Sentiment Analysis

Major project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Aanchal Thakur (181285)

Under the supervision of

Dr. Rakesh Kanji

Department of Computer Science and Engineering and Information Technology

Jaypee University of Information Technology Waknaghat,
173234, Himachal Pradesh, India

CERTIFICATE

This is to ensure that the work set out in the project report entitled "Sentiment Analysis" presented to the Department of Computer Science and Engineering to fulfill part of the B-Tech award requirements for Computer Science and Engineering, is accurate. Jaypee University of Information Technology, Waknaghat accurate record of work done by Aanchal Thakur (181285), under the direction of Dr. Rakesh Kanji, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat, between January 2022 and May 2022.

Aanchal Thakur (181285)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Rakesh Kanji

Supervisor Name: Dr. Rakesh Kanji
Designation: Assistant Professor (SG)
Department of Computer Science and Engineering and Information Technology
Jaypee University of Information Technology

ACKNOWLEDGEMENT

This acknowledgment is intended to express gratitude to all those who contributed to the success of this project. First and foremost, we express our heartfelt gratitude to Almighty God for his divine blessing, which has enabled us to accomplish this task successfully.

Many thanks to our mentor, Dr. Rakesh Kanji of the CSE Department at Jaypee University of Information Technology, Wakhnaghat, for guiding us with his extensive knowledge and deep interest in the field of "Machine Learning". This project would not have been complete without his constant patience, expert guidance, constant motivation, constant watchfulness and enthusiasm, constructive criticism, and helpful advice.

Many thanks to Professor Dr. Samir Dev Gupta, Head of the Department of Engineering, for giving us the opportunity to successfully complete our work. We would also like to thank anyone who has assisted us in any way, directly or indirectly, in making this project a success. In this context, we want to express our gratitude to the many staff, both educators and non-educators, who provided us with valuable assistance and further our work.

Finally, we must acknowledge and accept our parents' enduring love and patience.

Aanchal Thakur (181285)

TABLE OF CONTENT III

,

Abstract

Due to the popularity of online marketplaces in recent decades, online sellers and merchants have asked their customers to offer their thoughts on the things they have purchased. As a result, millions of evaluations are generated every day, making it difficult for a potential customer to decide whether or not to purchase the goods. For product makers, analyzing such a large number of comments is difficult and time-consuming.

Sentiment Analysis is also known as Opinion Mining and it is the systematic identification, extraction, evaluation, and study of important situations and independent information using natural language analysis and textual analysis. From advertising to client service to clinical medicine, sentiment analysis is frequently employed in survey reviews and replies, communication and communication platforms, and health care technologies. Sentiment analysis has a great impact on social media it tells how people trust about your brand online. Mainly sentiment analysis is beneficial for agent monitoring, handling multiple customers online, identifying key emotional triggers, adaptive customer service, live insights basically for businesses to gain insights about how customer feel about recent topics and find out the urgent issues in real time before they spiral out of control. Not only for customers reviews online but it is also beneficial in order to extract important and useful information about the teaching methodology of a teacher and also towards the course curriculum. It helps to identify students learning curve, understand student requirements, foresee their performances and make effective changes in the teaching style.

Sentiment Analysis of product-based reviews is the goal of this project. The data for this project was gathered from "amazon.com" online product updates. We are excited to conduct an evaluation of the review data, which we believe will yield positive outcomes. In this project we have dataset on amazon food review which will distinguish the reviews of the customers online and predict which come under positive statement and which comes under negative statement on the basis of scoring.

1 Introduction

Sentiment Analysis is the analysis of natural language (NLP) to assess whether information is positive, negative, or neutral. It is often used in textual information to assist the organization in tracking product and product sentiments in customer feedback and better understanding consumer needs. Indigenous language analysis (NLP), machine learning, and other data analysis techniques are used in sensory analysis to evaluate and obtain objective quantitative discovery from informal and unprocessed text data.

Sentiment Analysis is important because it allows firms to understand their customers' feelings about their product. Organizations are able to make better and more informed decisions by automatically distinguishing social media platforms, ratings, and more.

"It's a boring film but the scenes were good enough."
The line provided is a review of the film which means "it" (film) is boring but the scenes were good. Understanding such feelings requires effort.
Therefore, Sentiment Analysis is a form of textual division based on the Sentimental Orientation (SO) concept it contains.
Emotional analysis of product reviews recently has become very popular in the text mines and integrated language research.

- To begin, the review must be stripped of evaluative phrases that reflect opinions.
- Second, the polarity, or SO, of the opinions must be established.
- Finally, the strength of a viewpoint, or its intensity, should be determined.

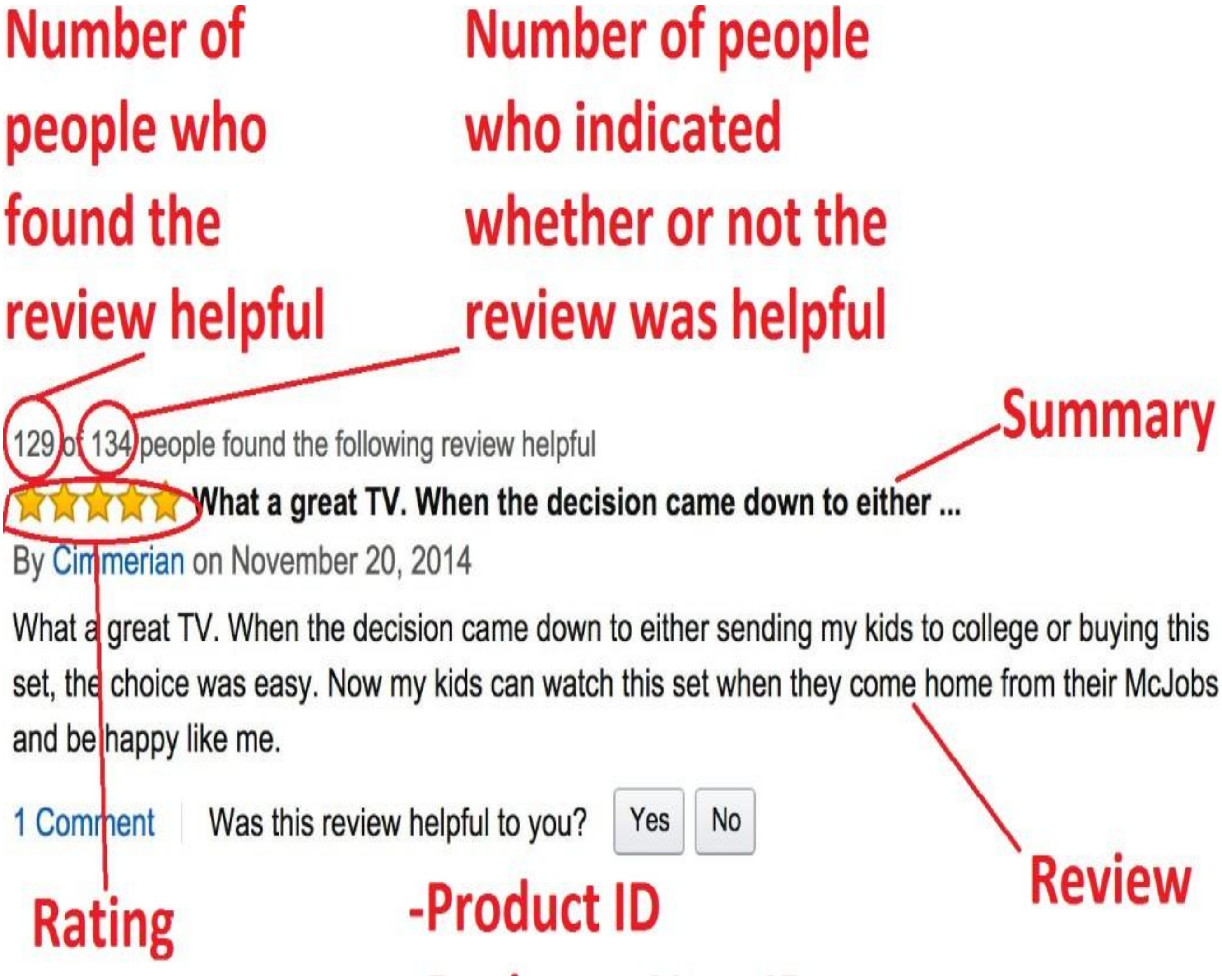
Finally, reviews are categorized according to emotional categories, such as Positive and Negative, based on SO of the ideas contained. Due to the popularity of online marketplaces in recent decades, online vendors and merchants now require their customers to provide feedback on the things they have purchased. Every day, millions of reviews of various products, services, and locations are generated over the Internet.

As a result, the Internet has become the most essential source of information and views about a product or service. However, as the quantity of product reviews available grows, it becomes more difficult for a potential customer to make an informed judgement about whether or not to purchase the product. Customers are further confused by differing perspectives about the same product on the one hand, and vague reviews on the other. The requirement to analyse these contents appears to be critical for all e-commerce enterprises in this case.

This analysis and classification is a computational study that seeks to solve this challenge by extracting subjective information such as views and sentiments from natural language texts. Natural language processing, text analysis, computational linguistics, and biometrics have all been utilized to address this issue. Machine learning approaches have gained popularity in the semantic and review analysis in recent years due to their simplicity and accuracy.

People use Amazon every day for online shopping since it is one of the e-commerce giants that allows them to browse thousands of evaluations left by other consumers about the things they want. These reviews contain vital information about a product, such as its features, quality, and suggestions, allowing buyers to comprehend practically every element. This is advantageous not just to consumers, but it also assists merchants who manufacture their own items in better understanding consumers and their demands.

The overall semantic of customer reviews is determined by classifying them into positive and negative sentiment in this research, which uses supervised techniques to determine the overall semantic of customer reviews.



Number of people who found the review helpful

Number of people who indicated whether or not the review was helpful

Summary

129 of 134 people found the following review helpful

★★★★★ **What a great TV. When the decision came down to either ...**

By Cimmerian on November 20, 2014

What a great TV. When the decision came down to either sending my kids to college or buying this set, the choice was easy. Now my kids can watch this set when they come home from their McJobs and be happy like me.

Rating

-Product ID

Review

2 Review of Literature

Sentiment Analysis is a rapidly expanding field of Natural Language Analysis, with studies ranging from document-class differentiation (Pang and Lee 2008) to learning the diversity of words and phrases (e.g., Hatzivassiloglou and McKeown 1997; Esuli and Sebastiani 2006). Because of the limited number of characters in tweets, dividing the sentiments of Twitter messages is similar to analysing sentence quality (e.g., Yu and Hatzivassiloglou 2003; Kim and Hovy 2004); however, the informal and special language used in tweets, as well as the nature of the microblogging domain, make Twitter Sentiment Analysis a very different task. How the features and procedures utilized in well-structured data will be moved to the microblogging arena is an outstanding subject.

A lot of emotive papers on Twitter have been published in the last year (Jansen et al. 2009; Pak and Paroubek 2010; O'Connor et al. 2010; Tumasjan et al. 2010; Bifet and Frank 2010; Barbosa and Feng 2010; Davidov, Tsur, and Rappoport 2010). Some researchers have started to experiment with speech difficulties, although the results are still a bit hazy. There has been minimal research on the utility of current sensory resources created in non-microblogging data, yet frequent elements in microblogging (e.g., emoticons) are also common.

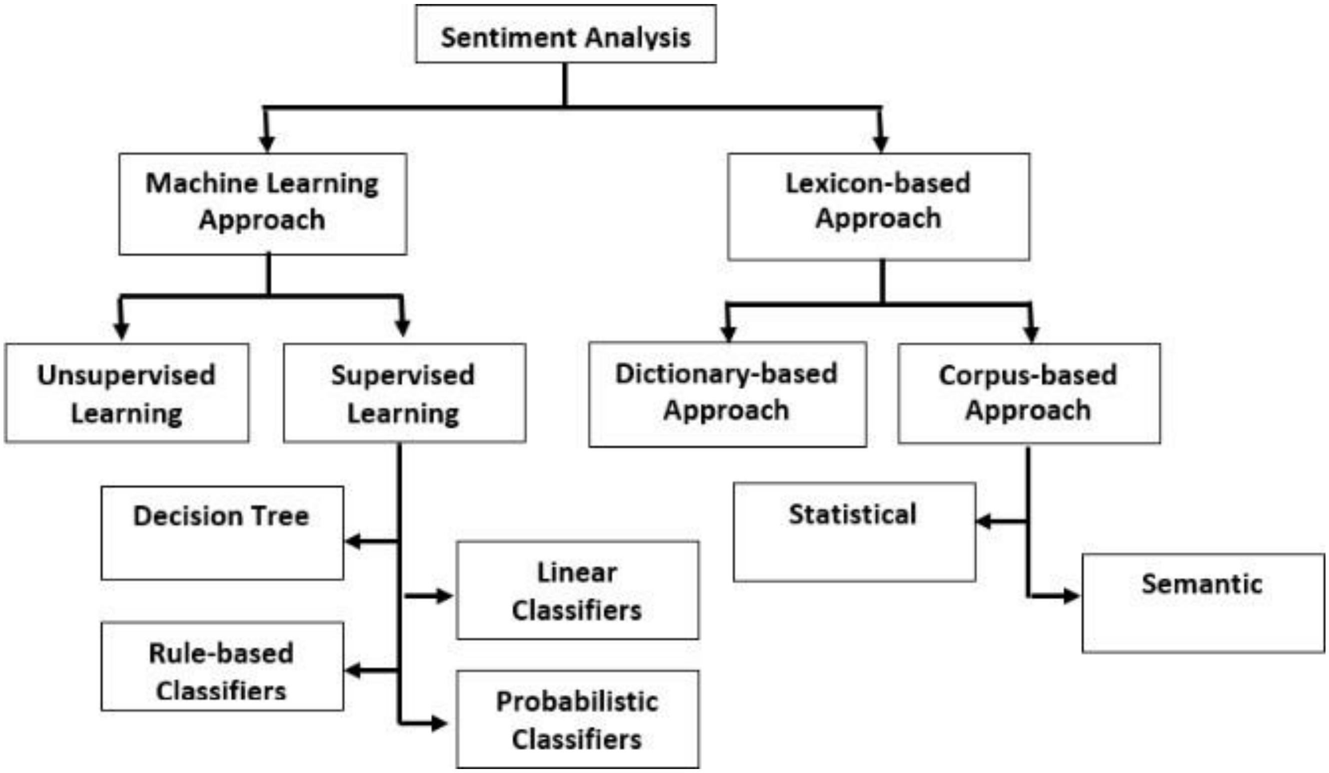
Researchers are also looking into different methods for automatically collecting training data. To convey their training data, several researchers used icons (Pak and Paroubek 2010; Bifet and Frank 2010). (Barbosa and Feng 2010) acquire training data using existing Twitter emotional sites. (Davidov, Tsur, and Rappoport 2010) utilize hashtags to construct training data as well, but they only test for emotional vs. non-emotional isolation rather than 3-way categorization, as we do.
To achieve the best result, we use WEKA and use the following Machine Learning algorithms for the second classification:

- K-Means Clustering
- Support Vector Machine
- Logistic Regression
- K Nearest Neighbors
- Naive Bayes
- Rule Based Classifiers

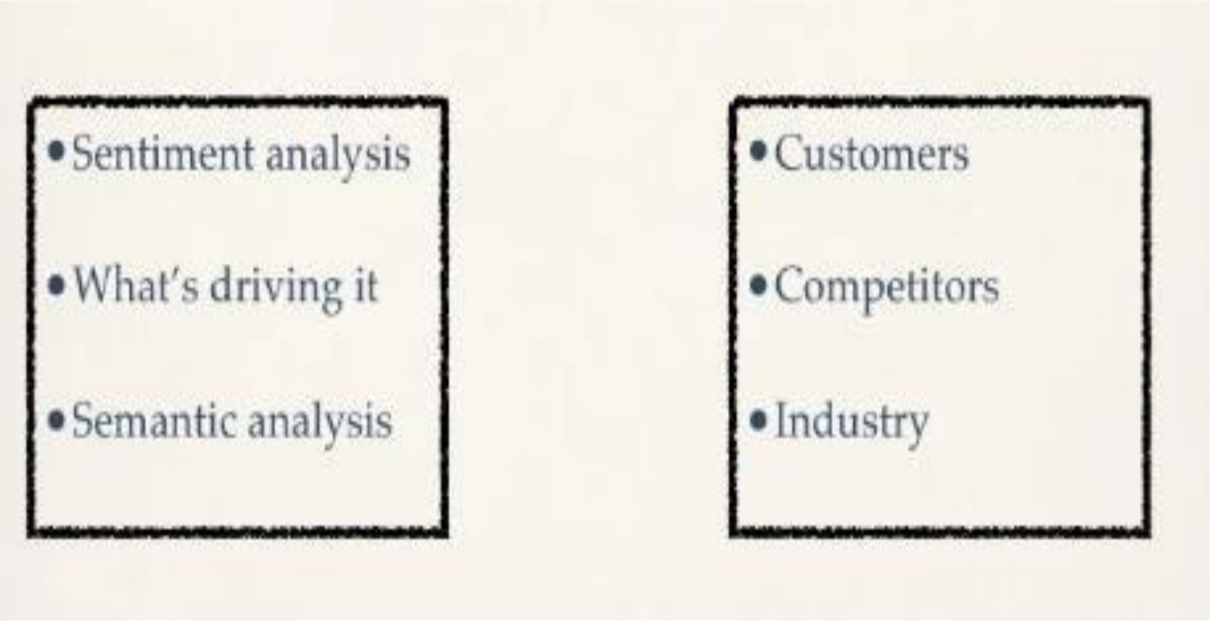
Approach

The main approach used in sentiment analysis is basically comparison of different machine learning technologies in order to get the best result out of these machines learning models and finding out the reviews based on their score. Reviews with score 1 and 2 is considered a negative review, review with score 3 is considered as neutral, review with score 4 and 5 are considered as positive review. Using the concepts of wordclouds , it becomes easy to identify which review is negative, positive or neutral.

After knowing the accuracy of model, then mainly focused on dataset in order to find out the number of positive, negative and neutral reviews which makes a company or organization to do particular analysis on the product or reviews. Positive reviews give a good feedback about the product or review.



Objective of the Project



- Sentiment analysis
- What's driving it
- Semantic analysis

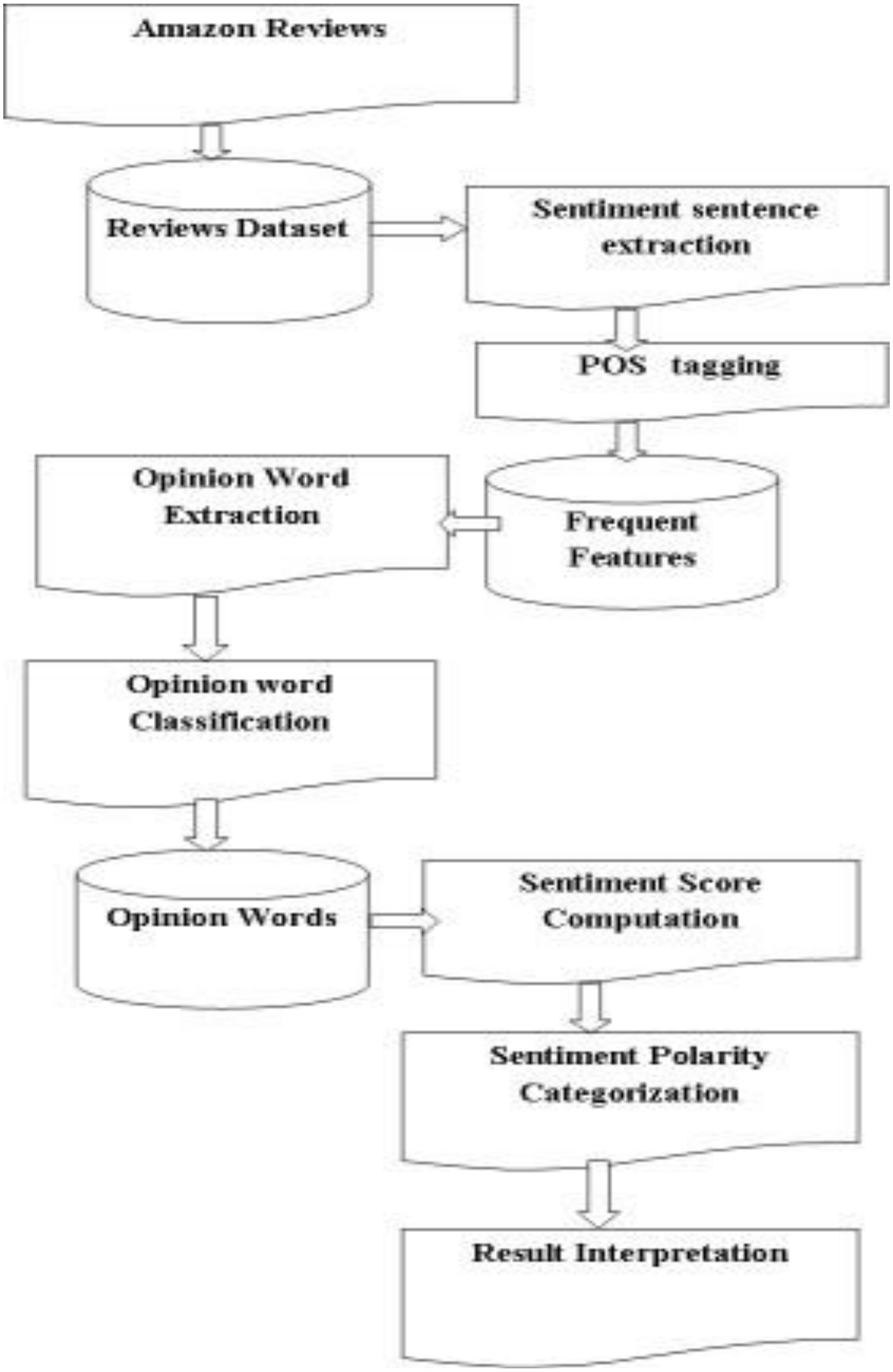
- Customers
- Competitors
- Industry

- Scraping product reviews from numerous websites, particularly amazon.com, that feature various products.

- Analyze and classify review data.

- Investigate the sentiment of the dataset at the document level (review level).

- Opinion sentiment is classified or categorise into-
    - Positive
    - Negative


Amazon Reviews
Reviews Dataset
Sentiment sentence extraction
POS tagging
Frequent Features
Opinion Word Extraction
Opinion word Classification
Opinion Words
Sentiment Score Computation
Sentiment Polarity Categorization
Result Interpretation

3 System Design

Hardware Requirements:

- Processor Intel Core i5/i7
- RAM of at least 8 GB is required.
- A minimum of 60 GB of usable hard disc space is required.

Software Requirements:

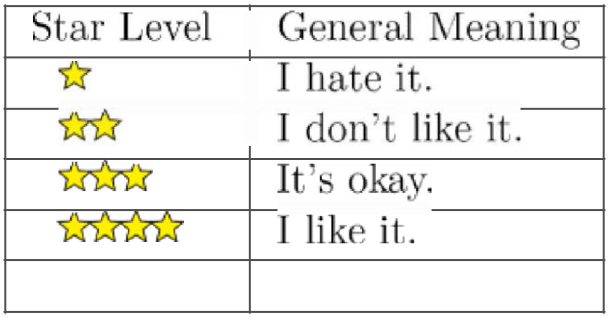
- Python 3.x
- Google colab
- NLTK Toolkit

Data Information:

- Updates from Amazon are included in the Amazon set data update. As of March 2013, the data covered 18 years and included 35 million reviews. Product and user information, ratings, and reviews of blank documents are all updated. Please see the following page for more information: Hidden objects and hidden topics: understanding measurement scores in review text, J. McAuley and J. Leskovec. RecSys, Inc., 2013.

- Xiang Zhang (xiang.zhang@nyu.edu) compiled a review of Amazon's comprehensive results database from the database above. It serves as a guide for separating text in the following paper: Yann LeCun, Xiang Zhang, Junbo Zhao Text Separating Letters Using Text Transforming Networks Neural 28 Information Processing System Development (NIPS 2015).

- Amazon updates the whole results database, which was constructed by capturing 200,000 samples for each review point from 1 to 5 on a scale of 1 to 5. There are 1,000,000 samples in all.



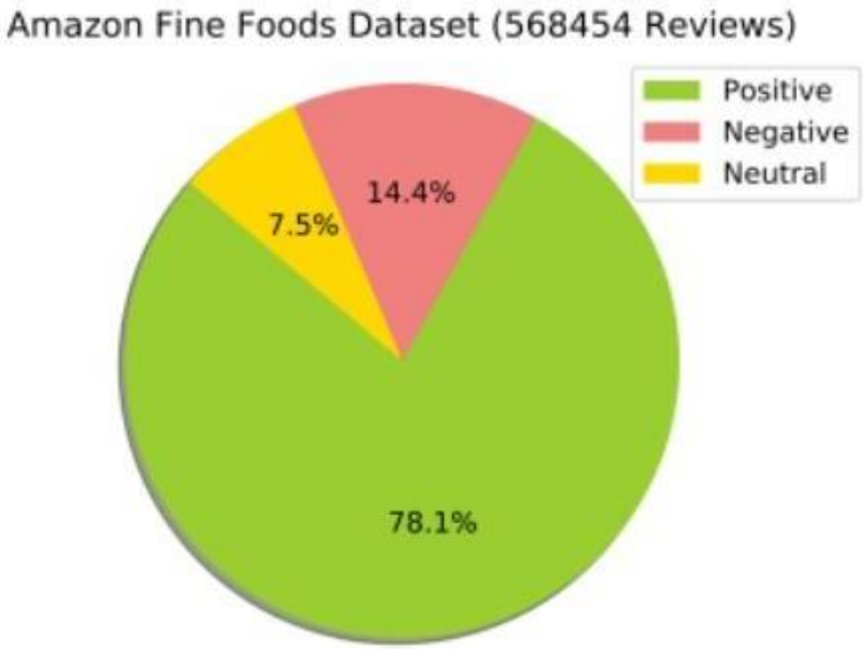
Star Level	General Meaning
★	I hate it.
★★	I don’t like it.
★★★	It’s okay.
★★★★	I like it.

Books	revie\•/s 22,507,155 revie\• s}	meta dala i”2,37D, 585products}	innage features
Electronics	revie\'is {7,824,482 reviews)	metadala {498.196 products,	image features
klovies nnd TV	revie\•/s{4,60?,g47 reviews)	metadala i”208.321 products]	image features
CDs and Vin'yl	revie\•/s 3,749,g04 reviews)	metadala i”492.799 products,	image features
Clothing, Shoes and Je\'Weir'	revie\'is{5,74i3,920 reviews)	metadala i”1,503.384 prod u cts}	innage fealures
Homa and Kitchan	revie\'is {4,253,926 reviews)	metadala {436.988 products,	image features
Kindle Store	revie\'is {3,205,467 reviews)	metadala i”434.702 products]	image features
Sports and Outdoors	revie\•/s'3,268,695 reviews)	metadala i”S32.197 products]	image features
Cell Phones nnd Accessories	revie\'is 3,44?,249 reviews)	metadala i”346.793 products,	image features
Health nnd Personal Care	revie\'is {2,982,326 reviews)	metadala i263.052 products,	image features
Toys and Games	revie\•/s'2,252,771 reviews)	metadala i”336.g7Z products]	image features
Video Games	revie\'is 1,324,753 reviews)	metadala i”S0,955 products”i	image features
Tool snnd Hons e Impro'7e m ent	revie\'is{1,926,047 reviews)	metadala i”2B9.120 products]	image fealures
Beaut,'	revie\'is {2,023,070 reviews)	metadala {259.204 products,	image features
Apps for Android	revie\•/s'2,638,173 reviews)	metadala i”o1.551 products”i	image features
Office Products	revie\'is 1,243,186 reviews)	metadala i”134.838 products,	image features
Pet Supplies	revie\'is{1,235,316 reviews)	metadala i”110.707 products]	image fealures
Automotive	revie\'is {1,373,768 reviews)	metadala {331.090 products,	image features
Grocery and Gourmet Food	revie\'is 1,29?,156 reviews)	metadala i”171.7s0 products,	image features
Patio. Law.'n and Garden	revie\'is {993,490 re'7iev.'s”i	metadala i”109.094 products]	image fealures
Bab'y	revie\'is {915,44s ra'7iev.'s i	metadala I”71,317 products i	image features
Digital Music	revie\•/s{836.0go re'ziev.'s”i	metadala i”279.£:99 products]	image features
klusical Instruments	revie\'is 500,17s re'7iev.'s”i	metadala i”84,9D1 products”i	image features
Amazon Instant Video	revie\'is {583,933 re'7iev.'s”i	metadala i”30,648 products”i	image fealures

4 Performance Analysis

About Data set

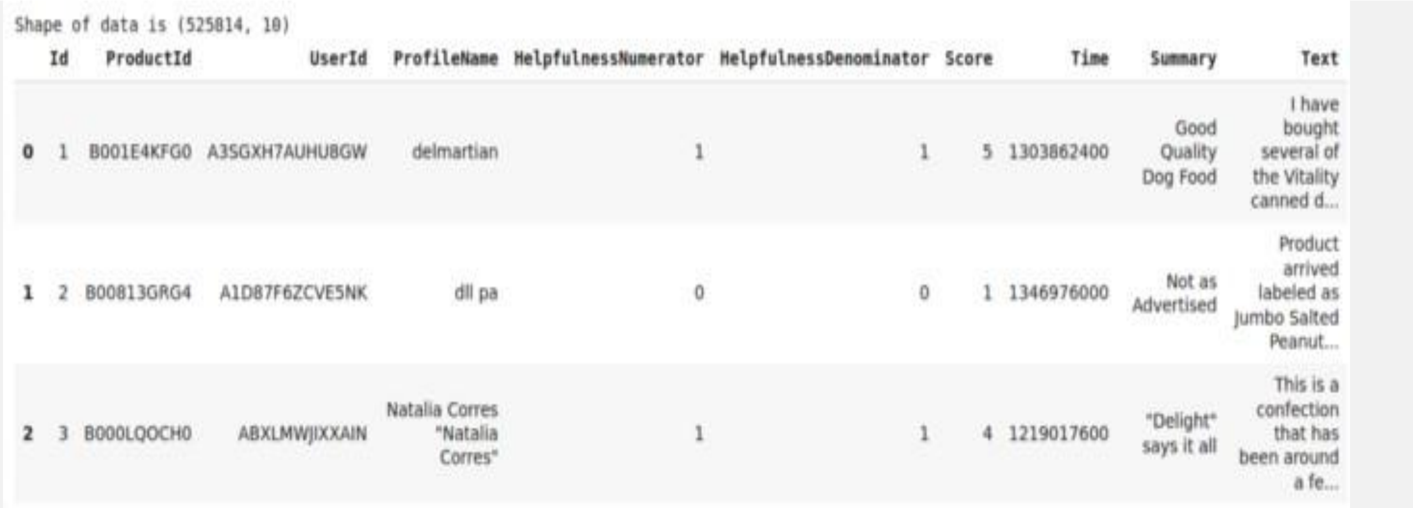
The data set includes 568,454 reviews of fine foods from Amazon over a 10-year span, with 568,454 reviews up until October 2012. Ratings, product and user information, and a plain text review are all included in reviews. It also contains reviews from all of Amazon's other categories.



The columns are as follows:

1. Product Id: The product's unique identification.
2. User Id: The user's unique identifier.
3. User Profile Name: The user's profile name.
4. Helpfulness Numerator:Number of people who found the review helpful as a numerator

5. Helpfulness Denominator: The number of people who said whether or not they found the review useful.
6. Score: A scale of 1 to 5 is used to assign a score.
7. Time: Hashcode
8. Summary:The following is a summary of the review.
9. Text: Review



```
Shape of data is (525814, 10)
```

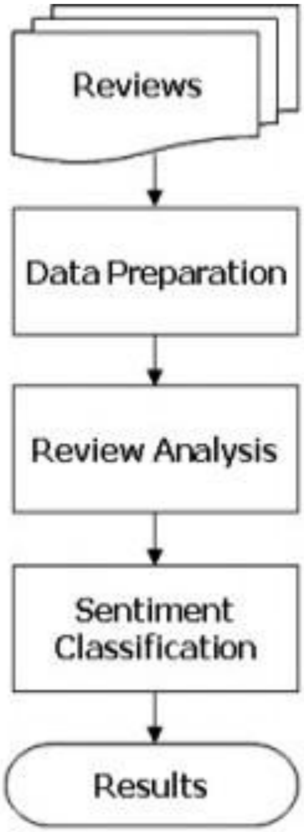
	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...

Main Aim

If you get a review, pick whether it's positive (4 or 5) or bad (5 or 6). (1 or 2 rating).

How can you tell if a review is positive or negative?

We can use the terms "School" and "Rating." A good review is one with four or five stars. A negative rating of 1 or 2 is possible. The rating of 3 is considered impartial, and such evaluations are not taken into account in our research. This is a hypothetical and tested approach for assessing a review's polarity (positive vs. unfavourable).



Background

Classification of Sentiment and analysis

Because e-commerce websites allow customers to post feedback on various products, electronic commerce is growing increasingly popular. Customers generate millions of evaluations every day, making it challenging for product manufacturers to keep track of client feedback on their products. To get usable information from a big set of data, it is necessary to classify such large and complex data. Such issues can be addressed using classification algorithms. The practice of classifying data into groups or classes based on common characteristics is known as classification (Pandey et al. 2016; Rain 2013). When large datasets are employed, the capacity to automate the classification process is a significant worry for businesses.

Sentiment analysis, often known as opinion mining, is an NLP task that involves finding and extracting subjective information from text sources. The goal of sentiment classification is to assess user reviews and categories them as positive or negative, without requiring the machine to fully comprehend the semantics of each phrase or document.

This isn't accomplished simply by categorizing words as positive or negative. There are several difficulties to overcome. It is not always possible to classify words and sentences based on their past positive or negative polarity. The word "wonderful," for example, has a prior positive polarity, but when combined with a negative word such as "not," the meaning might radically shift.

Different industries, such as movie reviews, travel location reviews, and product reviews, have attempted sentiment classification (Liu et al. 2007; Pang et al. 2009; Ye et al. 2009). The two most common approaches for sentiment categorization are lexicon-based methods and machine learning methods.

Analysis using Machine learning methods

In the field of machine learning, there have been a great number of articles published. Machine learning algorithms are one of the most widely utilized methodologies for sentiment classification. This section tries to address a few of them. Tom Mitchell (1997) presented one of the first definitions of machine learning in his book Machine Learning, which is as follows:

"With respect to some class of tasks T and performance measure P, a computer programmer is said to learn from experience E if its performance at tasks in T, as measured by P, improves with experience E."

Machine learning tries to create an algorithm that uses example data to optimize the system's performance. Machine learning gives a sentiment analysis method that consists of two basic components. The first stage is to "learn" the model using the training data, and the second step is to use the trained model to classify previously unseen data. There are several categories of machine learning algorithms:

1. Supervised learning

2. Semi-supervised learning

3. Unsupervised learning

1. The process of the algorithm learning from the training data is referred to as supervised learning, and it can be compared to a teacher supervising the learning process of their students (Brownlee 2016). The supervisor is educating the algorithm what conclusions it should produce as an output in some way. As a result, both input and output data are provided. It is also necessary for the training data to be labelled. The output will be more exact if the classifier receives more labelled data. The purpose of this method is for the algorithm to accurately forecast output for fresh input data. The supervisor can direct the algorithm back to the correct path if the output differs significantly from the predicted outcome.

When dealing with supervised people, however, there are several difficulties. As long as tagged data is available, supervised learning works properly. This means that if the machine encounters data it hasn't seen before, it will either classify it incorrectly or eliminate it because it hasn't "learned" how to identify it.

2 Unlike supervised learning, unsupervised learning is taught on unlabeled data with no associated output. The algorithm should figure out the data set's underlying structure on its own. This implies it must find comparable patterns in the data in order to decide the output without knowing the correct answers. Clustering is one of the most essential strategies in unsupervised learning challenges. Clustering is the process of finding similar groupings of data in a dataset.

Sentiment words and phrases are typically utilized for unsupervised sentiment classification. This indicates that a review's classification is determined by the average semantic orientation of its sentences . This is understandable because sentiment terms are frequently the most important feature in sentiment classification (Berk 2016). Turney's research incorporated this technique.

3. Finally, semi-supervised learning, which combines the advantages of both supervised and unsupervised learning, refers to problems in which only a small portion of the training data set is labelled and the remainder is unlabeled. This is useful when obtaining data is inexpensive but labelling it is time consuming and costly. This method is advantageous in theory and practise since having a large amount of unlabeled data throughout the training process tends to improve the accuracy of the final model while requiring much less time and money to construct (Zhu 2005). Dasgupta and Vincent Ng (2009) employed 2000 unlabeled papers and 50 randomly labelled documents in their semi-supervised learning experiment.

Search Vector Machine

SVMs (support vector machines) are supervised learning methods for handling sentiment categorization problems. This method uses a decision plane to place labelled training data, and then an algorithm to generate an optimum hyperplane that divides the data into groups or classes. The best hyperplane, as shown in figure 1, is the one that separates the classes by the greatest margin. This is accomplished by selecting a hyperplane with the greatest distance from the nearest data on each class.

Naive Bayes

Another machine learning technique noted for its power despite its simplicity is Nave Bayes. This classifier is based on the Bayes theorem and assumes that the characteristics (which in text classification are usually words) are mutually independent. Despite the fact that this assumption is incorrect (since the sequence of the words matters in some circumstances), Nave Bayes classifiers have proven to be quite effective (Rish 2001). Before applying the Nave Bayes model to text classification problems, the initial step should be feature extraction.

Feature extraction

The input (in this case text data) must be processed because machine learning algorithms only deal with fixed-length vectors of integers rather than raw text. The Bag of Words Model of Text, which is a widely used approach of feature extraction, is used to translate the texts into features. The method works by separating the words that appear in the training data set into various bags, each with its own number. This number represents the number of times each word appears in the document. Figure 2 shows a simple instance of the Bag of Words paradigm. Because the position of the words in the document is discarded, the model is called a bag of words.

Classification of Sentiments using Lexicon based methods

Another unsupervised method that depends on word and phrase annotation is the lexicon-based method. This method uses a vocabulary of sentiment words and phrases to compute a sentiment score for each text (Taboada et al. 2011). The easiest strategy for determining the sentiment of a review document in lexicon-based methodologies is to employ a count-based approach. We can assign the polarity of the review if we have a text and a lexical resource with positive and negative annotations of words and phrases. This signifies that the polarity of the evaluation is positive if the number of positive words exceeds the number of negative terms. When there are more negative than positive sentiment words in a text, the overall sentiment is negative.

However, classifying sentiments just on the basis of sentiment words and phrases is insufficient. Sentiment lexicon is necessary but not sufficient for sentiment analysis.
Liu raises the following concerns with this method:

1.In different fields, positive and negative sentiment terms may have distinct meanings. For example, the word "suck" has a bad connotation but can also have a pleasant connotation. "This camera sucks," for example, expresses a negative view, but "This vacuum cleaner truly sucks." offers a favorable opinion.

2.Sarcastic sentences, even when they contain sentiment words, are frequently difficult to deal with. "What a fantastic automobile!" for example. In two days, it ceased operating." Despite the fact that the word "good" is a positive term, this is a negative opinion.

3.It's possible that a phrase or opinion lacks a sentiment word, making it difficult for the machine to compute a sentiment score for the opinion." This washer uses a lot of water" has no sentiment words but it implies a negative opinion about the washer.

One of the earliest investigations on this strategy was Hu and Liu's (2004) work. They proposed a sentiment categorization method based on a vocabulary. They classified sentences at the sentence level because they considered that a review normally contains some sentences with unfavourable comments and some sentences with positive opinions. They determine if each sentence expresses a good or negative viewpoint, and then compile a final review summary.

Related work

Sentiment analysis has gotten a lot of interest in recent years thanks to the proliferation of online reviews. As a result, numerous studies have been conducted in this field. Some of the most relevant research works to this thesis are offered in this section.

SVM was tested for text classification by Joachims (1998), who found that it performed well in all experiments with lower error levels than other classification methods.

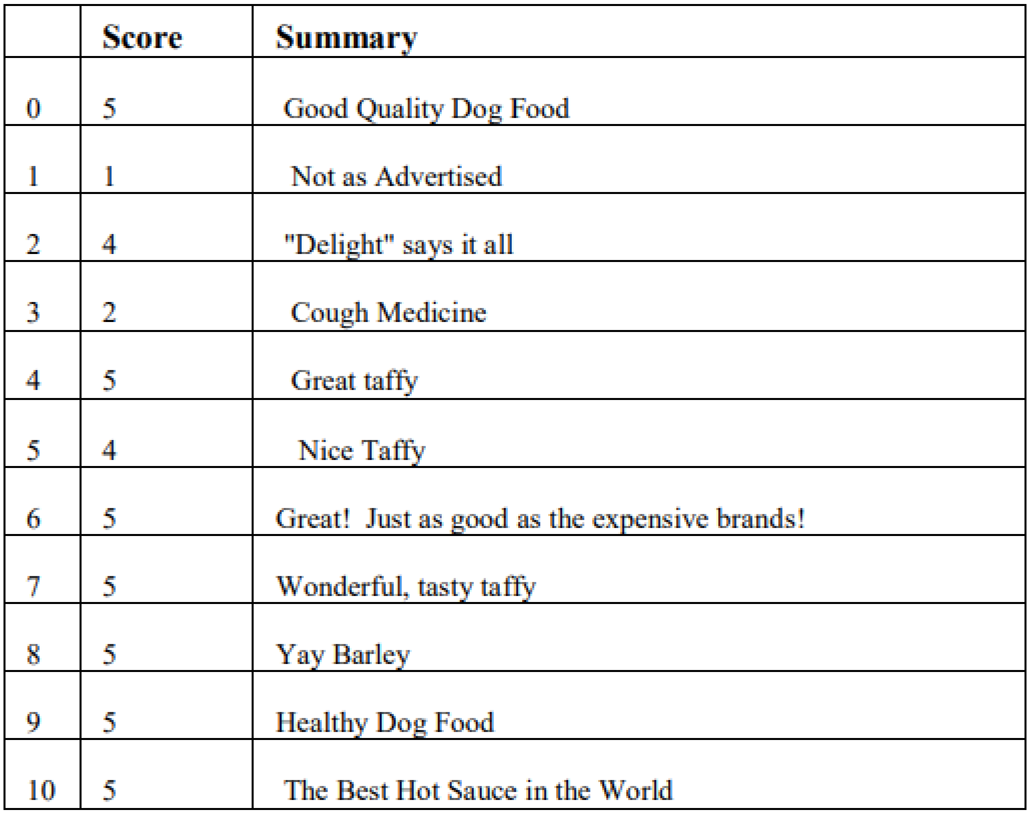
With the use of SVM and Nave Bayes and maximum entropy classification, Pang, Lee, and Vaithyanathan (2002) attempted supervised learning for classifying movie reviews into two classes, positive and negative. In terms of precision, all three approaches performed admirably. They experimented with numerous features and discovered that when a bag of words was utilised as a feature in the classifiers, the machine learning algorithms performed better.

Three supervised machine learning algorithms, Nave Bayes, SVM, and N-gram model, were tested using internet evaluations about various tourism sites throughout the world in a recent survey done by Ye et al. (2009). They discovered in this study that properly-trained machine learning algorithms perform exceptionally well for classification of vacation destination reviews in terms of accuracy. Furthermore, they revealed that the SVM and N-gram models produced superior results than the Nave Bayes technique. However, increasing the quantity of training data sets lowered the difference between the algorithms dramatically.

Chaovalit and Zhou (2005) compared the supervised machine learning technique to Semantic orientation, an unsupervised approach to movie assessment, and concluded that the supervised approach was more reliable.

Naive Bayes and SVM are two of the most often utilized algorithms in sentiment classification problems, according to various studies (Joachims 1998; Pang et al. 2002; Ye et al. 2009). As a result, this thesis attempts to use supervised machine learning methods such as Nave Bayes and SVM to Amazon's cosmetic product reviews.

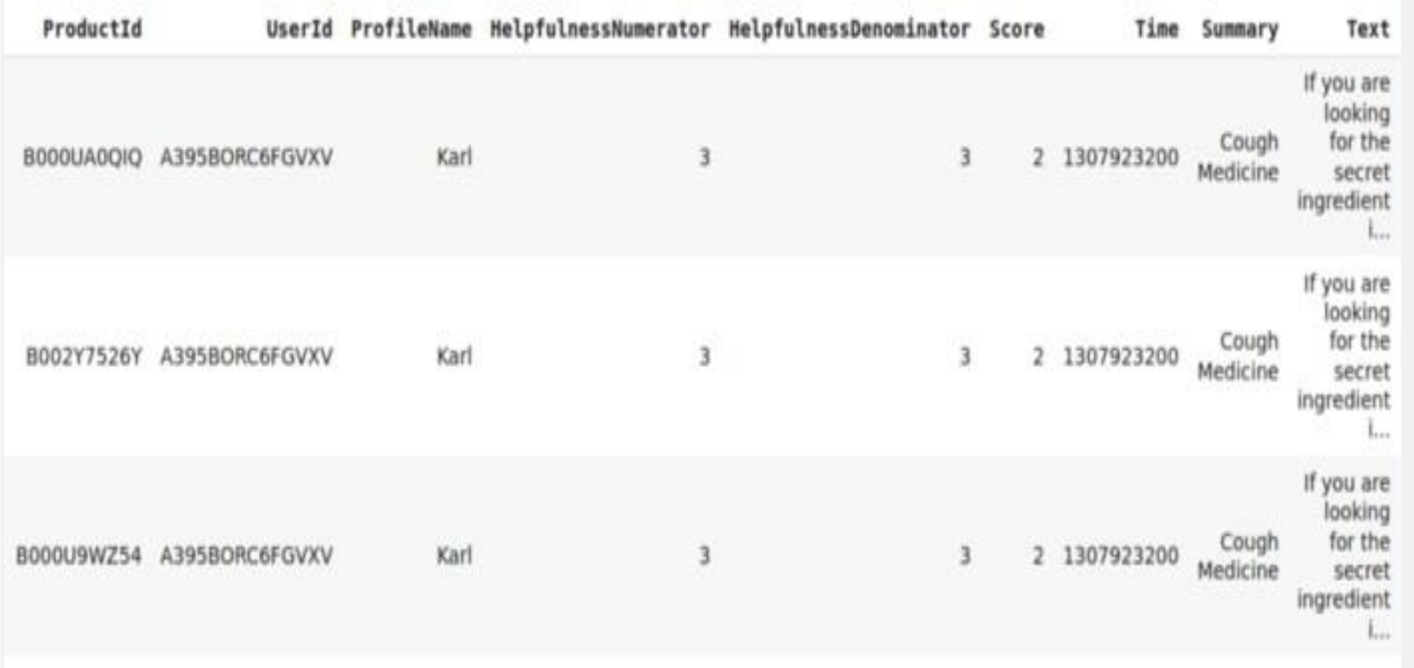
Exploratory Data Analysis



	Score	Summary
0	5	Good Quality Dog Food
1	1	Not as Advertised
2	4	"Delight" says it all
3	2	Cough Medicine
4	5	Great taffy
5	4	Nice Taffy
6	5	Great! Just as good as the expensive brands!
7	5	Wonderful, tasty taffy
8	5	Yay Barley
9	5	Healthy Dog Food
10	5	The Best Hot Sauce in the World

Basic Preprocessing

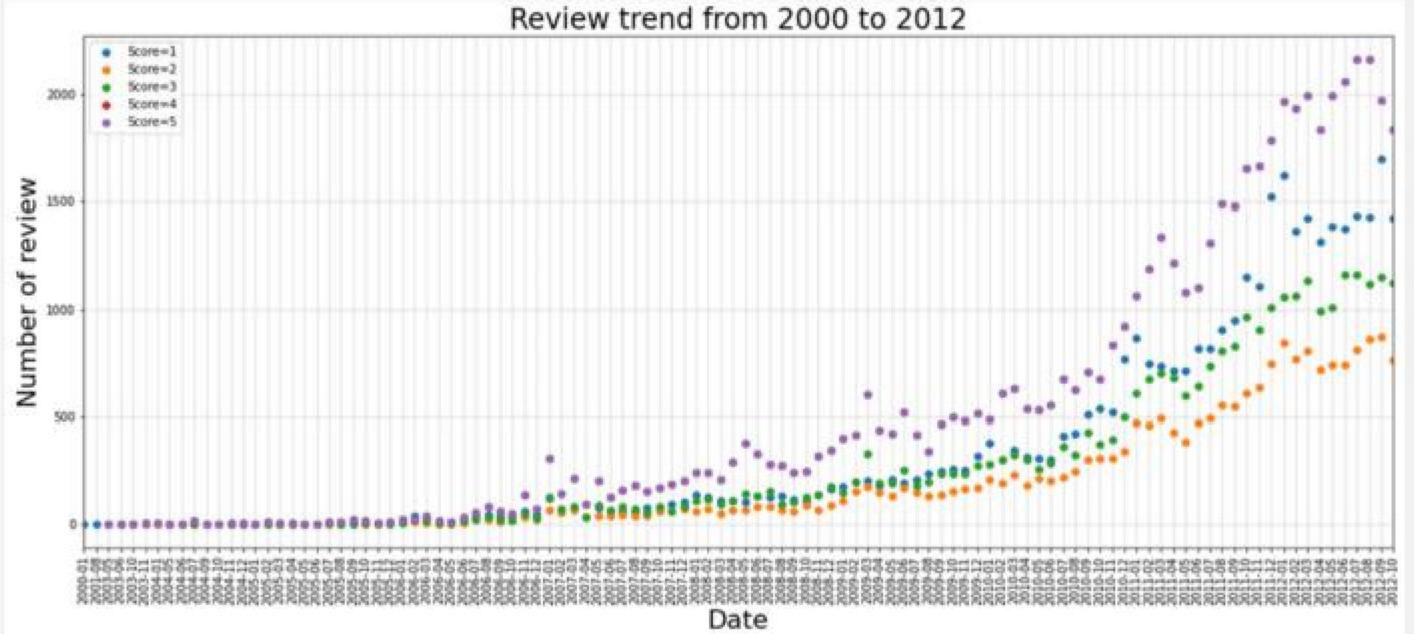
We examined any missing values as a preventive approach for basic data. We are fortunate in that there are no shortages. Then we'll look for duplicate entries. During our investigation, we discovered that the same update is delivered to the same user at the same time in various products. It doesn't make any sense. As a result, we'll just preserve the first one and delete a few duplicates.



	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
	B002Y7526Y	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
	B000U9WZ54	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...

Our data points have now been decreased by around 69 percent.

Examining the review pattern

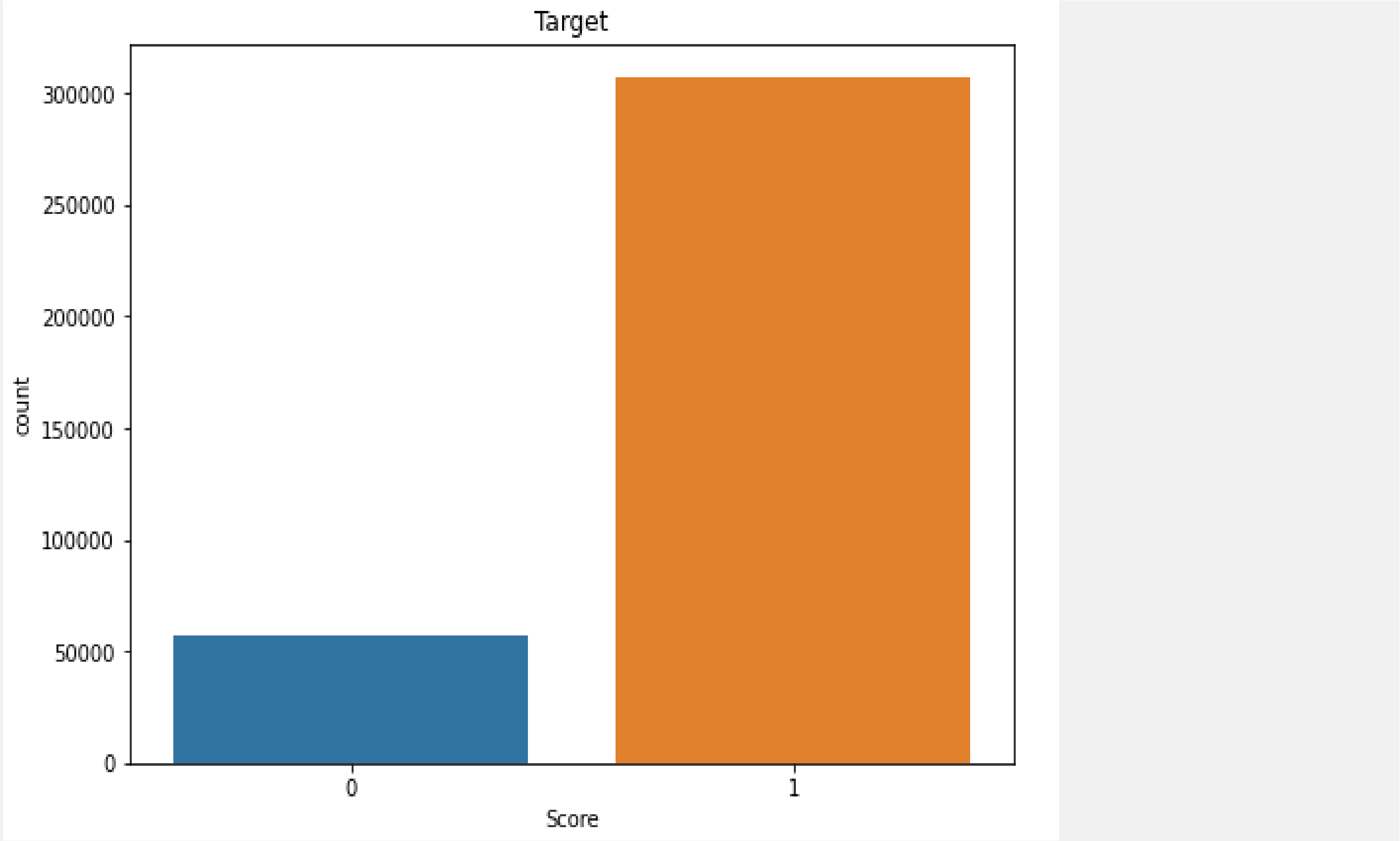


review pattern

The number of reviews has been steady between 2001 and 2006. However, the number of changes began to rise after that. The number of updates with a 5-star rating was the greatest of all. Perhaps the vendor is unfairly promoted by the unconfirmed accounts' bogus updates. Another factor could be the rise in the number of registered users.

Examining the target variable

As previously stated, we will classify any data points with a rating of 3 or higher as positive and those with a rating of 3 or lower as negative. The rest of the points will be ignored.



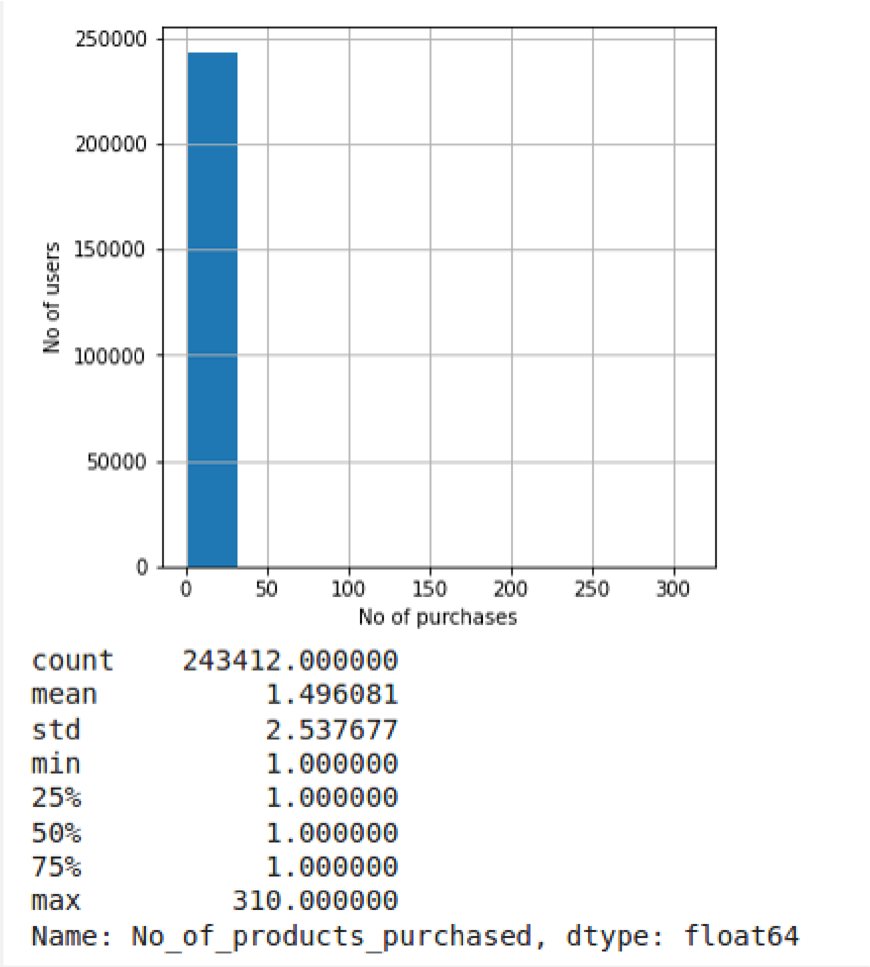
target

Observation: It is clear that the classification data set is imbalanced. As a result, we are unable to use accuracy as a metric. So let's begin with AUC (Area under ROC curve).

Why isn't accuracy possible with unbalanced datasets?

Consider the following scenario: we have a data set that is uneven. Consider credit card fraud detection, in which 98 percent of the points are non-fraud and the remaining 2% are fraud. Even if we foresee all of the points as non-fraud, we will achieve 98 percent accuracy in such cases. This is not the case, however. As a result, accuracy isn't a metric that may be employed

Analyzing User behavior



```
count    243412.000000
mean          1.496081
std           2.537677
min           1.000000
25%           1.000000
50%           1.000000
75%           1.000000
max         310.000000
Name: No_of_products_purchased, dtype: float64
```

- After looking at the number of things that customers brought, we observed that the majority of them only brought one.
- Because the helpfulness numerator is the number of users who found the review helpful, and the helpfulness denominator is the number of users who specified whether they found the review useful or not, the helpfulness denominator should always be greater than the numerator. In a few cases, however, this is not the case. As a result, those points are no longer available.

After preprocessing, the data was reduced from 568454 to 364162. Approximately 64% of the data is still available. Let us now go on to the heart of the topic. Data from the review is being processed.

Text data preprocessing

Before we can begin with the forecast analysis and modelling, we need to pre-process the text data. As a result, in the processing step, we do the following in the following order::-

- Starting with the Html tags, remove them.
- Remove all punctuation and a small number of special characters such as or. or #,!, and so on.
- Check to see if the term is alpha-numeric and made up of English letters.
- Change the capitalization of the word to lowercase.
- Finally, stopwords must be removed.

Train test split

After we've completed the pre-processing, we'll divide our data into train and test groups. Because changes in time can effect updates, we'll make the distinction after filtering the data by time.

Vectorizing text data

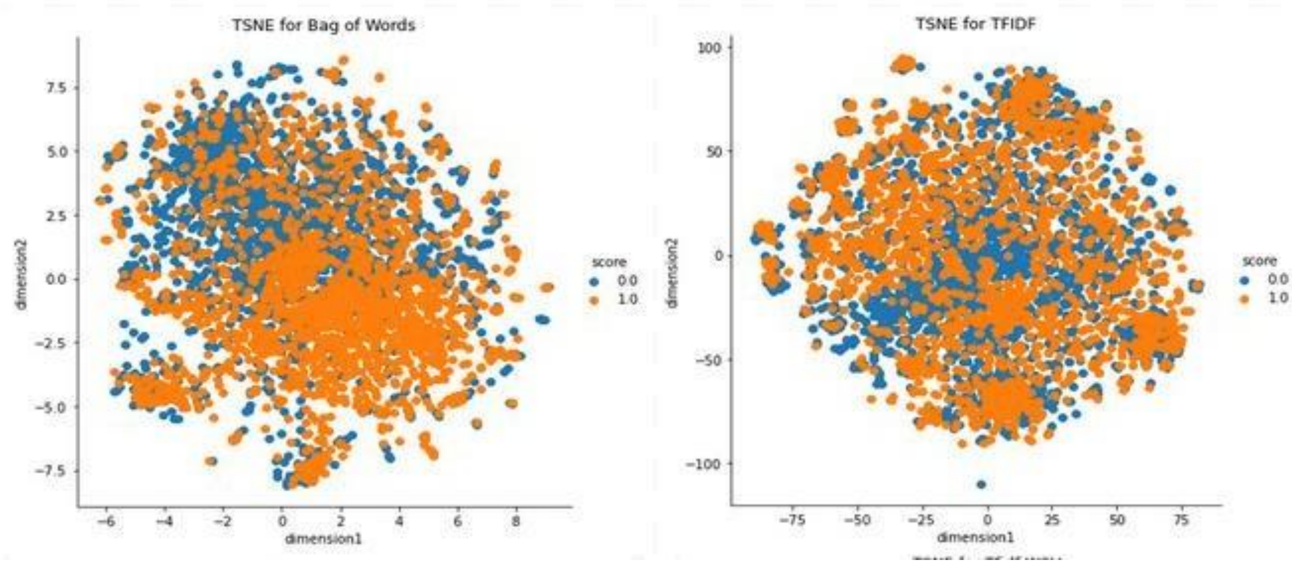
After that, I combined our text with bow vectorization, tfidf vectorization, word2vec averages, and tfidf word2vec procedures, keeping them as independent vectors. Because vectoring big amounts of data is costly, I performed it on a computer once and then decided I didn't want to do it again.

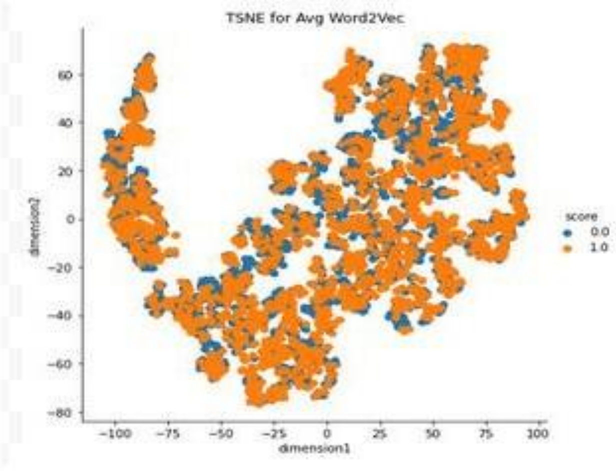
I used the word bag and tfidf unigram technique. Instead of using previously trained weights, I trained the model in the instance of word2vec. You can always utilise the n-gram bow / tfidf approach, and in word2vec matter, you can use pre-trained embedding.

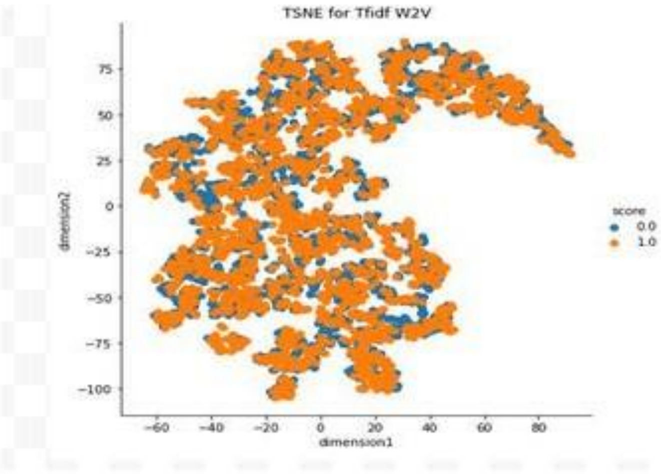
Always try to match your model to train data before converting it to test data. If you try to balance your vectorizer on test data, you can end up with data leakage issues.

TSNE visualization

One of the most popular approaches to reduce size is to use TSNE stands for t-distributed stochastic neighbour embedding. It's a popular tool for visualising small scales. I tried to envision you at a low level before I delved into machine learning models.







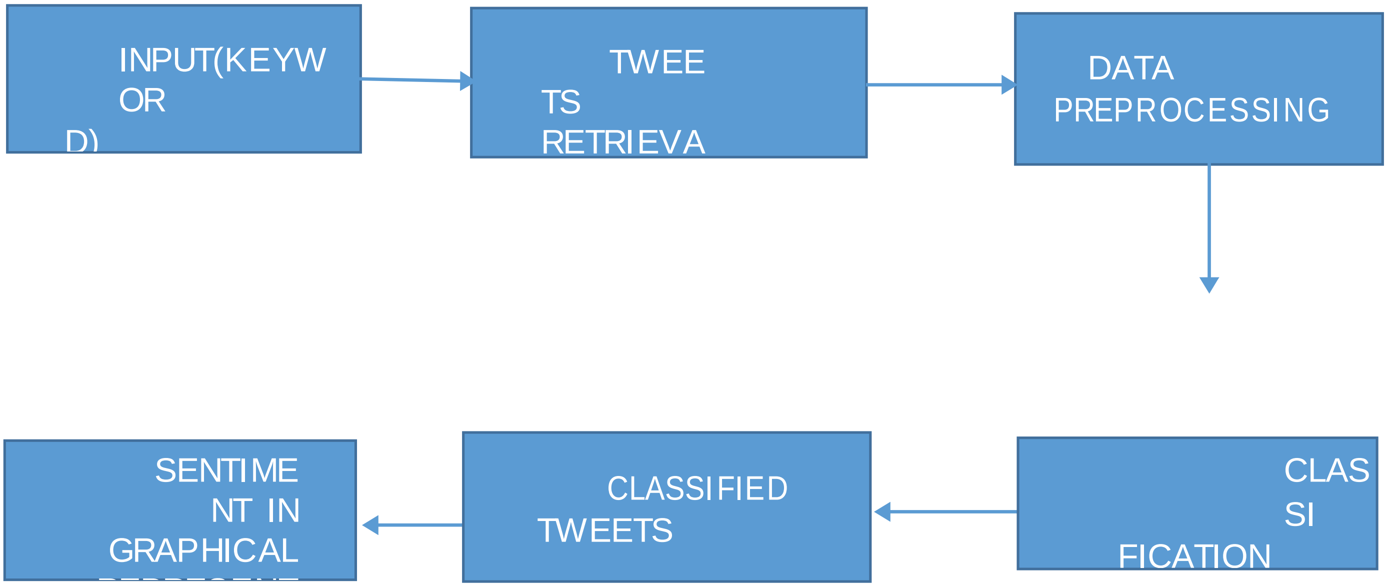
Steps I took to prepare for TSNE:

- I ran TSNE at several iterations while keeping perplexity constant and identified the best stable one.
- Keeping that iteration constant, I ran TSNE at several levels of perplexity to improve the results.
- I ran TSNE again with the same parameters once I had a consistent result.

However, I discovered that TSNE was unable to differentiate points at a lower level.

With 20000 random points, I tried TSNE (with equal class distribution). With a huge number of datapoints, the results can be improved.

Design



Data Preprocessing:

There are three steps to data preprocessing:

- Tokenization,
- Normalization, and
- Tagging of parts of speech (POS).

Tokenization:

The process of dividing a string of text into words, symbols, and other logical pieces known as "tokens" is known as tokenization. White space letters and/or punctuation marks can be used to separate tokens. It's built in such a way that tokens can be viewed as distinct components of a tweet. Icons and abbreviations (e.g., OMG, WTF, BRB) are treated as distinct tokens and are considered part of the token creation process.

Normalization:

The presence of abbreviations in the tweet is recognised throughout the customisation process, and the summaries are changed with their genuine meaning (for example, BRB -> be right back). We also note the existence of informal reinforcement items such as caps (e.g., I LOVE this show!!!) and character repetitions (e.g., I got a loan!! happyyyyy) All of the words claim to be written in lower case. A single letter has been used to replace repeated character shapes. Finally, any specific Twitter tokens (e.g., hashtags, usertags, URLs) are recorded, and placeholders indicating token type are altered. of POS, which is the final step in processing.

Part-of-speech:

POS-tagging is the process of assigning a tag to each word in a phrase that indicates which part of the word system it belongs to, such as noun, verb, adjective, adjective, link link, and so on. We have counting aspects in each tweet, such as the number of verbs, adverbs, adjectives, nouns, and any other parts of speech.

Implementation Methodology (Formulation/Algorithm)

DATA COLLECTION:

From May 1996 through July 2014, data for product reviews was collected from amazon.com. The following information is included in each review: 1) Reviewer ID; 2) Product ID; 3) measurement; 4) update time; 5) assistance; 6) text update With the exception of the existence of half a star or a star, all ratings are based on a 5-star rating, resulting in ratings ranging from 1 to 5 stars.

EXTRACTION OF SENTIMENT SENTENCES AND POS TAGGING:

A basic prerequisite for POS tagging is the production of tokens in updates after the removal of the phrase STOP, which implies nothing related to emotion. The remaining phrases are turned into tokens after the right elimination of STOP words such as "am, is, are, the, but," and so on. The POS tagging is aided by these tokens.

Part-of-speech (POS) taggers have been developed in natural language processing to classify words based on their parts of speech. A POS tagger is particularly beneficial for sentiment analysis for the following two reasons: 1) Nouns and pronouns are usually devoid of any sentiment. A POS tagger can be used to filter out such terms; 2) A POS tagger can also be used to distinguish between words that can be used in various parts of speech.

NEGATIVE PHRASE IDENTIFICATION:

With the use of negative beginnings, words like adjectives and actions can communicate negative feelings. Consider the following line from an electrical equipment review: "The built-in speaker also works, but nothing has changed thus far." The term "conversion" is an useful one to use when describing an internet listing. "Nothing is flexible," as a phrase, connotes negative or less pleasant sentiments.

As a result, it's critical to recognise these phrases. There are two types of phrases discovered in this study: adjective-negative (NOA) and negative-verb (NV) (NOV).

Discussion

The major purpose of this research was to see which of the SVM and Nave Bayes machine learning algorithms performed better at text classification. The Amazon beauty items data set was used to do this. The classifiers were assessed by comparing their accuracies in several experimental scenarios.

The overall accuracies of two machine learning methods in various experiments. The SVM approach had superior accuracy than the Nave Bayes approach in both scenarios where the algorithms were deployed on reviews and when they were applied on summaries, according to the results from the first set of tests presented in table 3. However, the difference in accuracy between these methods is negligible.

This experiment shown that properly-trained machine learning algorithms with a large enough training data set can do exceptionally well in categorization. SVM outperforms Naive Bayes in terms of accuracy, albeit the differences aren't significant, and the algorithms can properly classify more than 90% of the time.

In both situations, the findings from the second set of experiments, reported in Tables 4 and 5, showed that the Naive Bayes technique was more accurate than the SVM. This experiment had a significantly smaller data set than the last one, with only 300 reviews from 10 distinct products.

The size of the data sets could be one explanation for the disparity. The training data set in the first experiment is substantially larger than in the second experiment. This may lead to the conclusion that the SVM model performs better when there is more data. In their experiment, Fang and Zhan (2015) got a similar result. They use the Nave Bayes, SVM, and Random Forest models to handle the problem of sentiment analysis on Amazon reviews in their study. They showed that the SVM model outperforms the other classifiers when given more training data.

Another factor could be the three-star reviews, which are normally classified as impartial. However, in the second trial, they were deemed negative due to the tiny size of the data set. This may have had an impact on the outcome of the second experiment. It would be fascinating to see if categorising them as positive yields a different result in future research.

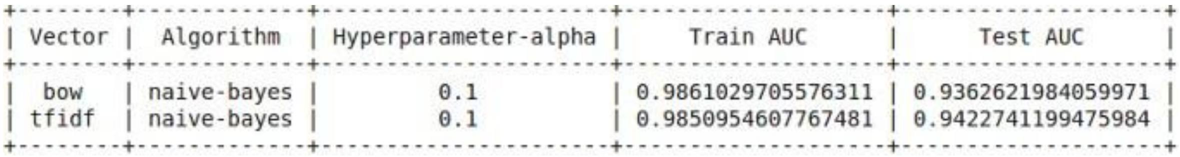
SENTIMENT CLASSIFICATION ALGORITHMS:

Naïve Bayesian classifier:

Another machine learning technique noted for its power despite its simplicity is Naive Bayes. This classifier is based on the Bayes theorem and assumes that the characteristics (which in text classification are usually words) are mutually independent. Despite the fact that this assumption is incorrect (since the sequence of the words matters in some circumstances), Naive Bayes classifiers have proven to be quite effective (Rish 2001). Before using the Naive Bayes model to text classification problems, the initial step should be feature extraction.

The following is how the Nave Bayesian classifier works: Assume you have a collection of training data, D, where each tuple is represented by an n-dimensional feature vector, X=x 1,x 2,...,x n, indicating n measurements taken on the tuple from n attributes or features. Assume m classes, C 1, C 2,..., C m. The classifier will predict that tuple X belongs to C I if and only if: P(C I |X)>P(C j |X), where i,j[1,m]a n d ij are the variables. P(C I |X) is calculated as follows:

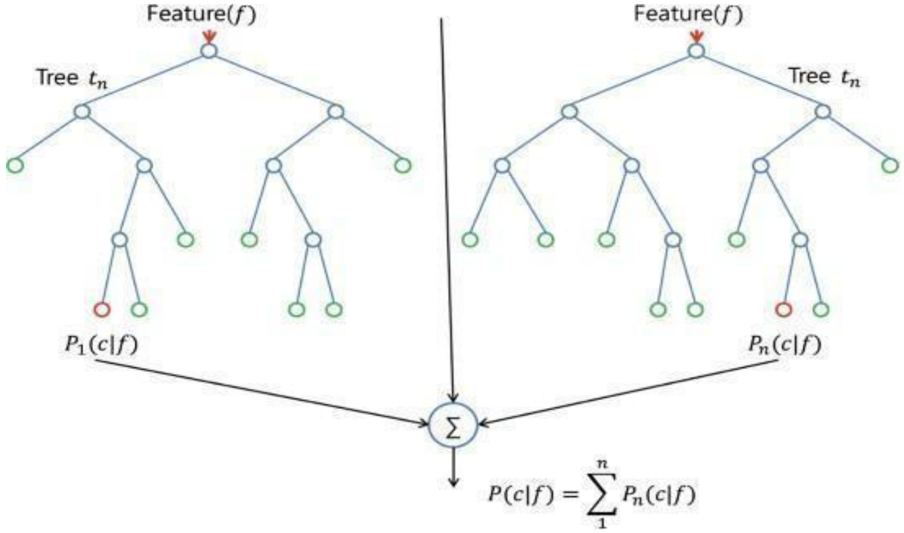
$$P(C_i|X) = \prod_{k=1}^{n} P(x_k|C_i)$$



```
+--------+-------------+----------------------+--------------------+--------------------+
| Vector |  Algorithm  | Hyperparameter-alpha |     Train AUC      |      Test AUC      |
+--------+-------------+----------------------+--------------------+--------------------+
|  bow   | naive-bayes |         0.1          | 0.9861029705576311 | 0.9362621984059971 |
| tfidf  | naive-bayes |         0.1          | 0.9850954607767481 | 0.9422741199475984 |
+--------+-------------+----------------------+--------------------+--------------------+
```

Random forest

Because a random forest category outperformed a single decision tree in terms of accuracy, it was picked. It's actually a bagging-based compilation method. The separator functions in the following way: The initial phase of D-given creates D-bootstrap samples of D, with each sample representing Di. The number of tuples transformed from D in a Di is the same as D. Because samples are replaced, some original copies of D may be missing from Di, while others may appear many times. After then, the classifier creates a decision tree based on each Di. As a result,



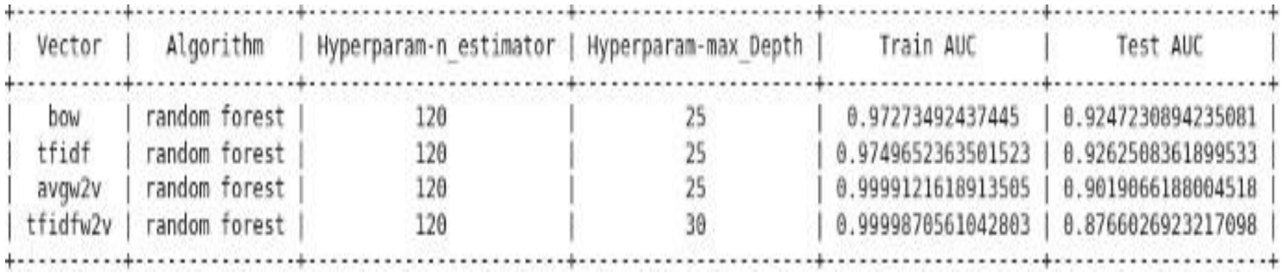
a “forest" that consists of *k* decision trees is formed.

Each tree presents its estimate of the class forecast as a single vote to separate the unknown tuple, X. The X class's final selection is made by the person who receives the most votes.

CART is the decision tree algorithm used in scikit-learn (Classification and Regression Trees). CART's tree induction is based on the Gini index. The Gini index for D is calculated as follows:

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

The probability that a tuple in D belongs to class C I is given by pi. The Gini index is a measure of D impurity. The better D was partitioned, the lower the index value.



```
+----------+---------------+------------------------+----------------------+--------------------+--------------------+
|  Vector  |   Algorithm   | Hyperparam-n_estimator | Hyperparam-max_Depth |     Train AUC      |      Test AUC      |
+----------+---------------+------------------------+----------------------+--------------------+--------------------+
|   bow    | random forest |          120           |          25          |  0.97273492437445  | 0.9247230894235081 |
|  tfidf   | random forest |          120           |          25          | 0.9749652363501523 | 0.9262508361899533 |
|  avgw2v  | random forest |          120           |          25          | 0.9999121618913505 | 0.9019066188004518 |
| tfidfw2v | random forest |          120           |          30          | 0.9999870561042803 | 0.8766026923217098 |
+----------+---------------+------------------------+----------------------+--------------------+--------------------+
```

Support vector Machine

SVM (Support Vector Machine) is a technique for separating direct and indirect data. SVM looks for the correct linear hyperplane (line kernel) while categorising data, which is the decision bar that separates data into different classes. A hyper plane divider can be calculated statistically as W X + b = 0, where W is a weight vector and W = w1, w2,..., w n. X is a training tuple. A scale is denoted by the letter b.
The problem effectively translates to the minimization of W in order to optimise the hyperplane, which is eventually computed as:

$$\sum_{i=1}^{n} \alpha_i y_i x_i ,$$

where αi are numeric parameters, and yi are labels based on support vectors, Xi .

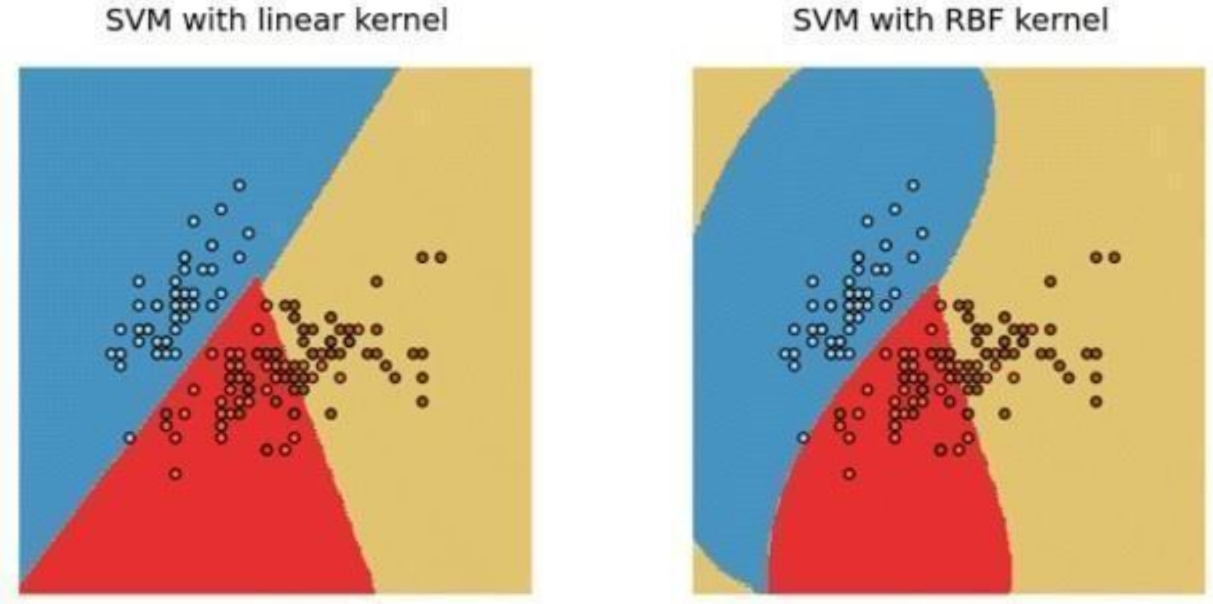
That is: if yi =1 then

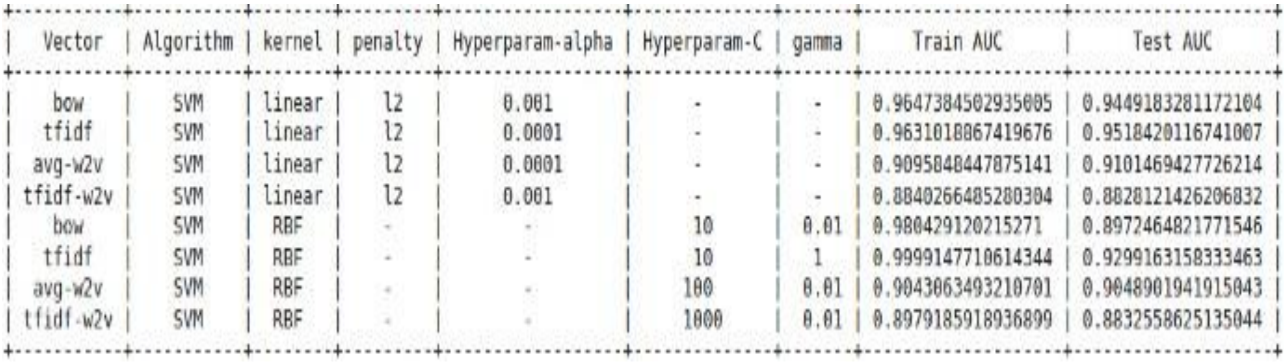
$$\sum_{i=1}^{n} w_i x_i \geq 1;$$

SVM employs an indirect map to turn data into a maximum size if it can't be split by a line. Then find a line hyperplane to solve the problem. Kernel functions are the tools used to make such changes. The Gaussian Radial Basis Function (RBF) was chosen as the kernel function for our experiment:

$$K(X_i, X_j) = e^{-\gamma \|X_i - X_j\|^2 / 2}$$

Xi are support vectors, X j are testing tuples, and is a free parameter in our experiment that utilises the default value from scikit-learn. On the following page, Figure illustrates a classification example of SVM based on the linear kernel and the RBF kernel.





```
+-----------+-----------+--------+---------+------------------+--------------+-------+--------------------+--------------------+
|   Vector  | Algorithm | kernel | penalty | Hyperparam-alpha | Hyperparam-C | gamma |     Train AUC      |      Test AUC      |
+-----------+-----------+--------+---------+------------------+--------------+-------+--------------------+--------------------+
|    bow    |    SVM    | linear |    l2   |      0.001       |      -       |   -   | 0.9647384502935005 | 0.9449183281172104 |
|   tfidf   |    SVM    | linear |    l2   |      0.0001      |      -       |   -   | 0.9631018867419676 | 0.9518420116741007 |
|  avg-w2v  |    SVM    | linear |    l2   |      0.0001      |      -       |   -   | 0.9095848447875141 | 0.9101469427726214 |
| tfidf-w2v |    SVM    | linear |    l2   |      0.001       |      -       |   -   | 0.8840266485280304 | 0.8828121426206832 |
|    bow    |    SVM    |  RBF   |    -    |        -         |      10      |  0.01 | 0.980429120215271  | 0.8972464821771546 |
|   tfidf   |    SVM    |  RBF   |    -    |        -         |      10      |   1   | 0.9999147710614344 | 0.9299163158333463 |
|  avg-w2v  |    SVM    |  RBF   |    -    |        -         |     100      |  0.01 | 0.9043063493210701 | 0.9048901941915043 |
| tfidf-w2v |    SVM    |  RBF   |    -    |        -         |     1000     |  0.01 | 0.8979185918936899 | 0.8832558625135044 |
+-----------+-----------+--------+---------+------------------+--------------+-------+--------------------+--------------------+
```

Logistic Regression

The likelihood of an outcome with only two possible values is predicted using logistic regression (i.e. a dichotomy). One or more predictors are used to make the prediction (numerical and categorical). For two reasons, linear regression is ineffective for predicting the value of a binary variable:

$$\sum_{i=1}^{n} w_i x_i \geq -1 .$$

A line break indicates values that are outside of the permitted range (e.g., predictors that may be outside the range 0 to 1)
The residues will not generally be dispersed in proportion to the anticipated line since dichotomous tests may only have one of two possible values per test.

Logistic retreat, on the other hand, results in a curve that is limited to integers between 0 and 1. Object reversal is identical to line reversal, but the curve is created using the natural logarithm of the target variety's "constraints," rather than possibilities. Furthermore, the predictions in each group do not need to be evenly distributed or even equally variable.

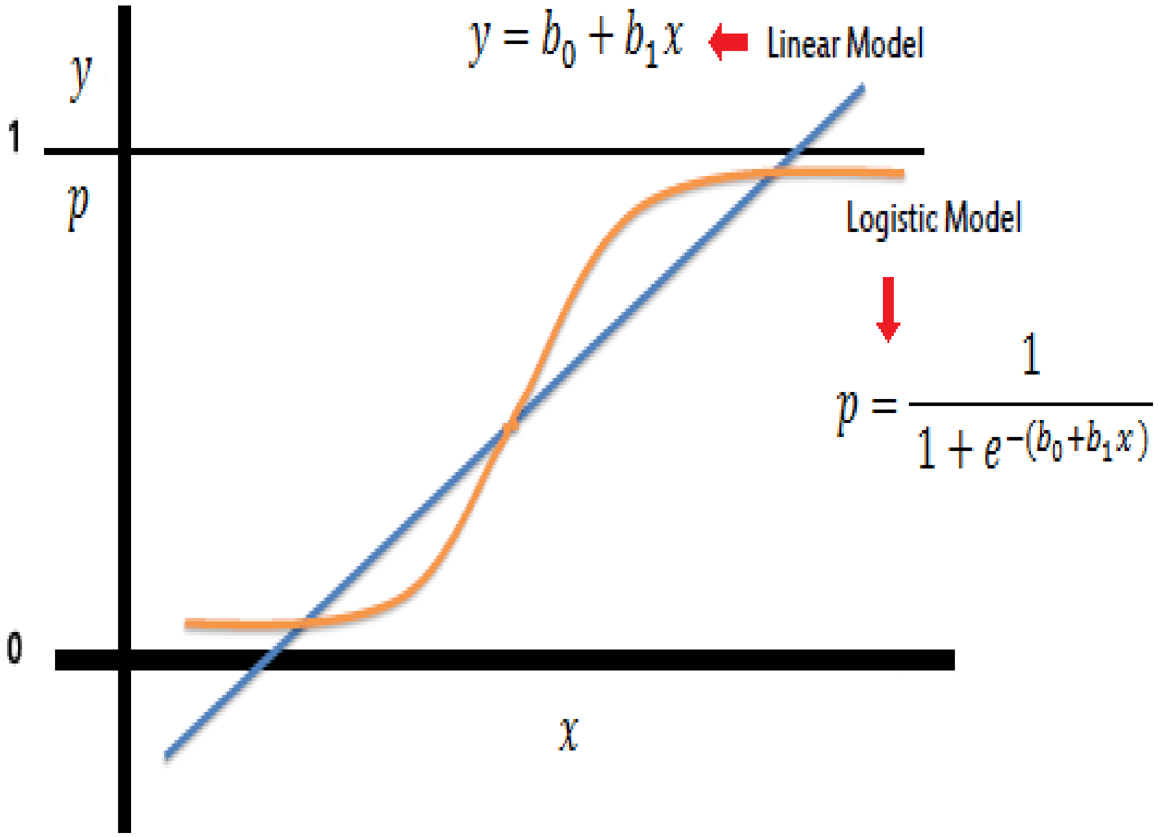
The model coefficient corresponds with predictions and target in the retrospective regression using high probability (MLE) estimations. The method is repeated until the LL (Log Likelihood) does not change appreciably after the initial function is constrained.

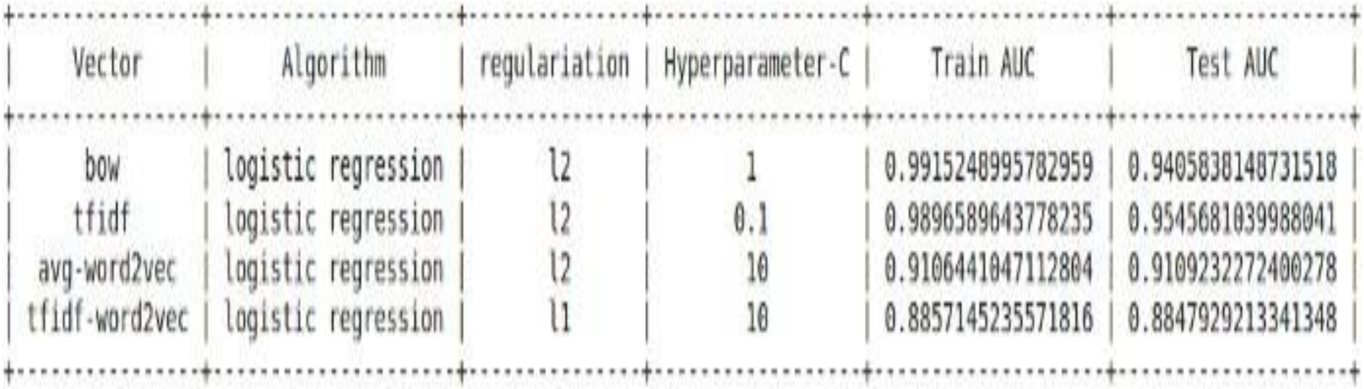
$$\beta^1 = \beta^0 + [X^T W X]^{-1} . X^T (y - \mu)$$

β is a vector of the logistic regression coefficients.

W is a square matrix of order N with elements $n_i \pi_i (1 - \pi_i)$ on the diagonal and zeros everywhere else.

μ is a vector of length N with elements $\mu_i = n_i \pi_i$.





```
+---------------+---------------------+---------------+------------------+--------------------+--------------------+
|     Vector    |      Algorithm      | regulariation | Hyperparameter-C |     Train AUC      |      Test AUC      |
+---------------+---------------------+---------------+------------------+--------------------+--------------------+
|      bow      | logistic regression |       l2      |        1         | 0.9915248995782959 | 0.9405838148731518 |
|     tfidf     | logistic regression |       l2      |       0.1        | 0.9896589643778235 | 0.9545681039988041 |
|  avg-word2vec | logistic regression |       l2      |        10        | 0.9106441047112804 | 0.9109232272400278 |
| tfidf-word2vec| logistic regression |       l1      |        10        | 0.8857145235571816 | 0.8847929213341348 |
+---------------+---------------------+---------------+------------------+--------------------+--------------------+
```

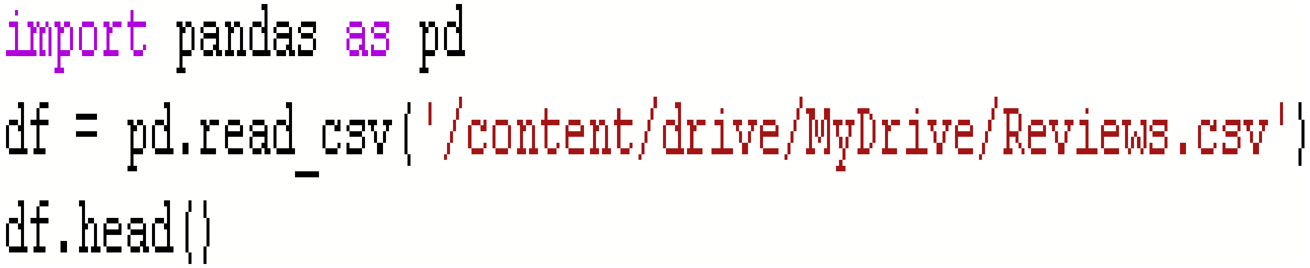
We can see that the dropdown and linear SVM in the central characteristics of word2vec provide the most common model after experimenting with a few machine learning approaches.

Implementation Details

To do the study, we will use a dataset from Kaggle, namely the “Amazon Fine Food” Reviews dataset.

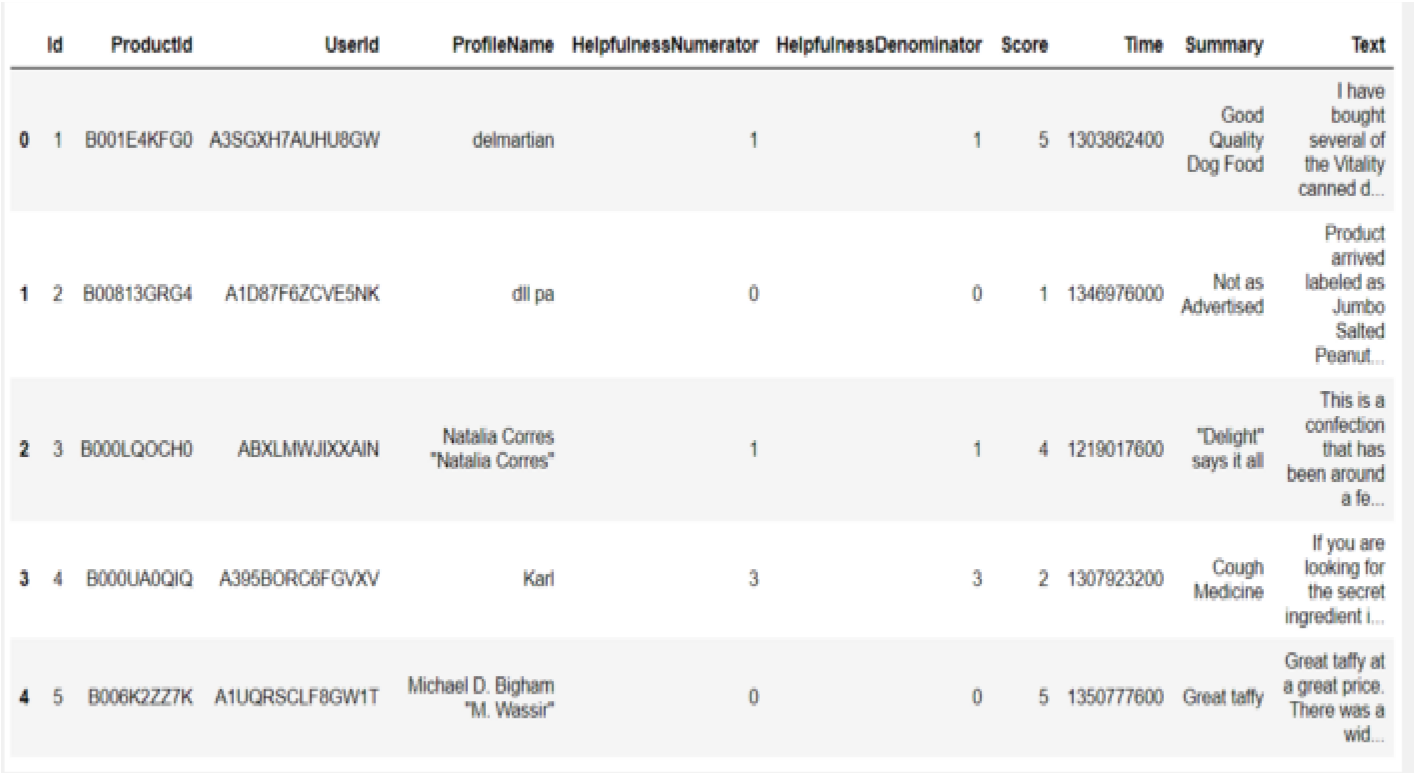
All analysis and visualisation will be done in Google Colab, however any Python IDE would suffice.

Step 1: Read the Data



```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Reviews.csv')
df.head()
```

Checking the dataframe's head:



	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

We can see that the dataframe contains information on the product, the user, and the reviews. "Summary," "Text," and "Score" are the most important data for this analysis.

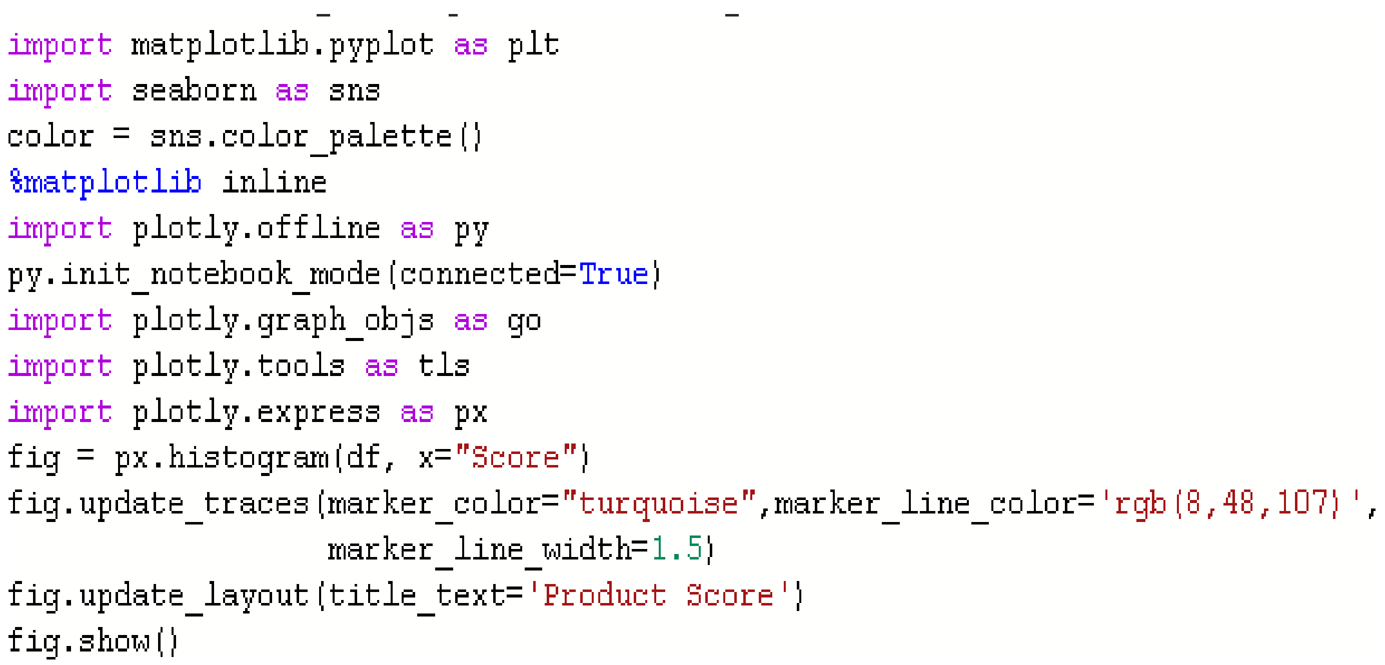
Text — This variable holds all of the information about the product review.

Summary — This is a condensed version of the whole review.

Score — The product rating provided by the customer.

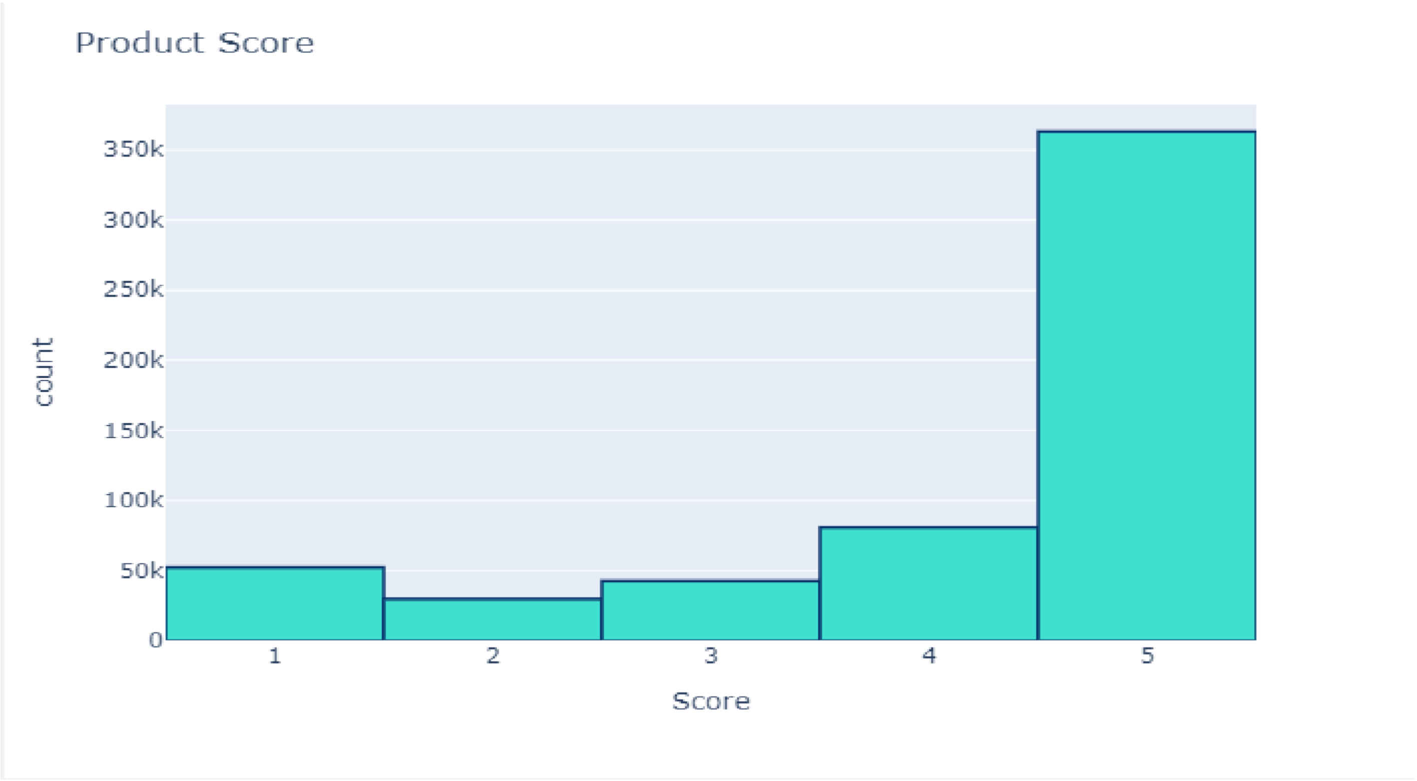
Step 2: Data Analysis

Now, we will take a look at the variable "*Score*" to see if majority of the customer ratings are positive or negative.



```python
import matplotlib.pyplot as plt
import seaborn as sns
color = sns.color_palette()
%matplotlib inline
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import plotly.express as px
fig = px.histogram(df, x="Score")
fig.update_traces(marker_color="turquoise",marker_line_color='rgb(8,48,107)',
                  marker_line_width=1.5)
fig.update_layout(title_text='Product Score')
fig.show()
```

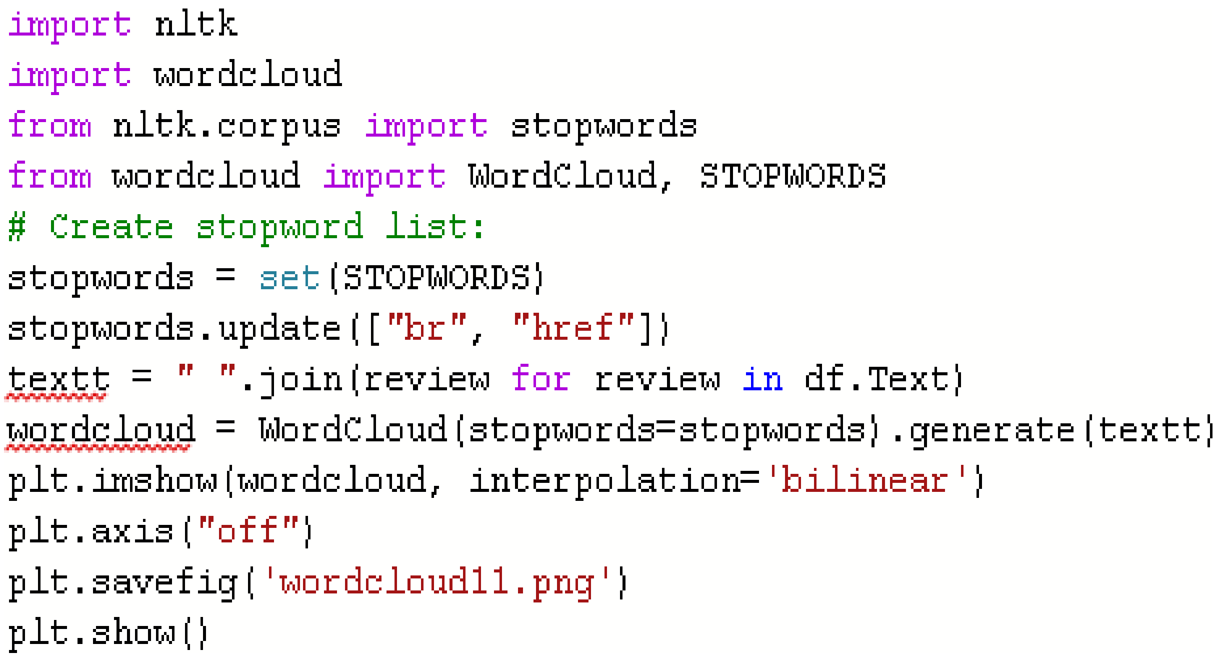
The resulting plot looks like this:



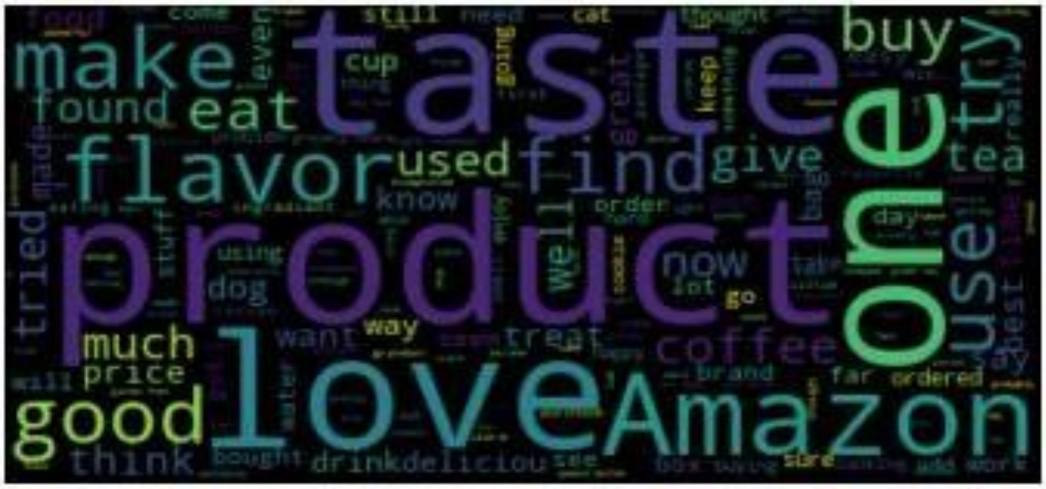
From here, we can see that most of the customer rating is positive. This leads me to believe that most reviews will be pretty positive too, which will be analyzed in a while.

Now, we can create some wordclouds to see the most frequently used words in the reviews.

The code above generates a word cloud that looks like this:



```
import nltk
import wordcloud
from nltk.corpus import stopwords
from wordcloud import WordCloud, STOPWORDS
# Create stopword list:
stopwords = set(STOPWORDS)
stopwords.update(["br", "href"])
textt = " ".join(review for review in df.Text)
wordcloud = WordCloud(stopwords=stopwords).generate(textt)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.savefig('wordcloud11.png')
plt.show()
```



Other popular names that can be seen here include “taste,” “product,” “love,” and “Amazon.” These terms are generally correct, and they indicate that most of the reviews in the database reflect positive emotions.

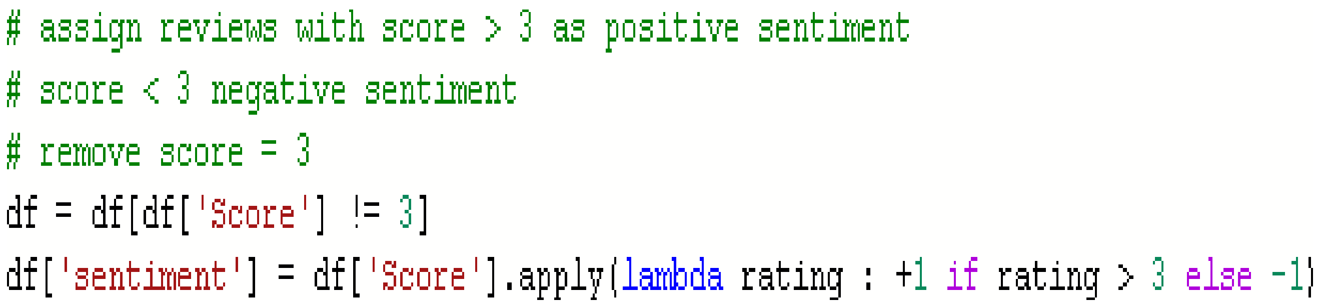
Step 3: Classifying Tweets

In this step, we will classify reviews into “positive” and “negative,” so we can use this as training data for our sentiment classification model.

Positive reviews will be classified as +1, and negative reviews will be classified as -1.

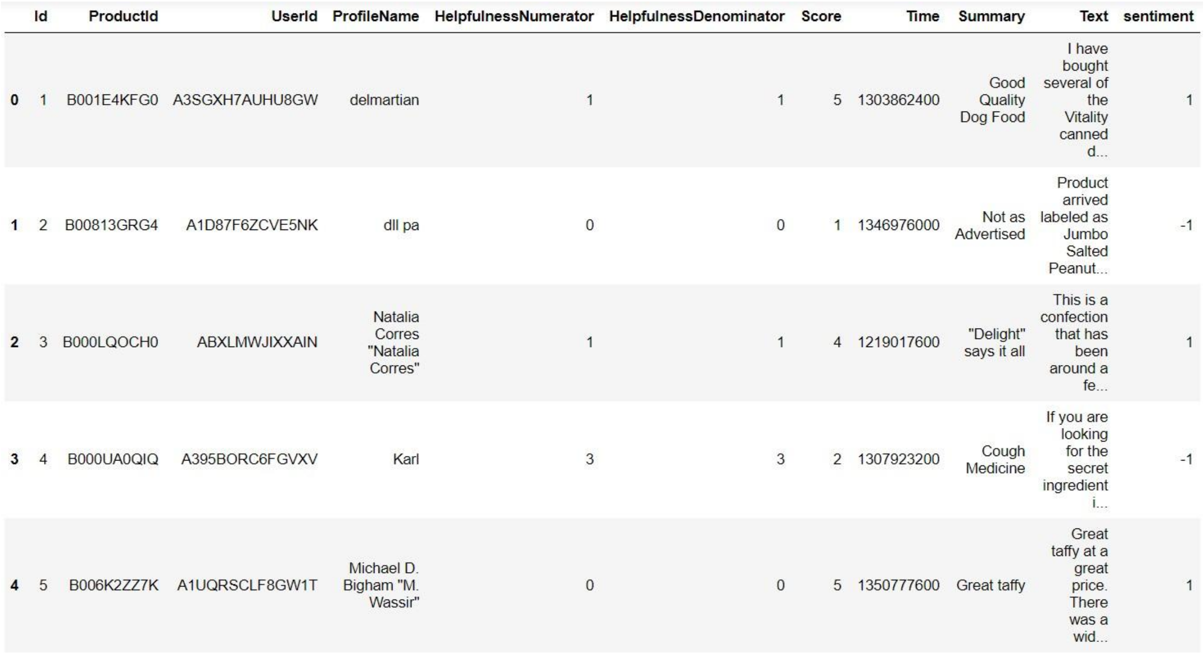
We will classify all reviews with ‘Score’ > 3 as +1, indicating that they are positive.

All reviews with ‘Score’ < 3 will be classified as -1. Reviews with ‘Score’ = 3 will be dropped, because they are neutral. Our model will only classify positive and negative reviews.



```
# assign reviews with score > 3 as positive sentiment
# score < 3 negative sentiment
# remove score = 3
df = df[df['Score'] != 3]
df['sentiment'] = df['Score'].apply(lambda rating : +1 if rating > 3 else -1)
```

Looking at the head of the data frame now, we can see a new column called ‘sentiment:’

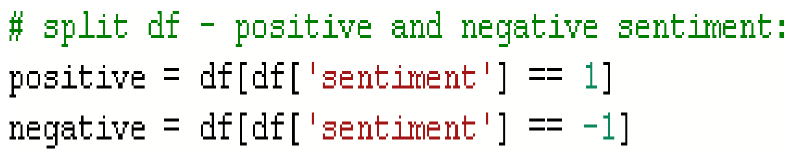


	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	sentiment
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	1
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	-1
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...	1
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...	-1
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...	1

Step 4: More Data Analysis

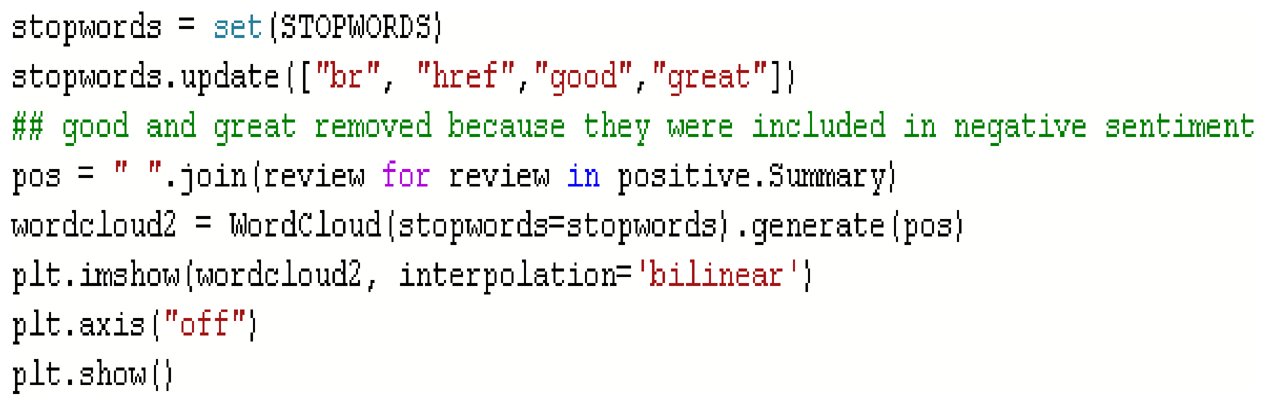
Now that we have classified tweets into positive and negative, let’s build wordclouds for each!

First, we will create two data frames — one with all the positive

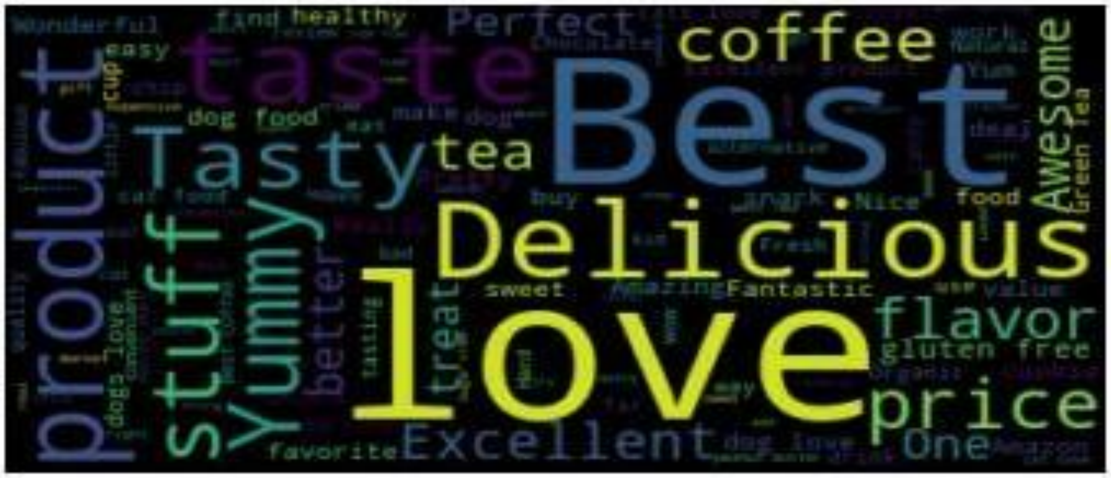


```
# split df - positive and negative sentiment:
positive = df[df['sentiment'] == 1]
negative = df[df['sentiment'] == -1]
```

Wordcloud — Positive Sentiment

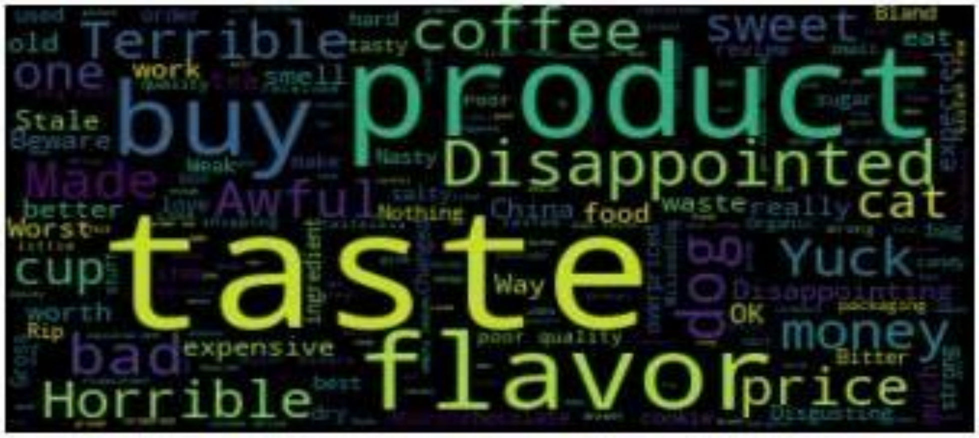


```
stopwords = set(STOPWORDS)
stopwords.update(["br", "href","good","great"])
## good and great removed because they were included in negative sentiment
pos = " ".join(review for review in positive.Summary)
wordcloud2 = WordCloud(stopwords=stopwords).generate(pos)
plt.imshow(wordcloud2, interpolation='bilinear')
plt.axis("off")
plt.show()
```



Wordcloud — Negative Sentiment

```
neg = " ".join(review for review in negative.Summary)
wordcloud3 = WordCloud(stopwords=stopwords).generate(neg)
plt.imshow(wordcloud3, interpolation='bilinear')
plt.axis("off")
plt.savefig('wordcloud33.png')
plt.show()
```

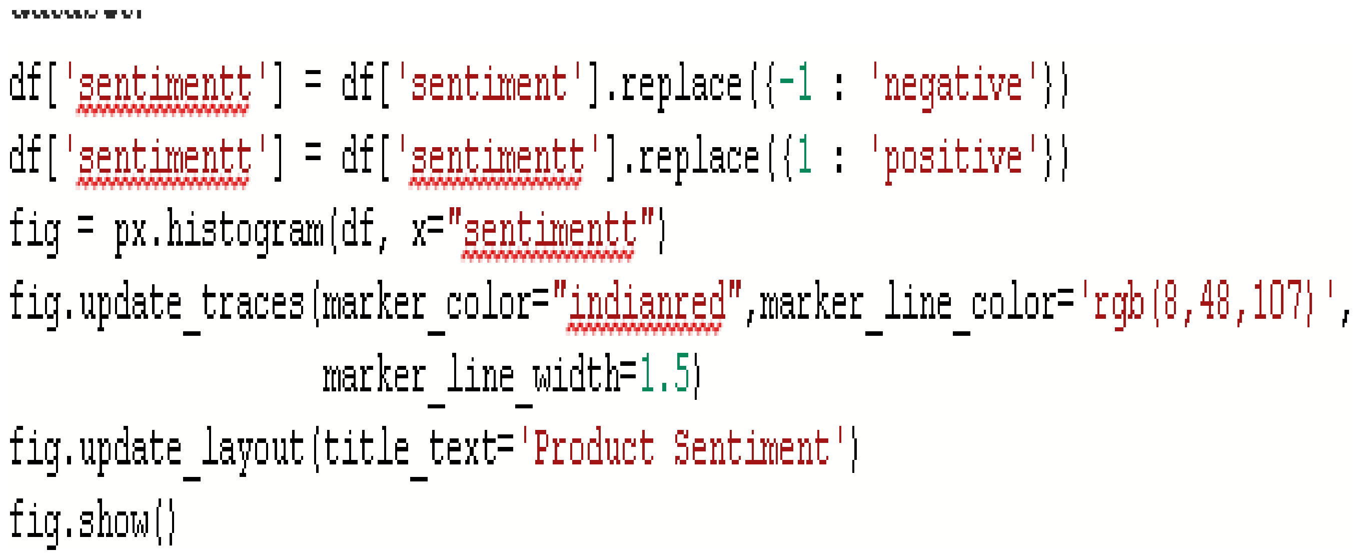


As noted above, the cloud of positive thoughts filled with positive words, such as “love,” “best,” and “pleasant.”

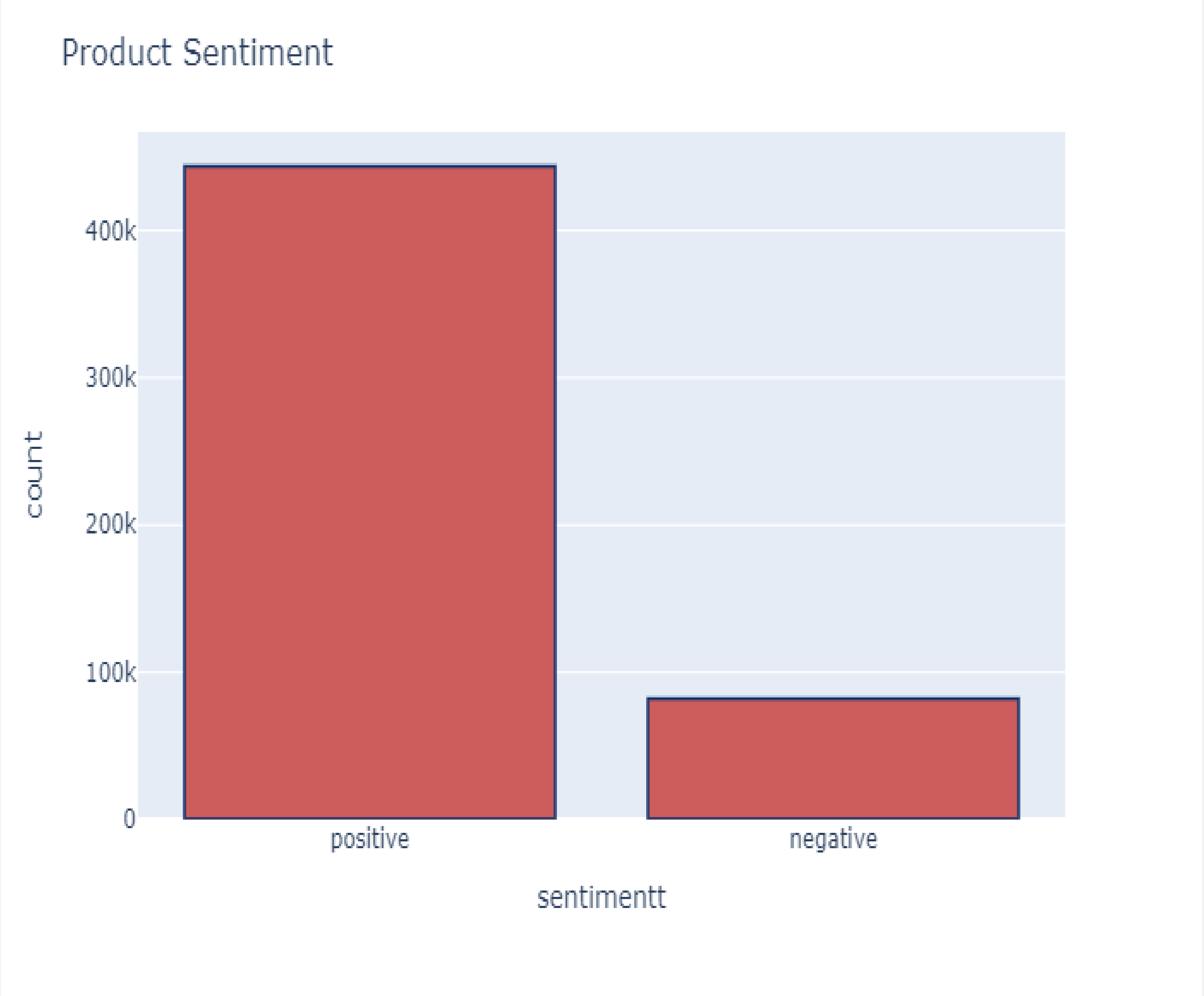
The cloud of bad ideas filled the air with many negative words, such as “disappointed,” and “yuck.”

The words “good” and “great” initially appear to be derogatory, an expression of endearment. This is because they are used in a bad situation, such as "they are not right." Because of this, I have removed those two words from the word cloud.

Finally, we can take a look at the distribution of reviews with sentiment across the dataset:



```
df['sentimentt'] = df['sentiment'].replace({-1 : 'negative'})
df['sentimentt'] = df['sentimentt'].replace({1 : 'positive'})
fig = px.histogram(df, x="sentimentt")
fig.update_traces(marker_color="indianred",marker_line_color='rgb(8,48,107)',
                  marker_line_width=1.5)
fig.update_layout(title_text='Product Sentiment')
fig.show()
```



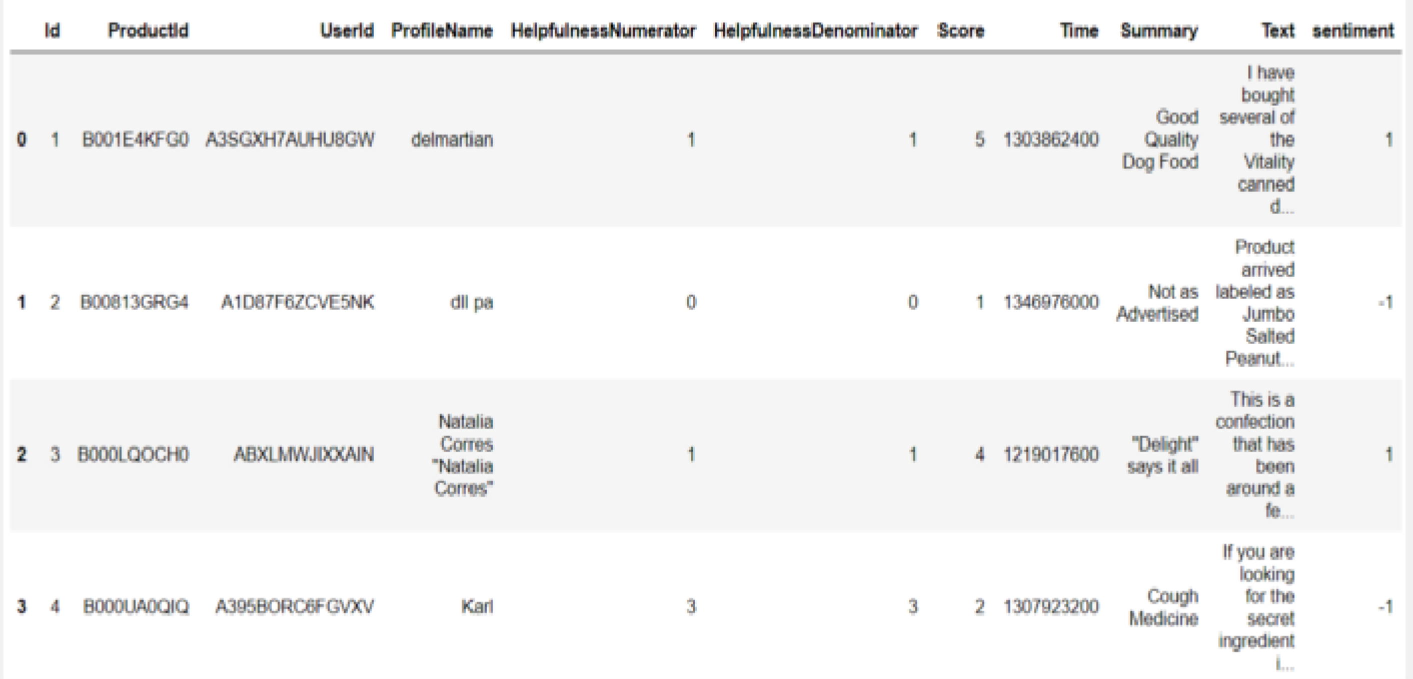
Step 5: Building the Model

Finally, we can build the sentiment analysis model!

This model will take the update as an installation. It will then come up with an estimate of whether the review is good or bad.

This is a segmentation job, so we're going to train a simple model to back things up for us to do.

For reference, take a look at the data frame again:



	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	sentiment
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	1
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	-1
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...	1
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...	-1

There are a few steps we need to take:

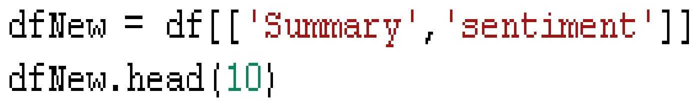
- **Data Cleaning**

We will use summary data to come up with predictions. First, we need to remove all punctuation marks from the data.

```
def remove_punctuation(text):
    final = "".join(u for u in text if u not in ("?", ".", ";", ":",  "!",'"'))
    return final
df['Text'] = df['Text'].apply(remove_punctuation)
df = df.dropna(subset=['Summary'])
df['Summary'] = df['Summary'].apply(remove_punctuation)
```

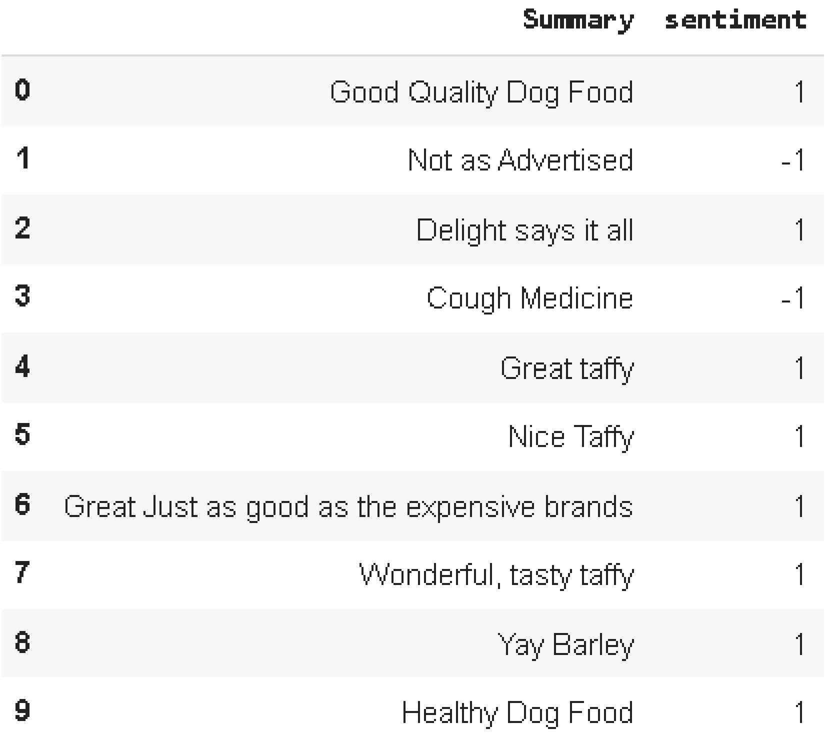
- **Split the Dataframe**

The new data frame should only have two columns — “*Summary*” (the review text data), and “*sentiment*” (the target variable).



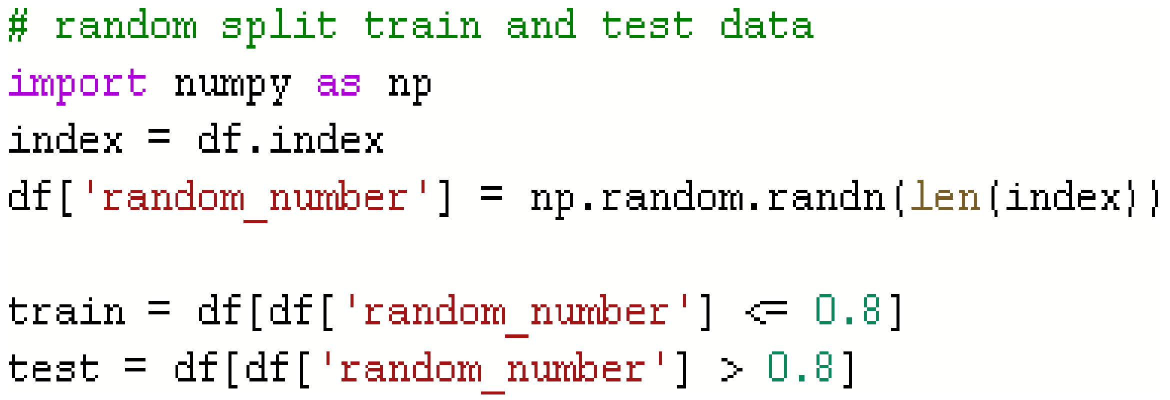
```
dfNew = df[['Summary','sentiment']]
dfNew.head(10)
```

Taking a look at the head of the new data frame, this is the data it will now contain:



	Summary	sentiment
0	Good Quality Dog Food	1
1	Not as Advertised	-1
2	Delight says it all	1
3	Cough Medicine	-1
4	Great taffy	1
5	Nice Taffy	1
6	Great Just as good as the expensive brands	1
7	Wonderful, tasty taffy	1
8	Yay Barley	1
9	Healthy Dog Food	1

We will now split the data frame into train and test sets. 80% of the data will be used for training, and 20% will be used for testing.



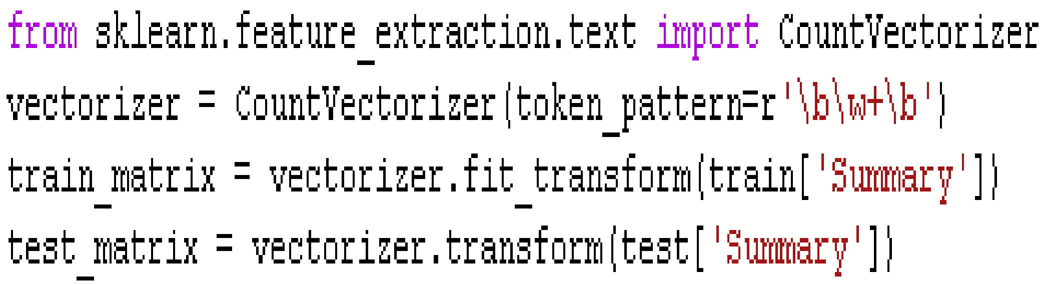
```
# random split train and test data
import numpy as np
index = df.index
df['random_number'] = np.random.randn(len(index))

train = df[df['random_number'] <= 0.8]
test = df[df['random_number'] > 0.8]
```

- **Create a bag of words**

Next, we will use a count vectorizer from the Scikit-learn library.

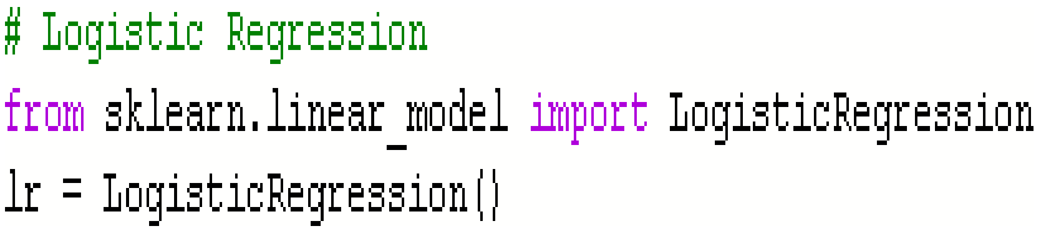
This will transform the text in our data frame into a word bag model, which will contain a small value matrix. The number of each word appearing will be calculated and printed.

We will need to convert the text into a word bag model as the retrieval algorithm can understand the text.



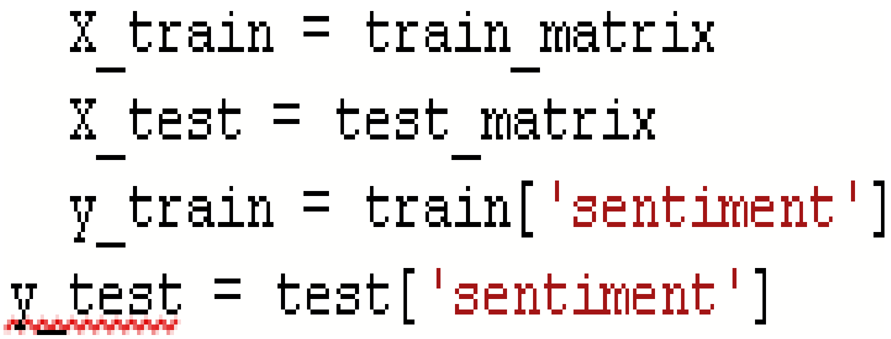
```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(token_pattern=r'\b\w+\b')
train_matrix = vectorizer.fit_transform(train['Summary'])
test_matrix = vectorizer.transform(test['Summary'])
```

Import Logistic Regression



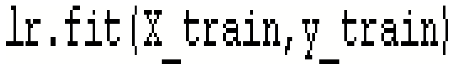
```
# Logistic Regression
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
```

Split target and independent variables



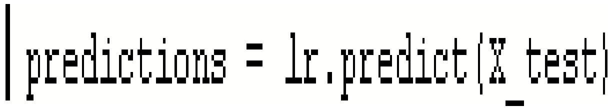
```
X_train = train_matrix
X_test = test_matrix
y_train = train['sentiment']
y_test = test['sentiment']
```

- **Fit model on data**



```
lr.fit(X_train,y_train)
```

- **Make predictions**

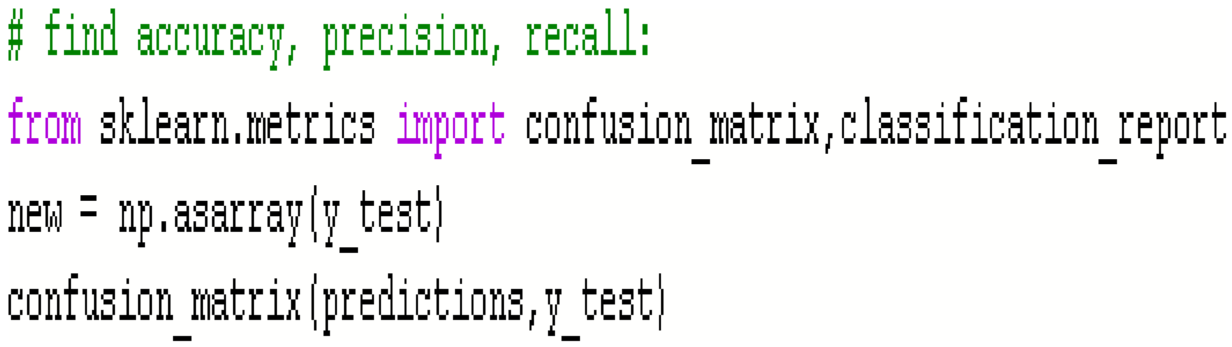


```
predictions = lr.predict(X_test)
```

We have successfully built a simple logistic regression model, and trained the data on it. We also made predictions using the model.

Step 6: Testing

Now, we can test the accuracy of our model!

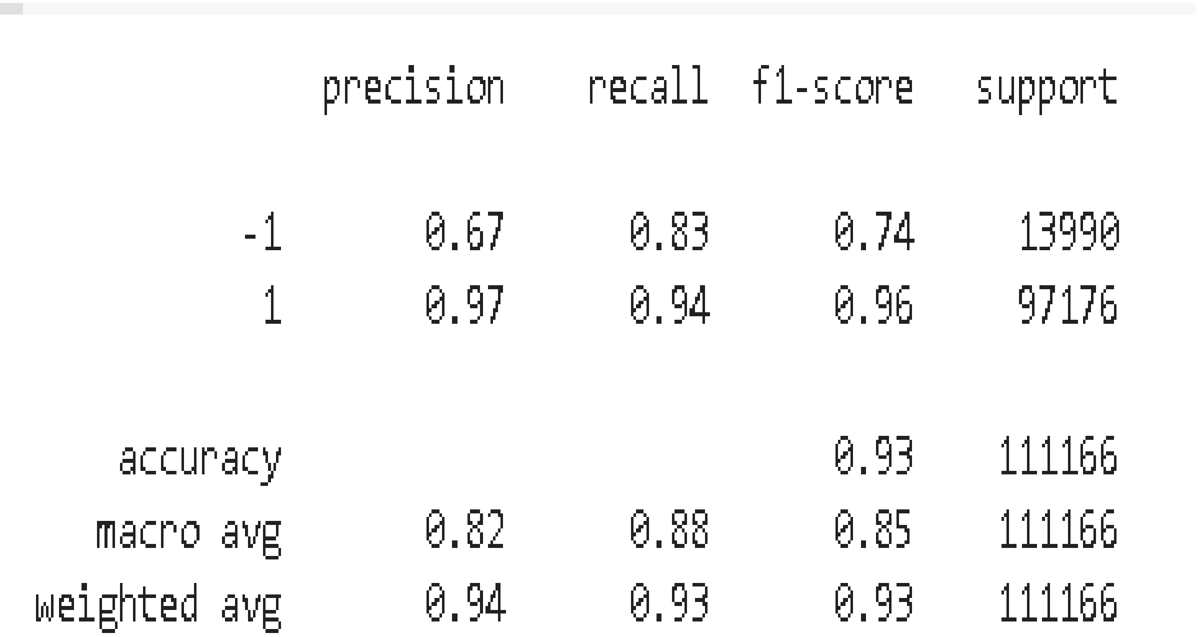


```
# find accuracy, precision, recall:
from sklearn.metrics import confusion_matrix,classification_report
new = np.asarray(y_test)
confusion_matrix(predictions,y_test)
```

Confusion matrix that looks like this:

§Ilh((ElBRRl(lEB(l0h IR§0I((§IR*lE(l0hR,§(RR(

The classification report:



```
              precision    recall  f1-score   support

          -1       0.67      0.83      0.74     13990
           1       0.97      0.94      0.96     97176

    accuracy                           0.93    111166
   macro avg       0.82      0.88      0.85    111166
weighted avg       0.94      0.93      0.93    111166
```

The final accuracy of the model = 93%, considerably good without any feature extraction and preprocessing.

Applications of Sentiment Analysis

- Sentiment analysis offers organizations the ability to monitor various social sites in real time and not accordingly.
- Utilization of sentiment analysis techniques in stock picking can lead to superior returns.
- Aspect level sentiment analysis is the most fine-grained analysis of review articles and social media snippets with respect to specific objects and their aspects.
- Augmentation to recommendation system.
- Detecting sensitive content webpages.
- Finding customers attitude and trends.
- Applications in politics, rulemaking and sociology.
- Prediction of sales performance.
- Public opinion polls.
- Stock market prediction.
- Box office revenues for movies.
- Election results predictions.
- Detecting heating languages in emails.
- Sentiment oriented question answering system.
- Sentiment analysis plays important role in market research.
- Useful in personalized medical diagnosis.
- Most commonly useful in online services.

Limitations and further study

The data set has not been pre-processed in order to execute the experiments in this project. Pre-processing is the process of cleaning and preparing data before it is delivered to the algorithms. Many irrelevant and uninformative aspects are common in online evaluations, and they may not even affect its orientation. Many processes, such as deleting white space and stop words, are involved in this procedure.

The findings of all of the studies show that when applied to review summaries, both approaches produce higher accuracy. The nature of the reviews could be one possible explanation for this outcome. Because the reviews themselves contain a significant amount of words, the bag of words features may be sparse. As a result, we can see that the algorithms' accuracies are higher for all experiments when applied to summaries that are more informative and have fewer words.

The text reviews were not pre-processed before being submitted to the classifiers in this investigation. Haddi et al. (2013), on the other hand, show in their study that pre-processing the data can improve the classifier's performance greatly. Their investigation showed that employing proper pre-processing methods can increase the accuracy of a classifier like SVM. More research should be done to see if pre-processing the training data set improves the outcomes.

Another drawback is that the bag of words approach ignores the position of words in a text, which can have a negative impact on the review's semantics. For example, despite the fact that the overall attitude of the review is negative, a computer might evaluate it as positive since it contains a large amount of positive terms (Pang et al. 2002). " This film sounds like it has a wonderful plot, great performers, and a good supporting cast, and Stallone is aiming to give a good performance. It will, however, fail."

This is the concept of "thwarted expectations," which has been identified by Pang et al. (2002) and Turney (2002), who stated that "the total is not always the sum of the parts." Another issue that needs be addressed in the topic of sentiment categorization is the identification of negation and its impact on sentence semantic interpretation. Future research could be useful in exploring this topic further and providing solutions.

In this study, only two major machine learning algorithms were investigated. Future research could look into other efficient sentiment classifiers, such as Decision trees.

Conclusion

Sentiment Analysis involves division of text on the basis of emotions it contains. Project on the emotional analysis that combines three key steps,i.e, data modification, review analysis and emotion analysis, and describes the strategies for representation involved.

Sentiment Analysis is the subject of one of the most emerging researches on text mining and integration languages, and contains a lot of research attention over the past few years. Future research shall involve sophisticated ways of finding an idea and extracting a product feature, and new classification models to address the structure of ordered labels in the scale. Applications using the outcomes of emotional analysis are expected to appear soon.

Sentiment analysis is the process of identifying the feeling expressed in the text or document. We proposed a methodology for mining the food reviews based on score combined with existing text analysing packages. The proposed system has produced a very good result using the score ratings. The limitation of this system is, it works better only for the open sentiments like rating or scores. The results were not promising for hidden sentiments. In Future work, prediction based methods will be implemented with existing approach. More features will be extracted to handle the implicit sentiment analysis.

References

- <div class="csl-entry">S., C., &#38; C., S. (2012). OPINION MINING AND SENTIMENT CLASSIFICATION: A SURVEY. <i>ICTACT Journal on Soft Computing</i>, <i>03</i>(01), 420–427.
- Callen Rain,”Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning” Swarthmore College, Department of Computer Science.
- Padmani P .Tribhuvan,S.G. Bhirud,Amrapali P. Tribhuvan,” A Peer Review of Feature Based Opinion Mining and Summarization”(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1), 2014, 247-250 ,www.ijcsit.com.
- Carenini, G., Ng, R. and Zwart, E. Extracting Knowledge from Evaluative Text. Proceedings of the Third International Conference on Knowledge Capture (K-CAP’05), 2005.
- Dave, D., Lawrence, A., and Pennock, D. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. Proceedings of International World Wide Web Conference (WWW’03), 2003.
- Zhu, Jingbo, et al. "Aspect-based opinion polling from customer reviews." IEEE Transactions on Affective Computing, Volume 2.1,pp.37-49, 2011.