NLP - Sentiment Analysis Project
Beauty Product Reviews

Engin Turkmen

March 15, 2019

Contents

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:

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 beauty products as their purchase inclination. For that, they gave their 9 years beauty products' reviews between 2005-2014 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 beauty 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 beauty product online purchases. Our dataset consists mainly of customers' reviews and ratings.

1.3 Data Set Description:

Beauty dataset revolving around the reviews written by customers. This is a real commercial data.

This data includes 28798 rows and 9 feature variables. Memory usage is 2.2+ MB.

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime
0	A6VPK7X53QNAQ	B0000CC64W	AmazonDiva "Keep Calm and Carry On."	[5, 5]	I am a devotee to this serum, it does wonders ...	5.0	If I had to choose only one product to take ca...	1245283200	06 18, 2009
1	A3CHMHGSJSQ02J	B0000CC64W	Anon. A. Non	[2, 2]	As a woman nearing 50, I need all the help I c...	5.0	Makes my skin lovely and smooth	1358467200	01 18, 2013
2	A1V1EP514B5H7Y	B0000CC64W	asiana	[0, 0]	I've used this regenerating serum for more tha...	5.0	Works well at a reasonable price	1322524800	11 29, 2011
3	A1X2LENOF84LCQ	B0000CC64W	D "D"	[62, 75]	I have tried so many products to just be total...	4.0	This does work ladies	1113350400	04 13, 2005
4	A2PATWWZAXHQYA	B0000CC64W	Farnoosh Brock	[1, 1]	I love Oil of Olay. My primary moisturizer is ...	1.0	Did not like the feel/texture of this serum	1387584000	12 21, 2013

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

I have downloaded the beauty product review file via link below and opened the file with coding in Jupyter notebook.

Data Source:
http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Beauty_10.json.gz

2. DATA WRANGLING

2.1 Inspecting the Data Set:

```
# Basic information on Dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 28798 entries, 0 to 28797
Data columns (total 9 columns):
reviewerID        28798 non-null object
asin              28798 non-null object
reviewerName      28576 non-null object
helpful           28798 non-null object
reviewText        28798 non-null object
overall           28798 non-null float64
summary           28798 non-null object
unixReviewTime    28798 non-null int64
reviewTime        28798 non-null object
dtypes: float64(1), int64(1), object(7)
memory usage: 3.4+ MB
```

Amazon beauty products data includes 28798 rows(observations) and 9 columns(feature variables) and its memory usage is 3.4+ MB. In the dataset, we have 7 object, 1 float64 and 1 int64 data types.

222 **'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.

We 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 we didn't need that column for our model.

We classified the **'overall'** (ratings) as good and bad in order to make sentiment analysis. For that, we dropped observations whose 'overall' columns' values are equal to 3 since that rating group doesn't give an exact opinion about product whether it is good or bad. We 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. We 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.

We 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 2084 reviews which have bad ratings whereas 22425 reviews which have good ratings.

We have 1340 unique customers and 733 products in this dataset. Each customer averagely gives 18 reviews for products and on the other hand, there is averagely 34 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.

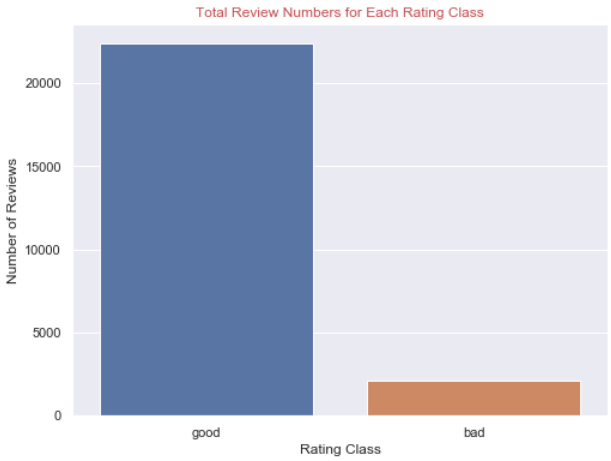
After applying text normalizer to ‘the review_text’ document, we applied tokenizer to create tokens for the clean text. As a result of that, we had 1706537 words in total with a vocabulary size of 25023. Max review length is 1090 whereas min review length is 1 as a word based.

Eventually, after completing all data wrangling and preprocessing phases, we save the dataframe to csv file as a ‘cleaned_dataset’.

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. DATA STORYTELLING

3.1 Target Variable ("rating_class")



Customers wrote reviews and gave ratings, which ranged between 1 to 5, for each beauty product they bought in the Amazon online market between 2005 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, 91.5% of them (22425) are classified as good, whereas 8.5% of them (2084) are bad.

3.2 Features

3.2.1 "Year" Feature

After 2012, good ratings' percentage is progressing over 90%. Before 2012, only 2009 also shows a slightly rapid increase in good ratings from 87.5% to 90.7%. Besides those, 2011 has the lowest good ratings with 85% overall. As it might be seen in the graph, the overall good rating is progressing between 85% and 93% in beauty products.

3.2.2 "Review Length" Feature

As it might be seen the graph, the highest percentage of good rating reviews lies between 0-50 bin with 96.4% whereas lowest percentage of good rating reviews lies between 400-450 bin with 88.3%. As the review length extends, the good rating tends to decrease slightly. The difference between good rating review percentage with shortest and longest of review length bin is 8.1%. Insightfully, the customers who have complains about the products are more willingly to write longer reviews than other customers who are satisfied with company's products.

3.2.3 "Text Review" Feature

Good Rating Words:

Words	Avg.	Words	Avg.	Words	Avg.	Words	Avg.
rosehip	1.0000	hyaluronic	0.9838	combo	0.9788	additional	0.9753
shany	1.0000	fantastic	0.9836	nail	0.9787	importantly	0.9750
adovia	1.0000	tree	0.9834	dropper	0.9786	amazing	0.9742
instanatural	0.9945	wonderfully	0.9832	win	0.9784	complimentary	0.9741
palette	0.9932	awesome	0.9829	gym	0.9782	reasonably	0.9739
express	0.9918	shipping	0.9828	compliment	0.9780	scar	0.9739
mud	0.9908	gift	0.9813	highly	0.9777	men	0.9737
mencare	0.9903	sea	0.9811	vera	0.9776	handy	0.9735
cucumber	0.9903	bonus	0.9810	youthful	0.9772		
pleasantly	0.9879	heel	0.9809	exchange	0.9764		
relief	0.9862	economical	0.9807	winner	0.9760		
excellent	0.9849	buildup	0.9803	polish	0.9757		
eczema	0.9842	conjuction	0.9801	bath	0.9754		
incredible	0.9840	suprisingly	0.9801	pleased	0.9754		

Fixing the rating count value is above 100, 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. For example, 'mencare' and 'men' words tell male beauty products are more appreciated in the reviews. On the other hand, 'eczema', and 'scar' tell some beauty products are praised for covering them. 'economical' and 'shipping' words might give the insight that products are accepted as reasonably priced and conveniently shipped to the customers.

Bad Rating Words:

Words	Avg.	Words	Avg.	Words	Avg.	Words	Avg.
sorry	0.4818	wax	0.7168	perhaps	0.7617	understand	0.7860
disappointed	0.5154	john	0.7169	skip	0.7627	ok	0.7863
unfortunately	0.5365	throw	0.7277	marketing	0.7642	intend	0.7865
fail	0.5865	glove	0.7298	direct	0.7658	shake	0.7867
attempt	0.6261	barrel	0.7401	dirty	0.7703	african	0.7867
awful	0.6328	wand	0.7401	gross	0.7714	possibly	0.7870
terrible	0.6328	weird	0.7412	initially	0.7737	odd	0.7877
horrible	0.6643	consumer	0.7445	waste	0.7757	idea	0.7878
toss	0.6730	frieda	0.7452	nothing	0.7785		
hop	0.6827	direction	0.7478	doubt	0.7800		
awkward	0.6959	miss	0.7500	okay	0.7812		
american	0.7045	description	0.7553	hope	0.7822		
whatsoever	0.7075	maybe	0.7583	instruction	0.7847		
impossible	0.7129	battery	0.7611	strange	0.7857		

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. For example, 'wax' products have mostly used to complain. 'description' word gives insight that the product's usage is not clearly depicted in the description or beauty product has side effects which the description fails to explain.

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).