# CIS 700
# Text Mining in Social Media

Project Report

# Sentiment Analysis on Amazon Fine Food Reviews

Karthikeya Bolla

SUID: 787064957

**Abstract**

Sentiment Analysis (also known as Opinion Mining) takes into account people's opinions and thoughts about specific products, organizations, events etc. and analyses the subjective information [1] in it. This analysis helps in making business decisions in real world scenarios viz. marketing a product, enhancing customer services that are being provided, understanding and identifying consumer market, providing product suggestions and many more. The project's scope focuses on one such domain viz. *Food* and tries to understand user opinions centered around food. Users opinions are gathered as reviews and are published by Amazon.com

# 1 Introduction

Humans evolve and enhance their cognitive ability by constant thought process and active reasoning. The thought process varies from one individual to another causing difference in opinion. We, the people have always consulted and counseled others about any important piece of information. What others think has always been important in critical analysis and decision making. Some examples include consulting parents about investments, asking friends for their opinion on vacation and many more, browsing through product reviews when buying a product and many more. The very consideration of others opinions has paved way for Sentiment Analysis. *Sentiment Analysis* also sometimes referred as *Opinion Mining* is defined as; *the study that analyzes people's sentiments, appraisals, opinions and emotions towards entities such as products, services, issues, organizations and topics* [excerpted from [2]]. Social websites such as Facebook and Twitter have always been benchmark sources for gathering user opinions on vivid variety of topics, events and organizations. Internet based e-commerce websites such as *Amazon*, *eBay*, *FlipKart* provide product reviews. Websites *Yelp* and *Zomato* provide food reviews, *AirBnB* provides property and apartment rental reviews. These all associate customer reviews with goodness of product, food and services being provided. Thus it is crucial to understand the paradigm of user reviews that drives potential customers' decision and consumer market.

## 2 Problem Definition

*Sentiment Analysis on Amazon Fine Food Reviews* focuses on food reviews and identifies polarity viz. *positive* and *negative* sentiment of reviews. A review is composed of *Summary* and *Text*. The project intends to understand the productivity of classifying a review solely based on review summary and later perform the same using review text i.e. *Is review summary more productive in classifying a review or is it review text that is productive in classifying a review ?* Reviews are processed and multiple classifiers are trained to classify reviews as positive and negative. Statistical measures are used to evaluate classification accuracy.

The dataset used is *Amazon Fine Food Reviews* provided by Amazon.com. The data is gathered over duration of 10 years with a timespan of October 1999 to October 2012. Reviews constitute user and product information along with provided ratings. The dataset is summarized [3] in below table:

| | |
|---|---|
| #Reviews | 568454 |
| #Users | 256059 |
| #Products | 74258 |

Table - 1

Sample entry from the dataset is showed below:

| | |
|---|---|
| ProductId | B001GVISJM |
| UserId | AJ613OLZZUG7V |
| ProfileName | Mare |
| HelpfulnessNumerator | 0 |
| HelpfulnessDenominator | 0 |
| Score | 5 |
| Time | 1304467200 |
| Summary | Twizzlers |
| Text | I love this candy. After weight watchers I had to cut back but still have a craving for it. |

Table - 2

# 3 Text Processing

The dataset available is composed of reviews that are of raw text. In order to perform sentiment analysis it is important to perform text processing. Prior to text processing, columns that are of least and no significance are removed. Columns viz. *ProductId, UserId, ProfileName,HelpfulnessNumerator, HelpfulnessDenominator, Time* are removed. *Score* is an ordinal attribute with value ranging from 1 to 5. Values are labeled as *positive* for scores 4 and 5, *negative* for scores 1 and 2. Reviews with score 3 are removed from the dataset as these reviews contain both positive and negative sentiment elements and their presence complicates classification process. Dataset after removing above mentioned columns looks as follows:

| | |
|---|---|
| Score | positive |
| Summary | Twizzlers |
| Text | I love this candy. After weight watchers I had to cut back but still have a craving for it. |

Table - 3

For text processing, the techniques used are:

- Numbers and punctuation marks are removed from the text
- Text is converted to lower case
- **Tokenization -** Text is broken down into finer strings where each string is a meaningful dictionary word that carries context in a language (here it is English)
- **Stemming -** Words obtained from tokenization are now stemmed. Stemming involves stripping affixes from a word and convert it to base form.
- **Stopwords -** Stopwords have been removed from the text as they do not add weight during classification process. There are two sets of stopwords that are being used. Initially, for the given dataset, stopwords provided by NLTK toolkit have been used which constitute 127 words. Classification is performed by removing stopwords from this set. Also, a new list of stopwords has been proposed as part of the project which

has 120 words (7 words being removed from NLTK based english stopwords list). On the original dataset, this customized list of stopwords has also been applied separately to understand how stopwords carry meaning in sentiment analysis of reviews. Thus by the end of data processing phase there are two stopword lists that have been used and stopword removal based on these two sets is applied on original dataset to obtain two different datasets which are text processed.

# 4 Classification Techniques

For the process of classifying reviews as *positive* and *negative*, classification algorithms have been used. The algorithms under consideration are:

## 4.1 Multinomial Naive Bayes

Multinomial Naive Bayes algorithm is one of the benchmark algorithms used for *Document Classification.* Multinomial Naive Bayes algorithm is based on Naive Bayes classifier [4] which is a probabilistic classifier based on Bayes theorem. According to multinomial Naive Bayes, the algorithm computes class probabilities for text documents and assigns them to one among the predefined set of classes denoted by $C$. Let $N$ be the vocabulary size, then for any test document $t_i$, multinomial Naive Bayes computes the probability value of assigning $t_i$ to $c$, where $c \in C$ using the equation [5] :

$$\mathrm{P}(c|t_i) = \frac{P(c).P(t_i|c)}{P(t_i)}$$

where, P$(c)$ is prior probability, P$(t_i|c)$ is the posterior probability and P$(t_i)$ is the evidence

$$\mathrm{P}(t_i|c) = (\Sigma f_{ni})!\ \Pi\ \frac{P(w_n|c)^{f_{ni}}}{f_{ni}!}$$

where $f_{ni}$ is word count of word $n$ of dcument $t_i$ and P$(w_n|c)$ is the probability of word $n$ given class $c$

$$\mathrm{P}(w_n|c) = \frac{1 + F_{nc}}{N + \Sigma_{x=1}{}^{N} F_{xc}}$$

where $F_{xc}$ is the count of word $x$ in taining document belonging to class $c$ The normalization term $P(t_i)$ is computed using below equation

$$\mathrm{P}(t_i) = \Sigma_{k=1}{}^{|C|}\ \mathrm{P}(k)\ \mathrm{P}(t_i|k)$$

## 4.2 Logistic Regression

Naive Bayes model and algorithms based on it viz. multinomial Naive Bayes model are *generative models.* A *generative model* estimates joint probability *p(x, y)*, where $x$ is input document and $y$ is the most likely class label which is assigned by applying Bayes rule *p(y | x)* on the learned joint probability model. On the other hand, *discriminative model* directly calculates posterior probability *p(y|x)* which is computationally effective [6]. Logistic regression is discriminative model.

Logistic Regression is used when the dependent varibale is categorical viz. *yes* or *no*, 1 or 0 and so on. Logistic regression models the relationship between categorical dependent variable and multiple predictor variables by calculating conditional probabilities using *logistic function.* Logistic function can be defined as a mathematical function that takes any input between negative infinity to positive infinity whilst output lies beteen zero and one [7]. Function capable of such transformation is *logistic sigmoid fucntion* [8] given by:

$$F(x) = \frac{1}{1+e^{-x}}$$

Where *F(x)* is the logistic function that maps the probability of occuracnce of an event to the continuous input variable $x$. Extending this to document classification, it can be written as:
For a binary classification problem where, for a given input document ***X*** which is the document's feature vector obtained by either bag of words model or TF-IDF model and the output is -1 or 1 is given by:

$$\mathrm{P(y{=}\pm1|x)} = \frac{1}{1+e^{-yw^Tx}}$$

where $y$ is the class label for which conditional probability is being estimated, $x$ is the input feature vector and $w$ is the weight vector, $w \in R^n$. $y_i \in$ (-1, 1). Logistic regression aims at minimizing the weight vector $w$ which leads to primal and dual forms [9] which are reduced forms of logistic function. The derivations of primal and dual forms are beyond the scope of this paper.

## 4.3 Support Vector Machines

Support Vector Machines are an intersting class of supervised learning algorithms proposed by Vladimir N. Vapnik and Alexey Ya. Chervonenkis [10]. They are used for data classification and regression analysis. As the name suggests, the algorithm builds support vectors that best seperate the data based on class information, thereby achieving higher classification accuracy.

Describing the algorithm, SVMs perform data classification by find *maximal marginal hyperplane*, this is the hyperplane that maximizes margin between two different classes repesented by data. The hyperplanes defined by vetors are called *Support Vectors*. For a given training set represented by $(X_1,y_1)$, $(X_2,y_2)$, ...... $(X_n,y_n)$, where $X_i$ is a feature vector (in the paradigm of document classification) of document $d_i$ formed by bag of words model or TF-IDF model and $y_i$ is the respective class to which the document $d_i$ belongs to (which in the current project scope is one of the two classes viz. *positive* or *negative*). The hyperplane that satisfies set of points represented by feature vector $X_i$ is:

$$\overline{w}.\overline{x} \text{ - b} = 0$$

where $\overline{w}$ is weight vector normal to the hyperplane. The parameter $\dfrac{b}{||\overline{w}||}$ is the offeset of hyperplane from the origin along normal vector $\overline{w}$ [Figure-1].

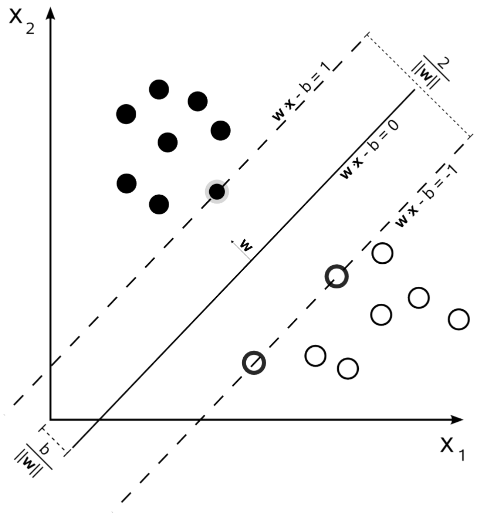
$x_2$
$x_1$
$\vec{w}\cdot\vec{x} - b = 1$
$\vec{w}\cdot\vec{x} - b = 0$
$\vec{w}\cdot\vec{x} - b = -1$
$\frac{2}{||\vec{w}||}$
$\vec{w}$
$\frac{b}{||\vec{w}||}$

Figure - 1

(Image credits: `https://en.wikipedia.org/wiki/Support_vector_machine#Linear_SVM`)

The equations representing support vectors is given by:

$$\overline{w}.\overline{x} \text{ - b} = 1$$
$$\overline{w}.\overline{x} \text{ - b} = \text{-}1$$

The distnace between these two support vectors is $\frac{2}{||\overline{w}||}$ and has to be maximized to achieve maximal marginal hyperplane. The above mentioned hyperplane and support vectors are used in linear SVM. SVMs can also also be extended to non-linear classification using mathematical kernel functions (non-linear) that transform the feature space to a higher dimensinal Euclidean space thus being non-linear. Some of the functions used for transformation are viz. *sigmoid fucntion*, *Gaussian radial basis fucntion*, *hyperbolic tangent* etc.

The efficiency of SVM depends upon choice of kernel and tuning parameters passed to kernel fucntion [10]. (Detail description about Support Vector Machines can be found at [11])

# 5 Results

The project has been implemented in Python 2.7. Natural Language Tool Kit (NLTK, version 3.1) has been used to perform tokenization, stopword removal, stemming and special character (numericals, punctuation) removal. Scikit-learn, a python based machine learning library (which is built on NumPy, Scipy and matplotlib) has been used to implement classifiers. Pandas (Python Data Analysis Library) has been used for data handling.

As mentioned earlier, the dataset *Amazon Fine Food Reviews* has been processed and resulting processed dataset is presented in section 3. The dataset contains 82037 reviews that are labelled as *negative* reviews and 443777 reviews labelled as *positive* reviews. During the intital tests there has been unprecedented results among classifers' accuracy measures. Detail investigations and multiple tests [12] have revealed that the difference in results is due to the inconsistency in class labels i.e. positve class of reviews are significantly higher that negative class reviews. Therefore, as part of text processing, data has been sampled such that equal umber of positive and negative class labeled data has been choosen. A total of 164074 reviews have been taken from the processed dataset of which 82037 reviews are labelled *negative* and 82037 are labelled *positive*. (82037 positive reviews have been randomly sampled from a total of 443777 positive reviews). Each review is considered as a document and each document is considered as group of words forming meaningful sentence. For feature extraction and classification, each document is processed and unique words are extracted from all (of 164074) documents. These words extracted form *bag of words* model. Upon the gathered bag of words, across each document, a feature vector is generated using TF-IDF (*Term Frequency - Inverse Document Frequency* [13]). As each document is of different length, the obtained TF-IDF vectors representing individual reviews have been length normalized. For model selection and evaluation, 5-fold cross validation has been used. Accuracy measure is the statistic used beacuse the labelled class data is unbiased (there are equal number of positive and negative labelled reviews).

Revising the problem statement, the project intends to understand the productivity of review summary as opposed to review text in classifying a review as positive and negative. Therefore there are two classes of test results under observation viz. accuracy measure of classifiers on review summary and accuracy measure of classifiers on review description. The results are tabulated below

% Accuracy on review summary

| Classifier used | not removing stopwords | removing stopwords | removing customized list of stopwords |
|---|---|---|---|
| Multinomial Naive Bayes | 87.101 | 84.899 | 87.123 |
| Logistic Regression | 89.604 | 85.752 | 89.422 |
| Support Vector Machines | 90.372 | 86.558 | 90.073 |

Table - 4

% Accuracy on review text

| Classifier used | not removing stopwords | removing stopwords | removing customized list of stopwords |
|---|---|---|---|
| Multinomial Naive Bayes | 88.78 | 88.333 | 88.49 |
| Logistic Regression | 90.942 | 90.432 | 90.767 |
| Support Vector Machines | 93.134 | 92.622 | 93.003 |

Table - 5

# 6 Analysis

Based upon the test results tabulated above in table - 4,5, solution for the problem statement is easily deducible. It is evident that among *review summary* and *review text*, *review text* helps better in understanding user sentiment and classifying a review as positive or negative. Also, among the three classifiers used, *Support Vector Machines* achieve better classification accuracy. The high accuracy can be attributed to variety of factors, some of them being kernel choice (linear kernel) and non-probabilistic mathematical model. Another important observation that can be made concerns *stopwords.* Looking at columns, *not removing stopwords* and *removing customized list of stopwords* from table - 4,5 it can be inferred that both these data processing techniques techniques provide relatively equal accuracy. As pointed out earlier in section 3 which described text processing involved in the project, it has been mentioned that a customized list of stopwords has been generated from standard english stopwords list provided by NLTK. The standard stopwords list provided by NLTK contained 127 english words considered as stopwords. Out of these 127 words, 7 words viz. *but*, *above*, *below*, *no*, *not*, *too*, *very* have been removed to generate customized stopword list. High classification accuracy achieved by not removing these words (as part of customized stopwords list) is easily understood by the behavior of these words in a sentence, most importantly reviews. It is clear that these words are highly polarized, that is, removing these words from a review alters a positve review to negative and vice-versa. Therefore it is important that these words should not be considered stopwords while classifying reviews as they carry potential information concerning user polarity.

# 7 Future work

Reviews have now become factors that assist users in making real time decisions. They alter consumers' perspective towards products and reviews. Sentiment analysis on such polarity indicators (reviews) is always challenging and equally an area of active research. The project intends to apply classification on unstructured text such as tweets and focus on variety of reviews datasets to derive patterns concerning sentiments. Also, behavior of stopwords needs critical analysis to make a final statement stating, *Is it time to reconsider stopwords?*

# 8 References

[1] https://en.wikipedia.org/wiki/Sentiment_analysis

[2] Morgan & Claypool Publishers, Sentiment Analysis and Opinion Mining, Bing Liu

[3] https://snap.stanford.edu/data/web-FineFoods.html

[4] https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[5] http://www.cs.waikato.ac.nz/ml/publications/2004/kibriya_et_al_cr.pdf

[6] http://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf

[7] https://en.wikipedia.org/wiki/Logistic_regression

[8] http://pages.cs.wisc.edu/~jerryzhu/cs838/LR.pdf

[9] http://www.csie.ntu.edu.tw/~cjlin/papers/maxent_dual.pdf

[10] https://en.wikipedia.org/wiki/Support_vector_machine

[11] http://cs229.stanford.edu/notes/cs229-notes3.pdf

[12] https://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf

[13] https://en.wikipedia.org/wiki/Tf%E2%80%93idf