PRODUCT REVIEWS CLASSIFICATION AND SPAM DETECTION

Name of Students:	Enrollment No.
Nisheeth Singh Choudhary	9913103458
Mukesh Kumar Mishra	9913103462
Saurabh Mishra	9913103660

Name of Supervisor: Mr.Ankur Kulhari
(Assistant Professor)

DECEMBER 2016

Submitted for the partial fulfillment of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

CONTENTS

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature of Student	Signature of Student
Name: Nisheeth Singh Choudhary Enrollment no: 9913103458	Name: Mukesh Kumar Mishra Enrollment no : 9913103462

Signature of Student

Name: Saurabh Mishra

Enrollment no: 9913103660

Date: 20-12-2016

CERTIFICATE

This is to certify that the work titled **Product Reviews Classification and Spam Detection** submitted by **Nisheeth Singh Choudhary, Mukesh Mishra, Saurabh Mishra** in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science and Engineering from Jaypee Institute of Information Technology University, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor:

Name of Supervisor Mr.Ankur Khulari

Designation: Assistant Professor

Date: 20-12-2016

ACKNOWLEDGMENT

We would like to place on record my deep sense of gratitude to Mr.Ankur Khulari (Mentor) for his constant support, encouragement, generous guidance and useful suggestions.

We would like to thank Jaypee Institute of Information Technology, Noida, for their invaluable guidance and assistance, without which the accomplishment of the task would have never been possible. We also thank them for giving this opportunity to explore into the real world and realize the interrelation of theoretical concepts and their practical application.

We also wish to extend my thanks to other classmates for their insightful comments and constructive suggestions to improve the quality of this project work.

Signature of Student

Name: Nisheeth Singh Choudhary

Enrollment no. 9913103458

Signature of Student

Name: Mukesh Kumar Mishra

Enrollment no. 9913103462

Signature of Student

Name: Saurabh Mishra

Enrollment no: 9913103660

Date: 20-12-2016

SUMMARY

Product review analysis is one of the key topics for study in recent time. In this project we focused on classification of reviews and grading them whether they are genuine or not by extracting reviews directly from the website using appropriate tools.

Reviews extracted from the website and then data cleaning was done by removing unwanted punctuations and text. Classification of reviews was done by segregating them as genuine and fake reviews. Classification of reviews along with sentimental analysis increases the accuracy of the system which in turn provides accurate reviews to the user. Next stage after classifying reviews is to rate the reviews using algorithm and presenting them graphically for the better understanding of the user.

Signature of Student

Name: Nisheeth Singh Choudhary

Date: 20-12-2016

Signature of Supervisor

Name: Mr. Ankur Kulh1ari

(Assistant Professor)

Date: 20-12-2016

Signature of Student

Name: Mukesh Mishra

Date: 20-12-2016

Signature of Student

Name: Saurabh Mishra

Date: 20-12-2016

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Acronym/Abbreviation	Expansion
GUI	Graphical User Interface
DB	Database
PU	Positive Unlabeled
PY	Python

Chapter 1 - INTRODUCTION

1.1 GENERAL INTRODUCTION

A **review** is an evaluation of a publication, service, company, a piece of hardware or an event performance such as a movie review, video game review, music review of a composition, book review, car, home appliance, computer; live music concert, play, musical theater show, dance show, or art exhibition. In addition to a critical evaluation, the review's author may assign the work a rating to indicate its relative merit. A compilation of reviews may itself be called a review.

A **user review** refers to a review written by a user or consumer for a product or a service based on his experience as a user of the reviewed product. E-commerce sites often have consumer reviews for products and sellers separately.Usually, consumer reviews are in the form of several lines of texts accompanied by a numerical rating. This text is meant to aid in shopping decision of a prospective buyer. A consumer review of a product usually comments on how well the product measures up to expectations based on the specifications provided by the manufacturer or seller. It talks about performance, reliability, quality defects, if any, and value for money. Often it includes comparative evaluations against competing products. Observations are factual as well as subjective in nature. Consumer review of sellers usually comment on service experienced, and dependability or trustworthiness of the seller. Usually, it comments on factors such as timeliness of delivery, packaging and correctness of delivered items, shipping charges, return services against promises made.

Consumer reviews online have become a major factor in business reputation and brand image. A negative review can damage the reputation of a business and this has created a new industry of reputation management where companies attempt to remove or hide bad reviews so that more favourable content is found when potential customers do research.

An **expert review** usually refers to a review written by someone who has tested several peer products or services to identify which offers the best value for money or the best set of features.

A **bought review** is the system where the creator (usually a company) of a new product pays a reviewer to review his new product**.**

Fake reviews - advertisers, marketers, and other stakeholders have motivation to produce fake positive user reviews for products they wish to promote or fake negative user reviews for products which they wish to disparage.In a fake user review, a person will create a user account and post a user review pretending to be a real person. This is a misuse of the user review system, which universally only invite reviews from typical users and not paid fake personalities.

1.2 OPEN PROBLEMS

The research in the field started with sentiment and subjectivity classification, which treated the problem as a text classification problem. Sentiment classification classifies whether an opinionated document (e.g.product reviews) or sentence expresses a positive or negative opinion. Subjectivity classification determines whether a sentence is subjective or objective . Many real-life applications, however, require more detailed analysis of reviews which doesn’t include any textual matter and only includes numeric rating or star ratings of product exact analysis of such reviews doesn’t provide accuracy in segregation of reviews.

1.3 PROBLEM STATEMENT

The project undertaken aims to develop an efficient system for detecting spam reviews. System aim is to process large number of reviews through data cleaning, feature extraction and sentiment analysis to segregate reviews and then using appropriate algorithm to detect spam reviews and then rating them on the scale of authenticity and finally representing them graphically.

1.4 PROPOSED SOLUTION APPROACH AND ITS NOVELTY AND BENEFITS

There can be various approach in detection of spam reviews. Firstly, we build a dataset of reviews using web scraping then cleaning of data is done for extracting raw useful data. Classification of reviews and rating of reviews is done using rate behavior approach.

Advantages of using proposed approach are:-

- Dataset collection is dynamically done by scraping which helps in collection of fresh updated dataset.
- Cleaning process eliminates the unwanted symbols, punctuations and replacement of repeated text by the required one for further analysis.
- Sentiment analysis of reviews helps in easy segregation and classification of reviews.
- Detection and rating of reviews by using rating review algorithm helps in better understanding of user about reviews and provides user an option for selecting products smartly.

Chapter 2 – BACKGROUND STUDY

2.1 LITERATURE SURVEY

2.1.1 SUMMARY OF PAPERS

[1] Amazon Review Classification and Sentiment Analysis

Aashuthosh Bhatt, Ankit Patel, Harsh Chedda, Kiran Gawande
2015
(IJCSIT) International Journal of Computer Science and Information Technologies, Vol.6(6)

Reviews on Amazon are not only related to the product but also the service given to the customers. If users get clear bifurcation about product reviews and service reviews it will be easier for them to take the decision, in this paper we propose a system that performs the classification of customer reviews followed by finding sentiment of the reviews. A rule based extraction of product feature sentiment is also done. Also we provide a visualization for our result summarization.

http://www.ijcsit.com/docs/Volume%206/vol6issue06/ijcsit2015060652.pdf

[2] Detecting Product Review Spammers using rating Behaviours
Ee-PengLim, Nitin Jindal, Hady W.Lauw, Bing Lin, Viet-An Ngygen

This paper aims to detect users generating spam reviews or review spammers
We identify several characteristic behaviors of review spammers and model these behaviors so as to detect the spammers. In particular, we seek to model the following behaviors. First, spammers may target special products or product groups in order to maximizetheir impact.Second, they tend to deviate from the other reviewers in their ratings of products. We propose scoring methods to measure the degree of spam for each reviewer and apply them on an Amazon review dataset. We then select a subset of highly suspicious reviewers for further scrutiny by our user evaluators with the help of a web based spammer evaluation software specially developed for user evaluation experiments. Our results show that our proposed ranking and supervised methods are erective in discovering spammers and outperform other baseline method based on helpfulness votes alone. We finally show that the detected spammers have more significantimpact on ratings compared with the unhelpful reviewer.

http://www.cs.uic.edu/~liub/publications/cikm-2010-final-spam.pdf

[3] Spotting Fake Reviews Using Positive Unlabeled learning
Huayi Li, Bing Liu, Arjun Mukherjee, Jidong Shao
Computaion y sistemas vol 18, No.3, 2014

Fake review detection has been studied by researchers for several years. However, so far all reported studies are based on English reviews. This paper reports a study of detecting fake reviews in Chinese. Our review dataset is from the Chinese review hosting site Dianping1, which has built a fake review detection system. They are confident that their algorithm has a very high precision, but they don't know the recall. This means that all fake reviews detected by the system are almost certainly fake but the remaining reviews may not be all genuine. This paper first reports a supervised learning study of two classes, fake and unknown. However, since the unknown set may contain many fake reviews, it is more appropriate to treat it as an unlabeled set. This calls for the model of learning from positive and unlabeled examples (or PU-learning). Experimental results show that PU learning not only outperforms supervised learning significantly, but also detects a large number of potentially fake reviews hidden in the unlabeled set that Dianping fails to detect.

https://www.cs.uic.edu/~hil/docs/pu_spam.pdf

[4] Spotting Fake Reviewer Groups in Consumer Reviews
Arjun Mukherjee, Bing Liu, Natalie Glance
ACM 978-1-4563-1229

Opinionated social media such as product reviews are now widely used by individuals and organizations for their decision making. However, due to the reason of profit or fame, people try to game the system by opinion spamming (e.g., writing fake reviews) to promote or demote some target products. For reviews to reflect genuine user experiences and opinions, such spam reviews should be detected. Prior works on opinion spam focused on detecting fake reviews and individual fake reviewers. However, a fake reviewer group of reviewers who work collaboratively to write fake reviews is even more damaging as they can take total control of the sentiment on the target product due to its size. This paper studies spam detection in the collaborative setting, i.e., to discover fake reviewer groups.The proposed method first uses a frequent itemset mining method to find a set of candidate groups. It then uses several behavioral models derived from the collusion phenomenon among fake reviewers and relation models based on the relationships among groups, individual reviewers, and products they reviewed to detect fake reviewer groups.
Additionally, we also built a labeled dataset of fake reviewer groups. Although labeling individual fake reviews and reviewers is very hard, to our surprise labeling fake reviewer groups is much easier. We also note that the proposed technique departs from the traditional supervised learning approach for spam detection because of the inherent nature of our problem which makes the classic supervised learning approach less effective. Experimental results show that the proposed method outperforms

multiple strong baselines including the state-of-the-art supervised classification, regression, and learning to rank algorithms.

https://www.cs.uic.edu/~liub/publications/WWW-2012-group-spam-camera-final.pdf

[5] Review Graph based Online Store Review Spammer Detection
Guan Wang, Sihong Xie, Bing Liu, Philip S. Yu

Online reviews provide valuable information about products and services to consumers. However, spammers are joining the community trying to mislead readers by writing fake reviews. Previous attempts for spammer detection used reviewers' behaviors, text similarity, linguistics features and rating patterns. Those studies are able to identify certain types of spammers, e.g., those who post many similar reviews about one target entity.However, in reality, there are other kinds of spammers who can manipulate their behaviors to act just like genuine reviewers, and thus cannot be detected by the available techniques. In this paper, we propose a novel concept of a heterogeneous review graph to capture the relationships among reviewers, reviews and stores that the reviewers have reviewed. We explore how interactions between nodes in this graph can reveal the cause of spam and propose an iterative model to identify suspicious reviewers. This is the first time such intricate relationships have been identified for review spam detection. We also develop an effective computation method to quantify the trustiness of reviewers, the honesty of reviews, and the reliability of stores. Different from existing approaches, we don't use review text information. Our model is thus complementary to existing approaches and able to find more difficult and subtle spamming activities, which are agreed upon by human judges after they evaluate our results.

https://www.cs.uic.edu/~liub/publications/ICDM-2011-final.pdf

[6] Review Ranking Method for Spam Recognition
Gunjan Ansari, Tanvir Ahmed, M.N. Doja
2016 9th International Conference on Contemporary Computing

E-commerce websites are becoming popular among customers who are buying products online. Online reviews play a major role in selling of online products. Reviews give the customer a complete overview of the product thus making it popular or unpopular among buyers thus increasing its sales. In order to increase sales of a product, reviewers are writing.Fake reviews. In this paper, a review ranking method is proposed. This method assigns a score to each review based different parameters. The reviews having high score are considered to be more helpful or genuine and thus are ranked higher than the reviews having lower score. The lower ranked reviews are fake reviews and thus they are non-useful to the users. The proposed approach is an effective approach which avoids heavy computation

Of learning. Evaluation on real-life flipkart review dataset shows a precision of 83.3% thus showing the effectiveness of proposed model.

https://www.researchgate.net/publication/288516654

2.1.2 INTEGRATED SUMMARY OF THE LITERATURE STUDIED

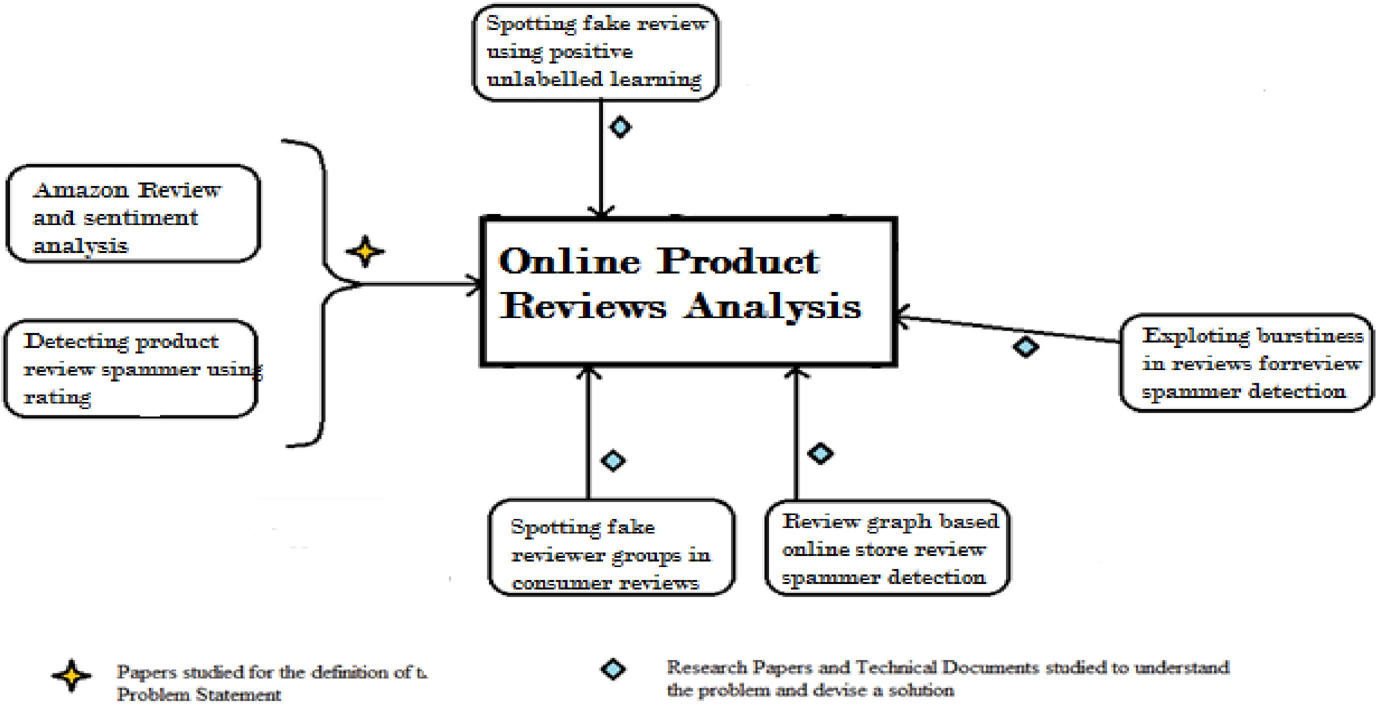

Spotting fake review using positive unlabelled learning
Amazon Review and sentiment analysis
Detecting product review spammer using rating
Online Product Reviews Analysis
Exploting burstiness in reviews forreview spammer detection
Spotting fake reviewer groups in consumer reviews
Review graph based online store review spammer detection
Papers studied for the definition of t. Problem Statement
Research Papers and Technical Documents studied to understand the problem and devise a solution

Figure 2.1: Summary of all papers studied.

Segregation of reviews as Positive, Negative and Product or Service reviews[1] classification of reviews along with sentimental analysis increases the accuracy of the system which in turn provides accurate reviews to the user. Detection of spam product reviews using rating behavior[2] scoring methods to rank reviewers according to the degree they demonstrate spamming behaviors. Removing reviewers with very high spam scores, the highly spammed products and product groups according to our approach will experience more significant changes in aggregate rating and reviewer count compared with removing randomly scored or unhelpful reviewers. Graph based spam review detection[5] method showed how the information in the review graph indicates the causes for spamming and reveals important clues of different types of spammers. Positive unlabeled learning [4] PU learning detects a large number of potential fake reviews hidden in the unlabeled set. Review ranking method [6] iterative review ranking algorithm has been proposed to give a buyer top ranked reviews based on their score thus providing honest reviews to the users. Also with this algorithm, spurious reviews can be detected as they are low ranked.

2.1.3 COMPARISON OF OTHER EXISTING APPROACHES TO THE PROBLEM

In the project, we used the approach of ranking reviews in spam detection as it is suits best for our project to rate reviews. There are other approaches for classification of reviews.
Comparison between rating behavior approach, unlabeled positive learning and rating behavior is documented below:

S.No.	Ranking Reviews Approach	Unlabled positive learning Approach	Rating Behavior approach
1.	In this each review is ranked on the basis of their authenticity	Reviews are divided into two groups of positive and unlabeled	Spam reviews and spammers are rated using statistical methods.
2.	Review ranked higher is genuine and reviews with lower ranks are fake reviews.	Unlabeled consist of both positive and fake reviews. Using statistical method helps in increasing accuracy of detection of fake reviews.	This method uses approach of detecting spammers behavior and rates spammer and their reviews.

Table 2.1: Comparison between various approaches for product review analysis.

2.2 EMPIRICAL STUDY

During the course of the project, following free and open source tools were used:

- **Beautiful Soup library 4.4.0**

It is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

- **Python Compiler**
  Python language is used in this project. Data cleaning, sentiment analysis is done using python inbuilt libraries and for running other tools.
  Following libraries are used :-

CSV:-

The so-called CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. There is no “CSV standard”, so the format is operationally defined by the many applications which read and write it. The lack of a standard means that subtle differences often exist in the data produced and consumed by different applications. These differences can make it annoying to process CSV files from multiple sources. Still, while the delimiters and quoting characters vary, the overall format is similar enough that it is possible to write a single module which can efficiently manipulate such data, hiding the details of reading and writing the data from the programmer.

The **csv** module implements classes to read and write tabular data in CSV format. It allows programmers to say, “write this data in the format preferred by Excel,” or “read data from this file which was generated by Excel,” without knowing the precise details of the CSV format used by Excel. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV formats.

The **csv** module’s **reader** and **writer** objects read and write sequences. Programmers can also read and write data in dictionary form using the **DictReader**and **DictWriter** classes.

Division:-

The current division (/) operator has an ambiguous meaning for numerical arguments: it returns the floor of the mathematical result of division if the arguments are ints or longs, but it returns a reasonable approximation of the division result if the arguments are floats or complex. This makes expressions expecting float or complex results error-prone when integers are not expected but possible as inputs.We propose to fix this by introducing different operators for different operations: x/y to return a reasonable approximation of the mathematical result of the division ("true division"), x//y to return the floor ("floor division"). We call the current, mixed meaning of x/y "classic division".Because of severe backwards compatibility issues, not to mention a major flamewar on c.l.py, we propose the following transitional measures (starting with Python 2.2):Classic division will remain the default in the Python 2 series; true division will be standard in Python 3.0.The // operator will be available to request floor division unambiguously. The future division statement, spelled "from __future__ import division", will change the / operator to mean true division throughout the module. A command line option will enable run-time warnings for classic division applied to int or long arguments; another command line option will make true division the default. The standard library will use the future division statement and the // operator when appropriate, so as to completely avoid classic division.

RE:-

This module provides regular expression matching operations similar to those found in Perl.

Both patterns and strings to be searched can be Unicode strings as well as 8-bit strings. However, Unicode strings and 8-bit strings cannot be mixed: that is, you cannot match a Unicode string with a byte pattern or vice-versa; similarly, when asking for a substitution, the replacement string must be of the same type as both the pattern and the search string.

Regular expressions use the backslash character ('\') to indicate special forms or to allow special characters to be used without invoking their special meaning. This collides with Python’s usage of the same character for the same purpose in string literals; for example, to match a literal backslash, one might have to write '\\\\' as the pattern string, because the regular expression must be \\, and each backslash must be expressed as \\ inside a regular Python string literal.

The solution is to use Python’s raw string notation for regular expression patterns; backslashes are not handled in any special way in a string literal prefixed with 'r'. So r"\n" is a two-character string containing '\' and 'n', while "\n" is a one-character string containing a newline. Usually patterns will be expressed in Python code using this raw string notation.

It is important to note that most regular expression operations are available as module-level functions and methods on compiled regular expressions. The functions are shortcuts that don’t require you to compile a regex object first, but miss some fine-tuning parameters

URLLIB:-

urllib is a package that collects several modules for working with URLs:

- urllib.request for opening and reading URLs
- urllib.error containing the exceptions raised by urllib.request
- urllib.parse for parsing URLs
- urllib.robotparser for parsing robots.txt files

Chapter 3 - ANALYSIS, DESIGN AND MODELING

3.1 REQUIREMENT SPECIFICATION

Software Requirements of a system on which the project can be simulated are:

- Beautiful Soup 4.4.0
- Python compiler (version 3.4.0)

3.2 FUNCTIONAL AND NON_FUNCTIONAL REQUIREMENTS

Functional Requirements

- The system should use different content formats for the source data.
- The system should support error recovery.
- The system should collect and store data that is conceptually correct.
- The system should make sure the semantics of the design is kept in accordance with the programming environment.

Non Functional Requirements

- Performance: system should be able to achieve an adequate performance with the given resources.
- Reliability: the system should be able to provide a certain performance under given preconditions over a certain time.
- Maintainability: The system or parts of the system should be able to be changed and modified subsequently.
- Interoperability: system should be able to interact with one or more other systems.
- Portability: The system should be able to be transferred to a different environment.
- Content: Database must be standardized, information stored should be complete, comprehensive and accurate and up-to-date.
- Usability: The system should be user-friendly, accessible, and easy to use and understand.
- Robustness: The system should be protected from technical break down and should enable to restore application data from a backup.
- Data protection: The system should be secure and privacy constraints should be applied.

3.2 ARCHITECTURE

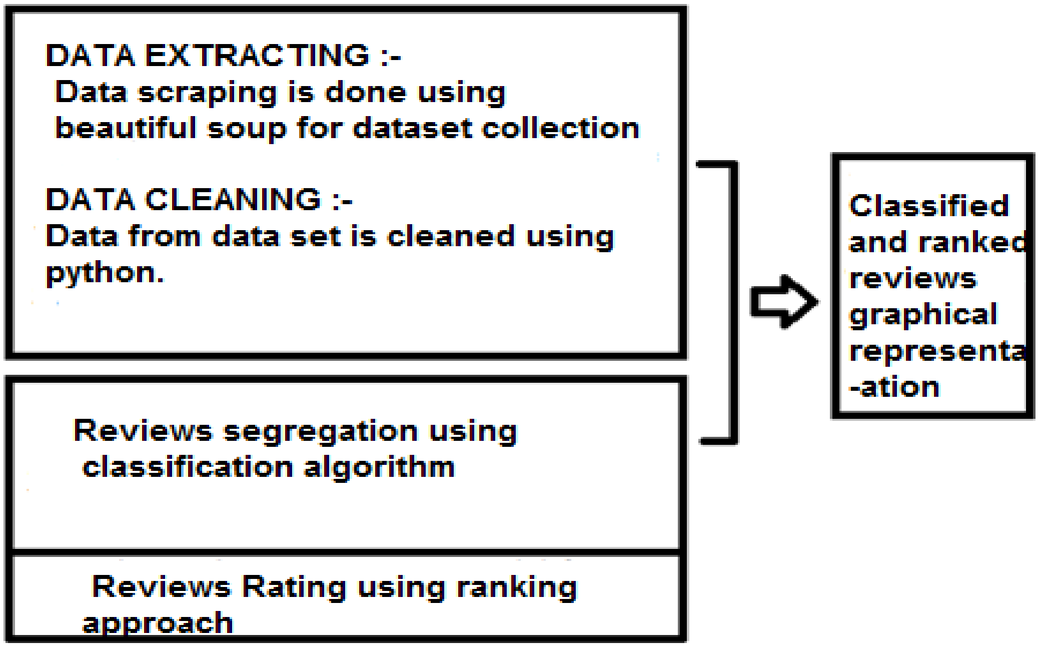
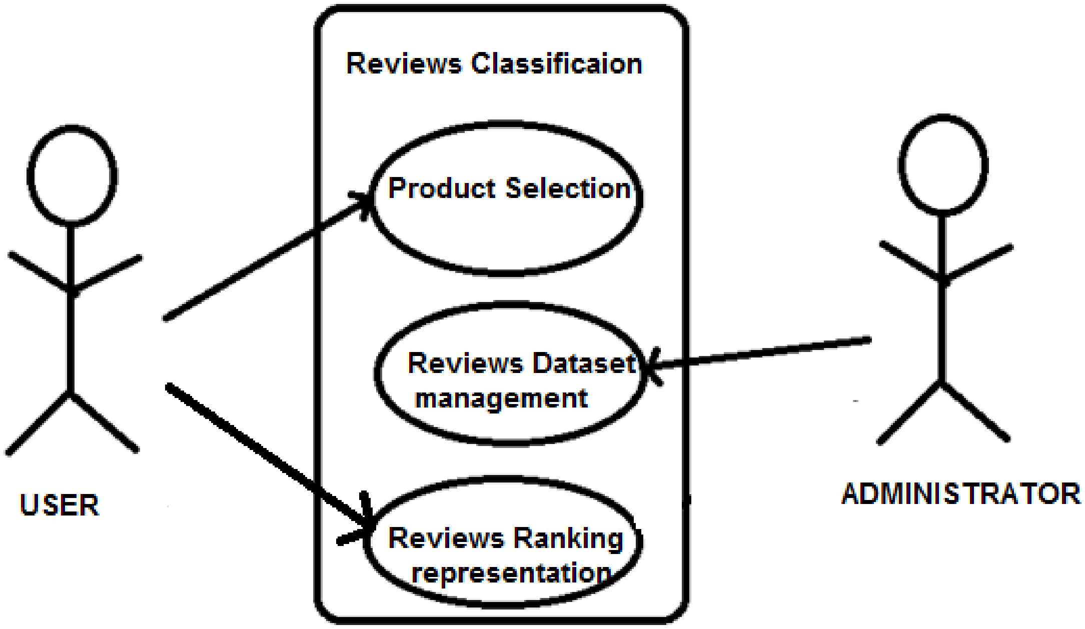

DATA EXTRACTING :-
Data scraping is done using beautiful soup for dataset collection
DATA CLEANING :-
Data from data set is cleaned using python.
Reviews segregation using classification algorithm
Reviews Rating using ranking approach
Classified and ranked reviews graphical representa -ation

Figure 3.1: Overall Architecture of the project

Project aims to implement algorithm to detect spam product reviews. This includes stepwise process of collecting data, storing in database, data cleaning, and classification of reviews and rating them on authenticity.

3.3 DESIGN DOCUMENTATION

3.3.1 USE CASE DIAGRAM



3.4 RISK ANALYSIS AND MITIGATION PLAN

Reviews classification is done previously also but there is no such tool for user to classify products reviews on the basis of authentication. So, there are some risks involved in the implementation.

Risk **Id**	**Description of Risk**	**Risk Area**	**Probability (P)**	**Impact (I)**	**RE (P*I)**	**Mitigation / Contingency Plan**
1.	Product Reviews are not complete	Required information.	0.6	1	0.6	Reviews without text should be rejected
2.	Reviews extraction from the website.	Required information.	0.2	2	0.4	Reviews extracted should be product specific
3.	Ranking of reviews	Analysis phase	.0.2	3	0.6	Ranking of reviews should not overlap

Table 3.2: Risk and Mitigation Plan

S. No.	**Risk Area**	**No. of risk statements**	**Priority**
1.	Extraction	6	2
2.	Sentiment analysis	8	1

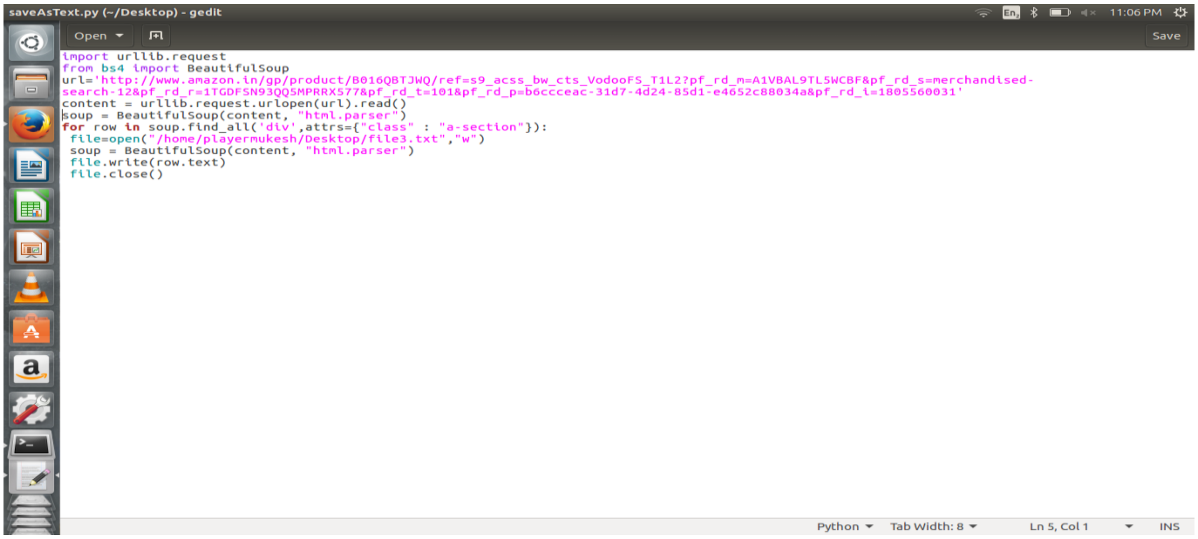
Table 3.3: Risk Priority

Chapter 4 – IMPLEMENTATION AND TESTING

4.1 IMPLEMENTATION DETAILS

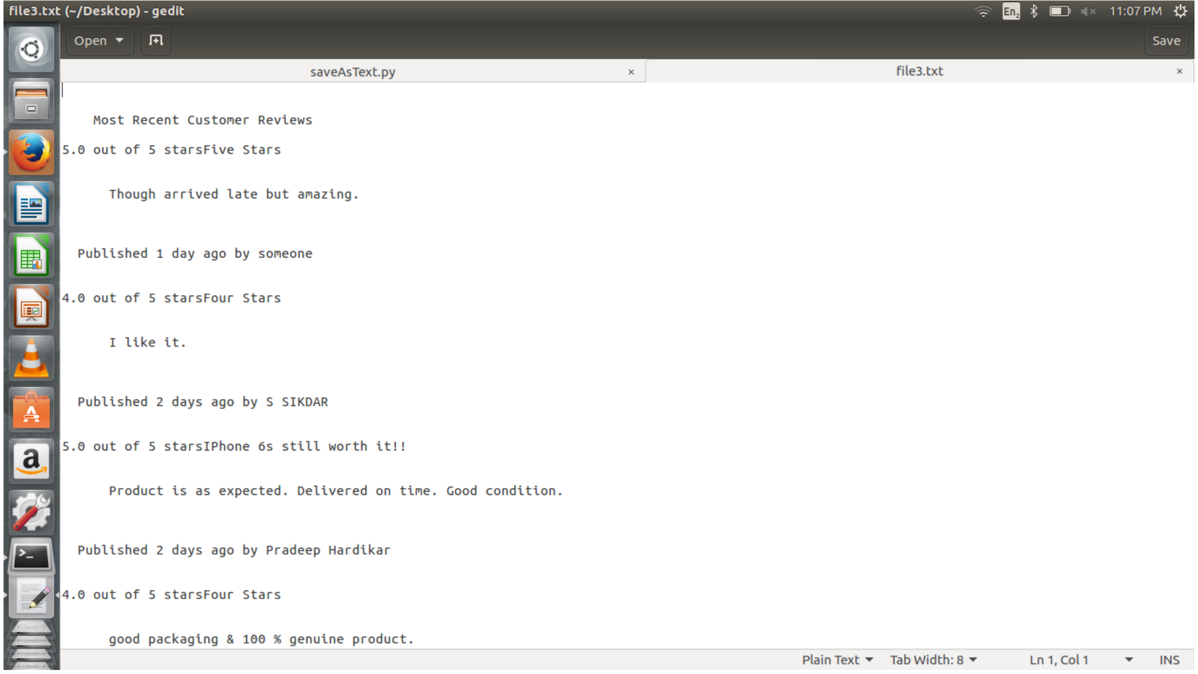
4.1.1 Data Scraping

Review data is extracted using beautiful soup.



```
import urllib.request
from bs4 import BeautifulSoup
url='http://www.amazon.in/gp/product/B016QBTJWQ/ref=s9_acss_bw_cts_VodooFS_T1L2?pf_rd_m=A1VBAL9TL5WCBF&pf_rd_s=merchandised-search-12&pf_rd_r=1TGDFSN93QQ5MPRRX577&pf_rd_t=101&pf_rd_p=b6ccceac-31d7-4d24-85d1-e4652c88034a&pf_rd_i=1805560031'
content = urllib.request.urlopen(url).read()
soup = BeautifulSoup(content, "html.parser")
for row in soup.find_all('div',attrs={"class" : "a-section"}):
 file=open("/home/playermukesh/Desktop/file3.txt","w")
 soup = BeautifulSoup(content, "html.parser")
 file.write(row.text)
 file.close()
```

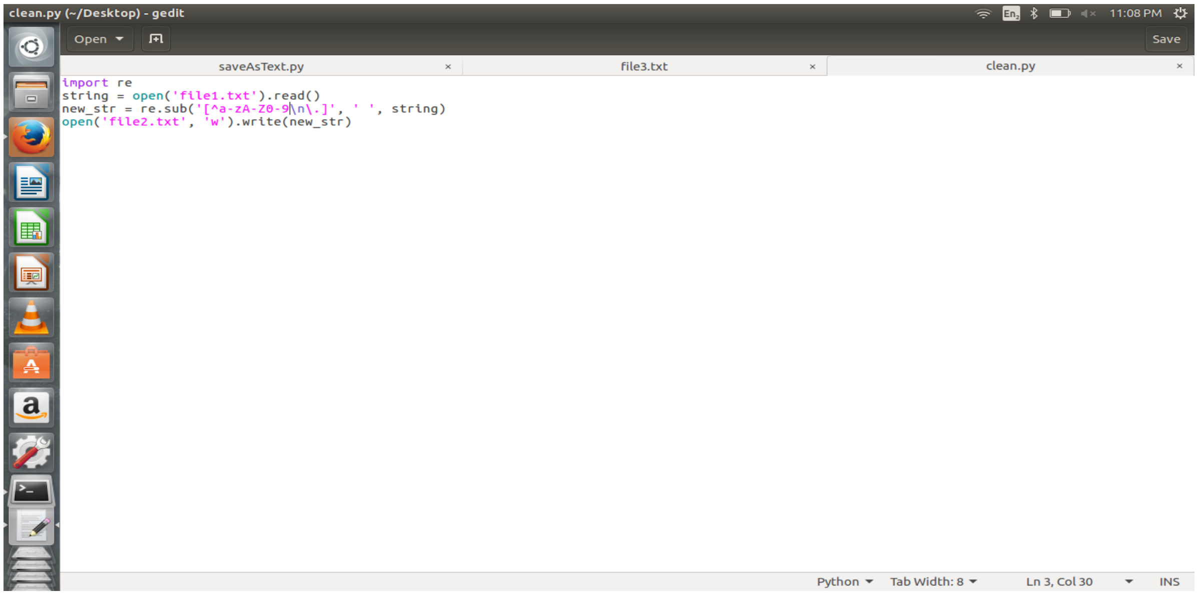
Extracted Reviews stored in savedastext.py



```

    Most Recent Customer Reviews
5.0 out of 5 starsFive Stars

      Though arrived late but amazing.


  Published 1 day ago by someone

4.0 out of 5 starsFour Stars

      I like it.


  Published 2 days ago by S SIKDAR

5.0 out of 5 starsIPhone 6s still worth it!!

      Product is as expected. Delivered on time. Good condition.


  Published 2 days ago by Pradeep Hardikar

4.0 out of 5 starsFour Stars

      good packaging & 100 % genuine product.
```

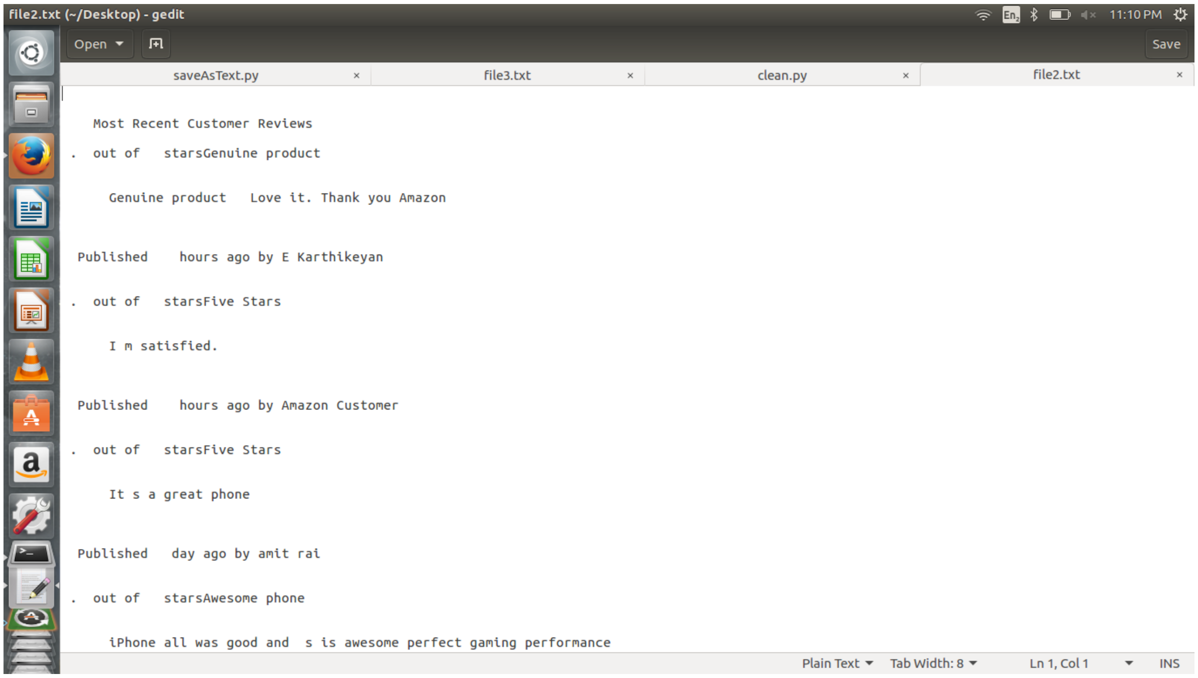
4.1.2 Data Cleaning

Using python libraries cleaning of dataset is done.

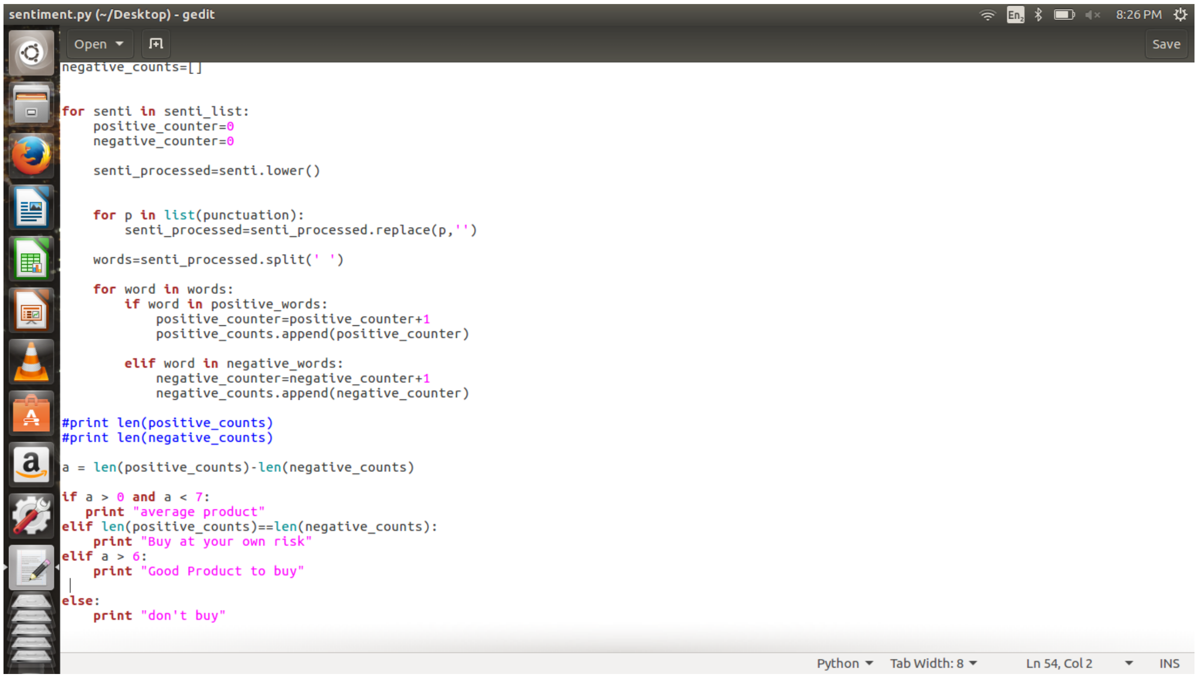


```
import re
string = open('file1.txt').read()
new_str = re.sub('[^a-zA-Z0-9\n\.]', ' ', string)
open('file2.txt', 'w').write(new_str)
```

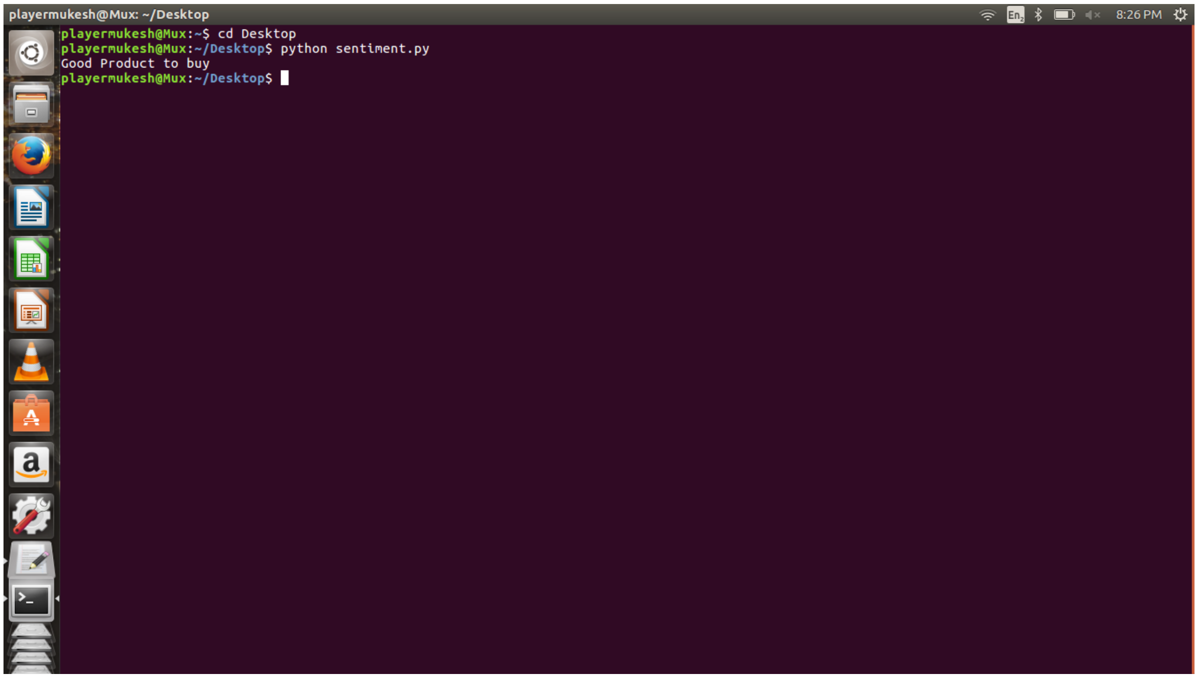
Cleaned dataset is saved in .txt file



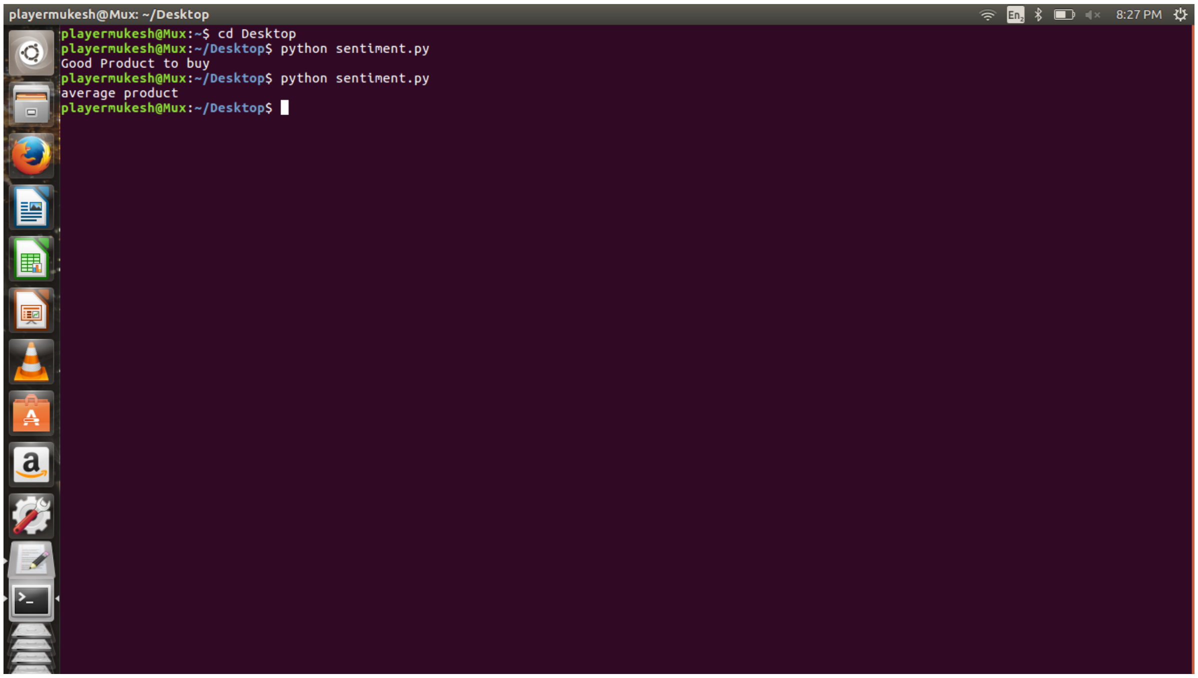
```

   Most Recent Customer Reviews

.  out of   starsGenuine product


     Genuine product   Love it. Thank you Amazon


 Published    hours ago by E Karthikeyan

.  out of   starsFive Stars


     I m satisfied.


 Published    hours ago by Amazon Customer

.  out of   starsFive Stars


     It s a great phone


 Published   day ago by amit rai

.  out of   starsAwesome phone


     iPhone all was good and  s is awesome perfect gaming performance
```



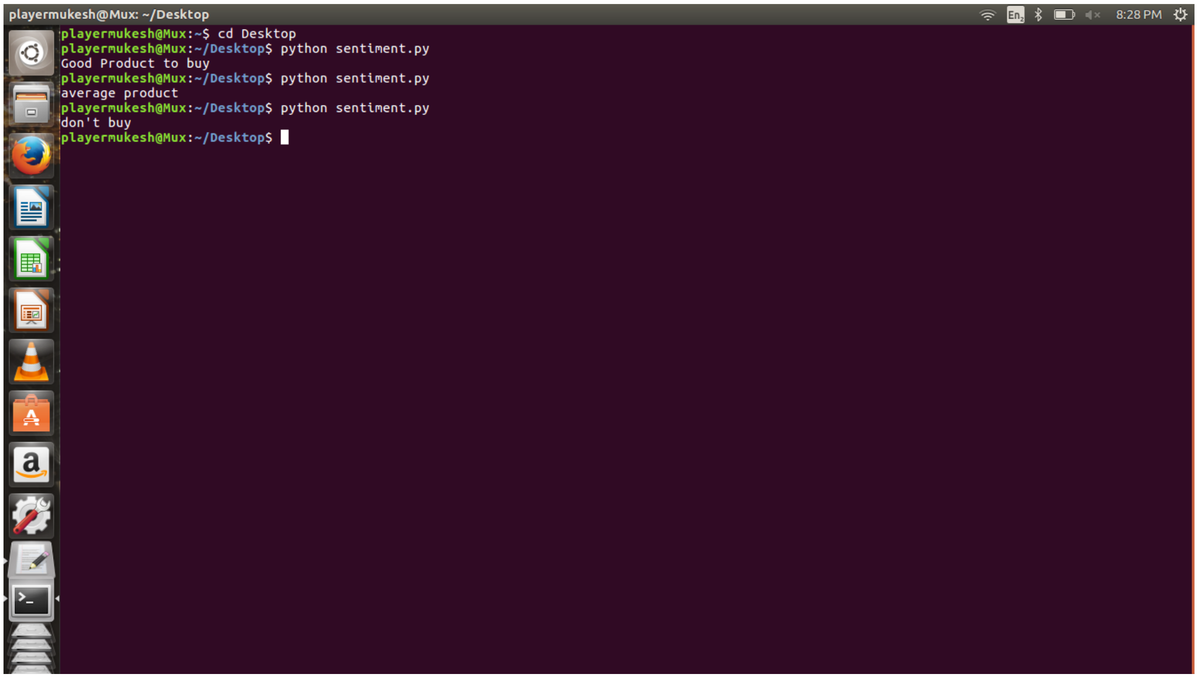
```
negative_counts=[]


for senti in senti_list:
    positive_counter=0
    negative_counter=0

    senti_processed=senti.lower()


    for p in list(punctuation):
        senti_processed=senti_processed.replace(p,'')

    words=senti_processed.split(' ')

    for word in words:
        if word in positive_words:
            positive_counter=positive_counter+1
            positive_counts.append(positive_counter)

        elif word in negative_words:
            negative_counter=negative_counter+1
            negative_counts.append(negative_counter)

#print len(positive_counts)
#print len(negative_counts)

a = len(positive_counts)-len(negative_counts)

if a > 0 and a < 7:
   print "average product"
elif len(positive_counts)==len(negative_counts):
    print "Buy at your own risk"
elif a > 6:
    print "Good Product to buy"

else:
    print "don't buy"
```



```
playermukesh@Mux:~$ cd Desktop
playermukesh@Mux:~/Desktop$ python sentiment.py
Good Product to buy
playermukesh@Mux:~/Desktop$
```



```
playermukesh@Mux:~$ cd Desktop
playermukesh@Mux:~/Desktop$ python sentiment.py
Good Product to buy
playermukesh@Mux:~/Desktop$ python sentiment.py
average product
playermukesh@Mux:~/Desktop$ 
```



```
playermukesh@Mux:~$ cd Desktop
playermukesh@Mux:~/Desktop$ python sentiment.py
Good Product to buy
playermukesh@Mux:~/Desktop$ python sentiment.py
average product
playermukesh@Mux:~/Desktop$ python sentiment.py
don't buy
playermukesh@Mux:~/Desktop$ 
```

4.2 TESTING

4.2.1 TESTING PLAN

Type of Test	Will it be performed?	Comments	Software Component
Requirements Testing	Yes	Requirements Testing must be done to supervise the process of gathering requirements, understanding them, creating test cases based on them, to verify that the representation satisfies the requirements.	Data cleaning, Sentiment analysis
Unit Testing	Yes	Unit testing must be done to test the smallest testable parts of the representation like classes, to determine if they are modeled correctly.	Data scraping, Data cleaning and sentiment analysis
Integration Testing	Yes	Integrating all phases of software and testing them together.	All phases (data scraping, cleaning and Sentiment Analysis)
Performance Testing	Yes	Performance testing needs to be performed to determine the speed or effectiveness of the system.	Data Scraping and sentiment analysis
Stress Testing	Yes	Stress Testing needs to be performed to determine the stability of the system by testing to a breaking point to observe the results.	Data scraping and sentiment analysis
Security Testing	No	Not required at this stage. Maybe required at later stages.	
Load Testing	Yes	Load testing needs to be performed to determine the system's behavior under both normal and anticipated peak load conditions.	Data cleaning, Data extraction
Volume Testing	Yes	Productre views data is stored in large volumes so components must be tested for large dataset.	Data cleaning and sentiment analysis

Table 4.1: Testing Plan

4.2.2 COMPONENT DECOMPOSITION AND TYPE OF TESTING REQUIRED

S. No.	List of Various Components that require testing	Type of Testing Required	Technique for testing
1.	Data scraping, Data cleaning, Sentiment analysis	1.1 Requirements Testing 1.2 Unit Testing 1.3 Integration Testing	Research, Black box testing
2.	Data scraping, Data cleaning, Sentiment analysis	2.1 Performance Testing 2.2 Stress Testing 2.3 Load Testing 2.4 Volume Testing	White Box Testing

Table 4.2: Component Decomposition and Type of Testing Required

4.2.3 TEST CASES

Test case ID	Input	Expected Output	Status
1.1	Conforming to previous researches and technical documents.	Correct understanding of requirements.	Pass
1.2	Product tag on particular website.	Reviews extracted and stored	Pass
1.2	Extracted dataset	Punctuations and other special character and phrases removed.	Pass
1.2	Cleaned Dataset	Segregation of positive and negative reviews	Pass
1.3	Raw Reviews from website.	Segregated reviews.	Pass
2.1	Dataset reviews	Fast extraction and segregation	Pass
2.2	Distorted reviews and images	Segregation	Fail
2.3	Mixed set of dataset	Segregated reviews	Pass
2.4	Large dataset	Storing and segregation	Fail

Table 4.3: Test Cases

4.2.4 LIMITATIONS OF THE SOLUTION

The solution worked out till now is:

- Prepration of architecture.
- Implementation of Data scraping
- Data cleaning
- Implementation of sentiment analysis
- Approach for classification of reviews.

Some limitations were realized during the project, as listed below:

- Reviews with various special characters are difficult to classify.
- Reviews with only images and characters cannot be classified.
- Review rankings can overlap so it will be difficult for classifying such reviews.
- Data extraction of reviews for some products differ from expected dataset.
- Product with few reviews are difficult to classify and accuracy also decreases in detection of authenticated product.

Chapter 5 – FINDINGS AND CONCLUSION

5.1 FINDINGS & CONCLUSION

- Product reviews are of great importance as they set the image of a particular product and also influence customer choices.
- Product reviews classification helps in segregation of genuine and fake reviews by proper analysis.
- Reviews extracted for dataset and cleaning is done using python as it most advanced language for machine learning.
- Based on reviews ranking customers can get clear picture of genuine products.
- Website can also use it as tool for detection of spammers.
- Ranking of reviews can be done using rating review approach [6] along with other appropriate approaches.
- Fake reviews and reviewer can be detected using this tool.
- Improvement in implementation of ranking phase is still required.

5.2 FUTURE WORK

- In the next phase deployment of a proper functioning tool is in focus.
- Some improvement is required in implementation of ranking system.
- Graphical representation of reviews using charts.
- comparison between various reviews of same product fromvarious websites
- Creation of proper GUI for representation of review analysis and making tool more user friendly is our prime objective.

C. REFERENCES

[1] Aasuthosh Bhatt, Ankit Patel, Harsh Chedda, Kiran Gawande, “Amazon Review Classification and Sentiment Analysis”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol.6(6) , 2015, 5107-5110 http://www.ijcsit.com/docs/Volume%206/vol6issue06/ijcsit2015060652.pdf

[2] Ee-PengLim, Nitin Jindal, Hady W.Lauw, Bing Lin, Viet-An Ngygen, “Detecting Product Review Spammers using Rating Behaviours” , ACM 978-1-4503-0099 http://www.cs.uic.edu/~liub/publications/cikm-2010-final-spam.pdf

[3] Arjun Mukherjee, Bing Liu, Natalie Glance “Spotting Fake Reviewer Group ConsumerReviews” ACM 978-1-4563-1229 https://www.cs.uic.edu/~liub/publications/WWW-2012-group-spam-camera.pdf

[4] Huayi Li, Bing Liu, Arjun Mukherjee, Jidong Shao “Spotting Fake Reviews Using Positive Unlabeled learning” Computaion y sistemas vol 18, No.3, 2014 https://www.cs.uic.edu/~hil/docs/pu_spam.pdf

[5] Guan Wang, Sihong Xie, Bing Liu, Philip S. Yu “Review Graph based Online Store Review Spammer Detection” https://www.cs.uic.edu/~liub/publications/ICDM-2011-final.pdf

[6] Gunjan Ansari, Tanvir Ahmed, M.N. Doja Review Ranking Method for Spam Recognition 2016 9th International Conference on Contemporary Computing https://www.researchgate.net/publication/288516654