

E-COMMERCE PRODUCT RATING BASED ON CUSTOMER REVIEW MINING

A Project Work
submitted in partial fulfillment of the
requirements for the degree of

Bachelor of Technology
in
Computer Science and Engineering

Vivek Kumar
1513101719

Submitted By
Vedika Arya
1513101676

Rajat Chaudhary
1513101468

Under the supervision of
Dr. Satyajee Srivastava
Associate Professor



SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA – 201306
MAY 2019

DECLARATION

Project Title: **E-Commerce Product Rating Based on Customer Review Mining**

Degree for which the project work is submitted: **Bachelor of Technology in Computer Science and Engineering**

I declare that the presented project represents largely my own ideas and work in my own words. Where others ideas or words have been included, I have adequately cited and listed in the reference materials. The report has been prepared without resorting to plagiarism. I have adhered to all principles of academic honesty and integrity. No falsified or fabricated data have been presented in the report. I understand that any violation of the above will cause for disciplinary action by the Institute, including revoking the conferred degree, if conferred, and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken.

Vivek Kumar

Enrollment No. 1513101719

Vedika Arya

Enrollment No. 1513101676

Rajat Chaudhary

Enrollment No. 1513101468

Date 8 may 2019

CERTIFICATE

It is certified that the work contained in this project entitled “**E-Commerce Product Rating Based on Customer Review Mining**” submitted by **Vivek Kumar (Enrollment No. 1513101719)**, **Vedika Arya (Enrollment No. 1513101676)**, **Rajat Chaudhary (Enrollment No. 1513101468)** for the degree of Bachelor of Technology in Computer Science and Engineering is absolutely based on his/her own work carried out under my supervision and this project work has not been submitted elsewhere for any degree.

Dr. Satyajee Srivastava

Associate Professor

School of Computing Science and Engineering

Galgotias University

Greater Noida, UP, India

Date:08 May 2019

Countersigned by

Prof. (Dr.) Sanjeev Kumar Pippal

Professor and Associate Dean

School of Computing Science and Engineering

Galgotias University

Greater Noida, UP, India

Abstract

E-commerce, is a type of online business that is mainly used for trading products and services through a website or portal that is available online or on internet. All the information regarding products and services are mentioned and described in detail such that user is not in any type of confusion with the product description and is able to understand the product. After selecting the products, the user may want that product to purchase and want it delivered to address of his own choice and pay for that product that he regards as the safest way either by COD or Online payment mode.

The Flourishment of these service may be seen as the market cap for E-commerce platform stand at 39 Billion dollars and is continuously rising. Though there are thousands of E-commerce website that are present and active but among those there are certain website that clearly has the dominance over another website. The difference that these site offers is the usability of the interface, detailing about the product, quality of the products, delivery of original product and secure payment methods. To better know about a product, we generally use the website given description and further more we look for the Reviews of the product and then we look at the text comment given by the user. We create a general perception about the product and we purchase them. But sometime the user just doesn't want to read the whole review rather just want the final outcome of the user comment as if it is Good or bad product. Also, sometime user can not just read all the user reviews as there might be many numbers of reviews available. Same is applicable on the case of vendors they will also not able to read all the reviews as there might be many numbers of product. Hence the product text reviews generally go in vain and can't be used efficiently, so in this project we attempt to text mine that text data and derive some useful data from this that might be useful for the customers as well as the vendor. [1]

In our agile world, we've learned that products are best built by prototyping early, soliciting feedback frequently, and continuing to iterate and improve. But for many product teams, soliciting frequent feedback can be the trickiest part. How do you narrow down which customer segment to ask? How do you sort through and weigh all their feedback? This is exactly where sentiment analysis can change the game. Whether by analyzing surveys,

customer support interactions, or social media, machine learning enables you to assess huge amounts of product feedback at once.

- Analyse large quantities of product feedback surveys
- Analyse all social media and online mentions about a product
- Filter comments by aspect and by sentiment, in order to see what to tweak and what to keep.
- Automatically route relevant comments to product teams.

Sentiment analysis or opinion mining is one of the major tasks of NLP (Natural Language Processing). Sentiment analysis has gained much attention in recent years. In this paper, we aim to tackle the problem of sentiment polarity categorization, which is one of the fundamental problems of sentiment analysis. A general process for sentiment polarity categorization is proposed with detailed process descriptions. Data used in this study are online product reviews collected from Amazon.com. Experiments for both sentence-level categorization and review-level categorization are performed with promising outcomes. At last, we also give insight into our future work on sentiment analysis.

Text mining and sentiment analysis have received huge attention recently, especially because of the availability of vast data in form of text available on social media, e-commerce websites, blogs and other similar sources. This data is usually unstructured and contains noise, therefore the task of gaining information is complex and expensive. There is a growing need for developing different methodologies and models for efficiently processing the texts and extracting apt information. One way to extract information is text mining and sentiment analysis, that include: data acquisition, data pre-processing and normalization, feature extraction and representation, labelling, and finally the application of various Natural Language Processing (NLP) and machine learning algorithms.

Acknowledgement

It is our privilege to express our sincerest regards to our project coordinator, Dr. Satyajee Srivastava for their valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project. We deeply express our sincere thanks to our Head of Department Dr Prof. Sanjeev Kumar Pippal for encouraging and allowing us to present the project on the topic **“E-Commerce Product Rating Based on Customer Review Mining”** at our department premises for the partial fulfilment of the requirements leading to the award of B-Tech degree. We take this opportunity to thank all our lecturers who have directly or indirectly helped our project. We pay our respects and love to our parents and all other family members and friends for their love and encouragement throughout our career. Last but not the least we express our thanks to our friends for their cooperation and support.

Table of Contents

1. Chapter 1: Introduction
 - (i) Overall Description
 - (ii) Purpose
 - (iii) Motivations and Scope
2. Chapter 2: Literature Survey
3. Chapter 3: Proposed Model
4. Chapter 4: Implementation
5. Chapter 5: Class Diagrams
6. Chapter 6: Results and Discussions
7. Chapter 7: Conclusions and Future Works
8. References

List of Figures

Figure No.	Description	Page No.
1.1	Demo Rating Figure	4
3.1	Proposed System Architecture	15
3.2	Start the project code	17
3.3	Home page	18
3.4	Admin panel	19
3.5	Feedback Demonstration	20
3.6	Dummy Wordset	21
4.1	The NLTK process	26
5.1	Block Diagram	32
5.2	Use Case Diagram	32
5.3	UML Diagram	33
5.4	Activity Diagram	35
5.5	Database Design	35
5.6	State Design	36
6.1	Registration page	37
6.2	Review page	38
6.3	Admin controls	39
6.4	Final Feedback Classified	39

CHAPTER 1

Introduction

(i) Overall Description

Internet has become another path for giving opinions the products and services. Many of the E-commerce sites containing such view are astronomically vast and it is promptly incrementing. The buyer reviews in web sites are truly useful for product recommendation in which fulfilled buyers tell other persons how much they like an originality of product. It begins to be the most credible forms of advertising because persons who do not understand to obtain privately by Recommending something put their good name on the line every time they make a proposal. Therefore, the computation method of sentiment and opinion has been observed as a challenging area of research that can benefit to different purposes. Product aspect ranking composed of three principal tasks: Identification of product aspect, classification based on sentiment and Product aspect ranking.[2]

Use of web and e-shopping web sites is developing very fast. Many products are available online. Most of the e-shopping sites inspire shopper to address their reviews about products to express their ideas on many aspects of the products. This gives rise to immense collection of feedbacks on web. These reviews contain rich and beneficial knowledge and have become a main resource for both buyers and firms. Buyers usually look for quality report from online reviews before purchasing a product and firms can use these reviews as feedback for better product development, buyer relationship management and for the development of new marketing approach.

Sentiment analysis, also known as opinion mining, in essence, is the process of quantifying the emotional value in a series of words or text, to gain an understanding of the attitudes, opinions and emotions expressed. Sentiment analysis can be applied to various sectors such as ecommerce, banking, mining social media websites like Facebook, Twitter and so on.

Using sentiment analysis and text mining, organizations can gain consumer insight from the response about their products and services. This can be further used to study customers' satisfaction with the services and in case of complaints and issues, finding the possible reasons for that. One of the applications of sentiment analysis is recommendation systems, for instance YouTube recommends on the basis of consumers likes, dislikes and comments provided by the user. Extensively study various text mining and sentiment analysis techniques applied to different areas in multi lingual format and from different resources.

A sentiment analysis and text mining

framework typically includes following subtasks: acquiring text data, data cleaning and pre-processing, data normalization, conversion of text to machine readable vectors, features selection, and finally applying NLP and machine learning algorithms. For instance, consumer review mining and application to tourism industry are the current successful

applications. Topic modelling is successfully combined with sentiment priors to generate topics and sentiment classes simultaneously. Emoji and emoticon sentiments are included in many of the studies to improve accuracy of results and so on.

Types of e-Commerce

- Business-to-Business (B2B): Electronic exchanges of products and ventures between companies.
- Business-to-Consumer (B2C): Electronic exchanges of products and ventures between companies and consumers.
- Consumer-to-Consumer (C2C): Electronic exchanges of products and ventures between consumers, generally through an outsider.
- Consumer-to-Business (C2B): Electronic exchanges of merchandise and ventures where people offer items or services to companies.

- Business-to-Administration (B2A): Electronic exchanges of products and ventures between companies and open organizations.
- Consumer-to-Administration (C2A): Electronic exchanges of products and ventures between people and open organizations.

Advantages of e-Commerce

- Global market reach
- Global choice for consumers
- Short item/service appropriation chain
- Lesser expenses and estimating

E-commerce website or portal aims to provide the most detailed description of the product. Even nowadays the e-commerce website is going towards giving the touch feel of the products mainly cloths and jewelry. But before that we have some of the things of which we can make a good use of the already available resources as the Text data of reviews. We make the Text data mine through the concepts of data mining. The data is mined through the basic process of data mining cycle. We make the Text mine by realizing the Keywords and classify them according to the Good, bad and neutral reviews. [3]

We make each classification category count and give that in respect with the product such that user can finally know about the general perception about the product. Along with these vendors can also know the mixed reviews of the product.

However, those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. For example, instead of sharing topic-related opinions, online spammers post spam on forums. Some spam are meaningless at all, while others have irrelevant opinions also known as fake opinions [10-12]. The second flaw is that ground truth of such online data is not always available. A ground truth is more

like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral. The Stanford Sentiment 140 Tweet Corpus [13] is one of the datasets that has ground truth and is also public available. The corpus contains 1.6 million machine-tagged Twitter messages. Each message is tagged based on the emoticons (☺as positive, ☹as negative) discovered inside the message.

Star Level	General Meaning
★	I hate it.
★★	I don't like it.
★★★	It's okay.
★★★★	I like it.
★★★★★	I love it.

Figure 1.1. **Demo Rating Figure**

(ii) Purpose

Showing product ratings when a customer hovers over an image provides crucial information instantly. They can see how many people purchased and reviewed the product, as well as the overall sentiment of the customers who bought it. By giving the right information at the right time, the product's rating will either confirm a customer's interest or highlight an item they may have otherwise overlooked.

Showing product ratings when a customer hovers over an image provides crucial information instantly. They can see how many people purchased and reviewed the product, as well as the overall sentiment of the customers who bought it. By giving the right information at the right time, the product's rating will either confirm a customer's interest or highlight an item they may have otherwise overlooked.

Most often, product ratings are displayed on a brand's site, but smart businesses are also using them to build social proof across their marketing. Whether showing product ratings in search results through Rich Snippets or Google Product Listing Ads or using them in ads on social, there are a variety of ways to boost the value of product ratings for your brand.

The purpose of syntactic analysis is to determine the structure of the input text. This structure consists of a hierarchy of *phrases*, the smallest of which are the *basic symbols* and the largest of which is the *sentence*. The structure can be described by a tree with one node for each phrase. Basic symbols are represented by values stored at the nodes. The root of the tree represents the sentence.

Here are key benefits to displaying product ratings with our customer reviews:

1. Increase on-site conversion

More stars really do equal more sales. The data shows that the higher rated the product, the higher the likelihood that a customer will buy. Out of all purchases, those with an average product rating of 5 stars make up 54% of the orders, while products with an average rating of 4 stars make up 40% of orders. This means that 94% of all purchases are made for products with an average rating of 4 stars and above.

2. Capture high-intent shoppers

Many shoppers online know exactly what they want to purchase. They create searches that are specific about the item that they have in mind and, when presented their options on search, need certain information to make their choice about where to purchase from. By displaying product ratings and reviews alongside your search results, you can give shoppers the social proof they need to trust your brand and make a purchase.

3. Showcase social proof at key conversion points

Displaying product ratings across your site can have a huge impact on sales at the exact moment that a shopper needs that extra push to make a purchase. From your homepage to category and product pages, there are tons of places where product ratings can make or break a customer's decision to stay on site and ultimately buy.

In addition to these ratings we make use of text that more briefly describe the product, and a user can easily know about the pros and cons of the product who have already used the product.

the user just doesn't want to read the whole review rather just want the final outcome of the user comment as if it is Good or bad product. Also, sometime user can not just read all the user reviews as there might be many numbers of reviews available. Same is applicable on the case of vendors they will also not able to read all the reviews as there might be many numbers of product. Hence the product text reviews generally go in vain and can't be used efficiently, so in this project we attempt to text mine that text data and derive some useful data from this that might be useful for the customers as well as the vendor.

Every word of a sentence has its syntactic role that defines how the word is used. The syntactic roles are also known as the parts of speech. There are 8 parts of speech in English: the verb, the noun, the pronoun, the adjective, the adverb, the preposition, the conjunction, and the interjection. In natural language processing, part-of-speech (POS) taggers [29-31] have been developed to classify words based on their parts of speech. For sentiment analysis, a POS tagger is very useful because of the following two reasons: 1) Words like nouns and pronouns usually do not contain any sentiment. It is able to filter out such words with the help of a POS tagger; 2) A POS tagger can also be used to distinguish words that can be used in different parts of speech. For instance, as a verb, "enhanced" may conduct

different amount of sentiment as being of an adjective. The POS tagger used for this research is a max-entropy POS tagger developed for the Penn Treebank Project [31]. The tagger is able to provide 46 different tags indicating that it can identify more detailed syntactic roles than only 8.

(iii) Motivation and Scope

Motivation for the Text mining of the data comes from the fact that almost every product has reviews from its final consumer, that is stored in the database, which certain user read and some just make use of the average star rating of the product. So, to use that important block of data we text mine that data. Sometime user can not just read all the user reviews as there might be many numbers of reviews available. Same is applicable on the case of vendors they will also not able to read all the reviews as there might be many numbers of product. Hence the product text reviews generally go in vain and can't be used efficiently, so in this project we attempt to text mine that text data and derive some useful data from this that might be useful for the customers as well as the vendor. [4]

Consumers are becoming more review-savvy, preferring businesses that receive high volumes of high-scoring reviews on a regular basis.

Consumers are also changing their habits regarding what they do after reading a positive online review. People are now less likely to go on to visit a business' website straight away — but they're far more likely to get directly in touch over the phone over via email, or by visiting the business. [5]

One very important task for the Ecommerce store is to maintain its reputation in the online market. Quite naturally, it takes a lot of effort to gain that reputation but not much to lose it: Product Reviews are the best ways to maintain their winning streak. Product Reviews and feedbacks have changed the game for online market since internet has become a very household thing. The Product Reviews are the factors which either make or break the relationship of the consumer with the store – they help build loyalty and trust and lets the potential consumer know the product much more clearly and the aspects that differentiate it from the rest of the products elsewhere. An Ecommerce store which has a good compilation of consumer reviews for the products shows the wide consumer base it incapacitates. The store, thus, anticipates positive reviews to gain more customers in the future.

However, the biggest advantage Product Reviews provide for an Ecommerce store is the increase in its Sales or the increase in the number of purchases from the consumers. Online reviews are so important to businesses because they ultimately increase the sales by giving the consumers the information, they need to make the decision to purchase the product. People are always more likely to buy the products which has already been recommended by other users. When customer reviews about the product has been added to the Ecommerce store, 42% of the site administrators have reported increases in average order value, versus only 6% that report a decrease with inclusion of reviews.

Chapter 2

Literature Survey

The opinion mining has become one of popular research area. The challenge is in process of opinion mining or sentiment analysis that is unstructured and noisy data on website. A part of opinion mining refers using of natural language processing (NLP) by proposed different method of dictionary for sentiment analysis of text as corpus, lexicon and specific language dictionary. They tried to extract word from sentences for removal stop word or unnecessary word automatically. In addition, various dictionaries are solved by machine learning methods V.B. Raut, D.D. Londhe, "Survey on opinion mining and summarization of user review on web", *International Journal of Computer Science and Information Technology*, vol. 5, no. 2, pp. 1026-1030, 2014. which try to rank scoring of various dictionaries. For example, the paper in J. Fiaidhi, O. Mohammed, S. Mohammed, S. Fong, T.H Kim, "Opinion Mining over twitterspace: Classifying tweets programmatically using the R approach", *Proceeding of the 7th International Conference on Digital Information Management*, pp. 313-319, 2012. used fuzzy logic algorithm to collect the ranking of different dictionary into rule for classify the opinion.

After word segmentation process is removal stop words by dictionary checking. The research in

A.H. Al-hamaami, S. H. Shahrour, "Development of an opinion blog mining system", *Proceeding of the 4th International Conference on Advanced Computer Science Application and Technologies*, pp. 74-79, 2015. focuses on the calculating polarity of words to trend in positive or negative in a cluster of interest's customers that are extracted from texts and compared the word occurrence of whole sentence. If the word extractions have weight from dictionary of emotional words, it is calculated to answer the comment as positive or negative.

However, the customer review has different behaviour with the product. The proposed classifier model is presented using association rule in N. Kumari, S. N. Singh, "Sentiment

analysis on E-commerce application by using opinion mining", *Proceeding of the 6th International Conference-Cloud System and Big Data Engineering (Confluence)*, pp. 320-325, 2016. From these researches are used classifier models that are the same objective to classified opinion. Our approach is different from them, this paper uses the advantage of classifier model to generate the rating value from classifier which is not only shown classify opinion as positive and negative and also factors analysis to impact the customer who posted or commented to positive and negative.

The product aspect ranking is to predict the ratings on individual aspects. Wang developed a latent aspect rating analysis model, which aims to infer reviewer's latent opinions on each aspect and the relative emphasis on different aspects. This work concentrates on aspect-level opinion estimation and reviewer rating behaviour analysis, rather than on aspect ranking. Snyder and Barzilay formulated a multiple aspect ranking problem. Justin Martineau and Tim Finin present Delta TFIDF, a general-purpose technique to efficiently weight word scores. This technique calculates the value of aspect in document but does not consider the frequency of words associated with aspect with it. In contrast, unsupervised approaches automatically extract product aspects from customer reviews without using training examples. Hu and Liu's works focuses on association rule mining based on the Apriori algorithm to mine frequent item sets as explicit product aspects. In association rule mining, the algorithm does not consider the position of the words in the sentence. In order to remove incorrect frequent aspects, two types of pruning criteria were used: compactness and redundancy pruning. The technique is efficient which does not require the use of training examples or predefined sets of domain-independent extraction patterns.

Previous studies have shown that the product rating grabs attention of the user and a product which is rated 4 stars plus is 75% more likely to be purchased. Along with these the user also, are very attentive and savvy about the product reviews which are given by the user who have finally used the product.

we are interested to answer the following research questions:

Q1(a). Is it possible to learn new positive words using a basic/extended list of positive words?

(b). Is it possible to learn new negative words using a basic/extended list of negative words?

Q2. Can we discover special groups of words that are associated with the list of positive and negative words?

Q3. What is the distribution of the sentences (neutral, positive, and negative)?

Q4. What are the scores of the top words associated with the positive and negative words and what can we learn from these scores?[7]

To answer these questions, we worked with two seed lists containing sentiment words in Hebrew. These lists were manually generated by us. Each one of these lists contains both positive and negative words. The first list is relatively a small list, containing only 45 words (22 positive and 23 negative). The second list, the largest list, contains 168 words (85 positive and 83 negative). Our motivation to perform experiments with two seed sentiment lists (basic and extended) is to check whether there is any difference in the results obtained by these two lists. An

example for a question is whether the use of the extended seed sentiment list can discover more positive and negative sentiment words than the use of the basic seed sentiment list.[8] More than 90% of Amazon buyers fail to leave feedback or review products they purchase. However, when they have a bad experience, customers will leave a negative review without any prompting. These reviews can harm your business and you should develop a strategy to help drown out the negatives and keep your seller account healthy. There are some great economical tools that can help you boost your feedback and review numbers. If you're not already using one of these tools, then we highly recommend the below customer feedback and review management software for Amazon sellers.[x] Of all these plugins are not yet integrated into the seller login system they have to first of all collect all the dataset and feed that dataset to tools like AMZfinder , Feedback express which are third party companies which provides the feedback services at cost of 1800 or 2000 per month. If we don't want

to opt for this option then we have to manually read all the reviews and aggregate them according to human potentials.[9]

The product selling companies often either don't use the product rating system like Myntra, Jabong or Paytm which led them to a loss of around 3000 Crores of Ruppes in the financial year of 2018. Even the PayTm has to shut its portal close for the e- commerce product site. [10]

Even the Amazon don't have its own module for analytics- they too use the third party tools like Google Analytics, Mixpanel or Visible.[11]

In [2] Kouloumpis et. al demonstrated the usefulness of linguistic features and existing lexical resources used in micro-blogging to detect the sentiments of twitter messages. From this paper the researchers concluded that microblogging features were more useful as compared to POS

(Part-of-Speech) features and features from existing sentiment lexicon. They also concluded that if they include micro-blogging features then the training data will be of less benefit. [3] consists of a new method formed by combination of rule based classification, supervised learning and machine learning which showed the improvement in micro and macro averaged F1. To get better effect, Prabowo et. al considered semi-automatic approach. From this paper they concluded that hybrid classification was better than the classification by any individual classifier. They also concluded that reduction of rules will produce less effect on F1. From [4], Mudinas et. al concluded that concept level sentiment analysis system (psenti) was better as compared to

pure lexicon based system and pure learning based system due to more precision in polarity classification and well structured, readable results. On experimenting, they confirmed that hybrid approach was better than sentistrength. From their paper, they concluded that psenti system obtained high precision than pure lexicon based system but near to pure learning based system. It also gave well structured, readable results and more resistance to writing style of text.

They also concluded that psenti system works better than sentistrength. In short, the proposed hybrid approach was capable in combining a carefully designed lexicon and a powerful supervised learning algorithm. In [5], Lin et. al identified subjective information using

automated tools and a novel probabilistic modelling framework called joint sentiment/topic model, which detects sentiment and topic together from text. They concluded that the proposed JST model was fully apart as compared to other machine learning approaches. Basically, they proposed this model on movie dataset to classify the sentiment polarity and to improve the sentiment classification accuracy. In this paper, a joint sentiment/topic (JST) model had been proposed with the help of which document level sentiment classification could be depicted and mixture of topics from text simultaneously could be extracted. On the other hand, existing approaches in sentiment classification were based on supervised learning, while the proposed JST model was fully unsupervised, hence comes up with more flexibility and could be easily combined with other applications. When the results were compared with existing supervised approaches then they found out that this model gave a competitive performance in document level sentiment classification. On other side it also had one limitation of classifying each document as a bag of words which results in ignoring the word ordering for example predicting sentiment of “not good movie” being positive and of “not bad movie” being negative. This leads them to include bigrams and trigrams in their model. Another step which would be included in future was to detect the polarity of text at several granularity levels, e.g. detecting sentiment labels for more fine-grained topics. Model performance on datasets from different domains were also evaluated. In their paper, Li et. al [6] studied online forums hotspot and forecast using sentiment analysis and text mining approaches. First of all, to inspect the sentiment polarity for each piece of text, an algorithm was created. Afterwards to develop unsupervised text mining approach the algorithm was joined with k-means clustering and support vector machine (SVM).

Chapter 3

Proposed Model

In our proposed work we develop a process of product aspect ranking consisting of three main Steps:

- (a) gathering the text data of customers
- (b) Opinion mining- mapping the keywords of review text with the predefined data set classifying them into positive and negative feedback
OR sentiment classification on aspects
- (c) generating overall review of the product

Given the buyer reviews of a product, first identify the aspects in the reviews and then analyse these reviews to find buyer opinions on the aspects via a sentiment classifier and finally rank the product based on importance of aspect by considering aspect frequency and buyers' ideas given to each aspect over their overall opinions. [6]

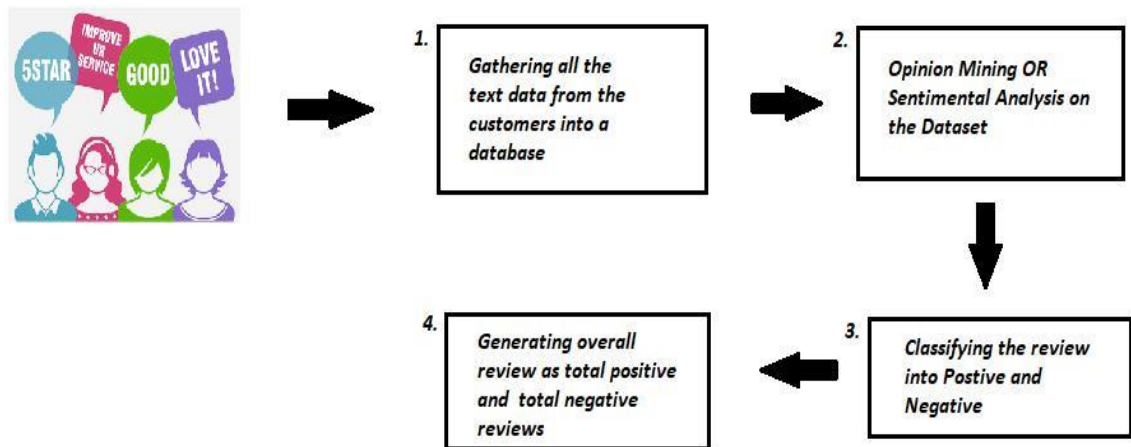


Figure.3.1 Proposed System Architecture

This project consists of mainly four modules which takes up the step of the proposed model accordingly as specified:

1. Admin module,
2. Seller module,
3. Customers module
4. Sentiment Classification module

1. admin module

Admin will create all types of product Categories. Besides the Admin will upload all the type of products based on categories respectively. In that we divide product into product aspects to store and retrieve in the server.

This module would enable the admin to have functionalities like

- a). users list
- b). add seller
- c). sellers list
- d). Feedback Analytics portal

This module can view as which product is selling more quickly and what are the reviews it is generating.

Steps to run the review system site on the local server of Django on a PC:

Step 1:

First of all, the product should be made and stored in a file from where we have to run our product.

For that we have to go inside the folder from CMD command and run the cmd code lines as shown:


```
Command Prompt - python manage.py runserver
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Vivek Kumar>cd desktop
C:\Users\Vivek Kumar\Desktop>cd python
C:\Users\Vivek Kumar\Desktop\python>productrating\Scripts\activate
(PRODUC~1) C:\Users\Vivek Kumar\Desktop\python>cd ecomproductrating
(PRODUC~1) C:\Users\Vivek Kumar\Desktop\python\ecomproductrating>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
January 23, 2019 - 21:34:47
Django version 2.1, using settings 'ecomproductrating.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[23/Jan/2019 21:35:52] "GET / HTTP/1.1" 200 20086
[23/Jan/2019 21:35:53] "GET /static/css/awemenu.css HTTP/1.1" 200 13890
[23/Jan/2019 21:35:53] "GET /static/css/awe-icon.css HTTP/1.1" 200 3342
[23/Jan/2019 21:35:53] "GET /static/css/owl.carousel.css HTTP/1.1" 200 3733
[23/Jan/2019 21:35:53] "GET /static/css/magnific-popup.css HTTP/1.1" 200 7805
[23/Jan/2019 21:35:53] "GET /static/css/awe-background.css HTTP/1.1" 200 1011
[23/Jan/2019 21:35:53] "GET /static/css/font-awesome.css HTTP/1.1" 200 28747
[23/Jan/2019 21:35:53] "GET /static/css/easyzoom.css HTTP/1.1" 200 984
[23/Jan/2019 21:35:53] "GET /static/css/bootstrap.css HTTP/1.1" 200 143867
[23/Jan/2019 21:35:53] "GET /static/css/star-rating-svg.css HTTP/1.1" 200 591
[23/Jan/2019 21:35:53] "GET /static/css/nanoscroll.css HTTP/1.1" 200 1366
```

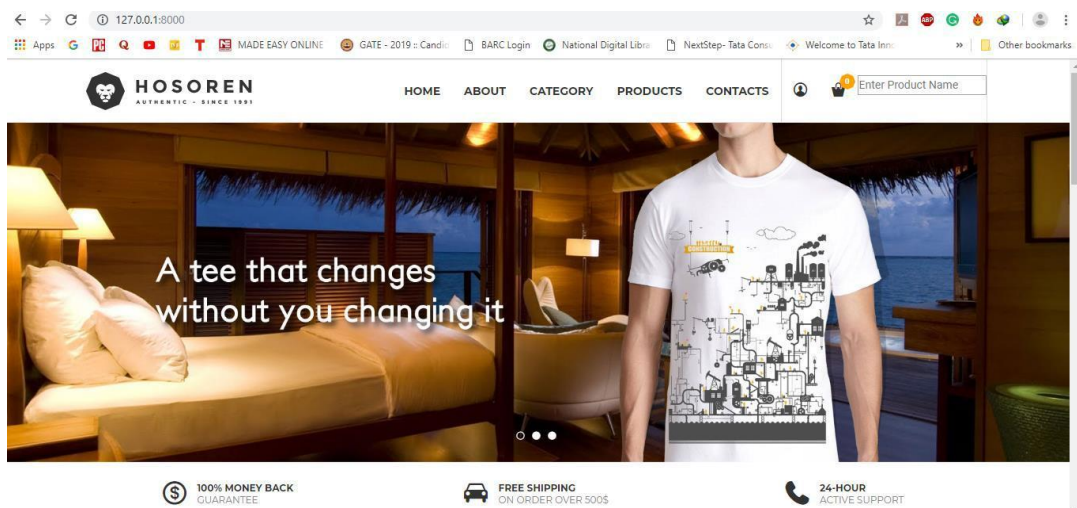
Figure 3.2: start the project code

After running this command, the CMD returns us the address of the local site which is generally:

<http://127.0.0.1:8000/>

Step 2:

Open this address in the google chrome tab bar and the home page occurs as follows:



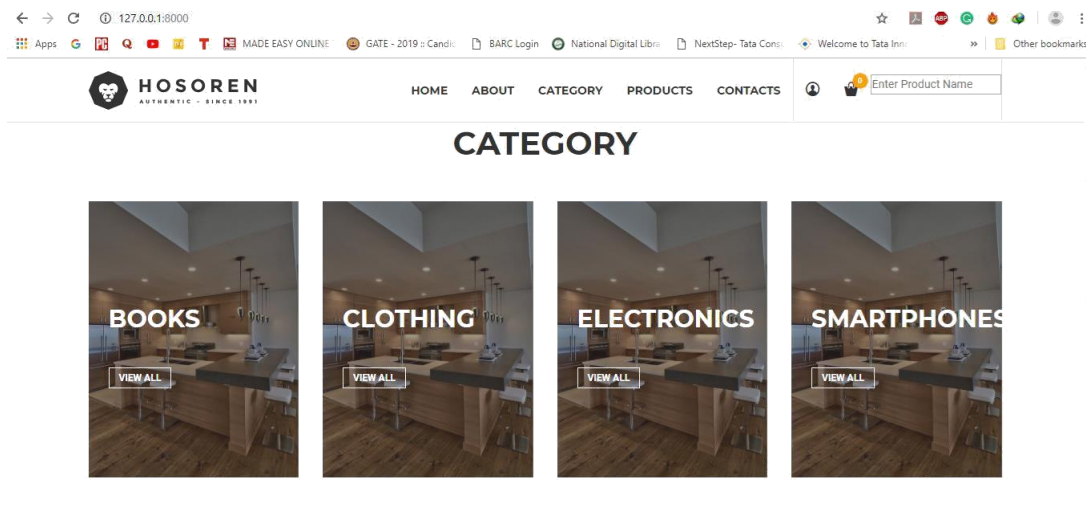
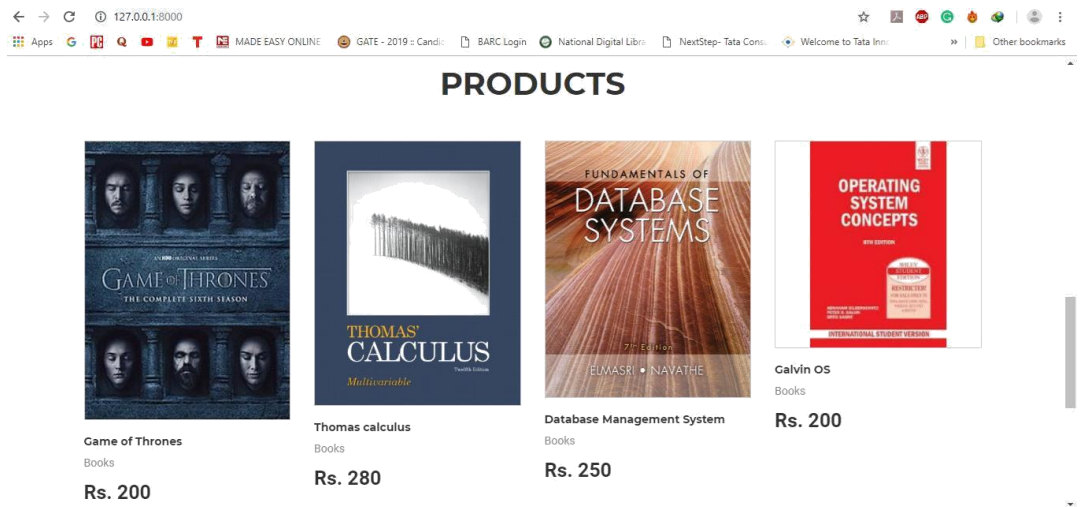


Figure 3.3: The Home Page

Here any user can register himself and purchase the product. After purchasing the product the user can review the product. The user can then submit the review which will be available to others users and furthermore that information will be available to the admin module or seller module where based on the product analysis words will be picked up from the review text and classified according to the site settings.

2. Seller/ Admin module

Seller would have this module where he can put all the products for sale. This module would have functionalities like

- a). Update profile
- b). Add product
- c). View product list
- d). Order details
- e). Change password

Step 4 that is generated output would be clearly visible to the vendor so that they can know about the main cons and pros of the product so that they can change or modify the product and can track the product progress in terms of sale.

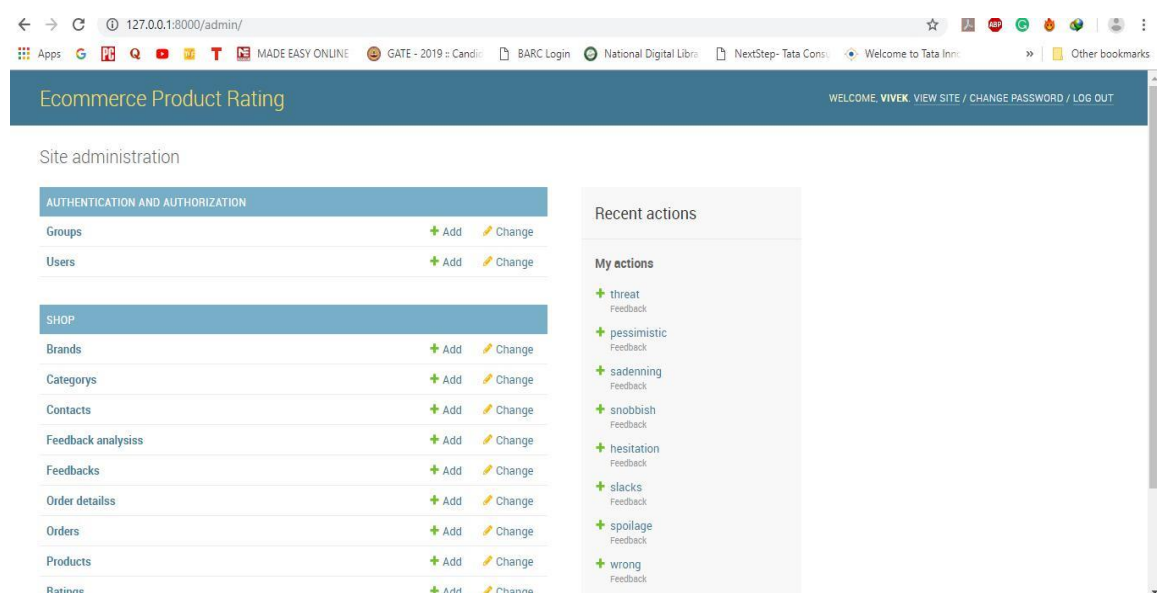


Figure.3.4: the admin module

3. Customers Module

This module belongs to the customers where they can purchase product and can do payment etc. and will give review of the product. This module have functionalities like

- a). update Profile
- b). view product
- c). add to cart
- d). payment
- e). Give Feedback

- f). Product rating
- g). Search Product
- h). Change password

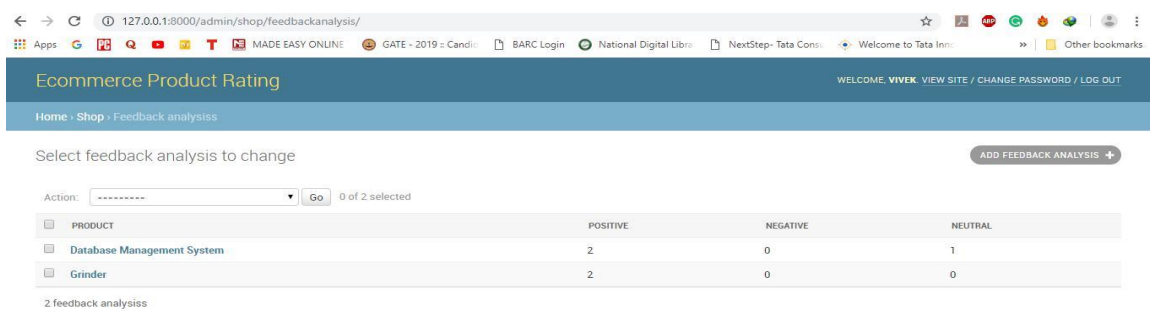
Step 1 is compiled in this module and gathers all the Text data from the users only after they purchase the product. It is because sometime the vendor or seller may add fake reviews of the product so we add this simple modification as only customers who have purchased the product can only write review about the product. This simple step provides security against the fake reviews.

4. Sentiment Classification module

The module takes review of various users, based on the review; module will specify whether the products and services provided by the E-Commerce enterprise are good, bad or worst. This uses text-based analysis along with positivity or negativity weight in database and then based on these sentiment keywords mined in customer review, product and services provided by the E-commerce enterprise is ranked. Text data mining used in this system to derive high quality information from text. High-quality information is typically derived through the devising of patterns and trends. This module scans a set of text written in natural language.

Step 2 and Step 3 belongs to this module. This module takes up the NLTK library in the python. It does the work of sentiment analysis.

Here the product analysis section will present us the review words in a concise form of classification as : Positive, Neutral and Negative section as follows:



2 feedback analysis

PRODUCT	POSITIVE	NEGATIVE	NEUTRAL
Database Management System	2	0	1
Grinder	2	0	0

Figure 3.5: Feedback demonstration

The Sentimental classification will perform its operations on a pre-defined set of words (which can be added and deleted) which are selected from a set of Researched words. The words are as follows:

1. Accomplishment	15. Perfect	29. Genius	43. Creative
2. Exceptional	16. Essential	30. Recommend	44. Renowned
3. Outstanding	17. Divine	31. Unbeatable	45. Skillful
4. Absolutely	18. Peak	32. Friendly	46. Blessed
5. Excellent	19. Exclusive	33. Amazing	47. Thriving
6. Authentic	20. Expert	34. Wholesome	48. Terrific
7. Helpful	21. Exquisite	35. Charming	49. Wow
8. Enjoy	22. Fantastic	36. Dynamic	50. Worthwhile
9. Prime	23. Positively	37. Fascinating	51. Sensational
10. Choice	24. Generous	38. Clarity	52. Promptly
11. Superb	25. Ideal	39. Impressive	53. Brilliant
12. Purify	26. Fully	40. Interesting	54. Favorite
13. Splendid	27. Inspiring	41. Memorable	55. Spectacular
14. Impeccable	28. Motivational	42. Exciting	56. Marvelous

Figure 3.6: **Dummy Word set**

Chapter 4

Implementation

The Implementation part begins with the install of all the software required.

Requirement

Python 3.5 or 3.6

Database- Mysql or SqlLight

Even if the database connectivity is not present by the third party application, the Django framework provides us with the default database as DbSqlite which store all the information on the device itself.

1. Installation

1. Install Python
2. Install Pip
3. Install VirtualEnv

2. Setup Django

1. Create virtual environments
eg: `virtualenvirontment productrating`
2. Activate Virtual Environments
eg: `productrating/Scripts/activate`

3. Install Project First Time

1. `virtualenvirontment productrating`
2. `productrating\Scripts\activate`
3. `cd ecomproductrating`
4. `pip instal -r requirements.txt`
5. `python manage.py migrate`

6. `python manage.py createsuperuser`
7. `python manage.py runserver`

4. Run Product Second Times

1. `productrating\Scripts\activate`
2. `cd ecomproductrating`
3. `python manage.py runserver`

The Front End

This constitutes of web pages that simulate an E-Commerce website is formed by the HTML, CSS and Python. All these are managed by the Django Framework.

Django is a high-level Python Web framework that encourages rapid development and clean pragmatic design. A Web framework is a set of components that provide a standard way to develop websites fast and easily. Django's primary goal is to ease the creation of complex database-driven websites. Some well-known sites that use Django include PBS, Instagram, Disqus, Washington Times, Bitbucket and Mozilla.

Django's template language is designed to feel comfortable and easy-to-learn to those used to working with HTML, like designers and front-end developers. But it is also flexible and highly extensible, allowing developers to augment the template language as needed.

The Back-end

This uses NLTK library for sentimental analysis. The NLTK module is a massive tool kit, aimed at helping you with the entire Natural Language Processing (NLP) methodology. NLTK will aid you with everything from splitting sentences from paragraphs, splitting up words, recognizing the part of speech of those words.

NLTK is a powerful Python package that provides a set of diverse natural languages algorithms. It is free, opensource, easy to use, large community, and well documented. NLTK consists of the most common algorithms such as tokenizing, part-of-speech tagging, stemming, sentiment analysis, topic segmentation, and named entity recognition. NLTK helps the computer to analysis, preprocess, and understand the written text.

There are many methods and algorithms to implement sentiment analysis systems, which can be classified as:

- **Rule-based** systems that perform sentiment analysis based on a set of manually crafted rules.
- **Automatic** systems that rely on machine learning techniques to learn from data.
- **Hybrid** systems that combine both rule based and automatic approaches.

1. Rule-based Approaches

Usually, rule-based approaches define a set of rules in some kind of scripting language that identify subjectivity, polarity, or the subject of an opinion.

The rules may use a variety of inputs, such as the following:

- Classic NLP techniques like stemming, tokenization, part of speech tagging and parsing.
- Other resources, such as lexicons (i.e. lists of words and expressions).

A basic example of a rule-based implementation would be the following:

1. Define two lists of polarized words (e.g. negative words such as *bad*, *worst*, *ugly*, etc and positive words such as *good*, *best*, *beautiful*, etc).
2. Given a text:
 - i. Count the number of positive words that appear in the text.
 - ii. Count the number of negative words that appear in the text.
3. If the number of positive word appearances is greater than the number of negative word appearances return a positive sentiment, conversely, return a negative sentiment. Otherwise, return neutral.

This system is very naïve since it doesn't take into account how words are combined in a sequence. A more advanced processing can be made, but these systems get very complex quickly. They can be very hard to maintain as new rules may be needed to add support for new expressions and vocabulary. Besides, adding new rules may have undesired outcomes

as a result of the interaction with previous rules. As a result, these systems require important investments in manually tuning and maintaining the rules.

2. Automatic Approaches

Automatic methods, contrary to rule-based systems, don't rely on manually crafted rules, but on machine learning techniques. The sentiment analysis task is usually modelled as a classification problem where a classifier is fed with a text and returns the corresponding category, e.g. positive, negative, or neutral (in case polarity analysis is being performed).

The bag-of-words model is a way of representing text data when modelling text with machine learning algorithms.

The bag-of-words model is simple to understand and implement and has seen great success in problems such as language modelling and document classification.

In this tutorial, you will discover the bag-of-words model for feature extraction in natural language processing.

- What the bag-of-words model is and why it is needed to represent text.
- How to develop a bag-of-words model for a collection of documents.
- How to use different techniques to prepare a vocabulary and score words.

Bag-of-Words Model:

The bag-of-words model is a way of representing text data when modelling text with machine learning algorithms. The bag-of-words model is simple to understand and implement and has seen great success in problems such as language modelling and document classification. In this, we will discover the bag-of-words model for feature extraction in natural language processing.

- What the bag-of-words model is and why it is needed to represent text.
- How to develop a bag-of-words model for a collection of documents.

- How to use different techniques to prepare a vocabulary and score words.

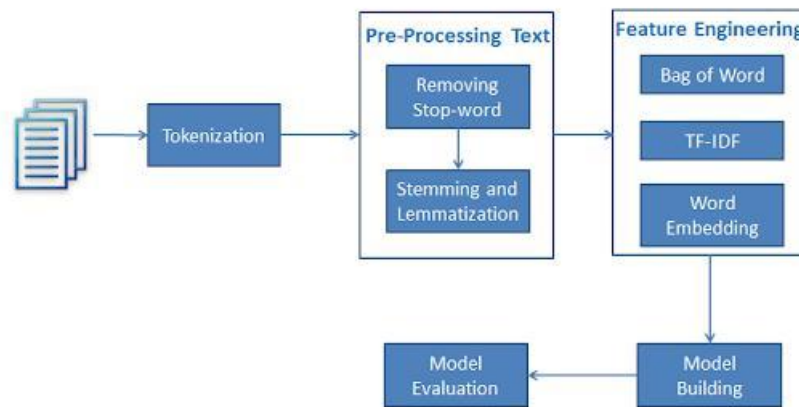


Figure 4.1: NLTK process

What is a Bag-of-Words?

A bag-of-words model, or BoW for short, is a way of extracting features from text for use in modelling, such as with machine learning algorithms. The approach is very simple and flexible, and can be used in a myriad of ways for extracting features from documents. A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:

1. A vocabulary of known words.
2. A measure of the presence of known words.

It is called a “*bag*” of words, because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.

A very common feature extraction procedures for sentences and documents is the bag-of-words approach (BOW). In this approach, we look at the histogram of the words within the text, i.e. considering each word count as a feature.

The intuition is that documents are similar if they have similar content. Further, that from the content alone we can learn something about the meaning of the document. The bag-of-words can be as simple or complex as you like. The complexity comes both in deciding

how to design the vocabulary of known words (or tokens) and how to score the presence of known words.

Example of the Bag-of-Words Model

Step 1: Collect Data

Below is a snippet of the first few lines of text from the book “A Tale of Two Cities” by Charles Dickens, taken from Project Gutenberg.

It was the best of times,
it was the worst of times,
it was the age of wisdom,
it was the age of foolishness,

For this small example, let’s treat each line as a separate “document” and the 4 lines as our entire corpus of documents.

Step 2: Design the Vocabulary

Now we can make a list of all of the words in our model vocabulary. The unique words here (ignoring case and punctuation) are:

- “it”
- “was”
- “the”
- “best”
- “of”
- “times”
- “worst”
- “age”
- “wisdom”
- “foolishness”

That is a vocabulary of 10 words from a corpus containing 24 words.

Step 3: Create Document Vectors

The next step is to score the words in each document.

The objective is to turn each document of free text into a vector that we can use as input or output for a machine learning model.

Because we know the vocabulary has 10 words, we can use a fixed-length document representation of 10, with one position in the vector to score each word.

The simplest scoring method is to mark the presence of words as a boolean value, 0 for absent, 1 for present.

Using the arbitrary ordering of words listed above in our vocabulary, we can step through the first document (*"It was the best of times"*) and convert it into a binary vector.

The scoring of the document would look as follows:

- "it" = 1
- "was" = 1
- "the" = 1
- "best" = 1
- "of" = 1
- "times" = 1
- "worst" = 0
- "age" = 0
- "wisdom" = 0
- "foolishness" = 0

As a binary vector, this would look as follows:

1[1, 1, 1, 1, 1, 1, 0, 0, 0, 0]

The other three documents would look as follows:

1."it was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]

2. "it was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0] 3. "it

was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

All ordering of the words is nominally discarded and we have a consistent way of extracting features from any document in our corpus, ready for use in modeling.

New documents that overlap with the vocabulary of known words, but may contain words outside of the vocabulary, can still be encoded, where only the occurrence of known words are scored and unknown words are ignored.

You can see how this might naturally scale to large vocabularies and larger documents.

Managing Vocabulary

As the vocabulary size increases, so does the vector representation of documents. In the previous example, the length of the document vector is equal to the number of known words.

You can imagine that for a very large corpus, such as thousands of books, that the length of the vector might be thousands or millions of positions. Further, each document may contain very few of the known words in the vocabulary. This results in a vector with lots of zero scores, called a sparse vector or sparse representation.

Sparse vectors require more memory and computational resources when modeling and the vast number of positions or dimensions can make the modeling process very challenging for traditional algorithms. As such, there is pressure to decrease the size of the vocabulary when using a bag-of-words model.

There are simple text cleaning techniques that can be used as a first step, such as:

- Ignoring case
- Ignoring punctuation
- Ignoring frequent words that don't contain much information, called stop words, like "a," "of," etc.
- Fixing misspelled words.
- Reducing words to their stem (e.g. "play" from "playing") using stemming algorithms.

A more sophisticated approach is to create a vocabulary of grouped words. This both changes the scope of the vocabulary and allows the bag-of-words to capture a little bit more meaning from the document.

In this approach, each word or token is called a “gram”. Creating a vocabulary of two-word pairs is, in turn, called a bigram model. Again, only the bigrams that appear in the corpus are modeled, not all possible bigrams.

An N-gram is an N-token sequence of words: a 2-gram (more commonly called a bigram) is a two-word sequence of words like “please turn”, “turn your”, or “your homework”, and a 3-gram (more commonly called a trigram) is a three-word sequence of words like “please turn your”, or “turn your homework”.

For example, the bigrams in the first line of text in the previous section: “It was the best of times” are as follows:

- “it was”
- “was the”
- “the best”
- “best of”
- “of times”

A vocabulary then tracks triplets of words is called a trigram model and the general approach is called the n-gram model, where n refers to the number of grouped words.

Often a simple bigram approach is better than a 1-gram bag-of-words model for tasks like documentation classification.

a bag-of-bigrams representation is much more powerful than bag-of-words, and in many cases proves very hard to beat.

Scoring Words

Once a vocabulary has been chosen, the occurrence of words in example documents needs to be scored. In the worked example, we have already seen one very simple approach to scoring: a binary scoring of the presence or absence of words.

Some additional simple scoring methods include:

- **Counts.** Count the number of times each word appears in a document.
- **Frequencies.** Calculate the frequency that each word appears in a document out of all the words in the document.

Word Hashing

You may remember from computer science that a hash function is a bit of math that maps data to a fixed size set of numbers. For example, we use them in hash tables when programming where perhaps names are converted to numbers for fast lookup.

We can use a hash representation of known words in our vocabulary. This addresses the problem of having a very large vocabulary for a large text corpus because we can choose the size of the hash space, which is in turn the size of the vector representation of the document. Words are hashed deterministically to the same integer index in the target hash space. A binary score or count can then be used to score the word. This is called the “*hash trick*” or “*feature hashing*”. The challenge is to choose a hash space to accommodate the chosen vocabulary size to minimize the probability of collisions and trade-off sparsity.

TF-IDF

A problem with scoring word frequency is that highly frequent words start to dominate in the document (e.g. larger score), but may not contain as much “informational content” to the model as rarer but perhaps domain specific words. One approach is to rescale the frequency of words by how often they appear in all documents, so that the scores for frequent words like “the” that are also frequent across all documents are penalized.

This approach to scoring is called Term Frequency – Inverse Document Frequency, or TF-IDF for short, where:

- **Term Frequency:** is a scoring of the frequency of the word in the current document.
- **Inverse Document Frequency:** is a scoring of how rare the word is across documents.

The scores are a weighting where not all words are equally as important or interesting. The scores have the effect of highlighting words that are distinct (contain useful information) in a given document. *Thus the idf of a rare term is high, whereas the idf of a frequent term is likely to be low.*

Chapter 5

Class Diagrams

Figure. 5.1 Block Diagram:

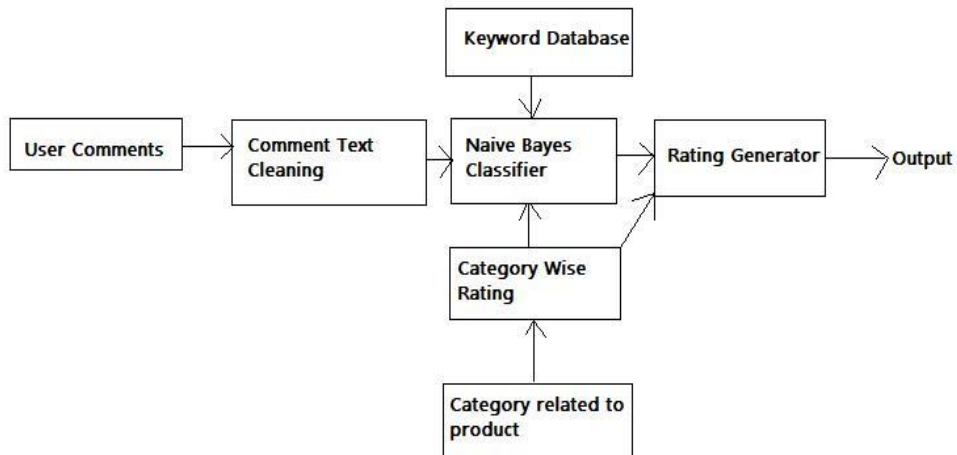


Figure. 5.2 Use Case Diagram:

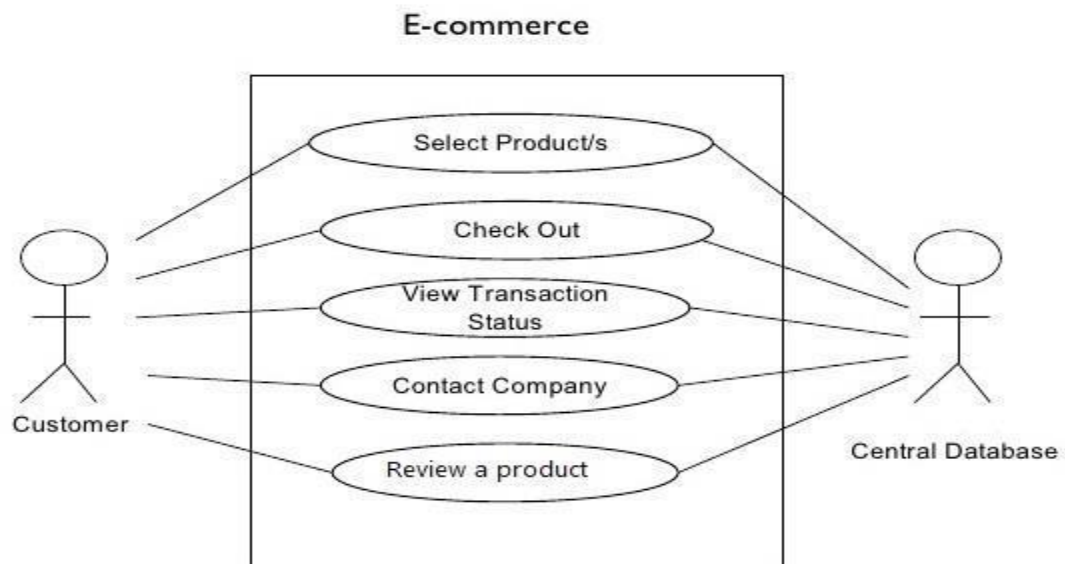
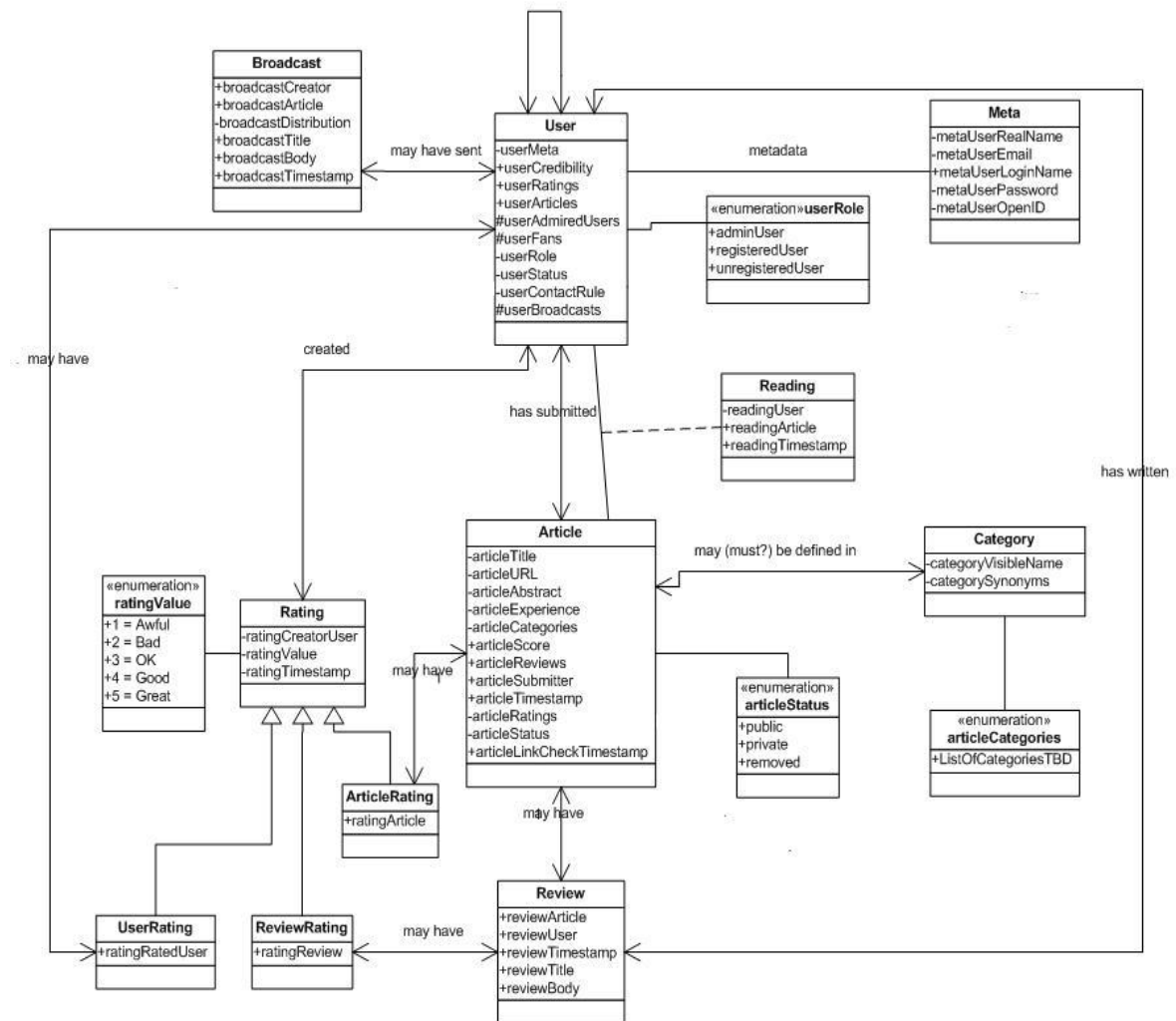


Figure. 5.3 UML Class Diagram:



Article:

articleScore. Each article will have a score that represents the collective perception of the quality of the article.

articleReviews. A list of all reviews / commentary on the article.

articleSubmitter. The user who submitted the article.

articleTimestamp. When the article was submitted.

articleRatings. [Private]. The individual ratings of the article, used to calculate the score.

articleStatus. [Private]. An article can be public, possibly private, or removed (no longer visible to users, but still in the database to prevent its re-appearance).

User

userRatings. All of the ratings submitted by the user.

userArticles. All of the articles submitted by the user.

userAdmiredUsers. Possibly have a list of users who you admire. Keeping this as a placeholder.

userFans. Possibly have a list of all users who admire you. Placeholder.

userRole. [Private?] A user can be an admin, can submit articles, and can review articles. Most likely, anyone who can review can submit. An admin can do both, but also has the ability to “clean house” in the system.

userContactRule. [Private?] A user may allow other users to contact her. The site may provide an abstraction/email interface that allows contact without revealing email addresses. And people may want to restrict who can contact them based on the fan-admired network, or allow any user to contact them. Another placeholder for future possibilities.

Rating

ratingArticle. The article being rated.

ratingUser. The user doing the rating.

ratingValue. The actual rating value assigned to the article by the user.

ratingTimestamp. When the rating was created.

Review

reviewArticle. The article being reviewed. reviewUser.

The user doing the reviewing. reviewTimestamp. When the review was created. reviewTitle. Title of the review

(e.g. “Best. Article. Ever.”)

reviewBody. The content of the review. Possibly “safe” html, analogous to comment field limitations on blogs.

Figure. 5.4 Activity Diagram:

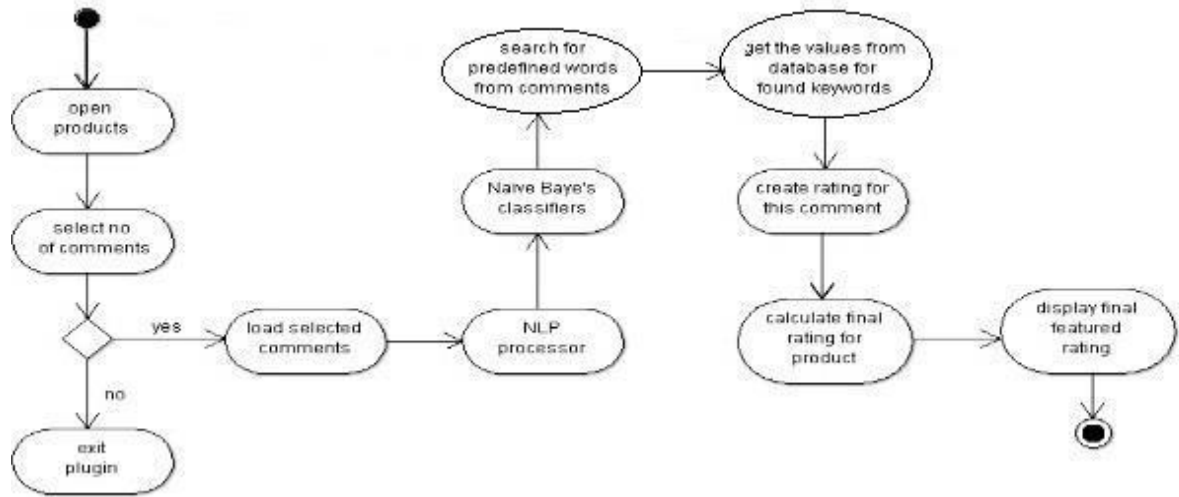


Figure. 5.5 Database Design:

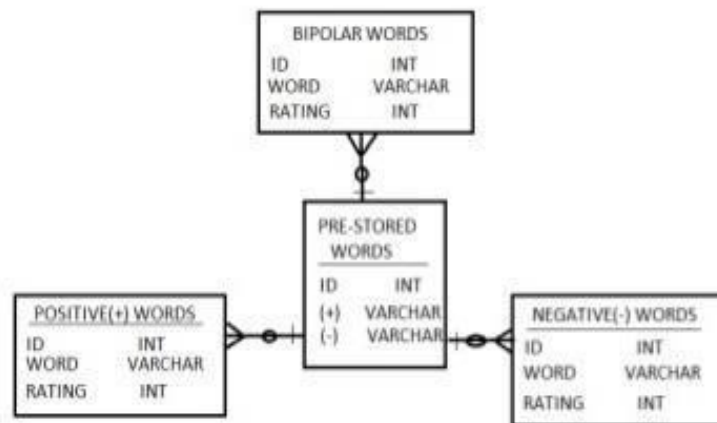
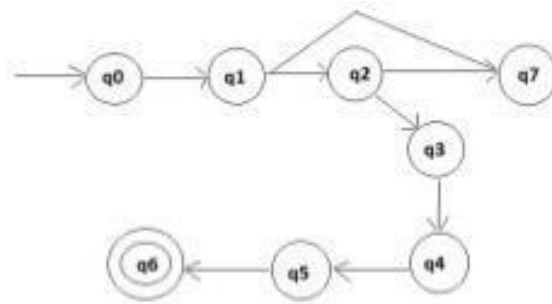


Figure 5.6 State Diagram:



q0= initial state(product page)
q1= plugin pop up(set comment limit)
q2= generate word pairs(noun&associated adjective)
q4= pre-stored words with associated rating
q5= rating calculated for each pair
q6= end state(final rating for the product)
q7= failure or if user quit

Chapter 6

Results and Discussions

The user registers itself on the website and buy the product otherwise he will not be able to give the feedback, so this constitutes for the fake review, so the site will be almost free of fake reviews. The user can register itself on the site as follows:

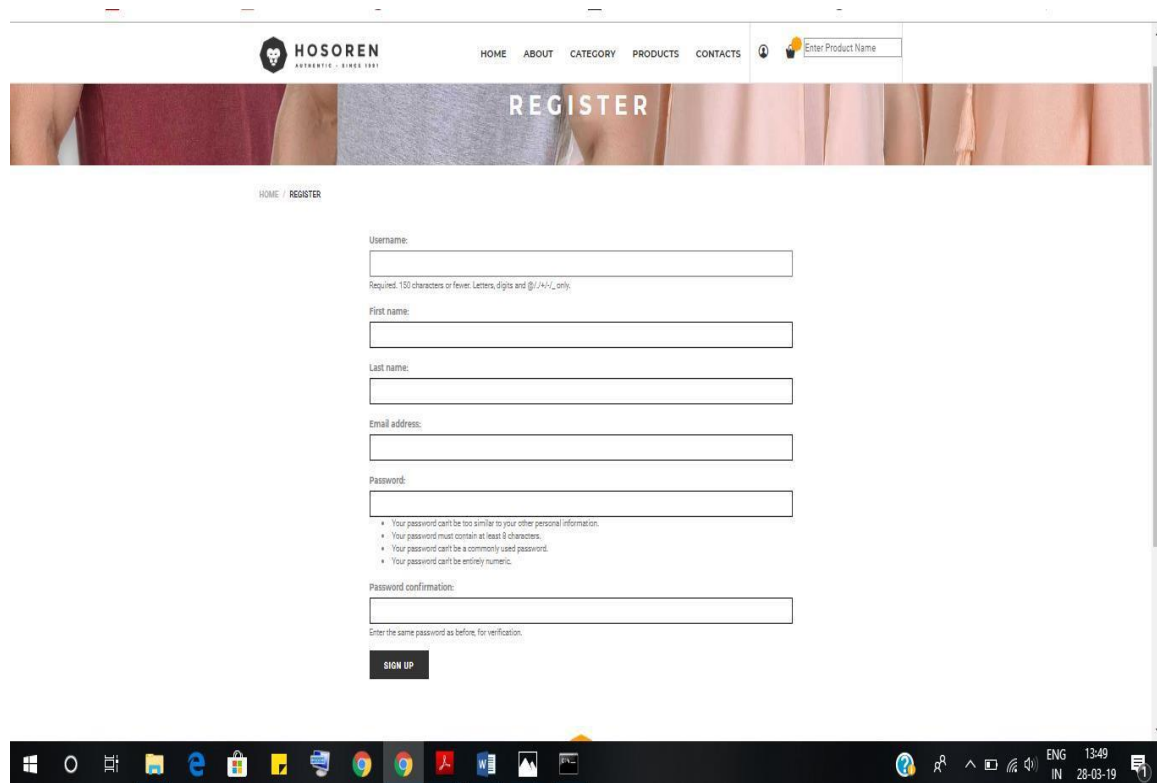
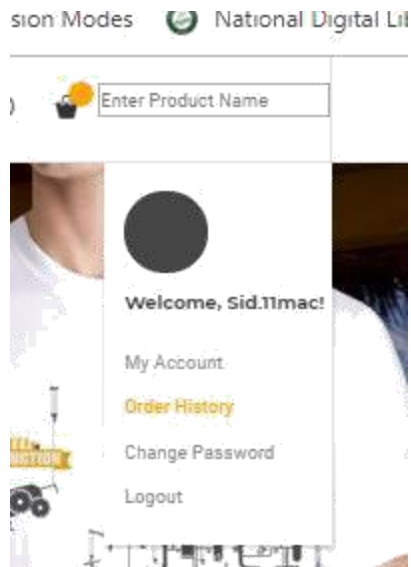


Figure 6.1 Registration page

Then the user can buy the product as desired and when the product is purchased as here we can just provide the dummy product purchase the user will be able to go on order history and see all the listed products. In this the list of product is displayed by the back end database DbSqlite. It fetches all the details as product details, timestamp etc.



After going to order history we can have all the product listed as follows:

HOME / ORDER HISTORY

INVOICE	NAME	EMAIL	ORDER DATE	AMOUNT	DISCOUNT	PAYMENT
INV0003	VIVEK KUMAR	SID.MAHAJAN4397@GMAIL.COM	MARCH 28, 2019, 7:54 A.M.	RS. 180	RS. 20	COD

After clicking to the desired product we can give a review:

Apps G Q Y T MADE EASY ONLINE... GATE - 2019 :: Cand... BARC Login Indian Statistical Ins... Admission Modes National Digital Lib... » Other bookmarks

HOSOREN
FOUNDED - 2015-2017

HOME ABOUT CATEGORY PRODUCTS CONTACTS

Enter Product Name

INVOICE	NAME	EMAIL	ORDER DATE	AMOUNT	DISCOUNT	PAYMENT
INV0003	VIVEK KUMAR	SID.MAHAJAN4397@GMAIL.COM	MARCH 28, 2019, 7:54 A.M.	RS. 180	RS. 20	COD

PRODUCT NAME	PRICE	DISCOUNT PRICE	DISCOUNT	QUANTITY
GAME OF THRONES	200	180	10	1

ADD A REVIEW

PRODUCT*
Select product for rate

TITLE*
Title

YOUR REVIEW*
Your review

YOUR RATING:
★★★★★

SUBMIT

Figure. 6.2 Review page

After submitting the review the if we go on the admin panel whose address is <http://127.0.0.1:8000/admin/>

We get several option as shown below:

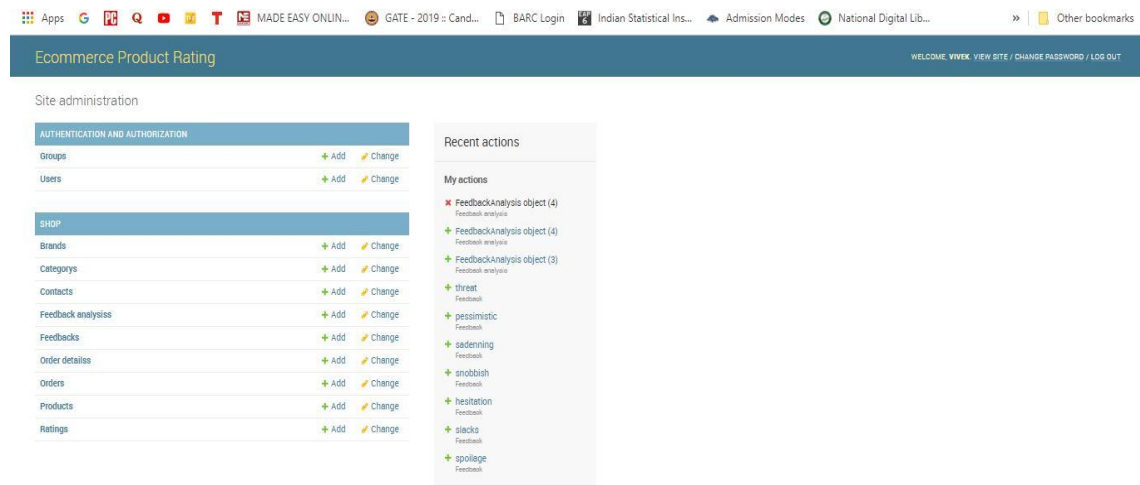


Figure 6.3. admin control

When we go to the Feedbacks we can see the product given ratings in the form of Positive, Negative and neutral as shown below. The result shown below is done by sentiment analysis module:

Ecommerce Product Rating

WELCOME VIVEK VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Shop > Feedback analysis

Select feedback analysis to change

ADD FEEDBACK ANALYSIS

Action: 0 of 4 selected

PRODUCT	POSITIVE	NEGATIVE	NEUTRAL
Game of Thrones	12	3	2
Thomas calculus	1	0	2
Database Management System	2	0	1
Grinder	2	0	0

4 feedback analysis

Figure 6.4. Feedback page

This gives the vendors a facility of not reading the whole review rather a just see with how much frequency users have given the review about the product about its features. This feature is not provided by any of the sites to their vendors. This analysis works on most of the products and improve the products accordingly.

1. The Problem with Text

A problem with modeling text is that it is messy, and techniques like machine learning algorithms prefer well defined fixed-length inputs and outputs. Machine learning algorithms cannot work with raw text directly; the text must be converted into numbers. Specifically, vectors of numbers. *In language processing, the vectors x are derived from textual data, in order to reflect various linguistic properties of the text.* This is called feature extraction or feature encoding.

Most of the work in sentiment analysis in recent years has been around developing more accurate sentiment classifiers by dealing with some of the main challenges and limitations in the field.

2. Subjectivity and Tone

The detection of subjective and objective texts is just as important as analyzing their tone. In fact, so called *objective* texts do not contain explicit sentiments. Say, for example, you intend to analyze the sentiment of the following two texts:

The package is nice.

The package is red.

Most people would say that sentiment is positive for the first one and neutral for the second one, right? All *predicates* (adjectives, verbs, and some nouns) should not be treated the same with respect to how they create sentiment. In the examples above, *nice* is more *subjective* than *red*.

3. Context and Polarity

All utterances are uttered at some point in time, in some place, by and to some people, you get the point. All utterances are uttered in context. Analyzing sentiment without context gets pretty difficult. However, machines cannot learn about contexts if they are not mentioned explicitly. One of the problems that arise from context is changes in polarity. Look at the following responses to a survey:

Everything of it.

Absolutely nothing!

Imagine the responses above come from answers to the question *What did you like about the event?* The first response would be positive and the second one would be negative, right? Now, imagine the responses come from answers to the question *What did you DISlike about the event?* The negative in the question will make sentiment analysis change altogether.

A good deal of preprocessing or postprocessing will be needed if we are to take into account at least part of the context in which texts were produced. However, how to preprocess or postprocess data in order to capture the bits of context that will help analyze sentiment is not straightforward.

4. Irony and Sarcasm

Differences between literal and intended meaning (i.e. *irony*) and the more insulting or ridiculizing version of irony (i.e. *sarcasm*) usually change positive sentiment into negative whereas negative or neutral sentiment might be changed to positive. However, detecting irony or sarcasm takes a good deal of analysis of the context in which the texts are produced and, therefore, are really difficult to detect automatically.

For example, look at some possible answers to the question *Have you had a nice customer experience with us?* below.

Yeah. Sure.

Not one, but many!

What sentiment would you assign to the responses above? Probably, you have listened to the first response so many times, you would have said negative, right? The problem is there is no textual cue that will make a machine learn that negative sentiment since most often, *yeah* and *sure* belong to positive or neutral texts.

How about the second response? In this context, sentiment is positive, but we're sure you can come up with many different contexts in which the same response can express negative sentiment.

5. Comparisons

How to treat comparisons in sentiment analysis is another challenge worth tackling.

Look at the texts below:

This product is second to none.

This is better than old tools.

This is better than nothing.

There are some comparisons like the first one above that do not need any contextual clues in order to be classified correctly.

The second and third texts are a little more difficult to classify, though. Would you classify them as *neutral* or *positive*? Probably, you are more likely to choose *positive* for the second one and *neutral* for the third, right? Once again, context can make a difference. For example, if the *old tools* the second text talks about were considered useless in context, then the second text turns out to be pretty similar to the third text. However, if no context is provided, these texts feel different.

6. Emojis

There are two types of emojis according to Guibon et al.. *Western emojis* (e.g. :D) are encoded in only one character or in a combination of a couple of them whereas *Eastern emojis* (e.g. ˊ \ _ (ヾ) _ / ˊ) are a longer combination of characters of a vertical nature. Particularly in tweets, emojis play a role in the sentiment of texts.

Sentiment analysis performed over tweets requires special attention to character-level as well as word-level. However, no matter how much attention you pay to each of them, a lot of preprocessing might be needed. For example, you might want to preprocess social media content and transform both Western and Eastern emojis into tokens and whitelist them (i.e. always take them as a feature for classification purposes) in order to help improve sentiment analysis performance.

7. Defining Neutral

Defining what we mean by *neutral* is another challenge to tackle in order to perform accurate sentiment analysis. As in all classification problems, defining your categories - and, in this case, the *neutral* tag- is one of the most important parts of the problem. What *you* mean by *neutral*, *positive*, or *negative* does matter when you train sentiment analysis models. Since tagging data requires that tagging criteria be consistent, a good definition of the problem is a must.

1. Objective texts. So called *objective* texts do not contain explicit sentiments, so you should include those texts into the neutral category.
2. Irrelevant information. If you haven't pre-processed your data to filter out irrelevant information, you can tag it neutral. However, be careful! Only do this if you know how this could affect overall performance. Sometimes, you will be adding noise to your classifier and performance could get worse.
3. Texts containing wishes. Some wishes like I wish the product had more integrations are generally neutral. However, those including comparisons, like I wish the product were better are pretty difficult to categorize.

It's a tremendously difficult task even for human beings. That said, sentiment analysis classifiers might not be as precise as other types of classifiers. Remember that inter-annotator agreement is pretty low and that machines learn from the data they are fed with (see above).

“Chances are that sentiment analysis predictions will be wrong from time to time, but by using sentiment analysis you will get the opportunity to get it right about 70-80% of the times you submit your texts for classification.”

Chapter 7

Conclusions and Future works

The bag-of-words model is very simple to understand and implement and offers a lot of flexibility for customization on your specific text data.

It has been used with great success on prediction problems like language modelling and documentation classification.

Nevertheless, it suffers from some shortcomings, such as:

- **Vocabulary:** The vocabulary requires careful design, most specifically in order to manage the size, which impacts the sparsity of the document representations.
- **Sparsity:** Sparse representations are harder to model both for computational reasons (space and time complexity) and also for information reasons, where the challenge is for the models to harness so little information in such a large representational space.
- **Meaning:** Discarding word order ignores the context, and in turn meaning of words in the document (semantics). Context and meaning can offer a lot to the model, that if modelled could tell the difference between the same words differently arranged (“this is interesting” vs “is this interesting”), synonyms (“old bike” vs “used bike”), and much more.

Sentiment analysis is useful to product analytics because it helps you do all of the following:

- Keep constant tabs on what people like and don’t like about your product.
- Zero in on which segments like which things, and how to appeal to those audiences.
- Empower your product development team with incredible insight into specifics of product performance.

Limitations:

1- dimensionality problem because the total dimension is the vocabulary size and it can easily over-fit your model.

2- Bag of words representation doesn't consider the semantic relation between words, it just focus on count of word and neglect the arrangement, n-grams and tagging in sentence.

Therefore, the productivity of this model limits to 60-70%. Since no model has yet achieved the efficiency even 80 %. This model serves as the most widely used that is able to parse the language but not syntactically. So further the work can be done to improve the model to syntactically accommodate the model.

References

- [1]. Anurag Manni, Naman Jaiswal- “Product Rating Based on Review Using Data Mining”-IJARIIT 2017
- [2].Ms.E.Aarthi,Ms.P.Yogalakshmi,Ms.P.Muthulakshmi-“E-Commerce Product Rating Based on Customer Review Mining”-IJAERD January 2018
- [3].Rutuja Tikait, Ranjana Badre and Mayura Kinikar, “Product aspect Ranking Techniques A Survey”, IJIRCCE, Nov 2014.
- [4]. L. Lin, J. Li, R. Zhang, W. Yu, C. Sun, "Opinion mining and sentiment analysis in social networks: A retweeting structure-aware approach", *Proceeding of the 7th International Confernece on Utility and Cloud Computing*, pp. 890-895, 2014.
- [5]. V.B. Raut, D.D. Londhe, "Survey on opinion mining and summarization of user review on web", *International Journal of Computer Science and Information Technology*, vol. 5, no. 2, pp. 1026-1030, 2014.
- [6]. N. Kumari, S. N. Singh, "Sentiment analysis on E-commerce application by Using opinion mining", *Proceeding of the 6th International Conference-Cloud System and Big Data Engineering (Confluence)*, pp. 320-325, 2016.
- [7] Positive and Negative Sentiment Words in a Blog Corpus Written in Hebrew Yaakov HaCohen-Kerner, 1*, Haim Badasha Dept. of Computer Science, Jerusalem College of Technology, 9116001 Jerusalem, Israel
- [8] Introduction Strategy for New Products with Positive and Negative Word-of-Mouth Article in Management Science 30(12):1389-1404 · December 1984ac
- [9] <https://www.amzfinder.com/blog/top-7-amazon-feedback-review-tools-help-optimize-listing/>
- [10].<https://www.businessinsider.in/I-keep-getting-the-same-phone-case-over->
- [11].https://www.researchgate.net/publication/227445457_Introduction_Strategy_for_New_Products_with_Positive_and_Negative_Word-of-Mouth

- [12] SAMIKSHA - Sentiment Based Product Review Analysis System Aarti
Potdara Pranav Patila Raunak Baglaa Rohitashwa Pandeya Nagesh Jadhav
Prof.b
- [13] E-COMMERCE RESEARCH AND APPLICATIONS A PROPOSAL FOR
CLASSIFICATION AND AN UPDATED LITERATURE REVIEW-Kasim
Banyal
- [14] Semantics-Preserving Bag-of-Words Models and Applications-“Lei
Wu; Steven C.H. Hoi ; Nenghai Yu
- [15] Asuncion, Arthur et al. (2009). “On Smoothing and Inference for
TopicModels”. In:Proc. UAI, pp. 27–34. Blei, David M., Andrew Y. Ng, and
Michael I. Jordan (2003)
- [16] Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). “Maximum
likelihood from incomplete data via the EM algorithm”
- [17] Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and
Opinion mining In: Proceedings of the Seventh conference on
International Language Resources and Evaluation.. European Languages
Resources Association, Valletta, Malta
- [18] Liu B (2010) Sentiment analysis and subjectivity In: Handbook of
Natural Language Processing, Second