Journal of Science and Technology ISSN: 2362-1443
2023, Vol. 3, No. 1, 18-22
DOI: https://doi.org/10.3126/jost.v3i1.69060

Opinion Mining of Customer Reviews for Online Products through Sentiment Analysis

Sachit Karki
Nepal College of Information Technology,
sachitjungkarki@gmail.com

Arun Timalsina*
IoE, Tribhuvan University
t.arun@ieee.org

* Corresponding author

Article History:
Received: 9 July 2023
Revised: 8 October 2023
Accepted: 3 December 2023

Keywords—Sentiment analysis, Opinion Mining, Product reviews, Naïve Bayes, Logistic Regression, Support Vector Machine, TF-IDF

***Abstract*—Sentiment essentially relates to feelings; attitudes, emotions and opinions. Sentiment Analysis refers to the practice of applying different Data Mining techniques to identify and extract subjective information from a piece of text. A person's opinion or feelings are for the most part subjective and not facts, which means to accurately analyze an individual's opinion or mood from a piece of text can be extremely difficult. Sentiment Analysis has gained much attention in recent years due to the importance of the automation in mining, extracting and processing information in order to analyze an individual's opinion or mood from a piece of text. These days, Internet has become a valuable place for exchanging ideas, learning skills, sharing reviews of a product, service or movies, it makes hard to understand or identify the user's emotion from the list of available online reviews. With Sentiment Analysis from a text analytics point of view, I am essentially looking to get an understanding of the attitude of a writer with respect to a review in a piece of text and its polarity; whether it's positive, negative or neutral. There are different techniques and algorithms that can be used for sentiment analysis on opinion mining. This paper performs the extraction of opinions and emotions of customer from product reviews using data mining and natural language processing techniques. It focuses on opinion mining from product reviews and discusses the characteristics of reviews and describes different methods to extract corresponding opinions.**

I. INTRODUCTION

Humans have always been dependent on the data and statistics for decision making process. People have always looked for suggestions or opinions for making life decisions even before the start of the Internet. People used to ask their friends who they are planning to vote for election or their travel destination plans. The pattern is still the same in Internet era, as people are still following the same footstep. People tend to check the other's review on the Internet before buying anything from the online store or check ratings before watching a movie or start reading a book.

Although it is easy to find a product with hundreds or thousands of opinions, it could be hard for them to analyze all of them. Also, it could be very repetitive and laborious work to find opinions from an experienced customer about some features of a particular product. Sometimes the textual representation or the language barrier makes it even harder to identify the correct opinion for a customer. So, a summarization mechanism such as Sentiment Analysis with the support of massive online data helps to analyze better reality of opinions and provides better way of users to draw conclusions out of them.

Sentiment Analysis is defined by Oxford dictionary [1] as "The process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral." In other words, it is the process of determining the feelings of a writer on a particular topic based on the writer's opinion conveyed in a text.

Sentiment analysis is a type of natural language processing for tracking the mood of the public about a particular product or topic. Sentiment analysis, which is also called opinion mining, involves in building a system to collect and examine opinions about the product made in blog posts, comments, reviews or tweets. Sentiment analysis can be useful in several ways. For example, in marketing it helps in judging the success of an ad campaign or new product launch, determine which versions of a product or service are popular and even identify which demographics like or dislike particular features [2].

Sentiment analysis uses different techniques to determine the sentiment of a text or sentence. The Internet is a large repository of natural language. People share their thoughts and experiences which are subjective in nature. Most of the times getting suitable information about a product can became tedious for customers. Companies may not be fully aware of customer requirements. Product reviews can be analysed to understand the sentiment of the people towards a particular topic.

II. RELATED WORK

Sentiment analysis or opinion mining is one of the growing areas of Natural Language Processing. There have been numerous researches done from document level classification to sentiment polarity categorization of word and phrases. Experiments for both sentence-level categorization and review-level categorization are performed with promising outcomes [3]. Producers can have better knowledge of their products and services through the sentiment analysis (ex. positive and negative comments or consumers likes and dislikes) which will help them to know their products status (ex. product limitations or market status). Sentiment analysis with the help of fuzzy logic (deals with reasoning and gives closer views to the exact sentiment values) have been used help the producers or consumers or any interested person for taking the effective decision according to their product or service interest [4]. In recent times, people share their opinions, ideas through social networking site, electronic media etc. Different organizations always want to find public opinions about their products and services. Individual consumers also want to know the opinions from existing users before purchasing product. Sentiment analysis is the computational treatment of user's opinions, sentiments and subjectivity of text. Experiments using R software has been used to analyze sentiment of users on Twitter data using Twitter API which involves in collecting of data from twitter, its pre-processing and followed by a lexicon-based approach to analyze user's sentiment [5].

III. METHODOLOGY

This research is carried out in the following sections. Each step is carried out precisely with the visualization of its state.

A. Data Collection

Consumers usually express their sentiments on public forums like the blogs, discussion boards, product reviews or social network sites such as Facebook and Twitter. Opinions and feelings are expressed in different way, with different vocabulary, context of writing, usage of short forms and slang, making the data huge and disorganized. Manual analysis of sentiment data is virtually impossible.

For this research, the dataset is used from amazon site which is provided by Kaggle that originally came from SNAP. This dataset contains fine food reviews from Amazon, which contains 568,453 reviews. This dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category, price, brand). [7]

B. Preprocessing

This involves filtering the extracted data before analysis. The text in the dataset is provided in json format. Unwanted columns in the dataset is dropped. After discarding all the unwanted columns, the data was computed with pre- processing filters like converting texts in lowercase, removal of tags and special characters and digits. Data preparation such as filtering short uninformative review text like "This is amazing!" for certain limit, categorization of review rating in excellent, good, neutral, bad or worst for future prediction will performed in the dataset to make the data more relevant for analysis. Removing of stop words, tokenization, normalization and stemming are performed before feature extraction.

C. Feature Extraction

Since text do not directly work with classification models, TF-IDF is used as feature extraction method to convert text into vectors or number of vectors. TF-IDF stands for term frequency-inverse document frequency, and the TF-IDF weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of TF- IDF weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. Typically, the TF-IDF weight is composed by two terms: the first computes the normalized Term Frequency (TF), aka. the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

- **TF:** Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

TF(t) = (Number of times term t appears in a document)/ (Total number of terms in the document).

- **IDF:** Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus, the frequent terms need to be weighed down while scaling up the rare ones, by computing the following:

IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

For both algorithms, the limitation has been set as the words occurring in at most 90% of reviews and in at least 10 reviews.

D. Rating Prediction

A predictive scoring model is built after looking at the words within reviews. When training these types of models, overfitting can occur where it becomes very good at predicting the result, but fails to predict on new result. To avoid this, the data is split in 70%/30% ratio where 30% data is reserved for gauging our final accuracy. 5 different approaches is used, each for one categories, to build for each classification model. One that predicts excellent, one that predicts good, one that predicts neutral, one that predicts bad and one that predicts the worst based on their score. For each review, calculations is made using all four

classification models. The model that scores the highest will tell us which kind of review it likely is.

The four classification models used to build the review predictions are: Logistic Regression, Naive Bayes, Support Vector Machines, Voters Classifier. This means the total of (4 models) x (5 ratings) = 20 total approaches are actually built. Since the data is limited, 10-fold cross-validation is used to split the data and measure accuracy in an unbiased way. The prediction is done using Python programming language.

E. Rating Comparison

The main idea of sentiment analysis is to convert unstructured text or raw user input into meaningful information. After the completion of analysis, the review results are compared with the models used in our research. Carrying out sentiment analysis is an important task for all the product and service providers today. The result is compared on the basis of various computational scores carried out in each approach.

IV. IMPLEMENTATION AND RESULT

It has been known that Amazon Product Reviews Matter to Merchants because those reviews have tremendous impact on how people make purchase decisions. So, this dataset here is used to analyze a collection of text documents. The data contains 568453 reviews, it has been categorized the dataset in 5 types based on Score, which spans from 1 to 5. The 1 is considered as least preferable review and 5 is considered as most preferable review. This review score has been categorized to 5 different categories, i.e. Worst, Bad, Neutral, Good and Excellent based on 1-5 ratings. There are 52268 worst, 29768 bad, 42640 neutral, 80655 good and 363122 excellent reviews in this dataset.

The dataset contains many columns, as mentioned in dataset section, where only 'Text' and 'Score' column is taken into account for this analysis. After discarding all the remaining columns, the data was computed with pre- processing filters like converting texts in lowercase, removal of tags and special characters and digits. Based on the score of the review the data is provided with a new column called "Category" that maps the Score value to Category, i.e. Worst for 1, Bad for 2, Neutral for 3, Good for 4 and Excellent for 5.

Then new columns are appended with the dataset for each review, which are Worst, Bad, Neutral, Good and Excellent where each review that falls into any of those categories is encoded as 1, example if the score is 5 then "excellent" column gets the value 1 and other category gets the value 0. This process is also known as One Hot Encoding. One Hot Encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.

Both the actual term frequency as well as the TF-IDF weighted term frequency has been calculated. Tokenization and stemming are also done while calculating TF-DIF value for each review. For both algorithms, the "max_df" and "min_df" were used. "max_df" is used for removing terms that appear too frequently, also known as "corpus-specific stop words". "min_df" is used for removing terms that appear too infrequently. The max_df used in this analysis is 90%, which means "ignore terms that appear in more than 90% of the documents". The min_df used in this analysis is 10, which means "ignore terms that appear in less than 10 document".

While the term-frequency matrix is just a word count, the IDF calculation adjusts for "boring" words that occur in many reviews. When training these types of models, overfitting can occur where it becomes very good at predicting the test data, but fail to predict on new data. To avoid this, the data has been split in 70%:30% ratio, where 30% of data was reserved for gauging the final accuracy. The use of n-gram feature used while computing TF-IDF are (1,1) meaning only unigrams, (1,2) meaning unigrams and bigrams, (1,3) meaning unigrams, bigrams and trigrams, (2,2) meaning bigrams only, (2,3) meaning bigrams and trigrams and (3,3) meaning trigrams only. In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application.

Then the split data was used to fit the Naïve Bayes, Support Vector Machine and Logistic Regression models. Since the original idea of this research was to analyze and compare the performance of Naïve Bayes, Logistic Regression and Support Vector Machine in the dataset, but one more model has been used to combine all these models using Voter Classifier Model. So, there are now 4 types of model for each category of review, total of 4 * 5 = 20 approaches.

As there is never enough data to train a model, removing a part of it for validation poses a problem of underfitting. By reducing the training data, there is a risk of losing important patterns/ trends in data set, which in turn increases error induced by bias. So, what requires is a method that provides ample data for training the model and also leaves ample data for validation. K Fold cross validation does exactly that.

So, the cross validation has been used for assessing the effectiveness of the model, especially for tackling overfitting and underfitting. In addition, it is useful to determine the hyper parameters of the model, in the sense that which parameters will result in lowest test error. Only K-fold cross validation has been used so far, where the number of folds is 10. Every data point gets to be in a validation set exactly once, and gets to be in a training set k-1 time. This significantly reduces underfitting since most of the data was being used for fitting, and also significantly reduces overfitting as most of the data is also being used in validation set.

TABLE I. LR ACCURACY SCORE COMPARISON FOR N-GRAM

N-grams Categories	(1,1)	(1,2)	(1,3)	(2,2)	(2,3)	(3,3)
worst	94	94.8	94.7	93.9	93.9	92.6
bad	94.8	95.2	95.2	95.3	95.3	95
neutral	92.7	93.4	93.5	93.4	93.4	93.1
good	86.3	87.9	88.3	88.1	88.2	87.6
excellent	82.2	86.1	86.4	85.3	85.5	79.1

TABLE II. SVM ACCURACY SCORE COMPARISON FOR N-GRAM

N-grams Categories	(1,1)	(1,2)	(1,3)	(2,2)	(2,3)	(3,3)
worst	94.2	95.9	95.9	95.4	95.5	93.5
bad	94.9	96.2	96.3	96.2	96.3	95.6
neutral	92.9	94.7	95	94.6	94.9	93.8
good	86.5	89.6	90.1	89.3	89.8	88.1
excellent	82.3	87.2	87.8	86	86.4	79.4

TABLE III. NB ACCURACY SCORE COMPARISON FOR N-GRAM

N-grams Categories	(1,1)	(1,2)	(1,3)	(2,2)	(2,3)	(3,3)
worst	91.8	92.5	91.9	92.7	92.5	92
bad	94.8	94.9	94.6	95.1	94.7	94.6
neutral	92.6	92.8	92.6	93.1	92.7	92.6
good	85.9	86.7	86.8	87.2	87.2	87.2
excellent	77.9	83.5	83.8	83.6	84	78.6

TABLE IV. VC ACCURACY SCORE COMPARISON FOR N-GRAM

N-grams Categories	(1,1)	(1,2)	(1,3)	(2,2)	(2,3)	(3,3)
worst	94	95	94.8	94.1	94.1	92.8
bad	94.8	95.3	95.3	95.4	95.3	95.2
neutral	92.8	93.5	93.6	93.5	93.5	93.3
good	86.4	88	88.4	88.2	88.3	87.8
excellent	82.3	86.4	86.8	85.7	85.9	79.4

Based on the results of n-grams vs categories accuracy score comparison, computed among various models such as LR, SVM, NB and VC. (1,3) seems to have highest score in most of all n-gram combination, (2,2) also seem to have better result for Naïve Bayes model. Therefore, this experiment is inconclusive as there are two different winners. But since (1,3) has major leading accuracy and contains all unigram, bigram and trigram combination, we take (1,3) as better resulting n- gram.

A. Comparison between Models based on scores

As the comparison between n-gram is done, further comparison between models based on scores for (1,3) needs to be done in order to identify the best model among LR, SVM, NB and VC.

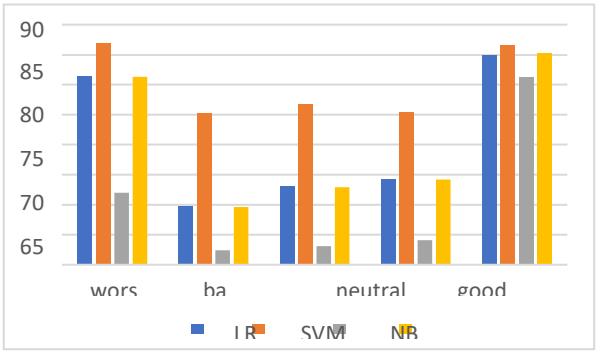

Fig. 1. F1 Score Comparison

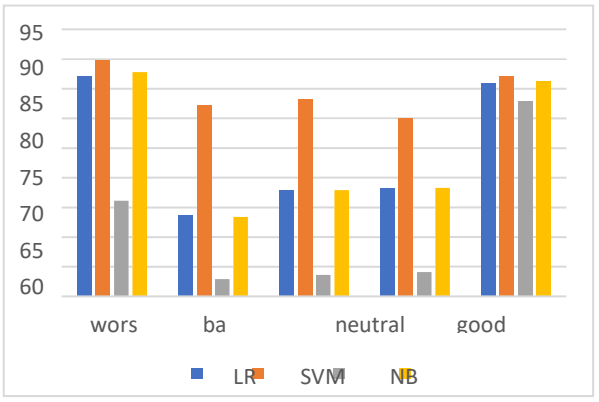

Fig. 2. Precision Score Comparison

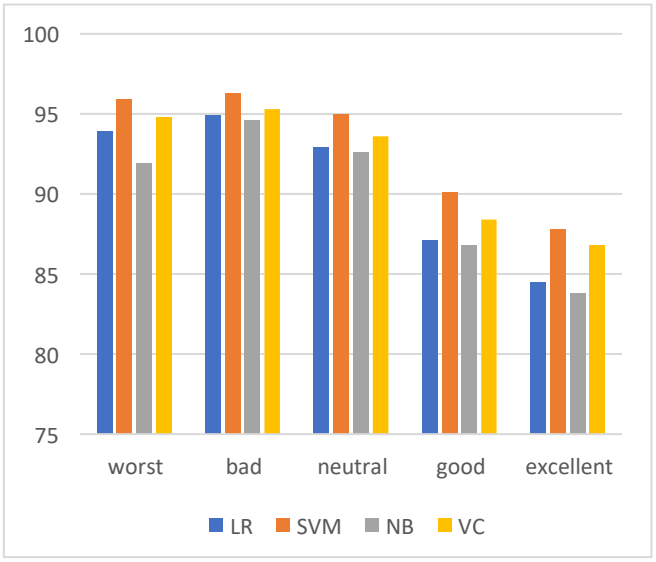

Fig. 3. Cross Validation Score Comparison

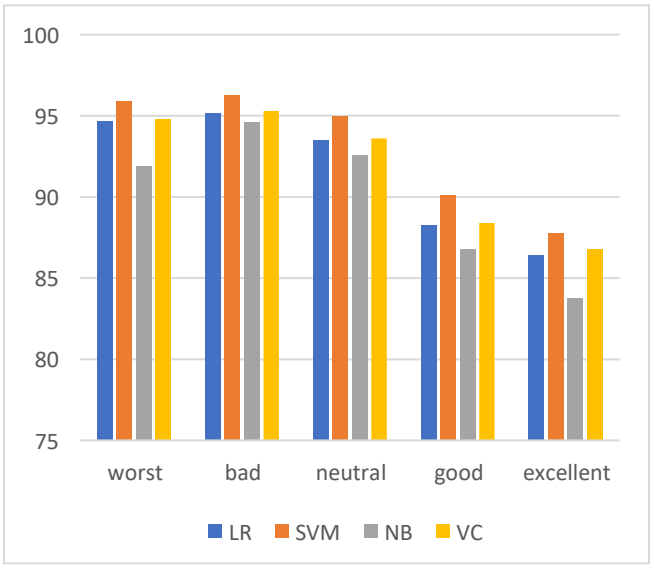

Fig. 4. Accuracy Score Comparison

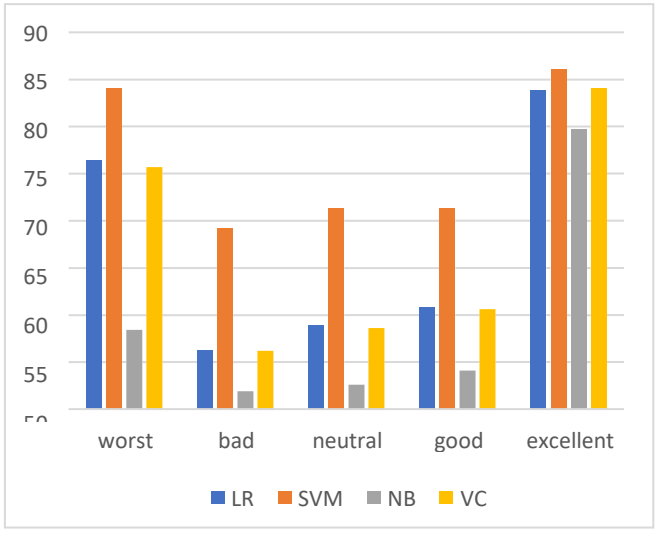

Fig. 5. Recall Score Comparison

Here, SVM has better result among other models, VC seems to be second better model for (1,3) result. But if comparison is to be made be based on accuracy score on other n-gram combination, SVM still proves to be yielding better result than the rest.

V. CONCLUSION

There has been done some improvements in order to identify the best model among these 4 models. Improvements like increasing the number of n in n-gram features. Using combination of uni-grams, bi-grams and tri-grams has led to more appropriate result. And even a combining model that can be used to combine Logistic regression, Naïve Bayes and SVM. A Voting classifier model combines multiple different models into a single model, which is (ideally) stronger than any of the individual models alone. Voting is the simplest and easiest way to combine classifiers, demanding no extra training for final prediction except for the pre-existing individual classifier classifications. Due to its ability to significantly improve predictions, voting spans many applications ranging from simple classification tasks to more complex implementations such as clustering, pairwise comparison and fuzzy systems.

The result clearly shows that SVM has dominated on all the scores type performed in this experiment. It has outperformed all the other selected algorithms in this dataset. There is only slight difference between the accuracy of LR and VC as compared to that of other models. So, it seems LR or VC has second better result. From the experiment, the study showed that as the score of accuracy, precision, recall, f1 and 10-fold cross validation of SVM is higher or equal in every n- gram test. Furthermore, it can be concluded that the SVM model has a higher score which makes it a most useful model for this type of sentiment analysis on product reviews in the future as well.

REFERENCE

[1] "Sentiment Analysis," in *Oxford Dictionary*.

[2] G. Vinodhini and R. Chandrasekaran, "Sentiment Analysis and Opinion Mining: A Survey," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 6, June 2012.

[3] P. Ray and A. Chakrabarti, "Twitter Sentiment Analysis for Product Review Using Lexicon Method," *International Conference on Data Management, Analytics and Innovation (ICDMAI)*, 2017.

[4] P. Kalaivani, "Sentiment classification of movie review by supervised machine learning approaches," *Indian Journal of Computer Science and Engineering*, vol. 4, no. 4, 2013.

[5] A. Haque and T. Rahman, "Sentiment Analysis By Using Fuzzy Logic," *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, vol. 4, no. 1, 2014.

[6] X. Fang and J. Zhan, "Sentiment analysis using product review data," *International Journal of Big Data*, 2015.

[7] R. D. et. al, "A Study of the Effects of Preprocessing Strategies on Sentiment Analysis for Arabic Text," *Journal of Information Science*, pp. 1-14, 2013.

[8] E. K. et al., "Twitter Sentiment Analysis: The good, the bad and the omg!," *Fifth International AAAI Conference on Weblogs and Social Media*, pp. 538-541, 2011.

[9] G. Chouhan, "Sentiment Analysis of Hindi Review based on Negation and Discourse Relation," *International Joint Conference on Natural Language Processing, Nagoya, Japan*, 2013.

[10] "Amazon Fine Food Reviews," Stanford Network Analysis Project,

[11] [Online]. Available: https://www.kaggle.com/snap/amazon-fine- food-reviews.