NLP Sentiment Analysis
Amazon Home and Kitchen Product Reviews

1. INTRODUCTION

1.1. General

Sentiment analysis ,which is a subtopics of Natural Language Processing (NLP), has been gradually becoming more and more popular. It is a contextual mining of text which identifies and extracts subjective information in source material and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations.

Sentiment Analysis has many applications ranging from ecommerce, marketing, to politics and any other research to tackle with text or unstructured text data. Companies, especially in e-commerce, also do sentiment analysis to collect and analyze customer feedback about their products. Besides that, potential customers prefer to review the opinions of existing customers before they purchase a product or use a service of a company. As seen here, there are two parts in e-commerce; one is the online retailer, which wants to maximize e-commerce sales or services, and the other is the consumers, who want to have the best product or service over alternatives.

1.2. Problem Statement

In this project, Amazon is our client. The company wants to develop a software tool that will identify the positive and negative words which customers use when they write reviews for the home and kitchen products as their purchase inclination. For that, they gave their 14 years home and kitchen products' reviews between 2000-2014 and asked us to develop a model which will identify positive and negative words used in the reviews as a component of customer's sentiment towards to the company's home and kitchen products.

According to the customer request, we will build a sentiment analysis model as part of natural language processing, based on their reviews on the home and kitchen product online purchases. Our dataset consists mainly of customers' reviews and ratings.

1.3. Data Set Description

Home and kitchen dataset revolving around the reviews written by customers. This is a real commercial data.

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime
0	A1115ST6F5CWYP	B00000JGRT	Amalfi Coast Girl	[29, 33]	I have had one of these for about 10 years. I...	4.0	good for a first ice cream machine	1148256000	05 22, 2006
1	A188JOXWF4EY1R	B00000JGRT	Ann B. Hibbard "anbee"	[4, 4]	We actually found this product on clearance sa...	4.0	Wonderful Product!	1282176000	08 19, 2010
2	AUAX1QWUCYKSX	B00000JGRT	Ashley S	[1, 1]	This product works great, if the unit kept in ...	5.0	Works as expected	1243555200	05 29, 2009
3	A2C27IQUH9N1Z	B00000JGRT	audrey	[12, 13]	After trying other ice cream makers with mixed...	5.0	this will be one of your favorite small applia...	1043712000	01 28, 2003
4	A2PN65B6BSTIYZ	B00000JGRT	B. A. Chaney	[1, 1]	I bought this ice cream maker last summer and ...	5.0	You'll be addicted to homemade ice cream!	1214179200	06 23, 2008

Each row corresponds to a customer review, and includes the variables:

reviewerID : ID of the reviewer, e.g. A2SUAM1J3GNN3B - type: object

asin : ID of the product , e.g. 0000013714 – type: object

reviewerName : name of the reviewer – type: object

helpful : helpfulness of the review, e.g. 2/3 – type: object

reviewText : text of the review – type: object

overall : Rating (1,2,3,4,5)– type: float64

summary : summary of the review – type: object

unixReviewTime : time of the review (unix time) – type: int64

reviewTime : time of the review (raw) – type: object

The data was in Standford Analysis Project webpage. The original data was in a JSON format there. In order to analyze the data, I should change the data format. For that, I import JSON and decode JSON file with using query in order to convert JSON file to csv file format.

Data Source:
http://seotest.ciberius.info/seo--snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Home_and_Kitchen_10.json.gz

2. DATA WRANGLING
2.1. Inspecting the Dataset

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 25445 entries, 0 to 25444
Data columns (total 9 columns):
reviewerID        25445 non-null object
asin              25445 non-null object
reviewerName      25276 non-null object
helpful           25445 non-null object
reviewText        25445 non-null object
overall           25445 non-null float64
summary           25445 non-null object
unixReviewTime    25445 non-null int64
reviewTime        25445 non-null object
dtypes: float64(1), int64(1), object(7)
memory usage: 1.9+ MB
```

Amazon home and kitchen products data includes 25445 rows (observations) and 9 columns(feature variables) and its memory usage is 1.9+ MB. In the dataset, we have 7 object, 1 float64 and 1 int64 data types.

169 'reviewerName' information is missing in the dataset. Since customers don't give their identity, it may not be reliable to make an analysis on their reviews and ratings. I would prefer to drop the missing values from dataset since we have enough observations to conclude a prediction for sentiment analysis.

I concatenated 'reviewText' and 'summary' since both gave the approximately same type of information about product in text format, and later dropped both 'reviewText' and 'summary' columns.

'helpful' feature was dropped since I didn’t need that column for our model.

I classified the 'overall' (ratings) as good (rating 3,4, and 5) and bad (rating 1 and 2) in order to make sentiment analysis. I created a new column named as ‘rate_class’ from ‘overall’ column and converted its’ values as ‘good’ and ‘bad’. Later, we dropped ‘overall’ column.

In the dataset, 'reviewerID' and 'reviwerName' were used both for identification of customers. I dropped one of them from the dataset. Preferably, I dropped 'reviewerName' since customer names were not standardized and there were lots of different style to represent them in it.

'unixReviewTime' was dropped since it has already been represented in 'reviewTime' feature in a more understandable format. Also, 'reviewTime' was converted to datetime data type and a new 'year' column was created to make analysis between other variables in the future work. After that, ‘reviewTime’ column was also dropped.

I renamed the columns in order to improve practicality/readability of coding:

reviewerID : "customer"

asin : "product"

reviewText: This will be concatenated with "summary" and renamed as "review_text"

overall: "rating_class"

reviewTime: "year"

2.2. Descriptive Statistics

In our dataset, we have 1276 reviews, which have bad ratings whereas 24000 reviews which have good ratings.

We have 1395 unique customers and 1171 products in this dataset. Each customer averagely gives 18 reviews for products and on the other hand, there is averagely 22 reviews for each product in the website.

2.3. Preprocessing the Text

Since, text is the most unstructured form of all the available data, various types of noise are present in it and the data is not readily analyzable without any pre-processing. The entire process of cleaning and standardization of text, making it noise-free and ready for analysis is known as text preprocessing. In this section, I apply the following text preprocessing respectively.

Removing HTML tags

We wrote a function to remove the HTML tags which typically does not add much value towards understanding and analyzing text.

Removing accented characters

We wrote a function to convert and standardize accented characters/letters into ASCII characters.

Expanding Contractions

We wrote a function to convert each contraction to its expanded, original form in order to help with text standardization.

Removing Special Characters

We used simple regular expressions (regexes) to remove special characters and symbols which are usually non-alphanumeric characters or even occasional numeric characters.

Lemmatization

We removed word affixes to get to the base form of a word, known as root word.

Removing stopwords

We wrote a function to remove stopwords, which have little or no significance in the text.

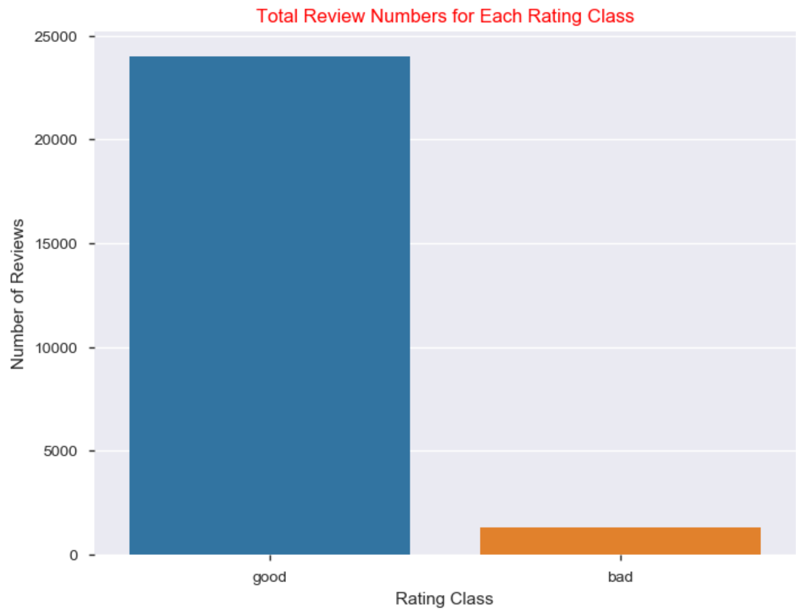
Building a Text Normalizer

Based on the functions which we have written above and with additional text correction techniques (such as lowercase the text, and remove the extra newlines, white spaces, apostrophes), we built a text normalizer in order to help us to preprocess the new_text document.

After applying text normalizer to 'the review_text' document, we applied tokenizer to create tokens for the clean text. As a result of that, we had 3070479 words in total.

Eventually, after completing all data wrangling and preprocessing phases, we save the dataframe to csv file as a 'Cleaned_Reviews_Home_and_Kitchen.csv. After cleaning, we have 25276 observations.

A clean dataset will allow a model to learn meaningful features and not overfit on irrelevant noise. After following these steps and checking for additional errors, we can start using the clean, labelled data to train models in modeling section.



3. EXPLORATORY DATA ANALYSIS

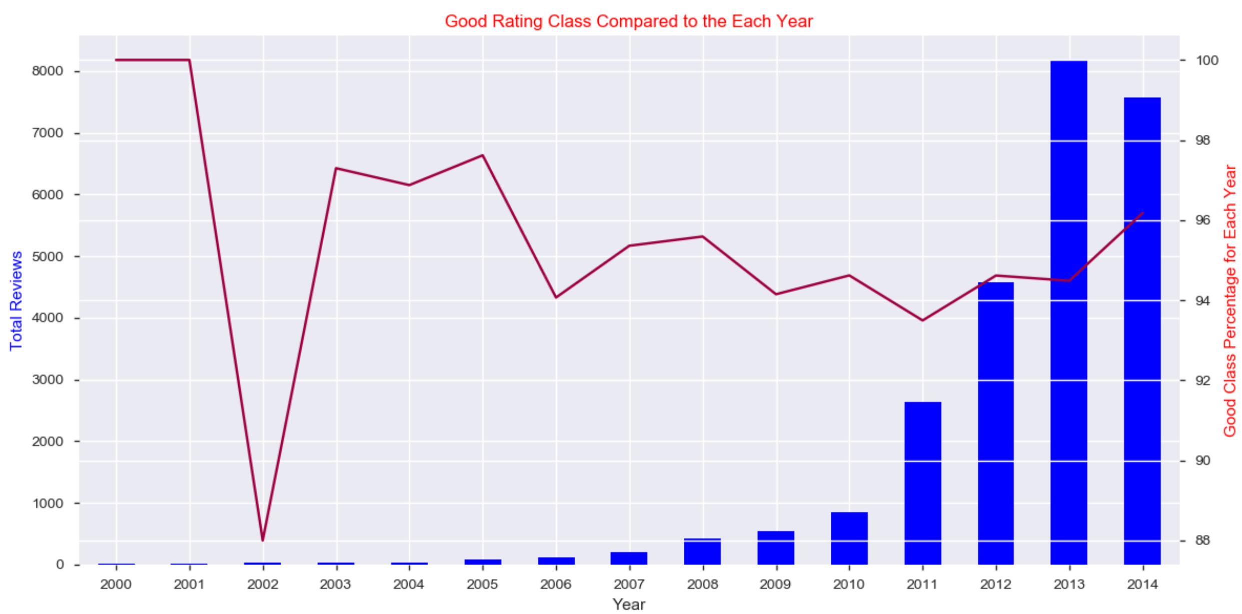
3.1. Target Variable :"rating_class" Feature

Customers wrote reviews and gave ratings, which ranged between 1 to 5, for each home and kitchen product they bought in the Amazon online market between 2000 and 2014. In overall, customers were seemed to be averagely satisfied with the products they purchased.

We diminished those 5 rating categories into two categories such as 'good' and 'bad' in order to develop a sentiment analysis model based on their reviews. According to those reviews, 95% of them (24000) are classified as good, whereas 5% of them (1276) are bad.

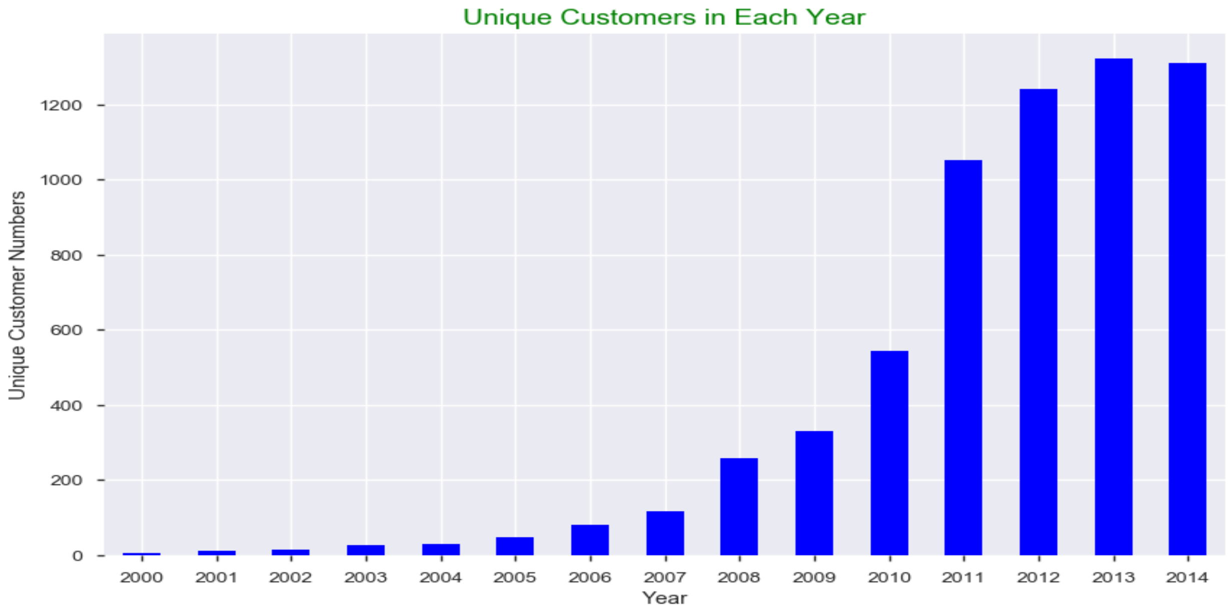
3.2. Other Features

3.2.1. "year" Feature



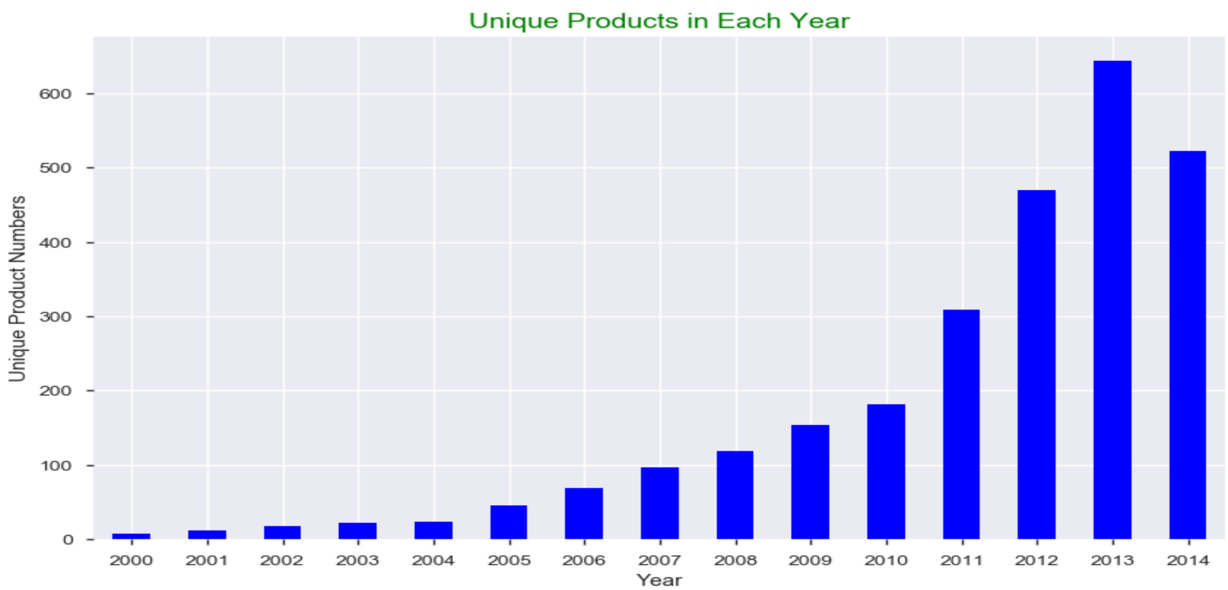
Except 2002, 'good ratings' percentage is progressing over 92%. 2002 has the lowest good ratings with 88% overall (There are only 25 reviews). 'good ratings' percentage is 100% in 2000 (10 reviews) and 2001 (16 reviews). As it might be seen in the graph, the overall good rating is progressing between 93% and 97% in home and kitchen products.

3.2.2. “customer” Feature



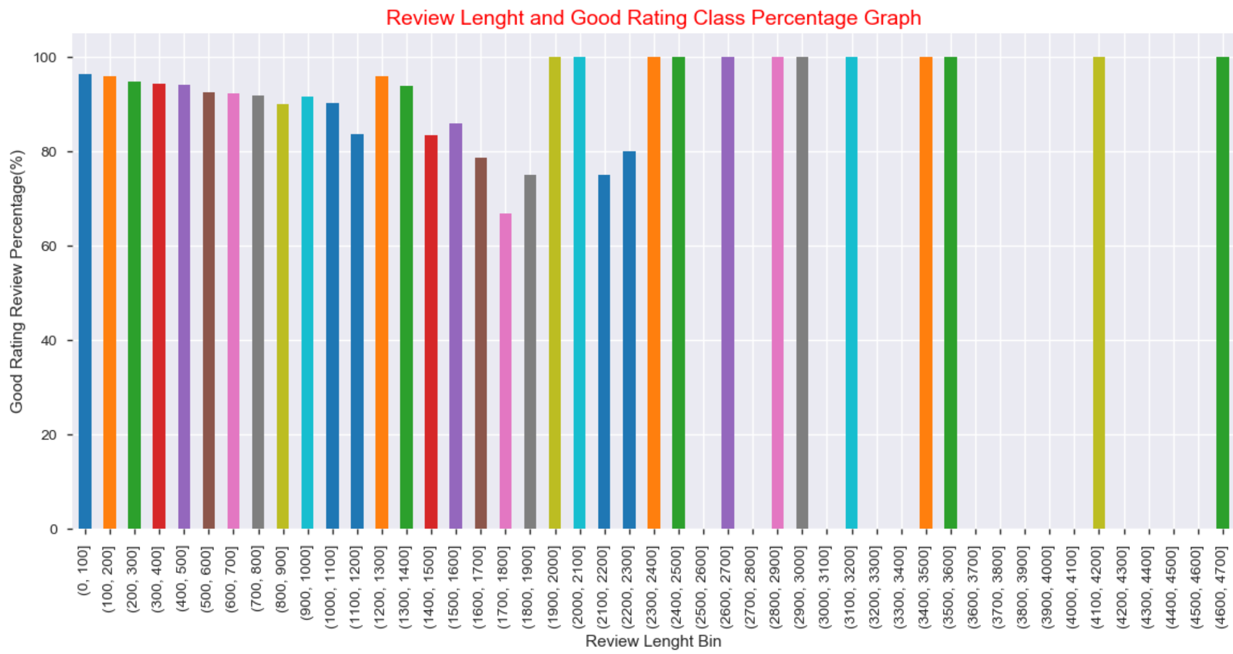
We have total 1395 unique customers who gave good reviews and 699 customers who gave bad reviews in the dataset. As it may be observed in the chart and table, the number of unique customers for each year has increased with the progress of the year.

3.2.3. "product" Feature



We have total 1171 unique products in the dataset which belongs to year between 2000 and 2014. As it may be observed in the chart and table, the number of unique products for each year has increased generally with the progress of the year except 2014. There is a slight decrease in 2014 but we have only data until June in 2014.

3.2.4. "review_ length" Feature



As it might be seen the graph, the highest percentage of good rating reviews lies between 0-1000 words with 96.2% whereas lowest percentage of good rating reviews lies between 1700-1800 words with 66.6%. As the review length extends, the good rating tends to increase. Generally, the customers who have write longer reviews (more than 1900 words) tends to give good ratings.

3.2.5 “Text Review” Feature

Good Rating Words:

	words	Avg		words	Avg		words	Avg
1	oven	0.91509	18	end	0.87255	35	nicely	0.85841
2	light	0.91045	19	comfortable	0.87129	36	second	0.85833
3	ever	0.90291	20	happy	0.87075	37	set	0.85827
4	cut	0.90099	21	side	0.87059	38	need	0.85787
5	especially	0.89655	22	new	0.86957	39	every	0.85714
6	might	0.89381	23	highly	0.86932	40	definitely	0.85714
7	done	0.89216	24	sturdy	0.86722	41	something	0.85714
8	find	0.89011	25	cooking	0.86705	42	baking	0.85577
9	try	0.8882	26	know	0.86458	43	kitchen	0.85488
10	dish	0.88288	27	room	0.86441	44	way	0.85484

11	shape	0.87851	28	le	0.86184	45	glass	0.85417
12	high	0.87845	29	food	0.8617	46	gift	0.85402
13	recommended	0.87805	30	cup	0.8617	47	like	0.85342
14	getting	0.87681	31	home	0.86066	48	best	0.85326
15	old	0.87662	32	everything	0.86014	49	fun	0.85294
16	could	0.8764	33	needed	0.85981	50	store	0.85276
17	another	0.87562	34	safe	0.85976			

The most common 50 words, which belong to good rating class, are shown in the table above. Each of these words define which products what kind of good impression have on the customers.

Bad Rating Words:

	words	Avg		words	Avg		words	Avg
1	machine	0.752212	18	say	0.79835	35	made	0.80941
2	perfectly	0.754717	19	cleaning	0.8	36	received	0.80952
3	bottom	0.76506	20	right	0.80091	37	first	0.80969
4	fit	0.776224	21	actually	0.80272	38	may	0.81035
5	big	0.776786	22	using	0.80451	39	problem	0.81035
6	space	0.779874	23	issue	0.80451	40	back	0.81068
7	attractive	0.784314	24	piece	0.80473	41	water	0.81108
8	although	0.784314	25	inside	0.80556	42	take	0.81139
9	three	0.787037	26	item	0.80591	43	nice	0.81191
10	amount	0.790476	27	holder	0.80734	44	long	0.81197
11	though	0.790698	28	day	0.80745	45	grip	0.8125
12	counter	0.79085	29	going	0.80791	46	used	0.81426
13	simple	0.792453	30	floor	0.80833	47	small	0.81461
14	wash	0.792593	31	stick	0.8087	48	enough	0.81544
15	pot	0.793103	32	worth	0.80909	49	worked	0.81553
16	give	0.793548	33	press	0.80916	50	house	0.816
17	little	0.797849	34	pan	0.80928			

Same standards as above, the most common 50 words, which belong to bad rating class, are shown in this table. Likewise, in good ratings, each of these words define which products what kind of bad impression have on the customers.

Controversial Cases:

The controversial case such as "I was expecting better - negative meaning" or "it was better than my expectation - positive meaning " will be handled in the modelling section via using deep learning technique (Keras with Word2Vec).