A Project Report on

RANKING ANAYASIS FOR ONLINE CUSTOMER REVIEWS

Submitted in the partial fulfilment of the requirements for the Major Project of

BACHELOR OF TECHNOLOGY
In
INFORMATION TECHNOLOGY

Submitted by

GAMPA SAI SHIVA	18071A1274
PERAM VARSHITHA	18071A12A0
SREEMENTH N	18071A12A6
SUMANTH V	18071A12B1

Under the esteemed guidance of

Mr. B. JALENDER
Associate Professor,
Dept. of Information Technology,
VNRVJIET

DEPARTMENT OF INFORMATION TECHNOLOGY

VNR Vignana Jyothi Institute of Engineering & Technology
(Autonomous Institute, Accredited by NAAC with 'A++' grade and NBA)
Bachupally, Nizampet (S.O.) Hyderabad- 500 090

June 2022

VNR Vignana Jyothi Institute of Engineering & Technology

Autonomous Institute, Accredited by NAAC with 'A++' grade and NBA)
Bachupally, Nizampet (S.O.) Hyderabad- 500 090

Department of Information Technology

Date: June 2022

CERTIFICATE

This is to certify that the project work entitled **"RANKING ANALYSIS FOR ONLINE CUSTOMER REVIEWS"** is being submitted by **GAMPA SAI SHIVA (18071A1274), PERAM VARSHITHA (18071A12A0), SREEMANTH N (18071A12A6), SUMANTH V (18071A12B1)** in partial fulfilment for the award of Degree of **BACHELOR OFTECHNOLOGY** in **INFORMATION TECHNOLOGY** to the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2021-22is a record of bonafide work carried out by them under our guidance and supervision.

The results embodied in this report have not been submitted by the students to any other University or Institution for the award of any degree or diploma.

GUIDE

Mr. B. Jalender
Associate Professor,
Dept. of IT,
VNRVJIET,
Hyderabad.

HEAD OF DEPARTMENT

Dr. D. SRINIVASA RAO
Head of Department,
Dept. of IT,
VNRVJIET,
Hyderabad.

VNR Vignana Jyothi Institute of Engineering & Technology
Autonomous Institute, Accredited by NAAC with 'A++' grade and NBA)
Bachupally, Nizampet (S.O.) Hyderabad- 500090.

Department of Information Technology

Date: June 2022

DECLARATION

I hereby declare that the project entitled "RANKING ANALYSIS FOR ONLINE CUSTOMER REVIEWS" submitted forthe B. Tech Degree is my original work and the project has not formed thebasis for the award of any degree, associateship, fellowship or any other similar titles.

Signature of the Student:

GAMPA SAI SHIVA	PERAM VARSHITHA	SREEMANTHN	SUMANTH V
18071A1274	18071A12A0	18071A12A6	18071A12B1

Place:
Date:

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved **President, Shri D. Suresh Babu, VNR Vignana Jyothi Institute of Engineering & Technology** for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved Principal, Dr C.D. Naidu for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved professor **Dr Srinivasa Rao Dammavalam, Associate Professor and Head, Department of Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad - 500090** for the valuable guidance and suggestions, keen interest and encouragement extended throughout the project work.

We take immense pleasure to express our deep sense of gratitude to our beloved Guide **Mr. B. Jalender, Associate Professor in Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad,** for his valuable suggestions and rare insights, for a constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed to the successful completion of our project work.

1. GAMPA SAI SHIVA _____
2. PERAM VARSHITHA _____
3. SREEMANTH N _____
4. SUMANTH V _____

Table of Contents

ABSTRACT

Online shopping websites are growing increasingly popular these days. Because of its ease, simplicity, reliability, and speed, customers are increasingly opting for online purchases rather than going to the stores. The rating of an online product is an important indicator for determining whether or not that product is acceptable to users. The rating is used by customers to assess the quality and excellence of a product. It aids a digital shopper in making a purchasing decision whether the product is good or not. It also aids the producer in making future changes during the manufacturing of the product. As a result, producers and sellers are very worried about client feedback because it has a direct impact on their operations. Unfortunately, spam reviews are produced to promote or demote certain products or services in order to obtain profit or fame. This is referred to as "review spamming." Although the subject of spam review identification has received a lot of attention from communities and scholars in recent years, there is still a need to conduct tests on real-world large-scale review datasets. This will aid in determining the impact of widespread opinion spam in internet reviews.

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Individuals nowadays use the World Wide Web (WWW) as their primary means of self-expression. Using electronic commerce websites, forums, and blogs, people may quickly share their opinions on any goods or service. Everyone on the internet now recognizes the value of online reviews for both customers and suppliers. Ahead purchasing a product or service, the majority of people check reviews. Using these reports vendors plan their upcoming manufacturing and marketing strategies. For example, if several consumers purchasing a specific model of laptop post reviews regarding concerns with the screen design, the manufacturer will be made aware of the problem and will work to remedy it in order to improve customer happiness.

Fake review or spam attacks have become more common these days because anyone can create and send spam reviews online. A spammer is someone who hires people to create a fictitious review of a product or service. Spam reviews are usually created to make money or promote a product or service. This is known as generating fake reviews or spamming fake reviews.

The biggest issue with opinion-sharing platforms is that spammers may easily inflate product buzz by generating fake reviews. These bogus reviews can significantly boost the value of a product or service. For example, if a consumer wants to buy a product online, he or she will generally scroll down to the review section to see what other customers have to say. If the majority of the reviews given by the people who purchased the products are positive, the user may decide to buy it, otherwise, he or she will not. All of this demonstrates that spam reviews have become a major issue in online commerce, resulting in losses for both buyers and manufacturers.

Fig 1.1.1: Fake review example

Fake review can have a financial repercussion on enterprises and create a sense of mistrust among the general public as a result, this issue has recently garnered the attention of the media as well as governments. "Nowadays, spam reviews are becoming quite widespread on websites, and, lately, a photographic company was exposed to thousands of bogus consumer reviews," according to recent media reports from the British Broadcasting Corporation and the New York Times International. As a result, detecting spam reviews is crucial, and if this important issue is not resolved, online review sites may become a place full of lies and, as a result, entirely useless. Major existing commercialized websites which showcase their product reviews, such as Yelp and Amazon, have already made significant advancement in recognizing bogus reviews to combat this issue. Researchers have been studying the subject of spam review for the past few years and have presented many solutions. Spam review detection algorithms utilising real-world datasets, on the other hand, have a lot of space for improvement.

Email and web spam are frequently associated with review spam. Web spam is used to attract visitors by changing the content of a web page in order for it to be highly rated by search engines. Spam email is primarily used for marketing goals. Spam reviews, on the other hand, are distinct in that they convey the erroneous impression of a product or service, and it is extremely difficult to spot spam reviews manually. As a result,

traditional web spam or email spam detection approaches are ineffective for detecting spam reviews. Spam review detection is a difficult task because no one can tell if a review is spam just by reading it.

Fig 1.1.2: Importance of online reviews

As a result, in the last twelve years, strategies for detecting phoney reviews has been actively investigated. However, a survey that can do the summarizing and analysis of various methodologies is still lacking. According to the literature assessment, existing approaches either adapted linguistic methods or used linguistic methods. Separate behavioural features are used to detect spammers as well as spam reviews to classify reviews, the majority of existing works have exclusively used the uni-gram linguistic technique. The uni-gram technique usually yields decent results, but it can sometimes fail in specific instances. For example, the popularity of the following review, "This hotel is not good," is neutral when assessed using the uni-gram technique, with one positive word, "good," and one negative word, "not." However, when the same review is examined using a bi-gram technique, the use of the phrase "not good" creates a negative impression

1.2 EXISTING SOLUTIONS

Currently, machine learning algorithms are used by companies like Amazon to choose pertinent features and determine a product's final rating. It does not, however, verify the veracity of a review using an algorithm. While some sites, such as Yelp.com and Fakespot.com, can be used to spot fraudulent reviews on the internet, there is no universally accepted algorithm for filtering reviews.

The work of Rafay et al.investigated how to predict ratings on business reviews using sentiment analysis and opinion mining algorithms. Based on their dataset, they obtained robust results using binary and multiclass techniques. Two feature extraction techniques were described by the authors, word2vec and Global Vector (Glove). These feature extraction results were then combined with the Multinomial Naive Bayes algorithm, the Deep Learning algorithm and the Convolution Long Short Term Memory algorithm. On CLSTM multiclass, they got 84% accuracy using word2vec and 83 percent accuracy with glove. As not mentioned earlier, Shah et al. proposed an abstract-level sentimental analysis of user reviews. The n-gram classifiers included MaxEnt, Naive Bayes, SVM, and Random Forest classes for POS tags. The new algorithm proved to be more efficient and effective than MaxEnt and Nave Bayes. They achieved an accuracy of 91% with their algorithm. The authors of Haji and al. proposed combining lexicon and machine learning for predictive modeling based on restaurant text reviews. Using their system improves Naive Bayes classification accuracy by 5% to 10%.

Tutubalina et al. introduced the AspeRa rating prediction algorithm, which predicts ratings based on the text of the reviews. Their proposed algorithm measured an accuracy of 87% on Instant Videos, 73% on Toys & Games datasets from Amazon. For a recommender system, Viard et al. proposed a rating prediction algorithm that uses link streams. They used XGBoost set of rules for their experiment cause and their set of rules achieved 78% accuracy. Kumar et al. proposed a version by meansof combiningEEG indicators andsentiment evaluation ofproduct opinions for cli ent products. They used synthetic Bee Colony (ABC) set of rules on EEG dataset and executed seventy two% accuracy.

An algorithm that may remove the intrinsic opinions from customer evaluations was

proposed by Cheng et al. and applied in various ways. Their suggested Stochastic Gradient Descent (SDG) method outperformed other algorithms when they compared the findings with other algorithms. Zhang et al. have suggested the Attention Convolution Collaborative Filtering (Att-ConvCF) technique to increase the feature's efficacy. By changing the feature vector's weights, they combined an attention mechanism with a collaborative filtering technique. They reach 77 percent accuracy using Convolutional Neural Networks (CNN). A method for predicting review ratings was put out by Verma et al . On the Amazon Electronics dataset, they employed long short term memory, gated recurrent neural networks, and deep sequential algorithms for review sentiment analysis. On the dataset, their suggested method had a 66 percent accuracy rate.

1.3 CHALLENGES

Identifying reviews as false or real is the main problem with fake review detection. Fake review identification relies heavily on machine learning. For example, one of the most common jobs in fake review detection is supervised learning, which requires labelled data to differentiate false reviews from genuine reviews based on specified features. When reading a large number of reviews, it might be difficult to distinguish between fake and real ones. Machine learning algorithms can distinguish between bogus and honest evaluations by highlighting linguistic patterns that the human eye may otherwise overlook. Existing fake review detection research may be divided into three categories: Identifying individual spammers, group of spammers, or false reviews in a single website or across many domains.

The dependability of internet material has recently seen significant improvement. There are still obstacles to go through despite the process made. The existing deficiencies in this study area, as well as probable future directions, are highlighted in this section.

- Group of scammers detection: According to the literature, identifying a spammer's gang is an important aspect of detecting false reviews. Because of the large number of spammers, false reviews are spread at precise real-time intervals. As a result, they observed a high level of accuracy in detecting bogus reviews by looking at research that focus on burst patterns. Future research on burst patterns utilizing new tools to detect spammers need more examination.

- A model for detecting explainable fake reviews: Deep learning played an important part in natural language processing, and the results were good. It is, nevertheless, regarded as a "Black Box," as it lacks declarative information for further explanations of the findings. All of the deep learning algorithms for detecting bogus reviews are unintelligible. As a result, it's tough to have faith in the model's performance and outcomes. For example, why do certain deep learning models beat others on one dataset but underperform others on another? What are the capabilities of deep learning models? Fundamental theories can be used to conduct interpretability. So yet, no study has been done to explain how

the false review detection mechanism works. As a result, explainable fake review detection models are required.

- Addressing concept of drift issue: Existing methods may not be suitable for detecting false reviews in a real-world application because the features of the reviews vary over time due to the dynamic nature of the reviews. Furthermore, in real-world applications, the prediction model must be updated often. As a result, an efficient model that can manage the idea drift problem in real-world circumstances is required.

- One model of class classification: In a real-world application, one class classification approach can provide a solution for dealing with unlabeled datasets. For example, one class condition, Random field, has been used to analyze anomalous information in Twitter datasets. Other one-class classification techniques that can deal with unlabeled real-world data include one-class support vector machine (OSVM) and Non-OSVM models. To address the lack of dataset issue in fake review identification, more research into unlabeled fake review datasets is required.

- Detection of cross domain bogus reviews: The problem of cross-domain communication must be appropriately handled. In the detection of bogus reviews, the lack of annotation datasets is a disappointment. A key research direction is applying a model trained in the source domain and tested in the target domains. Because the current literature mainly concentrated on one domain of fake review identification, many proposed models failed when trained and tested in other domains. The authors for example, trained the model in one domain and then tested it in another. When compared to performance in the same domain, the experimental results demonstrate a considerable decline in performance. For the detection of cross-domain fake reviews, more research and investigation are required.

- Detection of multilingual bogus reviews: The multilingual analysis is used to

detect fake reviews. Users can write reviews in any language they like, including English, Chinese, Malay, and Arabic. So far, only a few studies have employed false review datasets from various languages. Spammers are known for writing swiftly and copying text from another dataset. The spammer can also translate the English review to any other language using a language translation service. As a result, there is still a need to address this problem of multilingual false reviews detection.

Chapter 2

LITERATURE SURVEY

E-business destinations are the market's capital in today's world. For the internet buying sector, product reviews have a lot of commercial importance. Customer comments gleaned from product reviews can assist customers, sellers, and producers make better marketing decisions. Online shopping portals are growing in popularity these days. In order to enhance their item sales, businesses are ready to take into account their consumers' purchasing behaviours [1]. Because social networks have been increasingly popular in recent years, there is a risk that data expansion could become uncontrollable in the future as a result of those sites.

Online shopping is an effective way for consumers to exchange money and commodities while putting little effort into it. Due to dependability of sites, no one needs to go outside the market these websites are far more reliable. Additionally, businesses and suppliers may create fresh business plans depending on client input [2]. Online review analysis has become a valuable tool for determining consumer preferences and product innovation possibilities. However, assessing and extracting useful comments from a large number of internet reviews is difficult for both manufacturers and customers.

Customers and business managers can connect with one other through online product reviews. Online review services, like other types of social media, allow company owners and customers to connect on a common platform. According to brightlocal.com, 85% of shoppers examine product reviews before making a purchase choice [3]. Clients are entirely focused on e-commerce objectives, while they wait for the products to arrive. Since everyone is submitting comments on a daily basis, there has been a massive increase in the amount of internet data and information. As a result, it is extremely difficult to precisely extract important data as of now the online world.

It is necessary to conduct research on a product's characteristics and functionalities before purchasing it, and the internet is a great resource for doing so. Buyers can swiftly make purchasing decisions by focusing on the important components, while businesses

may think about enhancing the levels of these qualities and so accurately improve product positioning. In general, a product can have a variety of features. A Smart Phone, for example, offers thousands of features like "screen size," "camera," "memory space," and "sound quality." While some features are more vital than others. Customer reviews are one subset of the information acquired, in which we encounter a product firsthand as users. A method is required that summarises the ideas presented in all of the internet evaluations.

Example: On Amazon's website, users are invited to write comments and reviews. These submissions are evaluated, and the one with the most "useful" hits is shown on the front page. But what happens when the reviews aren't genuine? There have been a few instances when people have created false reviews to malign a brand, or where beautiful reviews have been posted to boost a product's sales. As a result, it's critical to identify key viewpoints and discard the others. Sentiment analysis is used to rate various items and assist consumers in making decisions by further separating the genuine evaluation into positive and negative remarks.

Importance of the Research

- By enabling companies to assess the degree of customer acceptance of their products, the research will offer a solution to the problems that ecommerce suffers as a result of competition.
- Additionally, it will result in more improvements to e-commerce products [4].

Study Motivation

(i) E-commerce offers a large area to explore study entries regarding everything from a client's click to the customer's routing around the E-business site to gathering client feedback.

(ii) One of the most significant hurdles is that we force clients to rely on word coordination. As a necessary step, we must demonstrate to them how to use it.

(iii) Web-based existence has global implications and is one of the explanations for information overload on the internet. A customer can deliver a product that is then uploaded on the Internet in a variety of ways.

(iv) The existing framework provides one component that allows anyone to provide feedback on any item. Individuals from the testing E-shopping site can make false accusations against the first site [5].

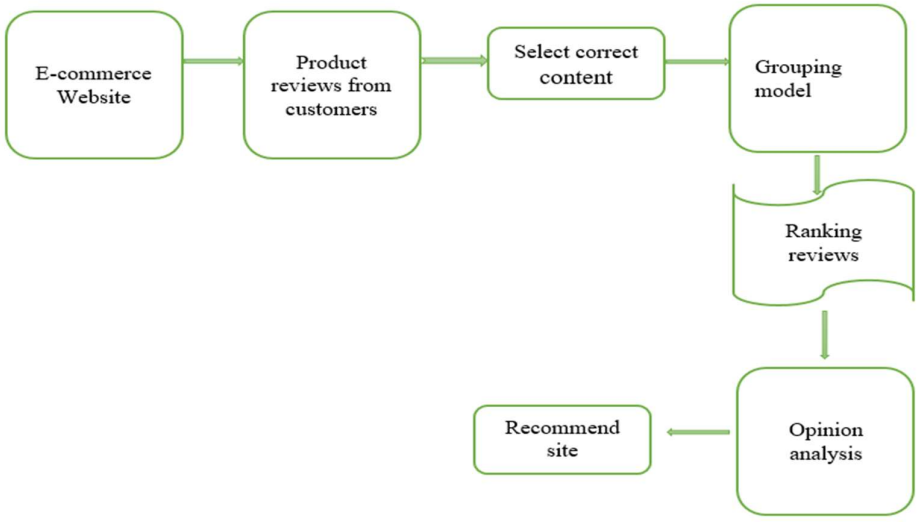

Fig 2.1: Opinion mining process

Algorithms

1. FCM Grouping strategy: With the help of subtle qualities that, in the eyes of customers, rank last, an exceptional features-based positioning approach determines the positioning ratings for a certain product. It uses Gaussian weights to determine the group focus, as well as extensive introduction models and processes for removing bunching. The major objective is to include people's opinions, behaviours, actions, and feelings about certain persons, issues, exercises, topics, and their highlights while analysing reviews of an item and rating its exactness. It does this by using a collection technique and an improving process based on swarms. Fuzzy c-means grouping

approach was utilised for data collecting [6].

2. Dragon fly: This new feature set is built on objectives updating methods with certain default phases such as separation, cohesiveness, alignment and attraction towards a food supply and they are prioritized in accordance with the optimization approach used, the dragon fly algorithm.[7].

3. Opinion mining: Reviewers' satisfaction with the product is checked when opinion mining is done on the reviews. As a consequence, feelings from the words used in natural language assessments are extracted. Supervised and unsupervised learning are the two main methods used in sentiment analysis. The text is manually annotated in supervised learning of sentiments, and the system classifies the sentiments based on this annotation. A sizable, labelled, domain specific training dataset is needed for this sort of study [8].Because of this, the model created using the supervised technique for one domain might not yield correct results when used with data from a different domain. Unsupervised sentiment analysis, on the other hand, generates a model that can be applied to data from other areas. We can utilise a dictionary of sentiments with polarities ascribed to each word in the lexicon-based method. Finally, the overall polarity of a sentence may be determined, and the opinion can be extracted to determine if the review is positive or negative.

Opinion mining is carried out on a database of authentic reviews. It comprises of two sub-processes: Natural Language Processing and Sentiment Analysis.
The reviews are first subjected to Natural Language Processing (NLP). Removing Stop-words, Part-of-Speech (POS) Tagging, Stemming, and Lemmatization are among the sub-processes. This procedure makes use of the Stanford Core-NLP library.

The second step in the opinion mining process is sentiment analysis. A lexicon-based method is used for sentiment analysis. The Sentiwordnet sentiment dictionary is used to determine the positive and negative sentiment scores of individual English words, and the average sentiment score of each review is created by adding the sentiment scores of all the synsets (words) in that review.

CHAPTER - 3

METHODOLOGY

3.1 PROPOSED SYSTEM

With the aid of a collecting technique and a swarm-based improvement system, the objective of this research is to analyse highly recommended web-based business sites. After gathering customer feedback on products from online marketplaces that had a few attributes, a fuzzy c-means (FCM) grouping approach was used to organize the features into categories for a less time-consuming process. Additionally, the Dragonfly Algorithm (DA), a revolutionary component of this study, detects perfect qualities of the things in sites, and an enhanced ideal feature-based placement technique will be directed to finally find the greatest and most intuitive web-based company webpage.. Figure 14 makes a clear representation of the approach used to solve the issue of ranking products based on internet reviews. There are three stages to the resolution process: 1) Fake review detection, 2) Opinion Mining, and 3) clustering

1. **Fake Review Detection**

   The first procedure, the Fake review finding, starts with the input of the original reviews database. The following are some logical guidelines that may be utilised to identify phoney reviews:

   • The same reviewer frequently posts reviews about the same product;

   • There are numerous reviews posted about the same product at the same time;

   • One user only assigns the highest or lowest rating to each product; and

   • There are numerous references to other people, such as "my family," "my sister," etc.

2. **Opinion Mining**

- After that, we switch to our second procedure, opinion mining. On the database of legitimate reviews, opinion mining is done. NLP and Sentiment Analysis are its two sub processes.

- The reviews are first subjected to Natural Language Processing (NLP). Stop-word elimination, Part-of-Speech (POS) tagging, stemming, and lemmatization are among the sub-processes. This procedure makes use of the Stanford Core-NLP library. Sentiment Analysis is the second step in the opinion mining process. For doing sentiment analysis, the lexicon-based method is used. a dictionary of emotions The average sentiment score of each review is determined by summing the sentiment scores of all the synsets (words) in that review. Sentiwordnet is used to obtain the positive and negative sentiment scores of various English words. The third database of review ratings is generated when opinion mining is finished, and it contains reviews together with their average sentiment scores.

3. Product Rating

- This database serves as the input for the final clustering algorithm. We must also provide a star rating to each item because Amazon uses a 5-star system. Depending on the average sentiment ratings of the reviews, we must first establish five separate groups
- Kmeans, hierarchical and DBScan are three distinct clustering techniques. Each reviews is given a cluster after clustering. Each review was given a rating between 1 and 5, depending on the review cluster, and the average was used to reach the final rating for each product.

3.2 REQUIREMENTS

3.2.1 SOFTWARE REQUIREMENTS

The product viewpoint and features, operating system and operating environment, graphics requirements, design limitations, and user documentation are all included in the functional requirements or overall description papers.

The project's broad overview, including its areas of strength and weakness and how to address them, is provided through the appropriate use of requirements and implementation restrictions.

- **Anaconda 3.7 (or)**
- **Google colab (or)**
- **Python idel 3.7 version**

3.2.2 HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- **Ram : min 4GB**
- **Hard disk : min 250GB**
- **Processor : min intel i3**
- **Operating system : windows, linux**

3.2.3 FUNCTIONAL REQUIREMENTS

1. Data Collection
2. Data Pre-processing
3. Training and Testing
4. Modelling
5. Predict

3.3 DATASET

Reviews of Amazon Products: Over 568455 consumer reviews for Amazon products have been collected by the Amazon product database. For each product, the dataset includes basic product id, user id, profile name, a rating score, summary, time when the review is posted, review text etc.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Id	ProductId	UserId	ProfileName	Helpfulne	Helpfulne	Score	Time	Summary	Text					
2	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1.3E+09	Good Quality Dog Foo	I have bought several of the Vitality canned dog food products					
3	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1.35E+09	Not as Advertised	Product arrived labeled as Jumbo Salted Peanuts...the peanuts					
4	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1.22E+09	"Delight" says it all	This is a confection that has been around a few centuries. It is					
5	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1.31E+09	Cough Medicine	If you are looking for the secret ingredient in Robitussin I believ					
6	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1.35E+09	Great taffy	Great taffy at a great price. There was a wide assortment of yu					
7	6	B006K2ZZ7K	ADT0SRK1MGOEU	Twoapennything	0	0	4	1.34E+09	Nice Taffy	I got a wild hair for taffy and ordered this five pound bag. The t					
8	7	B006K2ZZ7K	A1SP2KVKFXXRU1	David C. Sullivan	0	0	5	1.34E+09	Great! Just as good a	This saltwater taffy had great flavors and was very soft and che					
9	8	B006K2ZZ7K	A3JRGQVEQN31IQ	Pamela G. Williams	0	0	5	1.34E+09	Wonderful, tasty taff	This taffy is so good. It is very soft and chewy. The flavors are					
10	9	B000E7L2R4	A1MZYO9TZK0BBI	R. James	1	1	5	1.32E+09	Yay Barley	Right now I'm mostly just sprouting this so my cats can eat the					
11	10	B00171APVA	A21BT40VZCCYT4	Carol A. Reed	0	0	5	1.35E+09	Healthy Dog Food	This is a very healthy dog food. Good for their digestion. Also g					
12	11	B0001PB9FE	A3HDKO7OW0QNK4	Canadian Fan	1	1	5	1.11E+09	The Best Hot Sauce in	I don't know if it's the cactus or the tequila or just the unique c					
13	12	B0009XLVG0	A2725IB4YY9JEB	A Poeng "SparkyGoHome"	4	4	5	1.28E+09	My cats LOVE this "di	One of my boys needed to lose some weight and the other didn					
14	13	B0009XLVG0	A327PCT23YH90	LT	1	1	1	1.34E+09	My Cats Are Not Fans	My cats have been happily eating Felidae Platinum for more th					
15	14	B001GVISJM	A18ECVX2RJ7HUE	willie "roadie"	2	2	4	1.29E+09	fresh and greasy!	good flavor! these came securely packed... they were fresh and					
16	15	B001GVISJM	A2MUGFV2TDQ47K	Lynrie "Oh HELL no"	4	5	5	1.27E+09	Strawberry Twizzlers	The Strawberry Twizzlers are my guilty pleasure - yummy. Six p					

Fig 3.3.1: reviews of amazon product dataset

3.4 WORKFLOW DIAGRAM

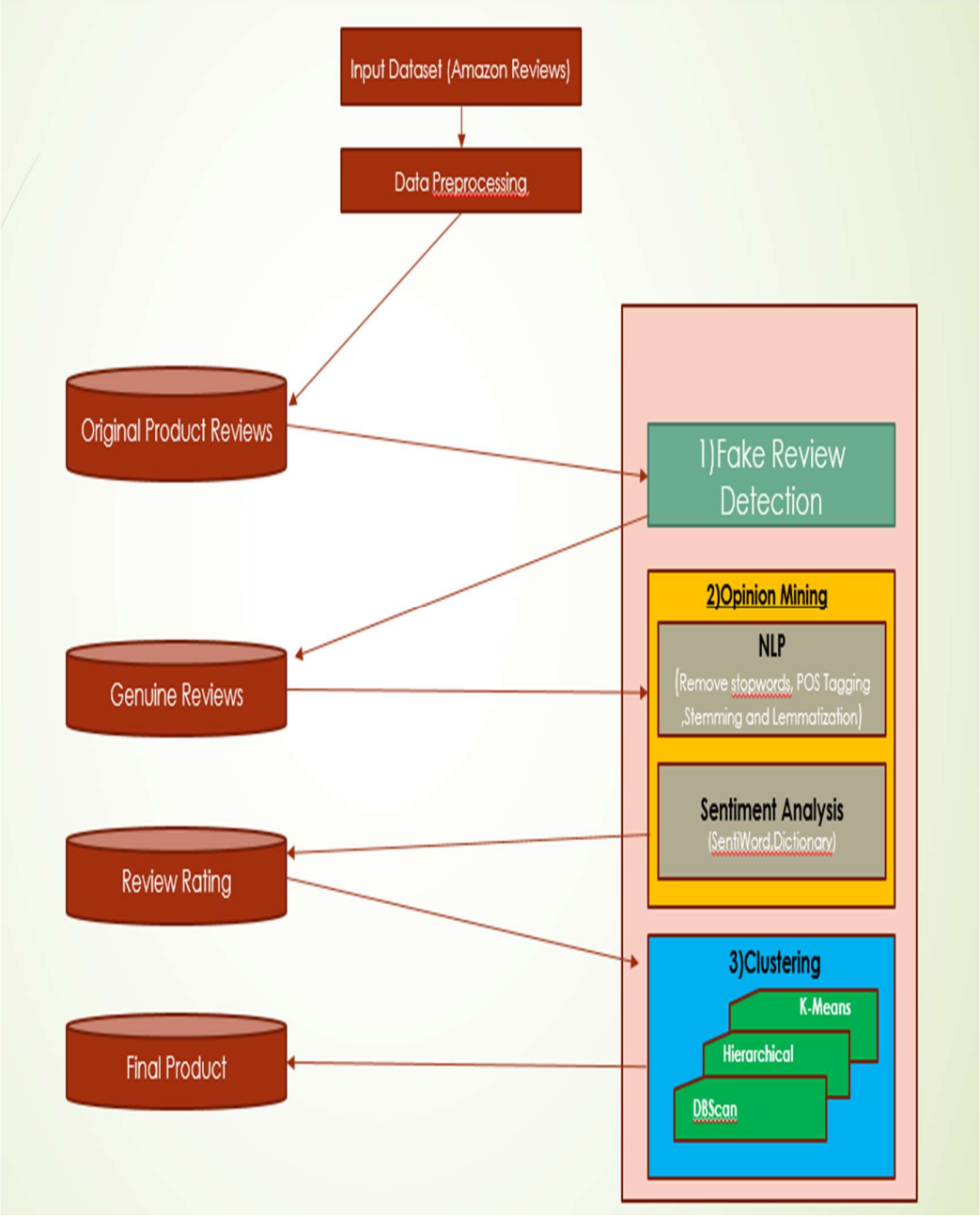

Fig 3.4.1: Proposed methodology flowchart

3.5 UML DIAGRAMS

3.5.1 USECASE DIAGRAM

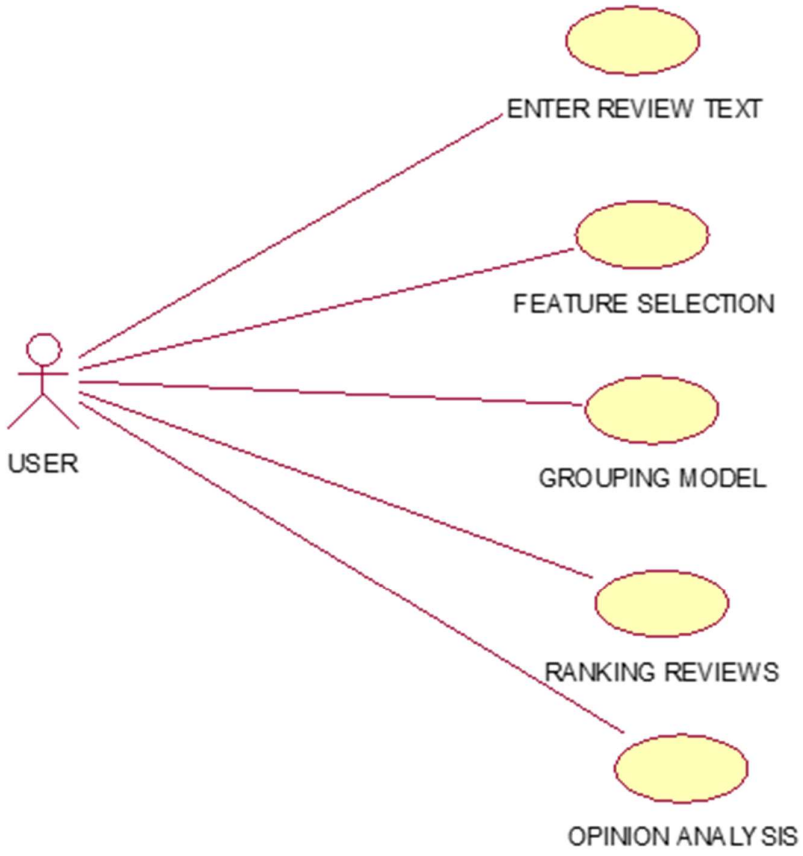

Fig 3.5.1.1: Use case diagram

3.5.2 CLASS DIAGRAM

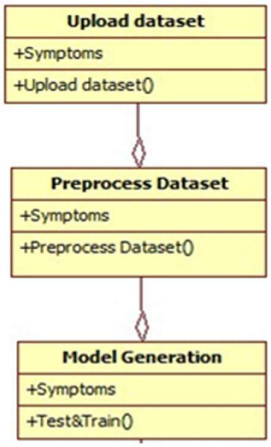

Fig 3.5.2.1: Class diagram

3.5.3 ACTIVITY DIAGRAM

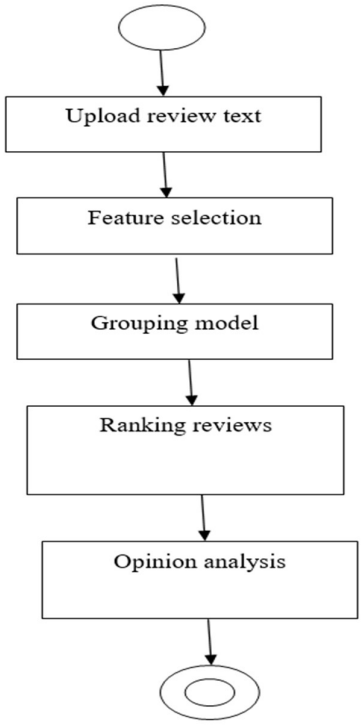

Fig 3.5.3.1: Activity diagram

3.5.4 SEQUENCE DIAGRAM

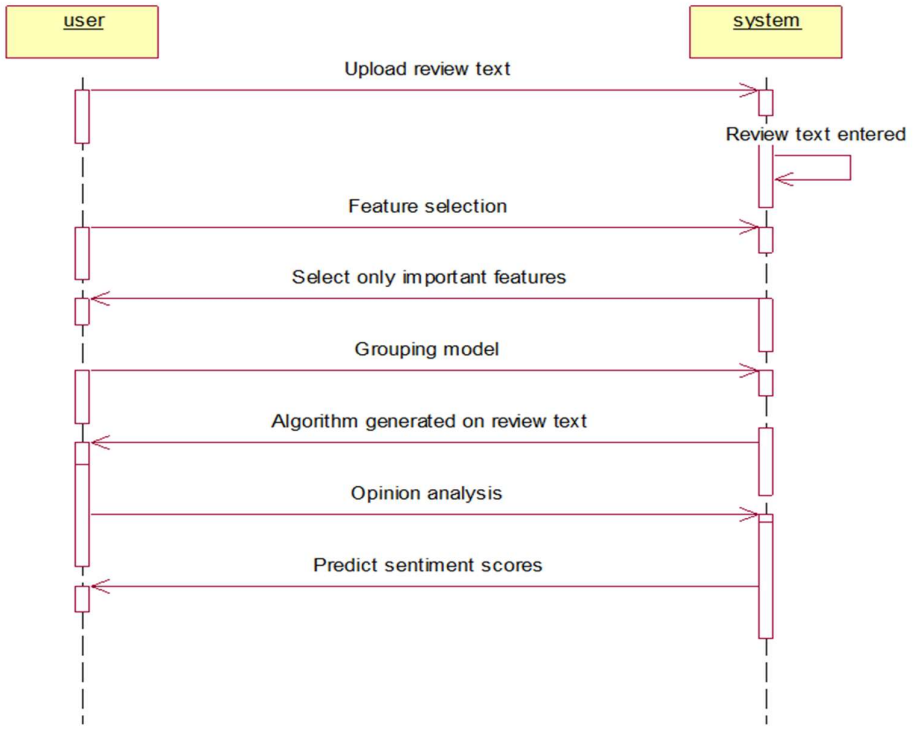

Fig 3.5.4.1: Sequence diagram

3.5.5 COLLABORATION DIAGRAM

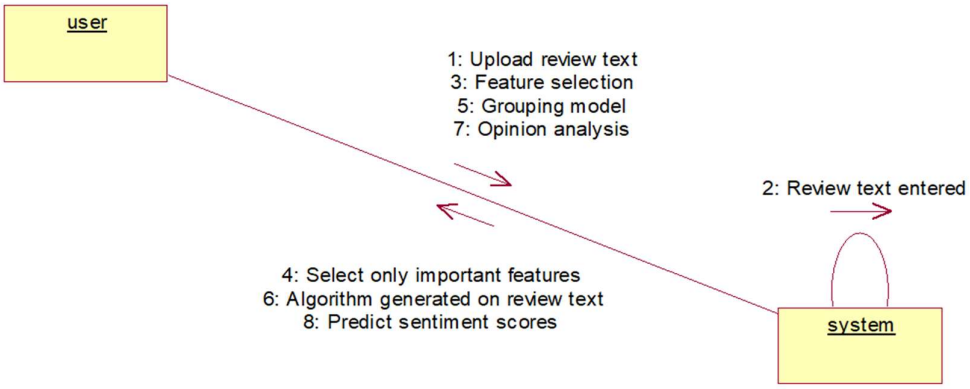

Fig 3.5.5.1: Collaboration diagram

CHAPTER – 4

IMPLEMENTATION

4.1 CODE

4.1.1 WEBSITE

There are six html files home, form, signin, result, index and signup pages. These html files lets user to login to his/her account and then review should be posted in the box given, upon submission of the review , it will be analysed and a score is generated for positive and negative factors.

SIGNIN PAGE CODE

```
<html>
    <head>
        <title></title>
        <style>
            html {
  height: 100%;
}

.button {
                  background-color: #4CAF50; /* Green */
                  border: none;
                  color: white;
                  padding: 15px 32px;
                  text-align: center;
                  text-decoration: none;
                  display: inline-block;
                  font-size: 16px;
                  margin: 4px 2px;
                  cursor: pointer;
                }
                .button2 {background-color: #008CBA;} /* Blue */

body {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  position: relative;
  min-height: 100%;
  background: #F1F1F1;
}

/* Animation Keyframes */
@keyframes scale_header {
  0%   {max-height: 0px; margin-bottom: 0px; opacity: 0;}
  100% {max-height: 117px; margin-bottom: 25px; opacity: 1;}
}

@keyframes input_opacity {
  0%   {transform: translateY(-10px); opacity: 0}
  100% {transform: translateY(0px); opacity: 1}
}

@keyframes text_opacity {
  0% {color: transparent;}
}

@keyframes error_before {
  0%   {height: 5px; background: rgba(0, 0, 0, 0.156); color: transparent;}
  10%  {height: 117px; background: #FFFFFF; color: #C62828}
```

```
  90%  {height: 117px; background: #FFFFFF; color: #C62828}
  100% {height: 5px; background: rgba(0, 0, 0, 0.156); color: transparent;}
}


/* Login Form */
.login-container {
  display: flex;
  flex-direction: column;
  align-items: center;
  position: relative;
  width: 340px;
  height: auto;
  padding: 5px;
  box-sizing: border-box;
}

.login-container img {
  width: 200px;
  margin: 0 0 20px 0;
}

.login-container p {
  align-self: flex-start;
  font-family: 'Roboto', sans-serif;
  font-size: 0.8rem;
  color: rgba(0, 0, 0, 0.5);
}

.login-container p a {
  color: rgba(0, 0, 0, 0.4);
}

.login {
  position: relative;
  width: 100%;
  padding: 10px;
  margin: 0 0 10px 0;
  box-sizing: border-box;
  border-radius: 3px;
  background: #FAFAFA;
  overflow: hidden;
  animation: input_opacity 0.2s cubic-bezier(.55, 0, .1, 1);
  box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.14),
              0 1px 5px 0 rgba(0, 0, 0, 0.12),
              0 3px 1px -2px rgba(0, 0, 0, 0.2);
}

.login > header {
  position: relative;



  width: 100%;
  padding: 10px;
  margin: -10px -10px 25px -10px;
  border-bottom: 1px solid rgba(0, 0, 0, 0.1);
  background: #009688;
  font-family: 'Roboto', sans-serif;
  font-size: 1.3rem;
  color: #FAFAFA;
  animation: scale_header 0.6s cubic-bezier(.55, 0, .1, 1), text_opacity 1s
cubic-bezier(.55, 0, .1, 1);
  box-shadow: 0px 2px 2px 0px rgba(0, 0, 0, 0.14),
              0px 1px 5px 0px rgba(0, 0, 0, 0.12),
              0px 3px 1px -2px rgba(0, 0, 0, 0.2);
}

.login > header:before {
  content: '';
  display: flex;
  justify-content: center;
  align-items: center;
  position: absolute;
  width: 100%;
  height: 5px;
  padding: 10px;
  margin: -10px 0 0 -10px;
  box-sizing: border-box;
  background: rgba(0, 0, 0, 0.156);
  font-family: 'Roboto', sans-serif;
  font-size: 0.9rem;
  color: transparent;
  z-index: 5;
}

.login.error_1 > header:before,
.login.error_2 > header:before {
  animation: error_before 3s cubic-bezier(.55, 0, .1, 1);
}

.login.error_1 > header:before {
  content: 'Invalid username or password!';
}

.login.error_2 > header:before {
  content: 'Invalid or expired Token!';
}

.login > header h2 {
  margin: 50px 0 10px 0;
}
```

```
.login > header h4 {
  font-size: 0.7em;
  animation: text_opacity 1.5s cubic-bezier(.55, 0, .1, 1);
  color: rgba(255, 255, 255, 0.4);
}

/* Form */
.login-form {
  padding: 15px;
  box-sizing: border-box;
}


/* Inputs */
.login-input {
  position: relative;
  width: 100%;
  padding: 10px 5px;
  margin: 0 0 25px 0;
  border: none;
  border-bottom: 2px solid rgba(0, 0, 0, 0.2);
  box-sizing: border-box;
  background: transparent;
  font-size: 1rem;
  font-family: 'Roboto', sans-serif;
  font-weight: 500;
  opacity: 1;
  animation: input_opacity 0.8s cubic-bezier(.55, 0, .1, 1);
  transition: border-bottom 0.2s cubic-bezier(.55, 0, .1, 1);
}

.login-input:focus {
  outline: none;
  border-bottom: 2px solid #E37F00;
}


/* Submit Button */
.submit-container {
  display: flex;
  flex-direction: row;
  justify-content: flex-end;
  position: relative;
  padding: 10px;
  margin: 35px -25px -25px -25px;
  border-top: 1px solid rgba(0, 0, 0, 0.1);
}

.login-button {
  padding: 10px;



  border: none;
  border-radius: 3px;
  background: transparent;
  font-family: 'Roboto', sans-serif;
  font-size: 0.9rem;
  font-weight: 500;
  color: #E37F00;
  cursor: pointer;
  opacity: 1;
  animation: input_opacity 0.8s cubic-bezier(.55, 0, .1, 1);
  transition: background 0.2s ease-in-out;
}

.login-button.raised {
  padding: 5px 10px;
  color: #FAFAFA;
  background: #E37F00;
  box-shadow: 0px 2px 2px 0px rgba(0, 0, 0, 0.137255),
              0px 1px 5px 0px rgba(0, 0, 0, 0.117647),
              0px 3px 1px -2px rgba(0, 0, 0, 0.2);
}

.login-button:hover {
  background: rgba(0, 0, 0, 0.05);
}

.login-button.raised:hover {
  background: #FDAB43;
}

        </style>
        <script>
            var form = document.getElementById('login');
var buttonE1 = document.getElementById('e1');

buttonE1.addEventListener('click', function () {
  form.classList.add('error_1');
  setTimeout(function () {
    form.classList.remove('error_1');
  }, 3000);
});
        </script>
    </head>
    <body>
        <div class="login-container">
            <section class="login" id="login">
              <header>

                <h4>Login</h4>
              </header>
```

```
            <form class="login-form" action="/signin" method="GET">
              <input type="text" class="login-input" name="user"
placeholder="User" required autofocus/>
              <input type="password" class="login-input" name="password"
placeholder="Password" required/>
              <div class="submit-container">
                <button type="submit" class="login-button">SIGN IN</button>
              </div>
            </form>
            <p style="font-size: smaller;">Register here! <a href="/logon">Sign
Up</a></p>
          </section>

        </div>

    </body>
</html>
```

SIGNUP PAGE

```
<html>
    <head>
        <title></title>
        <style>
            html {
  height: 100%;
}

.button {
                  background-color: #4CAF50; /* Green */
                  border: none;
                  color: white;
                  padding: 15px 32px;
                  text-align: center;
                  text-decoration: none;
                  display: inline-block;
                  font-size: 16px;
                  margin: 4px 2px;
                  cursor: pointer;
                }
                .button2 {background-color: #008CBA;} /* Blue */

body {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  position: relative;
  min-height: 100%;
  background: #F1F1F1;
}

/* Animation Keyframes */
@keyframes scale_header {
  0%   {max-height: 0px; margin-bottom: 0px; opacity: 0;}
  100% {max-height: 117px; margin-bottom: 25px; opacity: 1;}
}

@keyframes input_opacity {
  0%   {transform: translateY(-10px); opacity: 0}
  100% {transform: translateY(0px); opacity: 1}
}

@keyframes text_opacity {
  0% {color: transparent;}
}

@keyframes error_before {
  0%   {height: 5px; background: rgba(0, 0, 0, 0.156); color: transparent;}
  10%  {height: 117px; background: #FFFFFF; color: #C62828}
```

```
  90%  {height: 117px; background: #FFFFFF; color: #C62828}
  100% {height: 5px; background: rgba(0, 0, 0, 0.156); color: transparent;}
}


/* Login Form */
.login-container {
  display: flex;
  flex-direction: column;
  align-items: center;
  position: relative;
  width: 340px;
  height: auto;
  padding: 5px;
  box-sizing: border-box;
}

.login-container img {
  width: 200px;
  margin: 0 0 20px 0;
}

.login-container p {
  align-self: flex-start;
  font-family: 'Roboto', sans-serif;
  font-size: 0.8rem;
  color: rgba(0, 0, 0, 0.5);
}

.login-container p a {
  color: rgba(0, 0, 0, 0.4);
}

.login {
  position: relative;
  width: 100%;
  padding: 10px;
  margin: 0 0 10px 0;
  box-sizing: border-box;
  border-radius: 3px;
  background: #FAFAFA;
  overflow: hidden;
  animation: input_opacity 0.2s cubic-bezier(.55, 0, .1, 1);
  box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.14),
              0 1px 5px 0 rgba(0, 0, 0, 0.12),
              0 3px 1px -2px rgba(0, 0, 0, 0.2);
}

.login > header {
  position: relative;



  width: 100%;
  padding: 10px;
  margin: -10px -10px 25px -10px;
  border-bottom: 1px solid rgba(0, 0, 0, 0.1);
  background: #009688;
  font-family: 'Roboto', sans-serif;
  font-size: 1.3rem;
  color: #FAFAFA;
  animation: scale_header 0.6s cubic-bezier(.55, 0, .1, 1), text_opacity 1s
cubic-bezier(.55, 0, .1, 1);
  box-shadow: 0px 2px 2px 0px rgba(0, 0, 0, 0.14),
              0px 1px 5px 0px rgba(0, 0, 0, 0.12),
              0px 3px 1px -2px rgba(0, 0, 0, 0.2);
}

.login > header:before {
  content: '';
  display: flex;
  justify-content: center;
  align-items: center;
  position: absolute;
  width: 100%;
  height: 5px;
  padding: 10px;
  margin: -10px 0 0 -10px;
  box-sizing: border-box;
  background: rgba(0, 0, 0, 0.156);
  font-family: 'Roboto', sans-serif;
  font-size: 0.9rem;
  color: transparent;
  z-index: 5;
}

.login.error_1 > header:before,
.login.error_2 > header:before {
  animation: error_before 3s cubic-bezier(.55, 0, .1, 1);
}

.login.error_1 > header:before {
  content: 'Invalid username or password!';
}

.login.error_2 > header:before {
  content: 'Invalid or expired Token!';
}

.login > header h2 {
  margin: 50px 0 10px 0;
}
```

```
.login > header h4 {
  font-size: 0.7em;
  animation: text_opacity 1.5s cubic-bezier(.55, 0, .1, 1);
  color: rgba(255, 255, 255, 0.4);
}

/* Form */
.login-form {
  padding: 15px;
  box-sizing: border-box;
}


/* Inputs */
.login-input {
  position: relative;
  width: 100%;
  padding: 10px 5px;
  margin: 0 0 25px 0;
  border: none;
  border-bottom: 2px solid rgba(0, 0, 0, 0.2);
  box-sizing: border-box;
  background: transparent;
  font-size: 1rem;
  font-family: 'Roboto', sans-serif;
  font-weight: 500;
  opacity: 1;
  animation: input_opacity 0.8s cubic-bezier(.55, 0, .1, 1);
  transition: border-bottom 0.2s cubic-bezier(.55, 0, .1, 1);
}

.login-input:focus {
  outline: none;
  border-bottom: 2px solid #E37F00;
}


/* Submit Button */
.submit-container {
  display: flex;
  flex-direction: row;
  justify-content: flex-end;
  position: relative;
  padding: 10px;
  margin: 35px -25px -25px -25px;
  border-top: 1px solid rgba(0, 0, 0, 0.1);
}

.login-button {
  padding: 10px;



  border: none;
  border-radius: 3px;
  background: transparent;
  font-family: 'Roboto', sans-serif;
  font-size: 0.9rem;
  font-weight: 500;
  color: #E37F00;
  cursor: pointer;
  opacity: 1;
  animation: input_opacity 0.8s cubic-bezier(.55, 0, .1, 1);
  transition: background 0.2s ease-in-out;
}

.login-button.raised {
  padding: 5px 10px;
  color: #FAFAFA;
  background: #E37F00;
  box-shadow: 0px 2px 2px 0px rgba(0, 0, 0, 0.137255),
              0px 1px 5px 0px rgba(0, 0, 0, 0.117647),
              0px 3px 1px -2px rgba(0, 0, 0, 0.2);
}

.login-button:hover {
  background: rgba(0, 0, 0, 0.05);
}

.login-button.raised:hover {
  background: #FDAB43;
}

        </style>
        <script>
            var form = document.getElementById('login');
var buttonE1 = document.getElementById('e1');

buttonE1.addEventListener('click', function () {
  form.classList.add('error_1');
  setTimeout(function () {
    form.classList.remove('error_1');
  }, 3000);
});
        </script>
    </head>
    <body>
        <div class="login-container">
            <section class="login" id="login">
              <header>

                <h4>Register</h4>
              </header>
```

```
              <form class="login-form" action="/signup" method="GET">
                <input type="text" class="login-input" name="user"
placeholder="Username" required autofocus/>
                <input type="text" class="login-input" name="name"
placeholder="Name" required>
                <input type="text" class="login-input" name="email"
placeholder="Email" required>
                <input type="text" class="login-input" name="mobile"
placeholder="Mobile Number" required>
                <input type="password" class="login-input" name="password"
placeholder="Password" required/>
                <div class="submit-container">
                  <button type="submit" class="login-button">SIGN UP</button>
                </div>
              </form>
              <p style="font-size: smaller;">Already have an account? <a
href="/login">Sign in</a></p>
            </section>

          </div>

    </body>
</html>
```

HOME PAGE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

    <!-- SEO Meta Tags -->
    <meta name="description" content="Your description">
    <meta name="author" content="Your name">

    <!-- OG Meta Tags to improve the way the post looks when you share the page on
Facebook, Twitter, LinkedIn -->
        <meta property="og:site_name" content="" /> <!-- website name -->
        <meta property="og:site" content="" /> <!-- website link -->
        <meta property="og:title" content=""/> <!-- title shown in the actual
shared post -->
        <meta property="og:description" content="" /> <!-- description shown in the
actual shared post -->
        <meta property="og:image" content="" /> <!-- image link, make sure it's jpg
-->
        <meta property="og:url" content="" /> <!-- where do you want your post to
link to -->
        <meta name="twitter:card" content="summary_large_image"> <!-- to have large
image post format in Twitter -->
        <!-- Webpage Title -->
  <title>Form</title>

  <!-- Styles -->
  <link
href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,400;0,600;0,70
0;1,400&display=swap" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap"
rel="stylesheet">
  <link href="static/css/bootstrap.min.css" rel="stylesheet">
  <link href="static/css/fontawesome-all.min.css" rel="stylesheet">
  <link href="static/css/swiper.css" rel="stylesheet">
<link href="static/css/styles.css" rel="stylesheet">

<!-- Favicon  -->
  <link rel="icon" href="static/images/favicon.png">
</head>

<body>
  <nav id="navbarExample" class="navbar navbar-expand-lg fixed-top navbar-dark"
aria-label="Main navigation">
    <div class="container">

        <!-- Image Logo -->
```

```
        <a class="navbar-brand logo-image" href="/"><img
src="static/images/logo.svg" alt="alternative"></a>

        <!-- Text Logo - Use this if you don't have a graphic logo -->
        <!-- <a class="navbar-brand logo-text" href="index.html">Elma</a> -->

        <button class="navbar-toggler p-0 border-0" type="button"
id="navbarSideCollapse" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="navbar-collapse offcanvas-collapse" id="navbarsExampleDefault">
            <ul class="navbar-nav ms-auto navbar-nav-scroll">
                <li class="nav-item">
                    <a class="nav-link active" aria-current="page"
href="/form">Home</a>
                </li>

                <li class="nav-item">
                    <a class="nav-link active" aria-current="page"
href="#">About</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link active" aria-current="page"
href="/notebook">Cluster-Analysis</a>
              </li>
              <li class="nav-item">
                <a class="nav-link active" aria-current="page"
href="/login">Signout</a>
            </li>


            </ul>
            <span class="nav-item social-icons">
                <span class="fa-stack">
                    <a href="#your-link">
                        <i class="fas fa-circle fa-stack-2x"></i>
                        <i class="fab fa-facebook-f fa-stack-1x"></i>
                    </a>
                </span>
                <span class="fa-stack">
                    <a href="#your-link">
                        <i class="fas fa-circle fa-stack-2x"></i>
                        <i class="fab fa-twitter fa-stack-1x"></i>
                    </a>
                </span>
            </span>
        </div> <!-- end of navbar-collapse -->
    </div> <!-- end of container -->
  </nav> <!-- end of navbar -->
```

```
  <!-- end of navigation -->
   <!-- Header -->
   <header class="ex-header">
    <div class="container">
        <div class="row">
            <div class="col-xl-10 offset-xl-1">
      <h1>RAnking Analysis of Product Review based on Opinion Mining</h1>


                </div> <!-- end of col -->
              </div> <!-- end of row -->
          </div> <!-- end of container -->
      </header> <!-- end of ex-header -->
      <!-- end of header -->
   <!-- Basic -->
   <div class="ex-basic-1 pt-5 pb-5">
    <div class="container">
        <div class="row">
            <div class="col-lg-12">
            <form method="POST" action="/predict">
                <textarea name="text1" placeholder="Say Something: ...." rows="10"
cols="109"></textarea><br><br>

                <input class="example_a" type="submit">
            </form>
          </div> <!-- end of col -->
        </div> <!-- end of row -->
    </div> <!-- end of container -->
</div> <!-- end of ex-basic-1 -->
<!-- end of basic -->
<div class="ex-basic-1 pt-5 pb-5">
  <div class="container">
      <div class="row">
          <div class="col-lg-12">
            <div class="item active">
                  {% if final %}
                  <div>
                      <h2>The Opinion / Sentiment of</h2> '{{ text1 }}' <h2></h2>
                      <div class="item active">

                        {% if prediction_text == 1%}
                        <h2 class="heading-logo p-2"
style="color:rgb(0,0,0);">Review is not fake😀</h2>
                        {% elif prediction_text == 0%}
                        <h2 class="heading-logo p-2"
style="color:rgb(0,0,0);">Review is Fake☹</h2>
                        {% endif %}
                    </div>
                    <br>
                <h2>Score table</h2>
```

```
          <table style="width:100%">
            <tr>
              <th>SENTIMENT METRIC</th>
              <th>SCORE</th>
            </tr>
            <tr>
              <td>Positive</td>
              <td>{{text2}}</td>

            </tr>
            <tr>
              <td>Neutral</td>
              <td>{{text3}}</td>

            </tr>
            <tr>
              <td>Negative</td>
              <td>{{text5}}</td>

            </tr>
            <tr>
              <td>Compound</td>
              <td>{{text4}}</td>

            </tr>
          </table>
                      {% else %}
                      <p></p>
                      {% endif %}
                  </div>
                </div> <!-- end of col -->
              </div> <!-- end of row -->
          </div> <!-- end of container -->
        </div> <!-- end of ex-basic-1 -->
        <!-- end of basic -->
        <br><br><br><br>

              <!-- Footer -->
            <div class="footer">



              <!-- Copyright -->
              <div class="copyright">
                  <div class="container">
                      <div class="row">
                          <div class="col-lg-12">


                              <p class="p-small">Copyright © <a
href="#your-link">2022</a></p>
                          </div> <!-- end of col -->
                      </div> <!-- enf of row -->
                  </div> <!-- end of container -->
              </div> <!-- end of copyright -->
              <!-- end of copyright -->


              <!-- Back To Top Button -->
              <button onclick="topFunction()" id="myBtn">
                  <img src="static/images/up-arrow.png" alt="alternative">
              </button>
              <!-- end of back to top button -->

              <!-- Scripts -->
              <script src="static/js/bootstrap.min.js"></script> <!-- Bootstrap
framework -->
              <script src="static/js/swiper.min.js"></script> <!-- Swiper for image
and text sliders -->
              <script src="static/js/purecounter.min.js"></script> <!-- Purecounter
counter for statistics numbers -->
              <script src="static/js/scripts.js"></script> <!-- Custom scripts -->
          </body>
          </html>
```

4.1.2 PYTHON CODE

```
from flask import Flask, request, render_template
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import nltk
from string import punctuation
import re
from nltk.corpus import stopwords
import sqlite3
import pickle
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB, GaussianNB
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np

nltk.download('stopwords')

set(stopwords.words('english'))

app = Flask(__name__)

@app.route("/")
def home():
    return render_template("home.html")

@app.route('/logon')
def logon():
      return render_template('signup.html')

@app.route('/login')
def login():
      return render_template('signin.html')

@app.route("/signup")
def signup():

    username = request.args.get('user','')
    name = request.args.get('name','')
    email = request.args.get('email','')
    number = request.args.get('mobile','')
    password = request.args.get('password','')
    con = sqlite3.connect('signup.db')
    cur = con.cursor()
    cur.execute("insert into `info` (`user`,`email`,
`password`,`mobile`,`name`) VALUES (?, ?, ?, ?,
?)",(username,email,password,number,name))
    con.commit()
    con.close()
    return render_template("signin.html")

@app.route("/signin")
def signin():

    mail1 = request.args.get('user','')
    password1 = request.args.get('password','')
    con = sqlite3.connect('signup.db')
    cur = con.cursor()
```

```
    cur.execute("select `user`, `password` from info where `user` = ? AND
`password` = ?",(mail1,password1,))
    data = cur.fetchone()

    if data == None:
        return render_template("signin.html")

    elif mail1 == 'admin' and password1 == 'admin':
        return render_template("form.html")

    elif mail1 == str(data[0]) and password1 == str(data[1]):
        return render_template("form.html")
    else:
        return render_template("signup.html")



@app.route('/notebook')
def notebook():
     return render_template('notebook.html')

@app.route('/notebook1')
def notebook1():
     return render_template('NOtebook.html')


@app.route('/form')
def form():
    return render_template('form.html')

model=pickle.load(open('model.pkl','rb'))


@app.route('/predict',  methods=['GET','POST'])
def predict():
    stop_words = stopwords.words('english')

    df = pd.read_csv('data/deceptive-opinion.csv')
    df1 = df[['deceptive', 'text']]
    df1.loc[df1['deceptive'] == 'deceptive', 'deceptive'] = 0
    df1.loc[df1['deceptive'] == 'truthful', 'deceptive'] = 1
    X = df1['text']
    Y = np.asarray(df1['deceptive'], dtype = int)
    X_train, X_test, y_train, y_test = train_test_split(X, Y,
test_size=0.3,random_state=109)
    cv = CountVectorizer()
    x = cv.fit_transform(X_train)
    y = cv.transform(X_test)


    #convert to lowercase
    text1 = request.form['text1'].lower()

    data = [text1]
    vect = cv.transform(data).toarray()
    pred = model.predict(vect)

    if pred == 0:
```

```
        return render_template('result.html', prediction_text=pred)

    elif pred ==1:
        text_final = ''.join(c for c in text1 if not c.isdigit())

        #remove punctuations
        #text3 = ''.join(c for c in text2 if c not in punctuation)

        #remove stopwords
        processed_doc1 = ' '.join([word for word in text_final.split() if
word not in stop_words])

        sa = SentimentIntensityAnalyzer()
        dd = sa.polarity_scores(text=processed_doc1)
        compound = round((1 + dd['compound'])/2, 2)

        return render_template('form.html', prediction_text=pred,
final=compound,
text1=text_final,text2=dd['pos'],text5=dd['neg'],text4=compound,text3=dd[
'neu'])

if __name__ == "__main__":
    app.run(debug=True, host="127.0.0.1", port=5000, threaded=True)
```

4.1.3 COLAB FILES

```
#importing all the required libraries
import pandas as pd
import numpy as np
import sklearn as sk
import pickle
from sklearn.feature_extraction.text import CountVectorizer
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\feature_extraction\image.py:167: I
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/re
  dtype=np.int):
```

```
df = pd.read_csv('data/deceptive-opinion.csv')
```

```
df.head()
```

	deceptive	hotel	polarity	source	text
0	truthful	conrad	positive	TripAdvisor	We stayed for a one night getaway with family ...
1	truthful	hyatt	positive	TripAdvisor	Triple A rate with upgrade to view room was le...
2	truthful	hyatt	positive	TripAdvisor	This comes a little late as I'm finally catchi...
3	truthful	omni	positive	TripAdvisor	The Omni Chicago really delivers on all fronts...
4	truthful	hyatt	positive	TripAdvisor	I asked for a high floor away from the elevato...

```
df.tail()
```

	deceptive	hotel	polarity	source	
1595	deceptive	intercontinental	negative	MTurk	Problems started when I booked the InterC
1596	deceptive	amalfi	negative	MTurk	The Amalfi Hotel has a beautiful website
1597	deceptive	intercontinental	negative	MTurk	The Intercontinental Chicago Magnificent
1598	deceptive	palmer	negative	MTurk	The Palmer House Hilton, while it looks g
1599	deceptive	amalfi	negative	MTurk	As a former Chicagoan, I'm appalled at the

```
#Extracting only the requireed features
df1 = df[['deceptive', 'text']]
df1
```

	deceptive	text
0	truthful	We stayed for a one night getaway with family ...
1	truthful	Triple A rate with upgrade to view room was le...
2	truthful	This comes a little late as I'm finally catchi...
3	truthful	The Omni Chicago really delivers on all fronts...
4	truthful	I asked for a high floor away from the elevato...
...	...	...
1595	deceptive	Problems started when I booked the InterContin...
1596	deceptive	The Amalfi Hotel has a beautiful website and i...

```
#filling the categorical variable deceptive with 0 for fake review and 1 for real review
df1.loc[df1['deceptive'] == 'deceptive', 'deceptive'] = 0
df1.loc[df1['deceptive'] == 'truthful', 'deceptive'] = 1
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexing.py:205: SettingWithC
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/us
  self._setitem_with_indexer(indexer, value)
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: SettingWithCopyW
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/us

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:3: SettingWithCopyW
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/us
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
#Printing Dataframe1
df1
```

	deceptive	text
0	1	We stayed for a one night getaway with family ...
1	1	Triple A rate with upgrade to view room was le...
2	1	This comes a little late as I'm finally catchi...

```
#Taking the input and output features seperately
X = df1['text']
Y = np.asarray(df1['deceptive'], dtype = int)
```

```
#importing MultinomialNB
from sklearn.naive_bayes import MultinomialNB, GaussianNB


#splitting the data into training and testing set  with test size is 30%
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3,random_state=109)
```

1600 rows × 2 columns

```
X_test
```

```
1063    We stayed at the Ritz Carlton two weeks prior,...
21      We went to Chicago to see an exhibit at the Ar...
1480    I recently stayed in The James Hotel in Chicag...
1215    Hyatt Regency Hotel: Good ole Downtown, Chicag...
459     Me and my husband got married here. We loved t...
                              ...
133     Perfect location, clean and courteous staff al...
1252    If you want a 5-star hotel with 1-star service...
254     We had our hotel reservations at another hotel...
386     We became an Ambassador member just before spe...
1240    My experience as Fairmont Chicago Millennium P...
Name: text, Length: 480, dtype: object
```

```
y_test
```

```
array([1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1,
       0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1,
       1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1,
       0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1,
       1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1,
       1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0,
       1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0,
       0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0,
       0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1,
       1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
```

```
            1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1,
            0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
            0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0,
            1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0])
```

```
nb = MultinomialNB()
```

```
#Converting the review (text feature) to numerical features
cv = CountVectorizer()
x = cv.fit_transform(X_train)
y = cv.transform(X_test)
```

```
# Fitting the model
nb.fit(x, y_train)
pickle.dump(nb,open('model.pkl','wb'))
model=pickle.load(open('model.pkl','rb'))
```

```
nb.predict(y)
```

```
array([1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1,
       1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
       0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0,
       0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0,
       1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0,
       1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1,
       1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0])
```

```
# Training Accuracy
nb.score(x, y_train)
```

```
0.9714285714285714
```

```
# Testing Accuracy
nb.score(y, y_test)
```

```
0.85625
```

Implementing with the SVM

```
from sklearn import svm

#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:30: D
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  method='lar', copy_X=True, eps=np.finfo(np.float).eps,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:167: D
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  method='lar', copy_X=True, eps=np.finfo(np.float).eps,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:284: D
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  eps=np.finfo(np.float).eps, copy_Gram=True, verbose=0,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:862: D
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  eps=np.finfo(np.float).eps, copy_X=True, fit_path=True,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1101:
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  eps=np.finfo(np.float).eps, copy_X=True, fit_path=True,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1127:
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  eps=np.finfo(np.float).eps, positive=False):
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1362:
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  max_n_alphas=1000, n_jobs=None, eps=np.finfo(np.float).eps,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1602:
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  max_n_alphas=1000, n_jobs=None, eps=np.finfo(np.float).eps,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1738:
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  eps=np.finfo(np.float).eps, copy_X=True, positive=False):
```

```
#Train the model using the training sets
clf.fit(x, y_train)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

```
#Predict the response for test dataset
y_pred = clf.predict(y)
y_pred
```

```
array([1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1,
       0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0,
       0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1,
```

```
1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0,
1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0,
0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0,
0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,
1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1,
0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1,
0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1,
1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0,
1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,
1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1,
0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1,
0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,
0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0])
```

```
clf.score(x, y_train) #Training accuracy
```

```
1.0
```

```
clf.score(y, y_test)
```

```
0.8166666666666667
```

Notebook 2

Data Loading & Cleaning

```
import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer


# using the SQLite Table to read data.
con = sqlite3.connect('data/database.sqlite')

# filtering only positive and negative reviews i.e.
# not taking into consideration those reviews with Score=3
filtered_data = pd.read_sql_query("""
SELECT *
FROM Reviews
WHERE Score != 3
""", con)

# Give reviews with Score>3 a positive rating,
# and reviews with a score<3 a negative rating.
def partition(x):
    if x < 3:
        return 'negative'
    return 'positive'

#changing reviews with score less than 3 to be positive and vice-versa
actualScore = filtered_data['Score']
positiveNegative = actualScore.map(partition)
filtered_data['Score'] = positiveNegative
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\feature_extraction\image.py:167: D
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/re
  dtype=np.int):
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:30: De
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/re
  method='lar', copy_X=True, eps=np.finfo(np.float).eps,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:167: D
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/re
  method='lar', copy_X=True, eps=np.finfo(np.float).eps,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:284: D
```

```
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  eps=np.finfo(np.float).eps, copy_Gram=True, verbose=0,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:862: D
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  eps=np.finfo(np.float).eps, copy_X=True, fit_path=True,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1101:
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  eps=np.finfo(np.float).eps, copy_X=True, fit_path=True,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1127:
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  eps=np.finfo(np.float).eps, positive=False):
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1362:
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  max_n_alphas=1000, n_jobs=None, eps=np.finfo(np.float).eps,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1602:
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  max_n_alphas=1000, n_jobs=None, eps=np.finfo(np.float).eps,
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\least_angle.py:1738:
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/r
  eps=np.finfo(np.float).eps, copy_X=True, positive=False):
```

```
#Sorting data according to ProductId in ascending order
sorted_data=filtered_data.sort_values('ProductId',
   axis=0, ascending=True, inplace=False, kind='quicksort', na_position='last')

#Deduplication of entries
final=sorted_data.drop_duplicates(subset={
    "UserId","ProfileName","Time","Text"}, keep='first', inplace=False)
final.shape
```

```
(364173, 10)
```

```
final=final[final.HelpfulnessNumerator<=final.HelpfulnessDenominator]
final.shape
```

```
(364171, 10)
```

Data Pre-Processing

```
import re
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

stop = set(stopwords.words('english')) #set of stopwords
sno = nltk.stem.SnowballStemmer('english') #initialising the snowball stemmer

def cleanhtml(sentence): #function to clean the word of any html-tags
    cleanr = re.compile('<.*?>')
    cleantext = re.sub(cleanr, ' ', sentence)
```

```
    return cleantext
#function to clean the word of any punctuation or special characters
def cleanpunc(sentence):
    cleaned = re.sub(r'[?|!|\'|"|#]',r'',sentence)
    cleaned = re.sub(r'[.|,|)|(|\|/]',r' ',cleaned)
    return  cleaned


i=0
str1=' '
final_string=[]
all_positive_words=[] # store words from +ve reviews here
all_negative_words=[] # store words from -ve reviews here.
s=''
for sent in final['Text'].values:
    filtered_sentence=[]
    #print(sent);
    sent=cleanhtml(sent) # remove HTMl tags
    for w in sent.split():
        for cleaned_words in cleanpunc(w).split():
            if((cleaned_words.isalpha()) & (len(cleaned_words)>2)):
                if(cleaned_words.lower() not in stop):
                    s=(sno.stem(cleaned_words.lower())).encode('utf8')
                    filtered_sentence.append(s)
                    if (final['Score'].values)[i] == 'positive':
                #list of all words used to describe positive reviews
                        all_positive_words.append(s)
                    if(final['Score'].values)[i] == 'negative':
               #list of all words used to describe negative reviews reviews
                        all_negative_words.append(s)
                else:
                    continue
            else:
                continue
    #print(filtered_sentence)
    str1 = b" ".join(filtered_sentence) #final string of cleaned words

    final_string.append(str1)
    i+=1


#adding a column of CleanedText which displays
# the data after pre-processing of the review
final['CleanedText']=final_string
print(final['CleanedText'].head(3))
```

Save Cleaned Data

```
# store final table into an SQlLite table for future use.
conn = sqlite3.connect('final.sqlite')
c=conn.cursor()
conn.text_factory = str
```

```
final.to_sql('Reviews', conn, schema=None, if_exists='replace',
             index=True, index_label=None, chunksize=None, dtype=None)
```

Importing the Library for Clustering

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
# from sklearn.cross_validation import cross_val_score
from sklearn.model_selection import cross_val_score
from collections import Counter
from sklearn.metrics import accuracy_score
#from sklearn import cross_validation
from sklearn.cluster import KMeans
import sqlite3
import pdb
import pandas as pd
import numpy as np
import nltk
import string
import collections
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import DBSCAN
from sklearn.neighbors import NearestNeighbors

from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn import tree

from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer


# using the SQLite Table to read data.
con = sqlite3.connect('./final.sqlite')

#filtering only positive and negative reviews i.e.
# not taking into consideration those reviews with Score=3
final = pd.read_sql_query("""
SELECT *
FROM Reviews
```

```
""", con)

print(final.head(2))
```

```
   index      Id   ProductId          UserId      ProfileName  \
0  138706  150524  0006641040   ACITT7DI6IDDL  shari zychinski
1  138688  150506  0006641040  A2IW4PEEKO2R0U            Tracy

   HelpfulnessNumerator  HelpfulnessDenominator     Score        Time  \
0                     0                       0  positive   939340800
1                     1                       1  positive  1194739200

                                  Summary  \
0                   EVERY book is educational
1  Love the book, miss the hard cover version

                                                Text  \
0  this witty little book makes my son laugh at l...
1  I grew up reading these Sendak books, and watc...

                                         CleanedText
0  b'witti littl book make son laugh loud recit c...
1  b'grew read sendak book watch realli rosi movi...
```

Random Sampling

```
num_points_kmeans = 100000
num_points_hierarchical = 5000

# used to format headings
bold = '\033[1m'
end = '\033[0m'

# ignore yi's for unsupervised learning
d_unsampled = final.drop(['Score'], axis=1)

# dataset for kmeans & DBSCAN clustering is d_kmeans
# you can use random_state for reproducibility
d_kmeans = d_unsampled.sample(n=num_points_kmeans, random_state=2)

# dataset for hierarchical
d_hierarchical = d_unsampled.sample(n=num_points_hierarchical, random_state=5)
```

Mean Neighbourhood Distance

```
def Get_distanceMean(points,minPts,previous_distanceMean):


    if (minPts < len(points)):

        nbrs = NearestNeighbors(n_neighbors=minPts).fit(points)
```

```
        distances, indices = nbrs.kneighbors(points)
        d_mean = distances.mean()
        return d_mean

    else:
        return previous_distanceMean



def KNNdist_plot(points,minPts):

    epsPlot = []
    current_distanceMean = previous_distanceMean = 0
    knee_value = knee_found = 0

    for i in range (0,len(points),5):

        current_distanceMean = Get_distanceMean(points[i:],
                                        minPts,previous_distanceMean)
        df = current_distanceMean - previous_distanceMean

        if (df > 0.02 and i > 1 and knee_found == 0):
            knee_value = current_distanceMean
            knee_found = 1
            n_trainingData = i

        epsPlot.append( [i,current_distanceMean] )
        previous_distanceMean = current_distanceMean


    #Plot the kNNdistPlot
    for i in range(0, len(epsPlot)):
                plt.scatter(epsPlot[i][0],epsPlot[i][1],c='r',s=3,marker='o')

    plt.axhline(y=knee_value, color='g', linestyle='-')
    plt.axvline(x=n_trainingData , color='g', linestyle='-')
    plt.show()

    print("Knee value: x=" + str(n_trainingData) + " , y=" + str(knee_value))

    return knee_value
```

Elbow Method

```
def findK(d_vect_std):

    sse = {}
    for k in range(2, 20):
        kmeans = KMeans(n_clusters=k, max_iter=300).fit(d_vect_std)

        print(bold+"\nGroup Counter in Cluster %d is as follows:" % (k) +end)
        print(collections.Counter(kmeans.labels_))
```

```
        # Inertia: Sum of distances of samples to their closest cluster center
        sse[k] = kmeans.inertia_
    plt.figure()
    plt.plot(list(sse.keys()), list(sse.values()))
    plt.xlabel("Number of clusters")
    plt.ylabel("Loss Value")
    plt.show()
```

Cluster Analysis

```
def analyzeClusters(d_labels, k, algo='kmeans'):

    count = collections.Counter(d_labels)
    print("\n")
    print(type(count))
    print(count.items())
    print("cluster size = " + str(len(count.items())))
    k = len(count.items())

    if algo == 'kmeans':
        data = d_kmeans
        cluster_index_start = 1
    elif algo == 'hierarchical':
        data = d_hierarchical
        cluster_index_start = 1
    elif algo == 'dbscan':
        data = d_kmeans #change in last run
        cluster_index_start = 0

    print(bold+"\n*** CLUSTERS FORMED BY %s ALGORITHM is as follows: ***" %algo + end)

    for i in range(cluster_index_start, k+cluster_index_start):
        print("CLUSTER = " + str(i))
        # if point is noise then cluster index will be -1. hence exclude.
        if(count.get(i-1) > 1):

            print(bold+"\nThe Review Text in Cluster %d is as follows:" % (i-1) +end)
            print(data[d_labels == i-1].head(5)['Text'])

        else:
            print("Not enough datapoints to display in this cluster!")
```

K-Means & Hierarchical Clustering on BOW

```
from sklearn.random_projection import sparse_random_matrix
from sklearn.preprocessing import StandardScaler
```

```
count_vect = CountVectorizer(dtype="float") #in scikit-learn
d_kmeans_vect = count_vect.fit_transform(d_kmeans['CleanedText'].values)
d_kmeans_vect.get_shape()

# Standardisation. Set "with_mean=False" to preserve sparsity
scaler = StandardScaler(copy=False, with_mean=False).fit(d_kmeans_vect)
d_kmeans_bow_vect_std = scaler.transform(d_kmeans_vect)


count_vect = CountVectorizer(dtype="float") #in scikit-learn
d_hier_vect = count_vect.fit_transform(d_hierarchical['CleanedText'].values)
d_hier_vect.get_shape()

# Standardisation. Set "with_mean=False" to preserve sparsity
scaler = StandardScaler(copy=False, with_mean=False).fit(d_hier_vect)
d_hier_bow_vect_std = scaler.transform(d_hier_vect)

k = findK(d_kmeans_bow_vect_std)

# Hierarchical Clustering
hierarchichal = AgglomerativeClustering(n_clusters=2).fit(
                                    d_hier_bow_vect_std.toarray())
```

```
Group Counter in Cluster 2 is as follows:
Counter({0: 999, 1: 1})

Group Counter in Cluster 3 is as follows:
Counter({0: 998, 1: 1, 2: 1})

Group Counter in Cluster 4 is as follows:
Counter({0: 997, 1: 1, 3: 1, 2: 1})

Group Counter in Cluster 5 is as follows:
Counter({0: 996, 4: 1, 3: 1, 2: 1, 1: 1})

Group Counter in Cluster 6 is as follows:
Counter({1: 995, 5: 1, 2: 1, 4: 1, 0: 1, 3: 1})

Group Counter in Cluster 7 is as follows:
Counter({0: 994, 4: 1, 1: 1, 2: 1, 3: 1, 6: 1, 5: 1})

Group Counter in Cluster 8 is as follows:
Counter({0: 993, 6: 1, 1: 1, 7: 1, 2: 1, 3: 1, 4: 1, 5: 1})

Group Counter in Cluster 9 is as follows:
Counter({4: 992, 3: 1, 5: 1, 8: 1, 6: 1, 7: 1, 2: 1, 1: 1, 0: 1})

Group Counter in Cluster 10 is as follows:
Counter({0: 991, 6: 1, 1: 1, 2: 1, 9: 1, 5: 1, 3: 1, 7: 1, 4: 1, 8: 1})

Group Counter in Cluster 11 is as follows:
Counter({1: 990, 2: 1, 5: 1, 10: 1, 7: 1, 0: 1, 9: 1, 4: 1, 8: 1, 6: 1, 3: 1})

Group Counter in Cluster 12 is as follows:
Counter({2: 989, 9: 1, 11: 1, 6: 1, 8: 1, 3: 1, 4: 1, 10: 1, 7: 1, 1: 1, 5: 1, 0:

Group Counter in Cluster 13 is as follows:
```

```
pd.options.display.max_colwidth = 200

# Analyze clusters formed by kmeans clustering
kmeans = KMeans(n_clusters=10, max_iter=300).fit(d_kmeans_bow_vect_std)
analyzeClusters(d_labels=kmeans.labels_, k=10, algo='kmeans')

# Analyze clusters formed by hierarchical clustering
analyzeClusters(d_labels=hierarchichal.labels_, k=2, algo='hierarchical')
```

```
<class 'collections.Counter'>
dict_items([(8, 991), (4, 1), (7, 1), (3, 1), (6, 1), (2, 1), (9, 1), (5, 1), (0, 1),
cluster size = 10

*** CLUSTERS FORMED BY kmeans ALGORITHM is as follows: ***
CLUSTER = 1
Not enough datapoints to display in this cluster!
CLUSTER = 2
Not enough datapoints to display in this cluster!
CLUSTER = 3
Not enough datapoints to display in this cluster!
CLUSTER = 4
```

```
Not enough datapoints to display in this cluster!
CLUSTER = 5
Not enough datapoints to display in this cluster!
CLUSTER = 6
Not enough datapoints to display in this cluster!
CLUSTER = 7
Not enough datapoints to display in this cluster!
CLUSTER = 8
Not enough datapoints to display in this cluster!
CLUSTER = 9

The Review Text in Cluster 8 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recommend :
23280     I think this is probably as good as it gets for sugar free chocolate syrup
171368    I love these cornflakes.  I can't believe anyone would say they taste like
63408
245004    Never mind that this dog food is kind of gross looking - my dogs just love
Name: Text, dtype: object
CLUSTER = 10
Not enough datapoints to display in this cluster!


<class 'collections.Counter'>
dict_items([(0, 499), (1, 1)])
cluster size = 2

*** CLUSTERS FORMED BY hierarchical ALGORITHM is as follows: ***
CLUSTER = 1

The Review Text in Cluster 0 is as follows:
230993                              This was a gift for a coffee connoisseur friend
197476    Seeds of Change is a Santa Fe, New Mexico-based health foods company. Surpı
343150                                                                      I lo
77376     I have always loved ghee with everything I prepare and eat - be it my daily
112553    We have a 4 oz stand up electric popcorn popper and find 4 oz bags a littl
Name: Text, dtype: object
CLUSTER = 2
Not enough datapoints to display in this cluster!
```

K-Means & Hierarchical Clustering on tf-IDF

```
from sklearn.random_projection import sparse_random_matrix
count_vect = TfidfVectorizer(dtype="float") #in scikit-learn
d_kmeans_vect = count_vect.fit_transform(d_kmeans['CleanedText'].values)
d_kmeans_vect.get_shape()

# Standardisation. Set "with_mean=False" to preserve sparsity
scaler = StandardScaler(copy=False, with_mean=False).fit(d_kmeans_vect)
d_kmeans_bow_vect_std = scaler.transform(d_kmeans_vect)


count_vect = TfidfVectorizer(dtype="float") #in scikit-learn
d_hier_vect = count_vect.fit_transform(d_hierarchical['CleanedText'].values)
```

```
d_hier_vect.get_shape()

# Standardisation. Set "with_mean=False" to preserve sparsity
scaler = StandardScaler(copy=False, with_mean=False).fit(d_hier_vect)
d_hier_bow_vect_std = scaler.transform(d_hier_vect)


k = findK(d_kmeans_bow_vect_std)

# Hierarchical Clustering
hierarchichal = AgglomerativeClustering(n_clusters=2).fit(
                                        d_hier_bow_vect_std.toarray())
```

```
/home/user/test/lib/python3.6/site-packages/sklearn/feature_extraction/text.py:17
  UserWarning)

Group Counter in Cluster 2 is as follows:
Counter({1: 998, 0: 2})

Group Counter in Cluster 3 is as follows:
Counter({1: 954, 2: 45, 0: 1})

Group Counter in Cluster 4 is as follows:
Counter({1: 954, 0: 44, 2: 1, 3: 1})

Group Counter in Cluster 5 is as follows:
Counter({1: 995, 0: 2, 3: 1, 4: 1, 2: 1})

Group Counter in Cluster 6 is as follows:
Counter({2: 995, 3: 1, 4: 1, 5: 1, 1: 1, 0: 1})

Group Counter in Cluster 7 is as follows:
Counter({3: 989, 0: 6, 6: 1, 5: 1, 1: 1, 2: 1, 4: 1})

Group Counter in Cluster 8 is as follows:
Counter({2: 993, 1: 1, 4: 1, 6: 1, 3: 1, 7: 1, 0: 1, 5: 1})

Group Counter in Cluster 9 is as follows:
Counter({5: 990, 1: 3, 3: 1, 7: 1, 8: 1, 6: 1, 4: 1, 0: 1, 2: 1})
```

```
pd.options.display.max_colwidth = 200

# Analyze clusters formed by kmeans clustering
kmeans = KMeans(n_clusters=12, max_iter=300).fit(d_kmeans_bow_vect_std)
analyzeClusters(d_labels=kmeans.labels_, k=12, algo='kmeans')

# Analyze clusters formed by hierarchical clustering
analyzeClusters(d_labels=hierarchichal.labels_, k=2, algo='hierarchical')
```

```
<class 'collections.Counter'>
dict_items([(0, 989), (3, 1), (8, 1), (2, 1), (1, 1), (11, 1), (9, 1), (6, 1), (5
cluster size = 12

*** CLUSTERS FORMED BY kmeans ALGORITHM is as follows: ***
CLUSTER = 1

The Review Text in Cluster 0 is as follows:
297698    I have used it many times and the flavor is wonderful. I highly recomme
23280     I think this is probably as good as it gets for sugar free chocolate sy
171368    I love these cornflakes.  I can't believe anyone would say they taste l
63408
245004    Never mind that this dog food is kind of gross looking - my dogs just l
Name: Text, dtype: object
CLUSTER = 2
Not enough datapoints to display in this cluster!
CLUSTER = 3
Not enough datapoints to display in this cluster!
CLUSTER = 4
Not enough datapoints to display in this cluster!
```

```
CLUSTER = 5
Not enough datapoints to display in this cluster!
CLUSTER = 6
Not enough datapoints to display in this cluster!
CLUSTER = 7
Not enough datapoints to display in this cluster!
CLUSTER = 8
Not enough datapoints to display in this cluster!
CLUSTER = 9
Not enough datapoints to display in this cluster!
CLUSTER = 10
Not enough datapoints to display in this cluster!
CLUSTER = 11
Not enough datapoints to display in this cluster!
CLUSTER = 12
Not enough datapoints to display in this cluster!


<class 'collections.Counter'>
dict_items([(0, 499), (1, 1)])
cluster size = 2

*** CLUSTERS FORMED BY hierarchical ALGORITHM is as follows: ***
CLUSTER = 1

The Review Text in Cluster 0 is as follows:
230993                              This was a gift for a coffee connoisseur fri
197476    Seeds of Change is a Santa Fe, New Mexico-based health foods company. S
343150                                                                        I
77376     I have always loved ghee with everything I prepare and eat - be it my d
112553    We have a 4 oz stand up electric popcorn popper and find 4 oz bags a li
Name: Text, dtype: object
CLUSTER = 2
Not enough datapoints to display in this cluster!
```

K-Means, Hierarchical Clustering & DBScan on Word2Vec

```
import gensim
import re

w2v_dim = 100

def cleanhtml(sentence): #function to clean the word of any html-tags
    cleanr = re.compile('<.*?>')
    cleantext = re.sub(cleanr, ' ', sentence)
    return cleantext

#function to clean the word of any punctuation or special characters
def cleanpunc(sentence):
    cleaned = re.sub(r'[?|!|\'|"|#]',r'',sentence)
    cleaned = re.sub(r'[.|,|)|(|\|/]',r' ',cleaned)
    return  cleaned
```

```
def trainW2V_model(reviewText):
    #select subset of points for fast execution
    i=0
    list_of_sent=[]

    for sent in reviewText:
        sent = str(sent, 'utf-8')
        filtered_sentence=[]
        sent=cleanhtml(sent)
        for w in sent.split():
            for cleaned_words in cleanpunc(w).split():
                if(cleaned_words.isalpha()):
                    filtered_sentence.append(cleaned_words.lower())
                else:
                    continue
        list_of_sent.append(filtered_sentence)

    w2v_model=gensim.models.Word2Vec(list_of_sent,
                                     min_count=5,size=w2v_dim, workers=4)

    return w2v_model

def computeAvgW2V(w2vTrained_model, reviewText):
    sent_vectors = []; # the avg-w2v for each sentence/review is stored in this list

    for sent in reviewText: # for each review/sentence
        sent_vec = np.zeros(w2v_dim) # as word vectors are of zero length
        cnt_words =0; # num of words with a valid vector in the sentence/review
        sent = str(sent, 'utf-8')
        sent = re.sub("[^\w]", " ",  sent).split()

        for word in sent: # for each word in a review/sentence
            try:
                vec = w2vTrained_model.wv[word]
                sent_vec += vec
                cnt_words += 1
            except:
                pass
        sent_vec /= cnt_words
        sent_vectors.append(sent_vec)

    return np.nan_to_num(sent_vectors)


from sklearn.preprocessing import StandardScaler


w2v_kmeans_Model = trainW2V_model(d_kmeans['CleanedText'].values)
d_kmeans_vect = computeAvgW2V(w2v_kmeans_Model, d_kmeans['CleanedText'].values)

# Standardisation.
scaler = StandardScaler(copy=False).fit(d_kmeans_vect)
d_w2v_kmeans_vect_std = scaler.transform(d_kmeans_vect)
```

```
w2v_hier_Model = trainW2V_model(d_hierarchical['CleanedText'].values)
d_hier_vect = computeAvgW2V(w2v_hier_Model, d_hierarchical['CleanedText'].values)

# Standardisation.
scaler = StandardScaler(copy=False).fit(d_hier_vect)
d_w2v_heir_vect_std = scaler.transform(d_hier_vect)


print("***before finding k***")
## To find the best K for K-means
findK(d_w2v_kmeans_vect_std)

print("***before clustering***")
# Hierarchical Clustering
hierarchichal = AgglomerativeClustering(n_clusters=2).fit(d_w2v_heir_vect_std)

print("***after 1st clustering***")
# Hierarchical Clustering - different K
hierarchichal_test = AgglomerativeClustering(n_clusters=5).fit(d_w2v_heir_vect_std)

print("***after 2nd clustering***")

kneeValue = KNNdist_plot(d_w2v_heir_vect_std,200)
print("***ended clustering***")
```

```
WARNING:gensim.models.base_any2vec:under 10 jobs per worker: consider setting a s
WARNING:gensim.models.base_any2vec:under 10 jobs per worker: consider setting a s
***before finding k***

Group Counter in Cluster 2 is as follows:
Counter({1: 534, 0: 466})

Group Counter in Cluster 3 is as follows:
Counter({0: 501, 1: 274, 2: 225})

Group Counter in Cluster 4 is as follows:
Counter({3: 382, 1: 354, 0: 140, 2: 124})

Group Counter in Cluster 5 is as follows:
Counter({4: 308, 3: 282, 1: 204, 0: 107, 2: 99})

Group Counter in Cluster 6 is as follows:
Counter({3: 296, 0: 262, 4: 166, 1: 152, 2: 78, 5: 46})

Group Counter in Cluster 7 is as follows:
Counter({5: 255, 0: 254, 6: 181, 2: 141, 3: 98, 4: 45, 1: 26})

Group Counter in Cluster 8 is as follows:
Counter({0: 215, 6: 212, 4: 187, 1: 145, 2: 103, 5: 90, 7: 25, 3: 23})

Group Counter in Cluster 9 is as follows:
Counter({2: 205, 8: 185, 6: 173, 3: 140, 0: 105, 1: 88, 4: 68, 5: 23, 7: 13})

Group Counter in Cluster 10 is as follows:
Counter({6: 183, 1: 170, 4: 161, 9: 117, 2: 114, 3: 91, 5: 80, 7: 51, 8: 22, 0: 1

Group Counter in Cluster 11 is as follows:
Counter({3: 183, 8: 160, 7: 156, 1: 122, 0: 112, 2: 77, 5: 64, 10: 58, 9: 40, 4:

Group Counter in Cluster 12 is as follows:
Counter({2: 166, 8: 150, 4: 141, 6: 123, 0: 113, 11: 89, 10: 83, 1: 69, 9: 28, 5:

Group Counter in Cluster 13 is as follows:
Counter({5: 152, 3: 149, 9: 144, 11: 112, 0: 102, 8: 89, 1: 73, 12: 49, 6: 48, 4:

Group Counter in Cluster 14 is as follows:
Counter({10: 143, 7: 136, 6: 123, 0: 103, 3: 100, 1: 89, 9: 78, 8: 66, 13: 50, 5:

Group Counter in Cluster 15 is as follows:
Counter({8: 121, 7: 118, 13: 109, 2: 104, 0: 90, 9: 88, 14: 82, 6: 73, 5: 58, 4:

Group Counter in Cluster 16 is as follows:
Counter({0: 116, 10: 115, 3: 106, 15: 98, 8: 92, 13: 88, 6: 78, 5: 76, 2: 56, 11:

Group Counter in Cluster 17 is as follows:
Counter({3: 103, 12: 90, 10: 88, 5: 86, 13: 85, 0: 70, 15: 70, 8: 69, 2: 68, 11:

Group Counter in Cluster 18 is as follows:
Counter({6: 109, 13: 107, 2: 100, 16: 87, 0: 85, 8: 82, 14: 80, 9: 74, 1: 66, 3:

Group Counter in Cluster 19 is as follows:
```

```
pd.options.display.max_colwidth = 200
# analyzeClusters(d_vect_std=d_w2v_vect_std, k=15)

print("***before kmeans clustering***")
# Analyze clusters formed by kmeans clustering
kmeans = KMeans(n_clusters=15, max_iter=300).fit(d_w2v_kmeans_vect_std)
analyzeClusters(d_labels=kmeans.labels_, k=15, algo='kmeans')

print("***before heir clustering***")
# Analyze clusters formed by hierarchical clustering
analyzeClusters(d_labels=hierarchichal.labels_, k=2, algo='hierarchical')

print("***before 2nd heir clustering***")
# Analyze clusters formed by hierarchical clustering
print(bold + "%%% Hierarchical Clustering with 5 Clusters %%%" + end)
analyzeClusters(d_labels=hierarchichal_test.labels_, k=5, algo='hierarchical')

print("***ended heir clustering***")

dbscan = DBSCAN(eps=kneeValue, min_samples=200).fit(d_w2v_kmeans_vect_std)
analyzeClusters(d_labels=dbscan.labels_, k=len(set(dbscan.labels_)), algo='dbscan')
print("***after dbscan clustering***")

# To try out other Eps values
# by rule of thumb, min_samples should be 2*dimensionality = 200
print(bold + "%%% DBSCAN Clustering with Eps = 1 %%%" + end)
dbscan = DBSCAN(eps=1, min_samples=200).fit(d_w2v_kmeans_vect_std)
print(len(set(dbscan.labels_)))
analyzeClusters(d_labels=dbscan.labels_, k=len(set(dbscan.labels_)), algo='dbscan')

print("***after 2nd dbscan clustering***")

print(bold + "%%% DBSCAN Clustering with Eps = 50 %%%" + end)
dbscan = DBSCAN(eps=10, min_samples=200).fit(d_w2v_kmeans_vect_std)
print(len(set(dbscan.labels_)))
analyzeClusters(d_labels=dbscan.labels_, k=len(set(dbscan.labels_)), algo='dbscan')

print("***after 3rd dbscan clustering***")
```

```
***before kmeans clustering***


<class 'collections.Counter'>
dict_items([(5, 72), (1, 47), (2, 103), (4, 34), (11, 123), (14, 61), (0, 95), (1
cluster size = 15

*** CLUSTERS FORMED BY kmeans ALGORITHM is as follows: ***
CLUSTER = 1

The Review Text in Cluster 0 is as follows:
71765     This is my new favorite flavor of Lundberg Rice Chips. They are so tast
318265                                          I am a big fan of scharffen b
256827    Our labrador has an iron constitution. She has eaten a broad variety of
```

```
37071
346142    SF Bay Coffee's Fog Chaser was a nice "bold" surprise.  I am a big fan
Name: Text, dtype: object
CLUSTER = 2

The Review Text in Cluster 1 is as follows:
23280     I think this is probably as good as it gets for sugar free chocolate sy
245004    Never mind that this dog food is kind of gross looking - my dogs just l
110069
353742    This appeared to be the purest coconut cream I could find that was not
132203
Name: Text, dtype: object
CLUSTER = 3

The Review Text in Cluster 2 is as follows:
171368    I love these cornflakes.  I can't believe anyone would say they taste l
311440    I was very cautious and cut off a small corner of the pepper and set it
77439     Started buying this product 4 months ago after the whole "Canidae" issu
306512    The water is great but the price has me at a loss. I have been paying a
242252                                                                      Rece
Name: Text, dtype: object
CLUSTER = 4

The Review Text in Cluster 3 is as follows:
240280    I ordered Dark Sumatra Gayoland coffee bean from Coffee Bean Direct via
49234     I've always loved Thai yellow curries, but never had much luck trying t
10071     Although I grew up in NYC, I had not tried a Boylan's until recently wh
59253                                                                        I'
5172      i bought a whole bunch a syrups from this company because i love fruit
Name: Text, dtype: object
CLUSTER = 5

The Review Text in Cluster 4 is as follows:
63408
225583    Sorry to be so negative, but when something is SO bad on the market, yo
269634
146019    This poop scooper is overpriced and too small. The slits to filter the
195303
Name: Text, dtype: object
CLUSTER = 6

The Review Text in Cluster 5 is as follows:
```

CHAPTER – 5

OUTPUT SCREENSHOTS

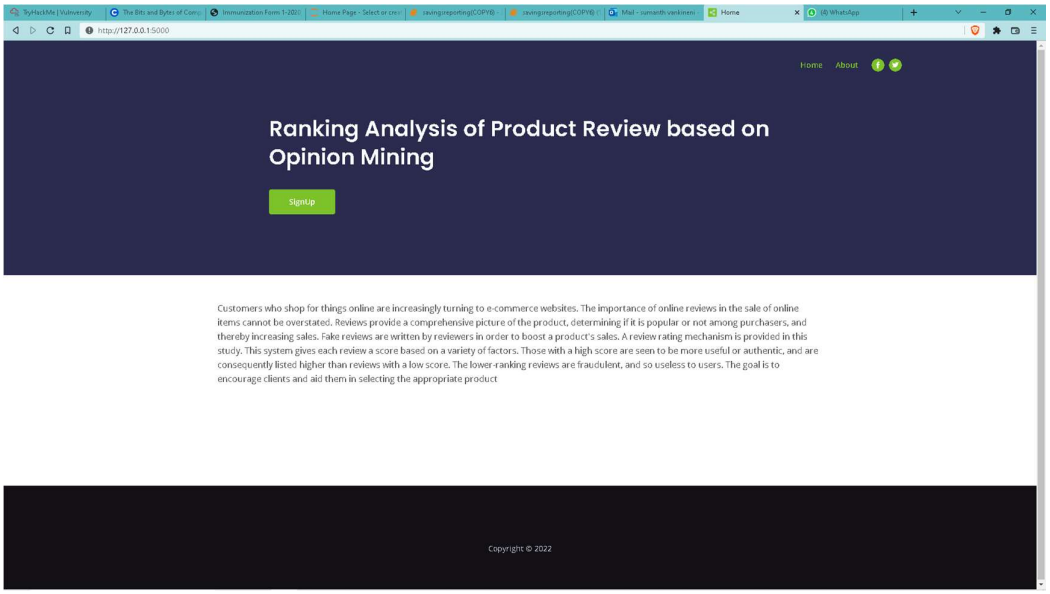

Fig 5.1: Home Page

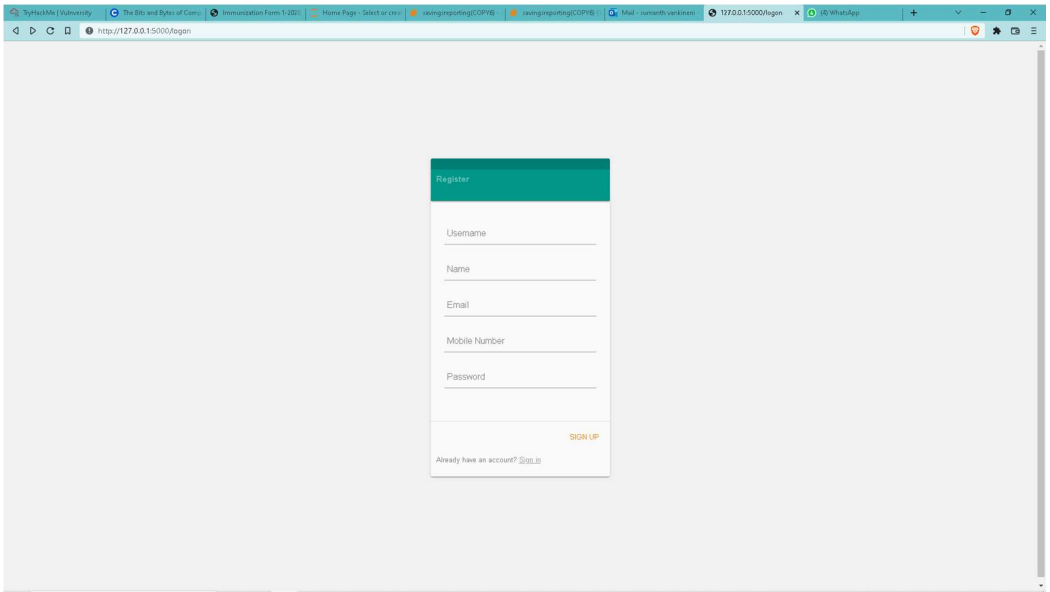

Fig 5.2: Sign up Page

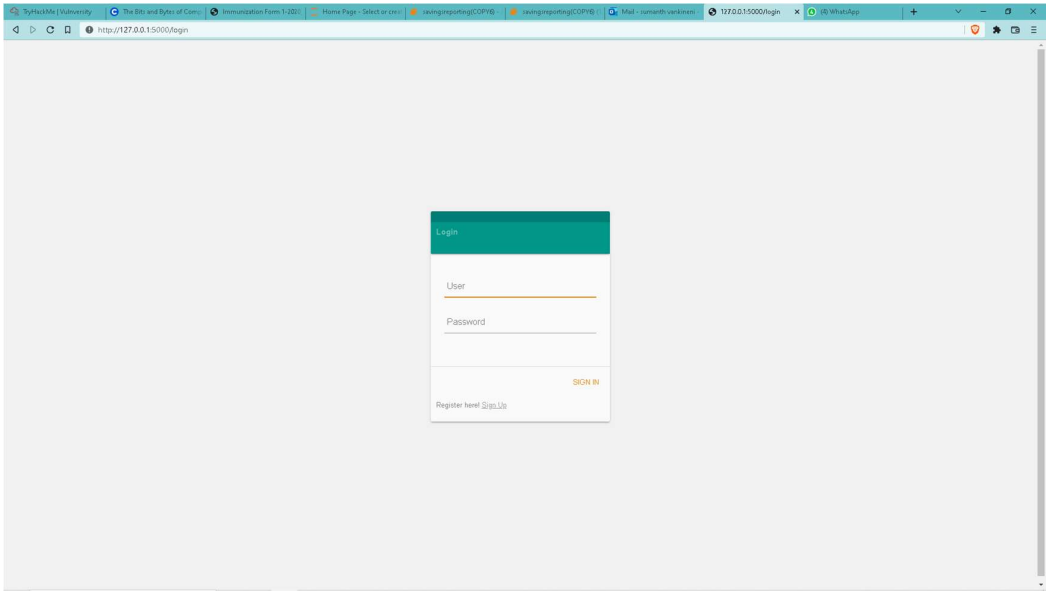

Fig 5.2: Login Page

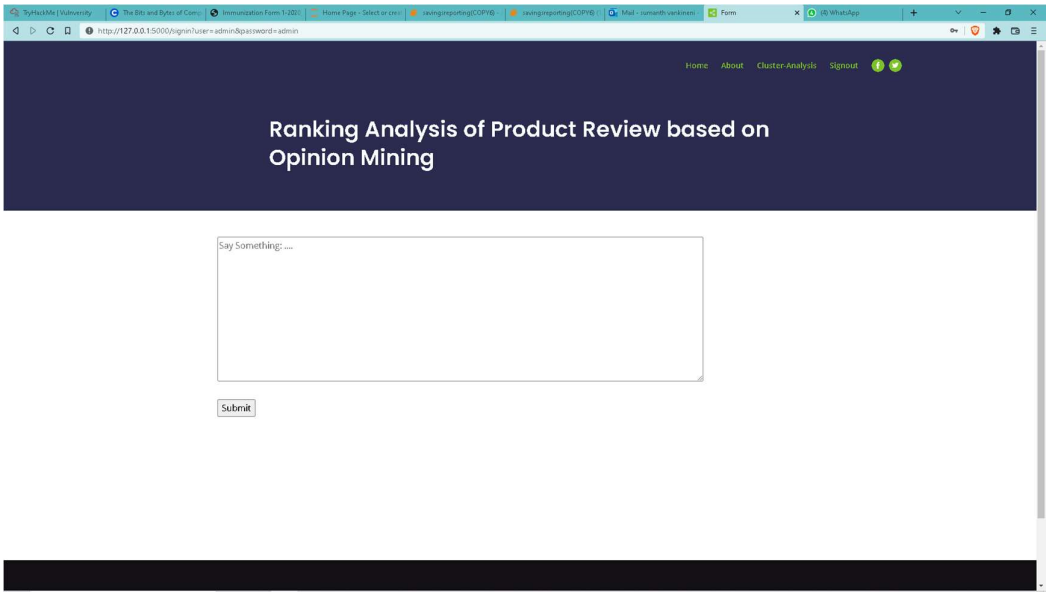

Fig 5.4: Form Page

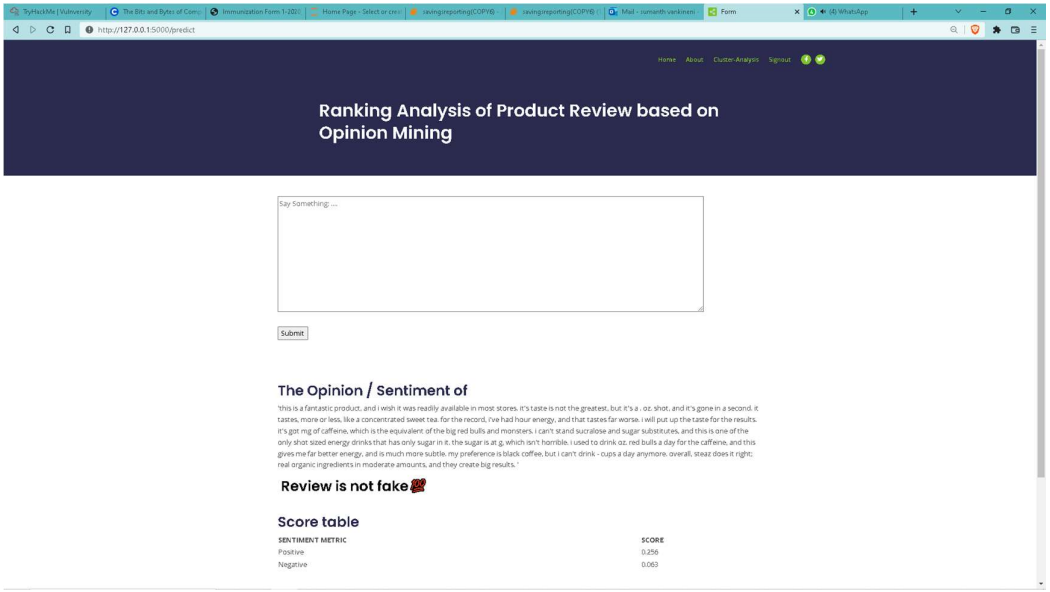

Fig 5.5: Form Output Page

CHAPTER - 6

CONCLUSION

We have made built a Fake review detector and given the ranking analysis for the product reviews. It is very crucial for an organization to know the feedback from its authentic clients about their products. Every establishment should know what do the customers find as the drawbacks of their product and should eliminate and further should enhance the features which will promote the product. The reviews of products on the e-commerce sites are a key basis for end users to select the right brand which suits their needs. The report portrays the process of ranking the online reviews to improve the give the customers or people a better deciding factor while selecting products. Our proposed system first removes fake reviews and then performs opinion mining on only genuine reviews to rate the products. Since we plan on using the 3 variations of the K-means, the precision of the result will be increased, and analysis will be more accurate. Since only genuine reviews are selected, our proposed system is expected to generate more dependable results for the customer to refer to and decide whether to buy a product or not.

CHAPTER – 7

FUTURE SCOPE

These days a few people are using acronyms and some short words with spelling mistakes which aren't present in the dictionaries we used in sentiment analysis, so we have to work on such content of words to get the better accuracy with the changes in slangs of shorts-words. With our current implementation we've built a web application under which out code runs but later on we can directly partner with the e-commerce websites to avoid the hassle of opening another websites to get the analysis of the reviews.

CHAPTER - 8

REFERENCES

[1] L. Xiao, F. P. Guo, and Q. B. Lu, "Mobile personalized service recommender model based on sentiment analysis and privacy concern, " Mobile Information Systems, vol. 2018, Article ID 8071251, 13 pages, 2018.

[2] A. Parashar and E. Gupta, "ANN based ranking algorithm for products on E-commerce website," in 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), pp. 362–366, Chennai, India, February 2017.

[3] BrightLocal, 2016. Local consumer review survey accessed from www.brightlocal.com/learn/localconsumer-review-survey/ Accessed on 22nd December 2016

[4] http://sersc.org/journals/index.php/IJAST/article/view/30626

[5]https://projectchampionz.com.ng/2018/11/14/e-commerce-product-rating-based-on-customer-review-mining/

[6] A. S. H. Basari, B. Hussin, I. G. P. Ananta, and J. Zeniarja, "Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization," Procedia Engineering, vol. 53, pp. 453–462, 2013.

[7] M. Malika, S. Habiba, and P. Agarwal, "A novel approach to web-based review analysis using opinion mining, " Procedia Computer Science, vol. 132, pp. 1202 –1209, 2018.

[8] V. Singh and S. K. Dubey, "Opinion mining and analysis: a literature review," in 2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), pp. 232–239, Noida, India, September 2014.

[9] https://flask.palletsprojects.com/en/2.1.x/

[10] A. S. H. Basari, B. Hussin, I. G. P. Ananta, and J. Zeniarja, "Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization," Procedia Engineering, vol. 53, pp. 453–462, 2013.

[12] N. Claypo and S. Jaiyen, "Opinion mining for thai restaurant reviews using K-means clustering and MRF feature selection," in 2015 7th International Conference on Knowledge and Smart Technology (KST), pp. 105–108, Chonburi, Thailand, January 2015.

PLAGIARISM REPORT

Batch-B8

ORIGINALITY REPORT

16%	2%	11%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Naveed Hussain, Hamid Turab Mirza, Ibrar Hussain, Faiza Iqbal, Imran Memon. "Spam Review Detection using the Linguistic and Spammer Behavioral Methods", IEEE Access, 2020 Publication	4%
2	Rami Mohawesh, Shuxiang Xu, Son N. Tran, Robert Ollington, Matthew Springer, Yaser Jararweh, Sumbal Maqsood. "Fake Reviews Detection: A survey", IEEE Access, 2021 Publication	3%
3	Md. Iqbal Hossain, Maqsudur Rahman, Tofael Ahmed, A. Z. M. Touhidul Islam. "Forecast the Rating of Online Products from Customer Text Review based on Machine Learning Algorithms", 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), 2021 Publication	3%

4	Submitted to National Institute of Technology, Silchar Student Paper	2%
5	Submitted to Gitam University Student Paper	1%
6	Submitted to VNR Vignana Jyothi Institute of Engineering and Technology Student Paper	1%
7	Submitted to Eiffel Corporation Student Paper	1%
8	Submitted to Higher Education Commission Pakistan Student Paper	<1%
9	Submitted to Kuala Lumpur Infrastructure University College Student Paper	<1%
10	Submitted to KDU College Sdn Bhd Student Paper	<1%
11	Palaiyah Solainayagi, Ramalingam Ponnusamy. "To Improve Feature Extraction and Opinion Classification Issues in Customer Product Reviews Utilizing an Efficient Feature Extraction and Classification (EFEC) Algorithm", Indonesian Journal of Electrical Engineering and Computer Science, 2018 Publication	<1%

12	coek.info Internet Source	<1%
13	Saleh Nagi Alsubari, Sachin N. Deshmukh, Ahmed Abdullah Alqarni, Nizar Alsharif et al. "Data Analytics for the Identification of Fake Reviews Using Supervised Learning", Computers, Materials & Continua, 2022 Publication	<1%
14	Naveed Hussain, Hamid Turab Mirza, Ibrar Hussain, Faiza Iqbal, Imran Memon. "Spam Review Detection Using the Linguistic and Spammer Behavioral Methods", IEEE Access, 2020 Publication	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		

SHOW AND TELL