SMU Data Science Review

Volume 1 | Number 4 Article 7

2018

Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches

Heidi Nguyen
Southern Methodist University, hqnguyen@smu.edu

Aravind Veluchamy
Southern Methodist University, aveluchamy@mail.smu.edu

Mamadou Diop
Southern Methodist University, mldiop@smu.edu

Rashed Iqbal
Ras Al Khaimah Academy, rashed.iqbal@gmail.com

Recommended Citation

Nguyen, Heidi; Veluchamy, Aravind; Diop, Mamadou; and Iqbal, Rashed (2018) "Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches," *SMU Data Science Review*: Vol. 1: No. 4, Article 7.
Available at: https://scholar.smu.edu/datasciencereview/vol1/iss4/7

Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches

Aravind Veluchamy[1], Heidi Nguyen[1], Mamadou L. Diop[1], Rashid Iqbal[2]
[1] Master of Science in Data Science, Southern Methodist University,
Dallas, TX 75275 USA
[2] Ras Al Khaimah Academy, Julphar Towers, Al Hisn Rd,
Ras Al Khaimah, United Arab Emirates
{aveluchamy, hqnguyen, mldiop}@smu.edu
rashed.iqbal@gmail.com

Abstract. In this paper, we present a comparative study of text sentiment classification models using term frequency inverse document frequency vectorization in both supervised machine learning and lexicon-based techniques. There have been multiple promising machine learning and lexicon-based techniques, but the relative goodness of each approach on specific types of problems is not well understood. In order to offer researchers comprehensive insights, we compare a total of six algorithms to each other. The three machine learning algorithms are: Logistic Regression (LR), Support Vector Machine (SVM), and Gradient Boosting. The three lexicon-based algorithms are: Valence Aware Dictionary and Sentiment Reasoner (VADER), Pattern, and SentiWordNet. The underlying dataset consists of Amazon consumer reviews. For performance measures, we use accuracy, precision, recall, and F1-score. Our experiments' results show that all three machine learning models outperform the lexicon-based models on all the metrics. SVM, Gradient Boosting, and LR models have accuracy of 89%, 87%, and 90%; precision of 90%, 88%, and 91%; recall of 98%, 98%, and 97%; F1-score of 94%, 92%, and 94%, respectively. Pattern, VADER, and SentiWordNet models have accuracy of 69%, 83%, and 80%; recall of 72%, 89%, and 88%, precision of 88%, 90%, and 90%; F1-score of 79%, 89%, and 88%, respectively. Our machine learning results are slightly better compared to recent text sentiment machine learning works while our lexicon-based result are worse compared to recent similar lexicon-based works.

1 Introduction

User-generated content such as product reviews on Amazon has huge power of shaping and influencing consumer's purchasing decisions, since buyers are highly motivated by other shoppers' recommendations and experiences. It is important to develop systematic methods to understand the information provided in user-generated content.

The most popular method of gaining insight into customers' text reviews is performing sentiment analysis to determine whether a review is positive or negative. In addition, the overwhelming magnitude of user-generated content repositories and their continuing fast growth make it very labor intensive to manually monitor and extract sentiment from user-generated content [1]. Automatic classification of textual content becomes the only practical method for effective data classification and insight. In recent years, there have been multiple machine learning and lexicon-based approaches along these lines, each with advantages and disadvantages, but the relative goodness of each approach is not well understood.

There has been substantial work on sentiment classification on reviews and comments from interactive websites using machine learning techniques at the document level [12]-[22]. In these methods, the model takes a review (a document), breaks it down into sentences, then examines each sentence for its structure and the contextual dependency of each word within the sentence to determine the sentiment orientation of the sentence [2]. Most studies on product review sentiment analysis are based on binary classification where the reviews are classified into "positive" and "negative." Moreover, even the best systems currently obtain F1-score, precision, accuracy of only about 80% [3][4].

There has not been as much work on the same topics using lexicon-based techniques at the document level. However, recently there has been progress on building lexicons for sentiment analysis. Comparing these new lexicon methods to machine learning techniques is the primary impetus for this project. In this paper, we present a comparative study of binary text sentiment classification using term frequency inverse document frequency (TF-IDF) vectorization in the three machine learning models and pre-processed texts in the three lexicon models. The three supervised machine learning techniques are Support Vector Machine (SVM), Logistic Regression (LR), and Gradient Boosting. The three lexicon-based techniques are Valence Aware Dictionary and Sentiment Reasoner (VADER), Pattern lexicon, and SentiWordNet lexicon. This is involved utilizing Amazon standard identification numbers (ASINs) and a Python library called Scrapy to collect text product reviews. A corpus consisting of 43,620 product reviews from 1,000 different products serves as the dataset of this study. These text reviews are pre-processed using various natural language processing (NLP) methods. Amazon allows its users to rate a product from 1 to 5 stars (1 is the lowest evaluation, and 5 is the best), and provide a text summary of their experiences and opinions about the product as well as the seller. We utilize this rating system to label the text reviews. Reviews receiving a 1-, 2-, or 3-star rating are labeled as 'negative', or '0' score, in the data, whereas reviews receiving 4 or 5 stars are labeled as 'positive, or '1' score. We notice that the final data set is imbalanced with 82% being labeled as positive.

The text reviews are represented as TF-IDF feature vectors that are generated from all the individual words in the reviews. Each of these feature vectors consists of TF-IDF scores. A TF-IDF score of a term is the product of that term's frequency and its relative importance score within a document. These TF-IDF vectors are the sole inputs into all three machine learning models while and pre-processed texts are the inputs into the three lexicon models. We utilize machine learning methods available in Python Scikit-learn library, such as SGDClassifier for SVM model, LogisticRegression for LR model, GradientBoostingClassifier for Gradient Boosting model. Because this project is a comparative study, we keep hyperparameter tuning to a minimum with default parameters for all these models.

The findings of our study show that all three supervised machine learning models perform well. In term of accuracy, SVM, Gradient Boosting, and LR models have results of 89%, 87%, and 90%, respectively. Lexicon-based models using Pattern, VADER, and SentiWordNet lexicons have accuracy as 69%, 83%, and 80%, respectively. In term of precision, all six models have high precision scores in the range of 88% to 91%. The three machine learning models also give high recall in the range of 97%-98%. However, the three lexicon-based models struggle with recall of only 72% with Pattern lexicon, 89%with VADER lexicon, and 88% with SentiWordNet lexicon. The F1 scores for SVM, Gradient Boosting, and LR are 94%, 92%, and 94%, respectively. The F1 scores for lexicon-based models are lower at 79%, 89%, and 88% for Pattern lexicon, VADER lexicon, and SentiWordNet lexicon, respectively.

The remainder of this paper is organized as follows: Section 2 provides a brief overview of Amazon, Amazon product reviews, Natural Language Processing, sentiment analysis and techniques in sentiment classification. Section 3 discusses previous work that this paper builds from. Section 4 describes how the data were collected and what the text dataset looks like. Section 5 describes the text preprocessing methods and the TF-IDF method that transform texts into numerical vectors. Section 6 describes the modeling of all six algorithms. Section 7 describes the model results. Section 8 discusses the ethical considerations when using acquired Amazon product review data. Section 9 summarizes our conclusions and discusses future work.

2 Amazon Product Reviews, Natural Language Processing, and Sentiment Analysis Background

The analysis detailed later in this paper requires an understanding of where the data were collected, what natural language processing (NLP) is and how it is used to pre-process our text data. In this section we will also provide a background on sentiment analysis and sentiment classification techniques.

2.1 Amazon and Its Product Reviews

Amazon.com is one of the largest e-commerce companies in the world. Amazon currently offers more than 12 million different products [6]. They sell books, music, games, phone apps, movie, clothes, electronics, toys, and many other goods. Since its creation as an online platform in 1994, Amazon.com has grown rapidly. As of the last reported period in February 2017, Amazon had 310 million active customers [5]. With this vast user base and huge product collection, Amazon has become a microcosm for user-supplied reviews. There is tremendous interest in sentiment analysis of these Amazon product reviews across a variety of domains such as commerce, health, and social behavior study [7].

Amazon allows its users to rate a product from 1 to 5 stars (1 is the lowest evaluation, and 5 is the best), and provide a text summary of their experiences and opinions about the product as well as the seller. This scoring system is universal, regardless of the product category. Since there is no guidance on how an Amazon web user should use this scoring system, Amazon product reviews are very personal and subjective. One can give a score of 1 for a good product, but bad purchasing experience, such as high price, or late delivery, and vice versa. This lack of guideline makes it challenging to determine the sentiment of a user toward different aspects of a product, different parts of a shopping experience, but at the same time makes Amazon product review a very rich source of data on how people perceive products and services.

2.2 Natural language Processing

Much of user-generated content is in the form of unstructured text. This vast amount of unstructured data has led to the creation of a collection of machine-based methods for computers to process content and understand text. This collection is referred to as natural language processing (NLP).

In this paper, we use various NLP approaches to process the comment fields of the reviews and turn them into machine readable vectors. We also utilize many Python libraries, such as Natural Language Toolkit (NLTK), SpaCy, and Pattern. NLTK provides more than 50 collections of text and lexical resources and many necessary tools, interfaces, and methods to process and analyze text data. NLTK contains the VADER lexicon and SentiWordNet lexicon that are used in our models. The Pattern library provides tools and interfaces for web mining, information retrieval, NLP, machine learning, and network analysis. The pattern.en module contains much of the same utilities as nltk, but they are generally more efficient. SpaCy is the newest library that provides the best implementation of each NLP technique and algorithm.

Other frameworks such as Python Scikit-learn, NumPy, Pandas, and SciPy stack libraries are used for converting text documents into vectors and for applying machine learning techniques to textual data.

2.3 Sentiment Analysis

Sentiment analysis is a discipline of text classification. Sentiment analysis refers to the practice of applying NLP and text analysis techniques to identify and extract subjective information from a piece of text. Sentiment analysis works better on text that has a subjective context than it does on text with only an objective context. This is due to the fact that if a body of text has an objective context or perspective to it, the text usually depicts some normal statements or facts without expressing any emotion, feelings, or mood [10] [11]. Subjective text contains text that is usually expressed by a human having typical moods, emotions, and feelings.

2.4 Machine Learning and Lexicon-Based Techniques in Sentiment Analysis

Various techniques are used to tackle sentiment analysis problems. One group of techniques is called supervised machine learning and uses classification algorithms to classify documents according to their associated sentiment. Supervised learning requires learning from a set of training data. Two widely used supervised machine learning algorithms for text classification are logistic regression (LR) and support vector machine (SVM) [8].

Logistic Regression (LR) is a classification algorithm, also called the logistic function, used to assign observations to a discrete set of classes. LR is a robust technique for two-class and multiclass classification. SVM is a supervised learning technique that uses hyperplanes to divide data into two groups. In recent years, SVM has been among the most widely used classifier. Also recently, researchers have applied the Gradient Boosting machine learning technique for sentiment analysis and have seen superior performance over SVM and LR. Gradient Boosting machine learning is an algorithm that is built on small decision trees. Each Gradient Boosting iteration fits a new model to get better class estimation. Each newly added model is correlated with the negative gradient of the loss function, and the loss is minimized using gradient descent [9].

There is also a surge in developing and using lexicons, which are dictionaries or vocabularies specifically constructed to be used for sentiment analysis. These allow researchers to compute sentiment without using any pre-classified corpus, a collection of words. Some of the most popular lexicons are VADER, Pattern, and SentiWordNet. VADER was specifically built for analyzing sentiment from social media resources with more than 9000 lexical features (words). SentiWordNet is the largest English lexical resource for sentiment analysis and opinion mining. The Pattern package has a sentiment module and other modules for analyzing mood and modality of a body of text [10].

3 Related Work

A major research field has emerged around the subject of how to extract the best and most accurate method and simultaneously categorize the customers' written reviews into negative or positive opinions. In a 2002 publication, Pang, Lee and Vaithyanathan were the first to propose sentiment classification using machine learning models on movie reviews dataset. They analyzed the Naïve Bayes, Max Entropy and Support Vector Machine models for sentiment analysis on unigrams and bigrams of data. In their experiment, SVM paired with unigram feature extraction produced the best results. They reported a result of 82.9% accuracy [12].

In a 2004 publication, Mullen and Collier performed sentiment classification on clothing, shoes and jewelry product review datasets [17]. They compared methods of hybrid SVM, Naïve Bayes, LR, and decision tree with feature extraction methods based on Lemmas and Osgood theory [18]. In their study, SVM produced the best results with an accuracy of 86.6%. In a 2015 publication, Lilleberg, Zhu, and Zhang performed a comparison study of TF-IDF and Word2vec feature extractions using SVM. They also compared the classification results with and without including stopwords. The best result of SVM with TF-IDF and without stopwords that they saw was 88% accuracy [19].

In recent years, the common classification techniques for document analysis include SVM and LR. In a 2017 publication, SVM and sentiment analysis were proposed by Elmurngi and Gherbi to detect fake movie reviews. They compared SVMs with Naïve Bayes, decision tree, and KNN classifications performance on a corpus with stopwords and a corpus without stopwords. In both cases, SVM performed the best, with accuracies of 81.75% and 81.35%, respectively [13]. In another publication, Ramadhan et al. conducted a sentiment analysis using logistic regression and TF-IDF feature extraction on a social media Twitter dataset. The classification accuracy was reported to be close to 83% [14]. In 2018, Das and Chakraborty conducted an experiment using SVM, TF-IDF model coupled with Next Word Negation on a Amazon product review dataset and reported accuracy 88.86% [20].

In a publication in 2018, Bhavitha, Rodrigues, and Chiplunkar also performed a comparative study of several machine learning methods, lexicon-based methods and sentiment analysis on movie reviews. For the SentiWordNet method, they reported an accuracy of 74%, and for the SVM method, they reported an accuracy of 86.40% [21]. In the same year, Athanasiou applied Gradient Boosting machine learning for sentiment analysis and found superior performance over SVM, Naive Bayes, and neural network for both balanced and imbalanced data sets. The Gradient Boosting machine learning performed best with an accuracy of 88.20% [22].

4 Data

This section details how the data sources were gathered, cleaned, and adjusted when necessary.

4.1 Collecting Amazon Product Reviews using Amazon Standard Identification Number

Amazon does not have an API to download reviews, but it has links for every review on every product through its product IDs, called Amazon Standard Identification Numbers (ASINs). In order to collect the ASINs of different products, we developed a Python script which used the Scapy library to go through 44 different product departments and collect products and their ASINs. Once we retrieved the ASINs, we traversed the site to collect the reviews. We collected 93,395 ASINs, which have a potential 14,448,400 reviews. Fig. 1 shows the distribution of reviews for products whose ASINs we were able to collect. Note that this does not include ASINs with no reviews or rating.

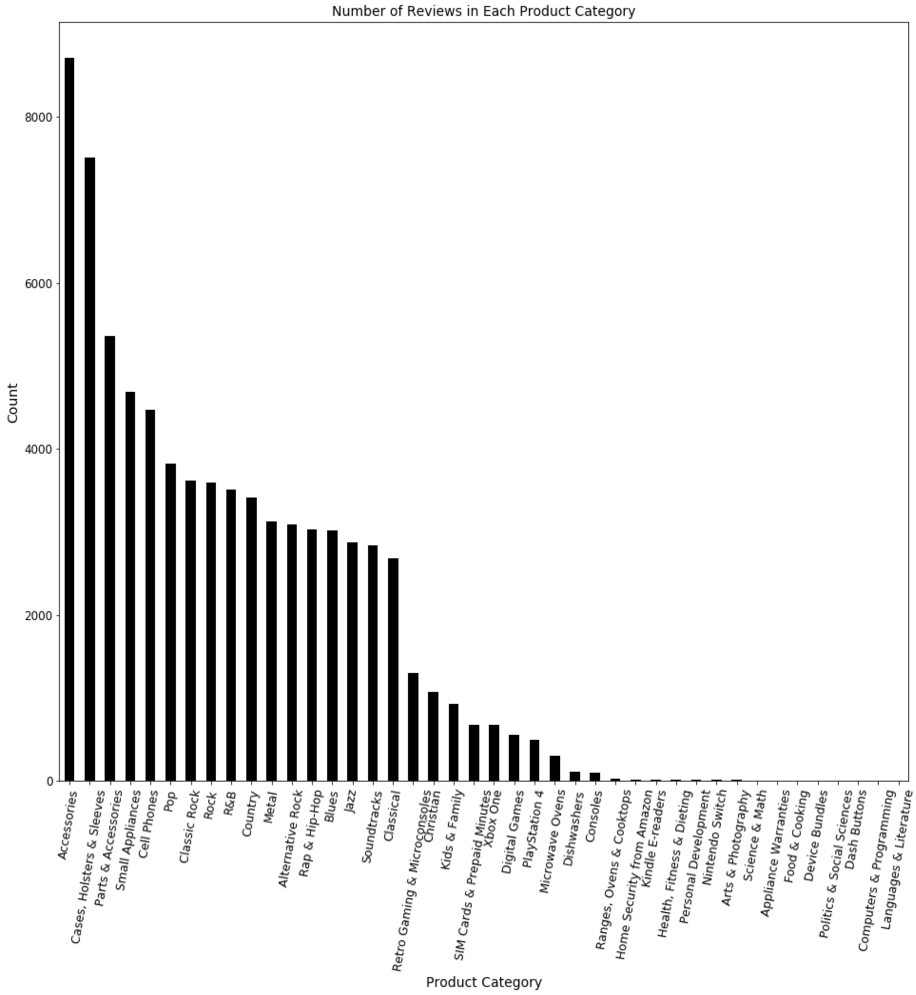


Fig. 1. Distribution of number of reviews across 44 product categories.

The majority of products have 10 to 400 reviews. Fig. 2 below is the graph of the distribution of the number of reviews per product. The x-axis is the logarithm of the number of reviews. The y-axis is the number of products having a given number of reviews.

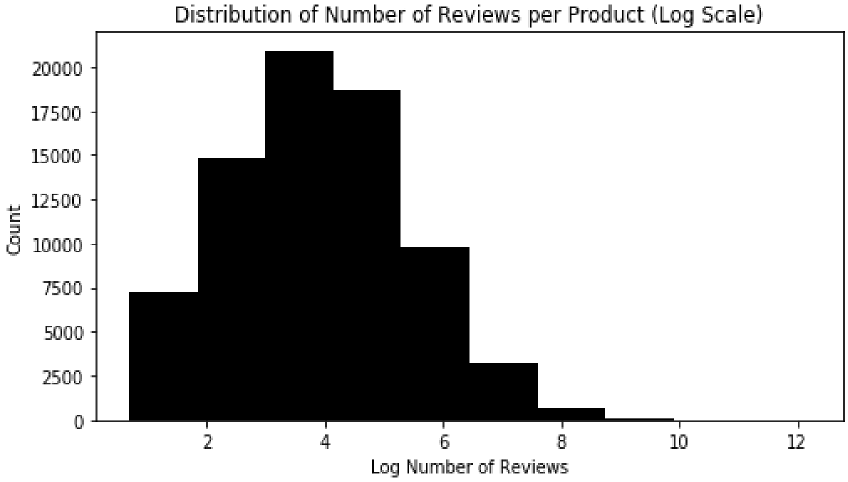


Fig. 2. Distribution of number of reviews in log scale.

Fig. 3 below shows the Kernel Density Estimation of the distribution of number of reviews per product in log scale. This plot confirms that the majority of products have 10 to 400 reviews and that 25% of the products have around 300-400 reviews.

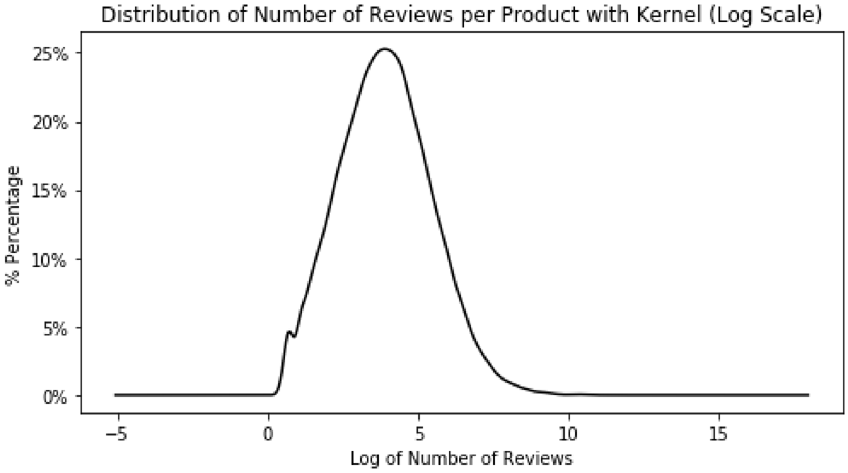


Fig. 3. Kernel Density Distribution of numbers of review per product in log scale.

4.2 Sampling Procedure

Even though there were potentially 14 million reviews on 94 thousand products, we randomly picked 1000 products (ASINs) to scrape product reviews to form our dataset. We used the sample() function in the Python’s random library to randomly sample, without replacement, a subset of 1000 ASINs. We then used two Python scripts to obtain the reviews. The first script downloads the entire HTML page for the product and the second searches the page for information about the review, such as review rating, review title, review date, and review text. Table 1 shows the summary of these attributes.

Table 1. Summary of data meaning and type of data in the raw dataset.

Category	Data Type	Description	Example
rating	Integer	Rating scored based on Amazon's 1-5 stars rating system.	1 out of 5 stars
title	String	Short description of the review.	"I did like the light feature
date	Date time	Date of the review.	6/18/18
body	String	The body of the text review.	"Wanted to be able to press a button to turn the fan on in the dark"

4.3 Data Exploration

For data manipulation, we removed duplicate reviews caused by the fact that Amazon allows cross reviews on similar products in the same category. For those reviews having a rating score and text title, but not having any text in the review body, we copied the text in the title to the review body. Table 2 is the summary of the final data set.

Table 2. Summary of data meaning and type in the final dataset.

Data	Number of Entries	Data Type
rating	43,620	Integer
title	43,619	String
date	43,620	Date time
body	43,620	String

Rating of reviews are skewed towards 4 and 5 stars. Fig. 4 and Fig. 5 show the distribution of rating scores, based on Amazon's 1 to 5-star rating scale, in our data set. The most frequent rating is 5 stars, with more than 40% in the entire data set.

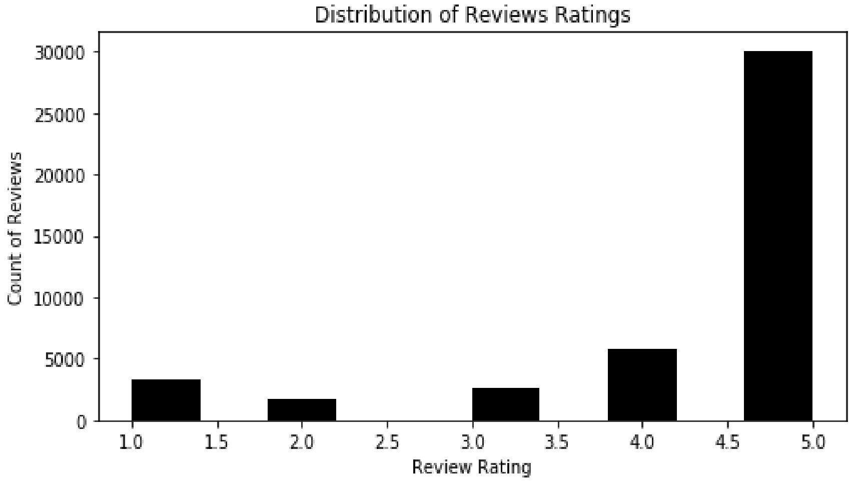


Fig. 4. Distribution of rating score based on Amazon’s 1-5 stars rating scale.

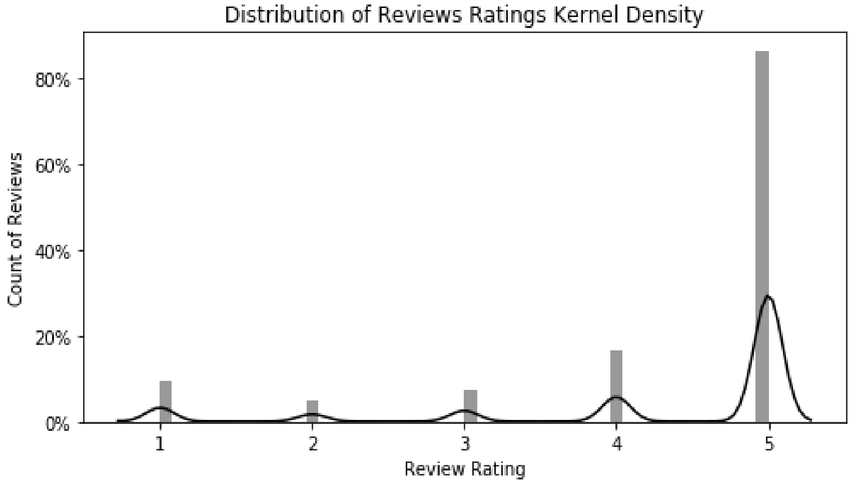


Fig. 5. Kernel Density Distribution of rating score based on Amazon’s 1-5 stars rating scale.

Fig. 6 shows the character count of the review text body grouped by rating score. The majority of reviews are less than 1000 characters long.

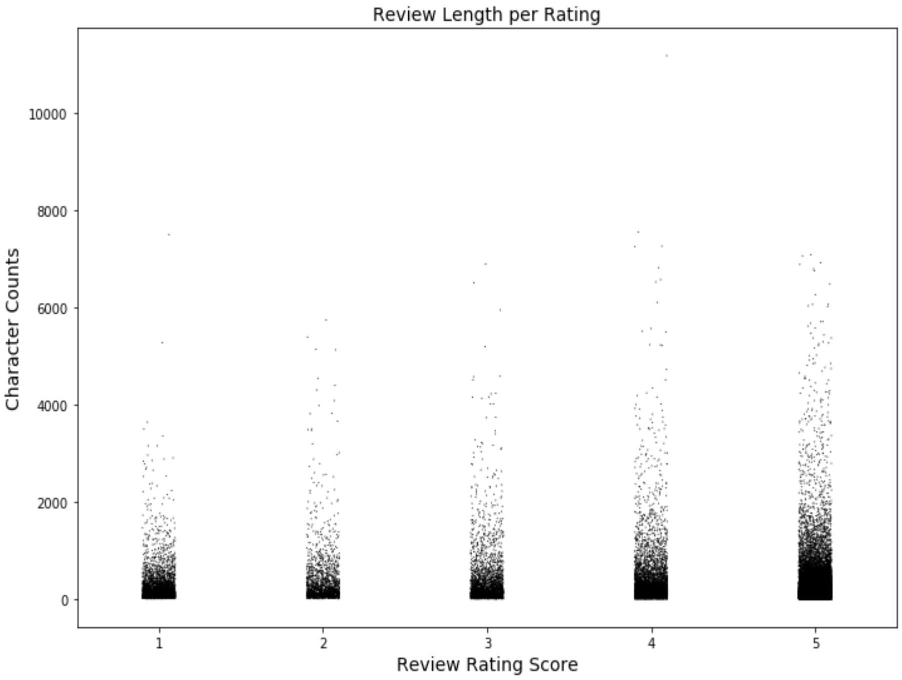


Fig. 6. Character count of the review text body grouped by rating score.

The dataset used in our experiment is separated into two groups. All reviews with ratings of 1 star, 2 stars, and 3 stars are labelled as negative, or “0” score, while all reviews with ratings of 4 stars and 5 stars are labelled as positive, or “1” score. Even though most researchers suggest to remove the neutral rating score, 3 stars [23], we decided to label all the 3-star ratings as negative. The rationale for this is that it is generally difficult to recognize words or sentences that are neutral. Moreover, the distribution of the rating of our dataset is skewed with a mean of 3.8, so it is reasonable to group 3-star rating reviews as negative. In practice, when firms collect their customer reviews, they usually consider neutral responses as negative. Fig. 7 below shows the distribution of negative and positive rating reviews in our dataset after labelling.

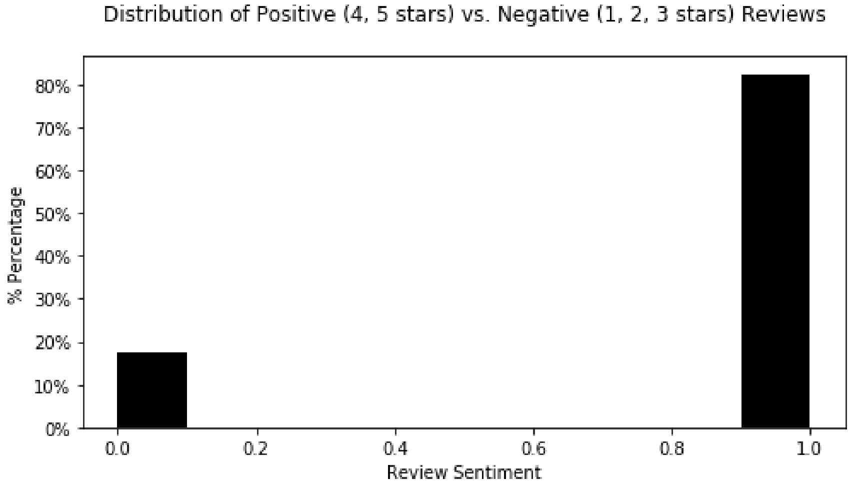


Fig. 7. Distribution of number of reviews across 1000 products.

5 Text Pre-processing and TF-IDF Vectorization

5.1 Text Pre-processing using NLP methods

Text preprocessing is an important aspect of sentiment Analysis. A model output is only as good as the data it is fed, therefore special emphasis is placed on text preprocessing [22]. User-generated content is typically unstructured. Therefore, certain steps are followed to normalize the data before feeding the data to a rule-based classifier or machine learning model. The main steps performed in text normalization include the following:

Perform sentence extraction. In this step, we read a text document, remove newline characters, parse the text, convert it into ASCII format, and break it down into its sentence constituents. Python scripts and various NLTK methods are utilized to complete this task.

Unescape HTML escape sequences. This step deals with unescaping special HTML characters. These characters prevent the text from being processed during subsequent steps like expanding contractions. So we use the Python module HTMLParser to unescape them and bring them back to their original unescaped form [10].

Expand contractions. Contractions need to be expanded to their individual words prior to removing stop words. This is achieved by using a function created using regular expressions.

Remove special characters and accented characters. The characters known as accent marks create a lot of issues in preprocessing steps like lemmatizing and expanding contractions; we remove these accent characters as a precautionary step. We also remove any other special characters and emoji in this step.

Lemmatize text. Lemmatization is the process of obtaining a morphological root of words [10]. In many cases, lemmatizing allows machines to recognize different tenses of the same word. We use the WordNet lemmatizer module available in the NLTK library.

Remove stopwords. Stop words are words that have little or no significance, like "I," "to," and "the" [10]. We use the NLTK stopwords corpus, but not excluding "no", "not", and "cannot".

Perform tokenization. Tokenization is the process of separating the words of a sentence into individual units, which are used for feature extraction. We use the tokenize module in NLTK library to complete this task.

5.2 Term Frequency Inverse Document Frequency (TF-IDF) Vectorization

Feature extraction is a process whereby we extract meaningful attributes from raw textual data that are fed into a statistical or ML algorithm. This process is also known as vectorization because the end result of this process is a set of numerical vectors. The step is needed because conventional algorithms work on numerical vectors and cannot work directly on raw text data. In this paper, we choose to use TF-IDF method because TF-IDF is recognized, by far, as the best feature extraction method for text analytics.

Term Frequency (TF). TF is the frequency of occurrence, of a word or group of words in a document. This is also called Bag of Words model. In this model, each document is represented as a vector of 0s and 1s. If a word exists in a document, its corresponding position in the vector is coded as a "1" and if it doesn't, it is coded as a 0. TF is calculated as follow:

$$\mathrm{TF(word)} = \frac{\text{Frequency of Word in the Document}}{\text{Number of Word in the Document}}. \quad (1)$$

Inverse Document Frequency(IDF). The IDF of a word is the measure of the relative importance of that word is in the whole corpus. IDF is calculated as follow:

$$\mathrm{IDF(word)} = \log\left(\frac{\text{Total Number of Documents}}{\text{Number of Documents Containing the Word}}\right). \quad (2)$$

Term Frequency Inverse Document Frequency(TF-IDF). TF-IDF is the product of TF and IDF score for specific words. In TF-IDF model, each document is represented as a vector that contains TF-IDF scores for each of the words in the document. TF-IDF scales down the impact of frequent but less informative features.

In this paper, we build our TF-IDF model using TfidfVectorizer and TfidfTransformer modules available in the Python Scikit-learn library. These modules fit and transform feature on the text data. The vectorized TF-IDF includes only unigrams (single words).

6 Modelling and Metrics

Sentiment classification algorithms were used to classify documents as positive or negative. In our study, we perform binary classification using three popular supervised classifiers, namely LR, SVM, and Gradient Boosting classifiers and three common lexicons in NLP, namely VADER, Pattern, and SentiWordNet. In our experiment, a single sentiment value is computed per document.

We used various frameworks, libraries, and computing platforms to build our models. Our models were implemented in Python 2.7 on a Jupyter notebook. The models were trained and tested locally on a 2.7 GHz Intel Core i5 machine with 8 GB of 1867 MHz DDR3 RAM. Python libraries were used including nltk 3.3.1, pattern 2.4, spacy 2.0.12, pandas 0.23.1, numpy 1.14.5, scikit-learn 0.19.1, matplotlib 1.5.1.

6.1 Metrics of Binary Classification

The result of binary classification consists of true positives, false positives, true negatives, and false negatives. True positives and true negatives accurately predict actual labels while false positives and false negatives are misclassifications. Accuracy (3) is the proportion of the total number of predictions that were correct. In addition, precision (4) is a measure of how good the classifier is classifying reviews as positive sentiment. Recall (5) measures how good the classifier is at correctly classifying reviews as negative sentiment. F1 score (6) is a metric that combines the trade-offs of precision and recall.

$$\text{Accuracy} = \frac{\text{True Positive (TP)} + \text{True Negative(TN)}}{\text{Total Number of Observations}}. \tag{3}$$

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive(FP)}}. \tag{4}$$

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative(FN)}}. \tag{5}$$

$$\text{F1 Score} = \frac{2 * \text{True Positive}}{2 * \text{True Positive} + \text{False Positive} + \text{False Negative}}. \tag{6}$$

Because our data is imbalanced with 82% positive and the costs of falsely predicting negative as positive (false positive) is markedly higher than incorrectly predicting positive as negative (false negative), predictive accuracy is not enough to measure the performance of a model. We want to use accuracy as the base metric to quickly evaluate the models. We want to classify sentiment with an accuracy better than the accuracy of an algorithm that simply assumes all reviews are positive, which would have an accuracy of 82% for our dataset. We will use the F1 score in comparing our 6 classifiers and deciding the overall goodness of the classifiers.

6.2 Supervised Machine Learning

We utilize the following supervised machine learning methods in Python Scikit-learn library: SGDClassifier for SVM model, LogisticRegression for LR model, and GradientBoostingClassifier for Gradient Boosting model. We fit all models using TF-IDF features of all words from all the review texts. Our models learn the vocabulary and frequency of each word in the TF-IDF training model. We keep hyperparameter tuning to a minimum with default parameters for all three supervised machine learning models.

A corpus, or collection of text reviews, contains 43,620 product reviews from 1000 different products from Amazon and serves as the dataset of study. The corpus undergoes text pre-processing and TF-IDF feature extraction. Stratified shuffled five-fold cross-validation is applied to the training procedure, allocating a fifth of the data for testing during each iteration. Each of the three machine learning classifiers is first trained and five-fold cross-validated during testing on the labeled data, generating the accuracy, precision, recall, and F1-score.

6.3 Lexicon-Based Learning

We fit all lexicon-based models using the pre-processed movie reviews. With VADER, we take in a movie review, perform initial pre-processing, including sentence extraction, unescaping HTML escape sequences, and expand contractions, and then tokenize the tokens. Since VADER rates each feature, or term, on a scale from -4 (extremely negative) to +4 (extremely positive) [10], the overall review sentiment is the summation of each sentiment score of words in the review. A summation of at least 0.1 is considered positive.

Similarly, with SentiWordNet, we take in a movie review, perform initial pre-processing, including sentence extraction, unescaping HTML escape sequences, and expand contractions, and then tokenize and POS tag the tokens. SentiWordNet rates each feature with one of three sentiment scores: a positive, a negative, and an objectivity (neutral) score [10]. We also sum the scores of individual words to arrive at an overall document score. A summation of at least 0.1 is considered positive.

With Pattern lexicon, we take in a movie review, perform pre-processing, including sentence extraction, unescaping HTML escape sequences, and expand contractions. Pattern lexicon computes the overall polarity and subjectivity score associated with a whole text document, not just individual words [10]. A threshold of 0.1 is recommended by Pattern lexicon to label a document as positive, and anything below it as negative. In our model, we adhere to these recommendations.

7 Results

In this section, we present our experimental results from all six different techniques to classify sentiment of our Amazon product reviews dataset. Table 3 shows our classification results on the testing dataset. The confusion matrix that classifies the reviews into positive and negative are also generated.

Table 3. Summary of experimental results of all evaluation parameters for all six classification algorithms

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Confusion Matrix [[TP, FN], [FP, TN]]
Pattern Lexicon	69	88	72	79	[[25748,10204], [3350,4318]]
VADER Lexicon	83	90	89	89	[[31828,4124], [3421, 4247]]
SentiWordNet Lexicon	80	88	88	88	[[31653,4299], [4238, 3430]]
Support Vector Machine	89	90	98	94	[[35153,799], [3808, 3860]]
Gradient Boosting	87	88	98	92	[[35120,832], [4894, 2774]]
Logistic Regression	90	91	97	94	[[34944,1008], [3549, 4119]]

The confusion matrix displays the number of positive (“1”) and negative (“0”) predictions acquired from the classification models in comparison with the actual counts in the dataset. Fig. 8 displays the confusion matrix for all the models.

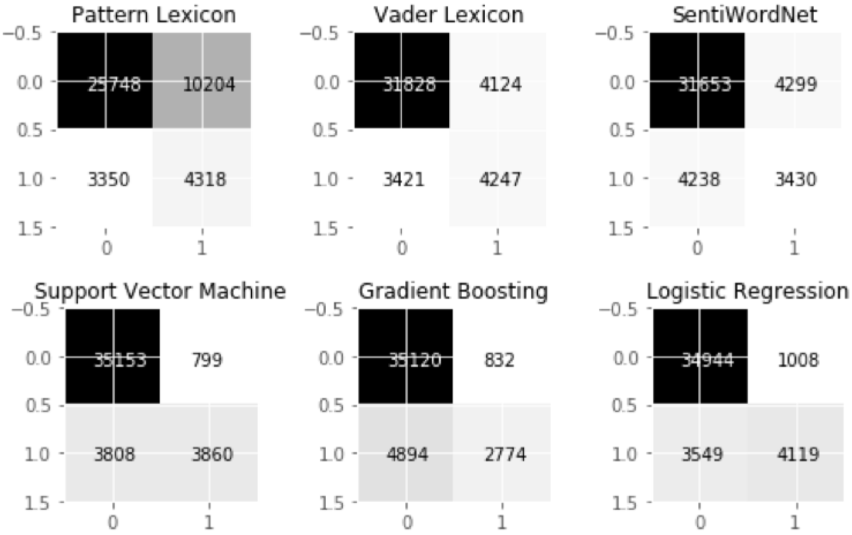


Fig. 8. Confusion matrices of all the models.

The comparison of accuracy of different classifiers on the reviews dataset indicates that machine learning algorithms outperform the lexicon-based techniques. Among the three machine learning algorithms, the LR algorithm outperforms the SVM and Gradient Boosting algorithms. Fig. 9 shows the model comparison.

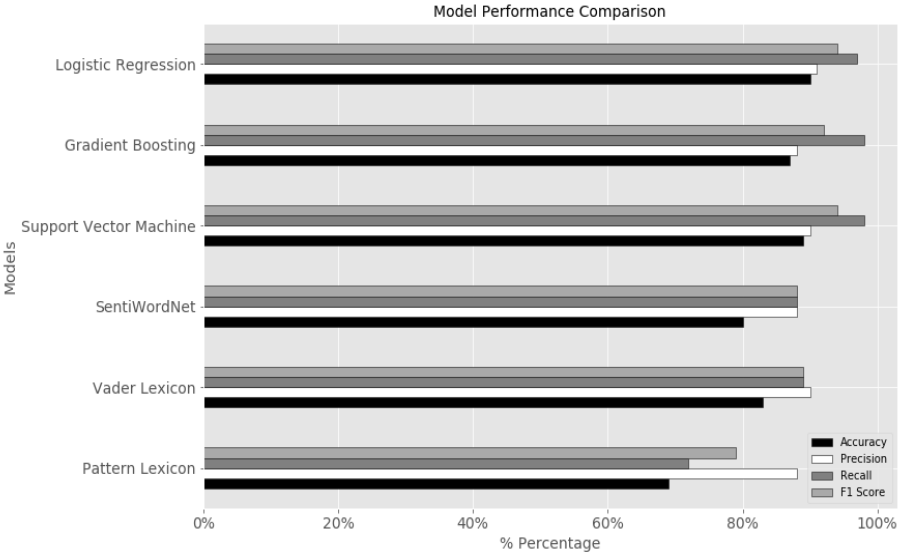


Fig. 9. Comparison of accuracy, precision, recall, and F1 score of six different classifiers.

The result shows that classifying negative reviews is more difficult than classifying positive reviews as our input dataset is imbalanced with 82% positive. This pattern holds true for both machine learning models and lexicon-based models. However, the lexicon-based models are better at classifying true negative reviews. The machine learning approaches may have performed well because the size of the input data is sufficiently large.

8 Ethics

Given the massive amount of user generated content on the internet, automated machine learning or lexicon-based methods could be applied in order to determine the overall sentiment for further analysis. This allows a small team of subject matter experts and data scientist to derive the sentiments from huge volumes of user generated content. Since, most of the text-based analytics are carried out on the fly, delivering precise and quality analysis is of the at most importance. For the ethical considerations we are referring to the ACM code of ethics and professional conduct [31]. In general terms, it is the professional responsibility of the professional to maintain high standards of professional competence, conduct, and ethical practice.

The cost of misclassification due to lack of training of supervised machine learning model or the failure to update the lexicon with relevant terms is substantial for an organization in terms of deriving accurate user feedbacks on a product. Therefore, delivering precise and quality analysis is of the at most importance. There are instances where new words or phrases which gets introduced might result in a misclassification as an unintended consequence. This could be avoided by simply retraining the supervised machine learning model with new training data or updating the lexicons with new terms along with its polarity. But the this might not be done in time due to other priorities or negligence.

The ethical considerations of this project are related to scraping product data and reviews from Amazon. We recognize that scraping the website for product reviews is carried out without getting prior permission from Amazon. From the license and access section of Amazon condition of use, it is clear the act of scraping of website and data mining becomes illegal when a third part benefits monetarily from utilizing the data collected [24]. Since our project does not have any commercial applications, it is well within the conditions of use laid out by Amazon and there are no ethical issues of concern.

9 Conclusions and Future Work

In this analysis, we compare six different sentiment classification methods, three supervised machine learning approach: SVM, Gradient Boosting, and LR algorithms and three lexicon-based techniques: VADER, Pattern, and SentiWordNet lexicons to analyze Amazon reviews datasets. We also carry out our experiments using Amazon product reviews with various NLP techniques including stopwords removal, word lemmatization, and TF-IDF vectorization. Our experimental approaches studied the accuracy, precision, recall, and F1 score of sentiment classification algorithms. Moreover, all our models were able to classify negative and positive reviews with relatively good accuracy and precision. The three supervised machine learning classifiers performed better than the lexicon-based classifiers on all the metrics. This could be attributed to the fact that the Lexicon based approaches uses a set list of words to identify positive or negative sentiment. Among the six models, the LR algorithm is the best classifier overall with the highest accuracy, precision, recall, and F1 score. Among the three lexicon-based models, the VADER lexicon model has the highest scores for all the metrics. Both groups of algorithms performed better in term of classifying positive class, and perform poorer in term of classifying negative class. The reason for this could have been due to certain stop words that might have a positive emotion associated with it and also due to the inherent class imbalance problem due to the dataset having a large proportion of reviews that have a positive sentiment. In conclusion, our machine learning results are slightly better compared to recent text sentiment machine learning works while our lexicon-based result are worse compared to recent similar lexicon-based works.

For future work, we wish to extend this work to include emoji in our texts. There has been an uptick in the usage of emoji in user-generated content. During preprocessing, all the emoji are removed from the texts. However, if emoji could be converted and processed then it could have improved the accuracy of the predictions. Another additional improvement would be to train using Word2vec, doc2vec, or pargraph2vec vectorization models instead of TF-IDF to improve our corpus. These models take longer texts into account compared to the words for TF-IDF.

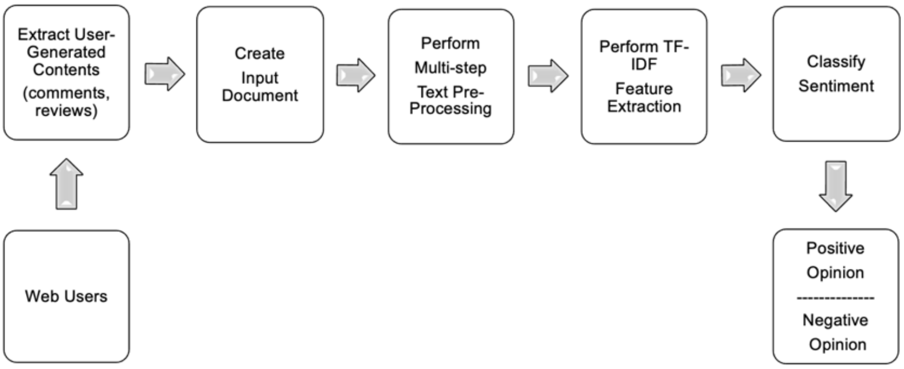
References

1. Guynn, Jessica. These are Facebook's secret rules for removing posts. USA Today. USA Today, Apr 24, 2018. Available at: https://www.usatoday.com/story/tech/news/2018/04/24/facebook-discloses-secret-guidelines-policing-content-introduces-appeals/544046002/
2. Xu, J., Xu, R., Lu, Q., Wang, X.: Coarse-to-fine sentence-level emotion classification based on the intra-sentence features and sentential context. In: Proceedings of CIKM-2012, poster, pp. 2455–2458 (2012)
3. Saqib, M.S., Kundi, F.M., Ahmad, S. Unsupervised Learning Method for Sorting Positive and Negative Reviews Using LSI (Latent Semantic Indexing) with Automatic Generated Queries. IJCSNS International Journal of Computer Science and Network Security, Vol.18 No. 1, 2018
4. Cambria E, Das D, Bandyopadhyay S, Feraco A. A practical guide to sentiment analysis. Switzerland: Springer, Cham; 2017.
5. Statista. Number of active Amazon customer accounts worldwide from 1st quarter 2013 to 1st quarter 2016 (in millions). Available at: https://www.statista.com/statistics/476196/number-of-active-amazon-customer-accounts-quarter/. Statista, 2018.
6. Reisinger, Don. Here's How Much Amazon Prime Customers Spend Per Year. Available at: http://fortune.com/2017/10/18/amazon-prime-customer-spending/. Fortune, 2018.
7. De Martino, B., Bobadilla-Suarez, S., Nouguchi, T., Sharot, T. & Love, B. C. Social information is integrated into value and confidence judgments according to its reliability. J. Neurosci. 37, 6066–6074 (2017). Available at: https://doi.org/10.1523/JNEUROSCI.3880-16.2017
8. Thorsten Joachims, Text Categorization with Suport Vector Machines: Learning with Many Relevant Features, Proceedings of the 10th European Conference on Machine Learning, p.137-142, April 21-23, 1998. Available at: https://doi.org/10.1007/BFb0026683.
9. Chen, T. & Guestrin, C. Xgboost: A scalable tree boosting system. In Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 785–794 (ACM, 2016). Available at: https://doi.org/10.1145/2939672.2939785.
10. Sarkar, Dipanjan. Text Analytics with Python. Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data. Apress, 2017.
11. Jeon J.H., Xia R., Liu Y. Sentence level emotion recognition based on decisions from subsentence segments. In ICASSP, Lyon, France (2011) 4940–4943
12. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment Classification Using Machine Learning Techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, (2002) 79–86, Philadelphia, PA.
13. Elmurngi, E., Gherbi, A., An empirical study on detecting fake reviews using machine learning techniques, In: Proceedings of the 2017 Seventh International Conference on Innovative Computing Technology (INTECH). (2017)
14. W. P. Ramadhan, S. T. M. T. Astri Novianty, and S. T. M. T. Casi Setianingsih, “Sentiment analysis using multinomial logistic regression,” 2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC), 2017.
15. Gautam, J. and Kumar, E. An Integrated and Improved Approach to Terms Weighting in Text Classification. International Journal of Computer Science Issues, 10, 1 (2013).
16. Martineau, J. and Finin, T. Delta TFIDF - an Improved Feature Space for Sentiment Analysis. Third AAAI International Conference on Weblogs and Social Media, San Jose CA (2009).
17. Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In Proceedings of EMNLP 2004, pp.412--418.

18. Charles E. Osgood, George J. Succi, and Percy H. Tannenbaum. The Measurement of Meaning. University of Illinois. 1957.
19. Lilleberg, J., Zhu, Y., Zhang, Y. Support vector machines and Word2vec for text classification with semantic features. 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC). (2015) 136-140.
20. Das, B. and Chakraborty, S. An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation. Cornell University Library, Computation and Language. (2018) arXiv:1806.06407. Available at: https://arxiv.org/abs/1806.06407v1.
21. Bhavitha, B.K., Rodrigues, A.P., Chiplunkar, N.N. Comparative Study of Machine Learning Techniques in Sentimental Analysis. In: Proceedings of International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE (2017) 216-221.
22. Vasileios Athanasiou and Manolis Maragoudakis. A Novel, Gradient Boosting Framework for Sentiment Analysis in Languages where NLP Resources Are Not Plentiful: A Case Study for Modern Greek. Algorithms 10 (2017), 34
23. Mishra, Prakhar. Natural Language Preprocessing – Hacker Noon. Hacker Noon. November 27, 2017. Available at: https://hackernoon.com/natural-language-preprocessing-630c28832fd1.
24. Rain, Callen. Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning. Swarthmore College, Department of Computer Science. 2016.
25. Amazon. Conditions of Use. Accessed November 05, 2018. Available at: https://www.amazon.com/gp/help/customer/display.html/?nodeId=508088.
26. Amazon Product Reviews. Accessed 2018. Available at: www.amazon.com.
27. JJ's World. Using Scrapy in Jupyter notebook. Accessed 2018. Available at: https://www.jitsejan.com/using-scrapy-in-jupyter-notebook.html.
28. Khan A., Baharudin B., Khan K. Sentiment Classification Using Sentence-level Lexical Based Semantic Orientation of Online Reviews. Trends in Applied Sciences Research, Vol. 6 (2011) 1141-1157.
29. Max Woolf. A Statistical Analysis of 1.2 Million Amazon Reviews. Available at: https://minimaxir.com/2014/06/reviewing-reviews/. 2014.
30. Janet Williams. How to Scrape Amazon Product Reviews using Python. Available at: https://www.promptcloud.com/blog/how-to-scrape-amazon-reviews-python. 2018.
31. 2018 Code, Draft 1. (2016, November 22). Retrieved December 11, 2018, from https://ethics.acm.org/2018-code-draft-1/

Appendix:

Fig. 10 shows the pipeline of our sentiment analysis.


Extract User-Generated Contents (comments, reviews)
Create Input Document
Perform Multi-step Text Pre-Processing
Perform TF-IDF Feature Extraction
Classify Sentiment
Web Users
Positive Opinion

Negative Opinion

Fig. 10. Pipeline of sentiment analysis