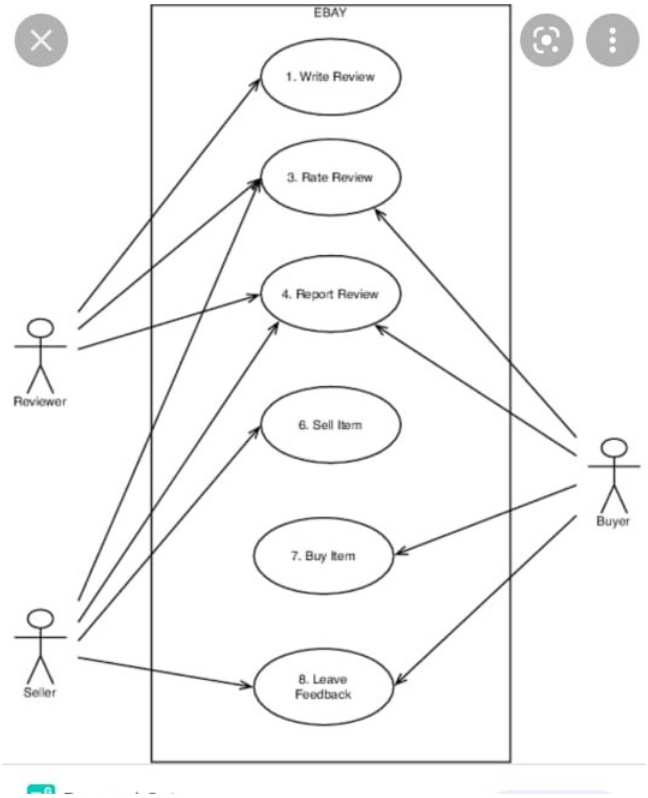


4.3 UML Diagram

Unified Modeling language(UML) is a standardized modeling language enabling developers to specify, visualize, construct and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure and robust in execution. UML is an important aspect involved in object-oriented software development. It uses graphic notation to create visual models of software systems.

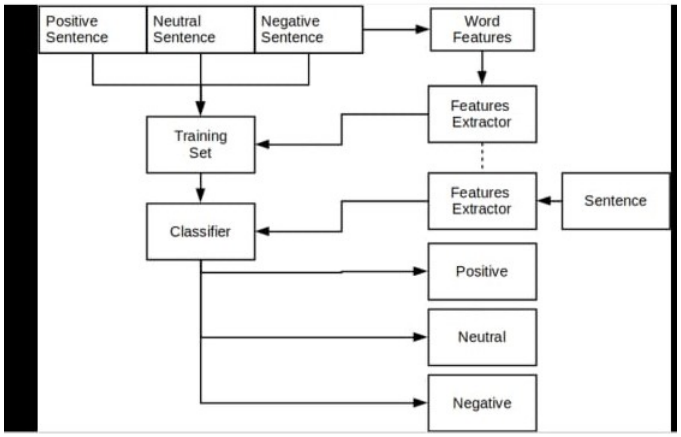
4.4 Use Case Diagram

A UseCase diagram shows the interaction between the system and entities external to the system. These entities are called actors which have specific role in the system.The figure shows the use case diagram for proposed system. Purpose of UseCase Diagram is to know or show functionality of the system



4.5 Class Diagram

A Class diagram is used to represent the static view of the system. It mainly use classes,interfaces and their relationships. The Figure shows the class diagram for proposed system. Purpose of Class Diagram is to show structural aspects of the system



4.6 Summary

In the chapter, design and architecture of sentiment analysis of text feedback is described.

Chapter 5

Coding/ Implementation

The chapter includes coding and implementation of spam words detection using machine learning algorithms. Provides information about coding and implementation part of system.

5.1 Algorithm

The Steps section of the project describe the algorithms that are used in the project for classification and the steps required for the overall implementation of the design

5.1.1 Classifiers

In the following chapter the algorithm used for development of project are discussed.

- Support Vector Machine

Support Vector Machine (SVM)is a method for the classification of both linear and non linear data. If the data is linearly separable, the SVM searches for the linear optimal separating hyperplane (the linear kernel), which is decision boundary that separates data of one class from another. Mathematically, a separating hyperplane can be written as: W . X + b = 0, where W is a weight vector and W = is a weightr and W = w1,w2, ...,wn. X is a training tuple. b is a scalar. In order to optimize the hyperplane, the problem essentially transforms to the minimization of W, which is eventually computed as: n i=1 iyixi, where i are numeric parameters, and yi are labels based on support vectors, Xi. That is: if yi = 1 then n i=1 wixi 1; if yi = 1 then n i=1 wixi 1. If the data is linearly inseparable, the SVM uses nonlinear mapping to transform the data into a higher dimension. It then solve the problem by finding a linear hyperplane

- Naive Bayesian Classifier

The Naïve Bayesian classifier works as follows: Suppose that there exist a set of training data, D, in which each tuple is represented by an n-dimensional feature vector, X = x1, x2, .., xn, indicating n measurements made on the tuple from n attributes or features. Assume that there are m classes, C1,C2, ...,Cm. Given a tuple X, the classifier will predict that X belongs to Ci if and only if: P(Ci—X) ¿ P(Cj—X), where i, j [1, m] andi = j

- Logestic Regression

Logistic regression predicates the probability of an outcome that can only have two values (that is a dichotomy).The predition based on the use of one or more predictors(numerical and categorial).

5.1.2 Steps of Implementaton

- Reviews

We took the dataset of musical instruments product reviews from Kaggle.

- Data Prepartion

Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It is an important step prior to processing and often involves reformatting data, making corrections to data and the combining of data sets to enrich data.

- Review analysis

Review analysis is the process of transforming unstructured review data to structured data that can be used to guide decision-making.

- Sentiment Classification

Sentiment classification is the automated process of identifying opinions in text and labeling them as positive, negative, or neutral, based on the emotions customers express within them.

- Result

The final outcome will be the accuracy score of all three machine learning model.

5.2 Implementation Details

5.2.1 Dataset Collection

The training of dataset consists of the following steps:â€< Unpacking of data:-The huge dataset of reviews obtained from amazon.com comes in a .csv file format. A small python code has been implemented in order to read the dataset from those files .

```
import numpy as np # linear algebra
import pandas as pd
```

```
df=pd.read_csv('/content/Musical_instruments_reviews.csv')
df
```

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime
0	A2IBPI20UZIR0U	1384719342	cassandra tu "Yeah, well, that's just like, u...	[0, 0]	Not much to write about here, but it does exac...	5.0	good	1393545600	02 28, 2014
1	A14VAT5EAX3D9S	1384719342	Jake	[13, 14]	The product does exactly as it should and is q...	5.0	Jake	1363392000	03 16, 2013
2	A195EZSQDW3E21	1384719342	Rick Bennette "Rick Bennette"	[1, 1]	The primary job of this device is to block the...	5.0	It Does The Job Well	1377648000	08 28, 2013
3	A2C00NNG1ZQQG2	1384719342	RustyBill "Sunday Rocker"	[0, 0]	Nice windscreen protects my MXL mic and preven...	5.0	GOOD WINDSCREEN FOR THE MONEY	1392336000	02 14, 2014
4	A94QU4C90B1AX	1384719342	SEAN MASLANKA	[0, 0]	This pop filter is great. It looks and perform...	5.0	No more pops when I record my vocals.	1392940800	02 21, 2014
...	...	...	...	...	...	...	...	...	...
10256	A14B2YH83ZXMPP	B00JBIVXGC	Lonnie M. Adams	[0, 0]	Great, just as expected. Thank to all.	5.0	Five Stars	1405814400	07 20, 2014
10257	A1RPTVW5VEOSI	B00JBIVXGC	Michael J. Edelman	[0, 0]	I've been thinking about trying the Nanoweb st...	5.0	Long life, and for some players, a good econom...	1404259200	07 2, 2014
10258	AWCJ12KBO5VII	B00JBIVXGC	Michael L. Knapp	[0, 0]	I have tried coated strings in the past (incl...	4.0	Good for coated.	1405987200	07 22, 2014

5.2.2 Preparing Data

Preparing Data for Sentiment Analysis :- 1) The dataset which is loaded contains a lot of coloumns we only need reviewtext and overall rating coloumn for sentiment analysis . All the other coloums are droped.

```
#drop the unnecessary features from the dataset
df=df.drop(['reviewerID','asin','reviewerName','unixReviewTime','reviewTime','summary'],axis=1)
```

```
df.isnull().sum()
```

```
helpful       0
reviewText    7
overall       0
dtype: int64
```

```
df=df.drop('helpful',axis=1)
```

```
df.isnull().sum()
```

```
reviewText    7
overall       0
dtype: int64
```

```
df=df.dropna() # used to remove rows and columns with Null/NaN values.
```

2) After that we create a binary rating i.e '1' if overall rating is greater than equal to 3 and '0' if less than 3.

```
#lets create a binary rating i.e '1' if overall rating is greater than equal to 3 and '0' if less than 3
df['overall']=np.where(df['overall']>=3,1,0)
```

```
df
```

	reviewText	overall
0	Not much to write about here, but it does exac...	1
1	The product does exactly as it should and is q...	1
2	The primary job of this device is to block the...	1
3	Nice windscreen protects my MXL mic and preven...	1
4	This pop filter is great. It looks and perform...	1
...	...	...
10256	Great, just as expected. Thank to all.	1
10257	I've been thinking about trying the Nanoweb st...	1
10258	I have tried coated strings in the past (incl...	1
10259	Well, MADE by Elixir and DEVELOPED with Taylor...	1
10260	These strings are really quite good, but I wou...	1

10254 rows × 2 columns

5.2.3 Preprocessing Data

This is a vital part of training the dataset. Here we clean the text. We make text lowercase, remove text in square brackets,remove links,remove special characters and remove words containing numbers.

5.2.4 Training Data/ Evaluation

The main chunk of code that does the whole evaluation of sentimental analysis based on the preprocessed data is a part of this.

The following are the steps followed:

(a) The Accuracy score is calculated and displayed

(b) Navie Bayes, Logistic Regression, Linear SVM are applied on the dataset for evaluation of sentiments

(c) Total positive and negative reviews are counted.

(d) review like sentence is taken as input on the console and if positive the console gives 1 as output and 0 for negative input.

5.3 Required Software and Hardware for development

The section deals with the type of hardware that is used and the software to support the same. Selection and identification of suitable software is also taken into account.

5.3.1 Software Requirement

Software Requirements defines a requirement is requires for development of the proposed system. The description of software is required for developing the system. It is written for developers and users both because developer need the requirement for develop the system and user need it requirement for use on the client side. By using software requirement knows the nature of system in environment it run or executed. Software Requirements for system is given as follows

- Operating system Window 7 or above
- python 3.x
- NLTK Toolkit

5.3.2 Hardware Requirement

Hardware Requirements defines a requirement is requires for development of the proposed system. The requirement of hardware require for computer system for execution. It is use for speed, efficiency, quality, time and storage. The hardware requirement is minimum for the system. Hardware requirements for system is given as follows:

- At least 60GB of usable hard Disk space
- core i5/i7 processor
- 8GB RAM

5.4 Summary

The chapter includes the Algorithms used for implementation and Hardware and Software Requirements for development, and the Modules in Project. In the next chapter, Testing part is described.

Chapter 6

Testing

Testing is the process to prove that the software works correctly. It is used to test whether a particular software satisfies most of the possible conditions. Test cases are designed for the purpose that include certain conditions based on boundary values, module functions, etc that are unfavorable for the software and it is observed whether all the test cases are passed by the software or not.

6.1 White Box Testing

White box testing, sometimes called as glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineers can derive test cases that:

(a) Guarantee that all independent paths within a module have been exercised at least once.

(b) . Exercise all logical decisions on their true and false sides.

(c) Exercise all loops at their boundaries and within their operational bounds.

(d) Exercise internal data structures to ensure their validity

6.2 Black Box Testing

Black box testing also called as behavioral testing focuses on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box techniques. Rather, it

is a complementary approach that is to uncover a different class of errors than white box methods. Black box testing attempts to find errors in the following categories:

(a) Incorrect or missing functions.

(b) Interface errors.

(c) Errors in data structures or external data base access

(d) Behavior or performance errors.

6.3 Manual Testing

Manual testing is testing of the software where tests are executed manually by a QA analyst. It is performed to discover bugs in software under development. In manual testing, the tester checks all the essential features of the given application or software. In the process, the software testers execute the test cases and generate the test reports without the help of any automation software testing tools. It is a classical method of all testing types and helps find bugs in software systems. It is generally conducted by an experienced tester to accomplish the software testing process.

- The initial investment in the Manual testing is comparatively lower.
- Manual testing is not as accurate because of the possibility of the human errors.
- No need for programming in Manual Testing.
- Manual testing proves useful when the test case only needs to run once or twice.
- While testing a small change, an automation test would require coding which could be time-consuming. While you could test manually on the fly

6.4 Automation Testing

In Automated Software Testing, testers write code/test scripts to automate test execution. Testers use appropriate automation tools to develop the test scripts and validate the software. The goal is to complete test execution in a less amount of time. Automated testing entirely relies on the pre-scripted test which runs automatically to compare actual result with the expected results. The helps the tester to determine whether or not an application performs as expected. Automated testing allows you to execute repetitive task and regression test without the intervention of manual tester. Even though all processes are performed automatically, automation requires some manual effort to create initial testing scripts.

- The initial investment in the automated testing is higher.
- Automated testing is a reliable method, as it is performed by tools and scripts. There is no testing fatigue.
- Programming knowledge is a must in automation testing.
- Automation testing is useful when frequently executing the same set of test cases
- Testing coverage can be increased because automation testing tool never forgets to check even the smallest unit

6.5 Test cases Identification and Execution

Test case is the set of inputs along with the output and some additional information like

6.5.1 Test CaseID 1 :-

- Input :- Good Music Product
- Expected Output :-support vector machine:-positive , Naive Bayes Classifier :- positive, Logestic Regression:- Positive
- Actual Output :-support vector machine:-positive , Naive Bayes Classifier :- positive, Logestic Regression:- Positive

6.5.2 Test CaseID 2 :-

- Input :- I am not statisfied by these muscial instrument
- Expected Output :-support vector machine:- Negative, Naive Bayes Classifier :- Negative, Logestic Regression:- Negative
- Actual Output :-support vector machine:-Negative , Naive Bayes Classifier :- Negative, Logestic Regression:- Negative

6.6 Summary

All these is the Information about the testing in the project.

Chapter 7

Result

The following are the result of the project.

In this project we are using three algorithms that is Support Vector Machine,Logestic regression ,Naive Bayes .

Classifier	Support Vector Machine	Logestic Regression	Naive Bayes
Accuracy score	0.9570	0.960019	0.9570

Chapter 8

Conclusion and Future Work

Sentiment analysis deals with the classification of texts based on the sentiments they contain. This article focuses on a typical sentiment analysis model consisting of three core steps, namely data preparation, review analysis and sentiment classification, and describes representative techniques involved in those steps.

Sentiment analysis is an emerging research area in text mining and computational linguistics, and has attracted considerable research attention in the past few years.Future research shall explore sophisticated methods for opinion and product feature extraction, as well as new classification models that can address the ordered labels property in rating inference. Applications that utilize results from sentiment analysis is also expected to emerge in the near future.

The future of sentiment analysis is going to continue to big deeper, far past the surface of thr number of likes , comments and shares , and aim to reach and truly understand , the significance of social media instruction and what they tell us about consumers behind the screens.As a result , sentiment analysis is becoming more important for businesses as the data underlying those interaction grows larger and more complex .

Bibliography

- S. ChandraKala1 and C. Sindhu2, "OPINION MINING AND SENTIMENT CLASSIFICATION: A SURVEY,".Vol .3(1),Oct 2012,420-427.
- Callen Rain,"Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning" Swarthmore College, Department of Computer Science
- G.Angulakshmi , Dr.R.ManickaChezian ,"An Analysis on Opinion Mining: Techniques and Tools". Vol 3(7), 2014
- Kim S-M, Hovy E (2004) Determining the sentiment of opinions. In: Proceedings of the 20th international conference on Computational Linguistics, page 1367. Association for Computational Linguistics, Stroudsburg, PA, USA
- Liu B (2010) Sentiment analysis and subjectivity. In: Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca
- Sarvabhotla K, Pingali P, Varma V (2011) Sentiment classification: a lexical similarity based approach for extracting subjectivity in documents. Inf Retrieval 14(3):337–353
- . Zhou S, Chen Q, Wang X (2013) Active deep learning method for semi-supervised sentiment classification. Neurocomputing 120(0):536–546. Image Feature Detection and Description
- Mukharji A, Liu B, Glance N (2012) Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 21st, International Conference on World Wide Web, WWW '12. ACM, New York, NY, USA. pp 191–200
- Kristina T (2003) Stanford log-linear part-of-speech tagger.
- (2014) Scikit-learn. http://scikit-learn.org/stable