Problem Chosen	2020 MCM/ICM Summary Sheet	Team Control Number
C		2015970

Prudence of Versatile Online Company: We Just Know Consumers'Minds

Summary

In recent years, expansion of online transaction spurs the circulation of network information exchange between merchants and customers. Not only regarded as compensation mechanism in customer services, ratings & reviews also play dominant roles in market feedback chains that facilitate enterprise soundness. A variety of advanced methods have been applied to figure out the internal correlation between ratings and reviews. However, a comprehensive and unequivocal set of strategies are needed for customizing enterprise online business programs targeted at improving product desirability.

We manage to develop a model that effectively explains the relationship between 3Rs (ratings, reviews and ratings of reviews) and directly offer practical strategies. The main works of this paper can be concluded as follows:

- We utilize the combination of model TF-IDF and Word2Vec to quarry the information of text data and turn them into sentimental ratings to serve as text-based measures.
- In light of the ratings-based patterns and text-based approaches, we rate reliability of each review in LightGBM model to clarify reviews that worth tracking the most.
- We design the time-based measures that can reflect the fluctuation of the products' reputation.
- We adopt Association Rule Analysis to find the combinations of text-based measures and ratings-based measures that can reveal the success or failure of the products.
- We use the Dynamic Time Warping(DTW) model to determine that specific star ratings indeed incite more related reviews.

Our model illustrates inner relationship between the 3Rs and the potential of the products. Analyzing from multiple perspectives, we elucidate the measures to maximize customer information availablity. By visualizing the reputation of each product, we can monitor the fluctuation of reputation with respect to time. In line with these time-based measures, we study the following trend phenomenon of the reviews and ratings.

To evaluate the sensitivity of our model, we calibrate the parameters of the text preprocessing model and input the cleaned text-data and the uncleaned ones respectively. The results confirm the stability and dependability of our model.

In conclusion, our models cover a wide range of analysis in response to the concrete requests of Sunshine Company. We also point out the need for future applications in relevant areas. We believe our models will benefit Sunshine Company in a profitable way.

keyword: Logistic Regresion; LightGMB; Apriori algorithm; DWT

A Letter to Market Director of Sunshine Company

Dear Market Director,

As consultants of Sunshine Company, it is our obligation to assist you in data analysis regarding the relationship and connections of 3Rs (Ratings, Reviews and Ratings of Reviews) of online shopping platforms. We are writing this letter to report our latest findings.

We design Sentiment Prediction Model to investigate the correlation between ratings and reviews. As ratings denote the quality of products in a general sense, the ratings of reviews signify whether a review can represent the experience of a common customer. To address the problem of variety of reviews, we conduct two operations, stemming and merging, to standardize the input. Two false scenarios, five stars on products of poor quality and one star mistakenly marked on genuine reviews, are 95% filtered out by Logistic Regression. After an in-depth study of sentiment prediction, we conclude that certain descriptors of quality are remarkably bound up withs specific ratings, which is commensurate to our common sense.

In view of already available sentimental scores, we establish Dynamic Time Warping Model to research the connections between particular rating levels and the amount of reviews. The difference between the warped signals and original signals are trivial, which is the mathematical proof that specific rating levels are more likely to incite more reviews. We can also induce that among three products, pacifiers cling to corresponding reviews to a greater extent.

Additionally, we develop Association Rule Analysis to examine the text-base and rating-based measures that represent the soundness of the products. We use gradient descent decision-making tree to filter the word vectors. This method not only accelerates the algorithm, but it also help us gain parameters shown in Appendix 11.

In conclusion, our model successfully explains the correlations between ratings and reviews. Because the models and evaluation approaches are data-propelled, its applications in future online marketing problems, especially in customer services, are promising.

Integrating the analysis above, we propose following strategies for your consideration:

1. **Trace the source of quality texts.** Synthesizing customer advice can be tough. The list regarding quality descriptors and coefficient can represent the importance level and similarities of different customers' voice. We recommend you retrieve the reasons why customers leave their comments by locating the source of customers

which is feasible owing to trading recordings. Refined feedback chart in accordance with coefficient (importance) can better facilitate the improvements of your company.

2. **Treat three products with different perspectives.** We find that in the list of coefficients concerning three products, though the top positive coefficients are almost identical, the absolute value of negative coefficient of pacifier is saliently greater. The compatibility of our parameter indicates that for the sake of long-term soundness of your enterprise, more attention and subtlety are required in ameliorating the quality of pacifier.

3. **View text-based and time-based measures in time-series patterns.** Views of customers can alter in time. We deem it plausible to conduct our operations in time-series patterns for the generality of business practices.

We have a strong belief that our model can effectively enhance the efficiency of online customer service and provides an appropriate as well as feasible way to best improve the business performance of your products.

Sincerely yours,
MCM 2020 Team #2015970

Contents

1 Introduction

1.1 Background

Platforms for the public to swap purchase experiences are designated to spur the proceedings of trade process. Rating systems, especially in terms of reviews, are of significance in opinion alteration for prudent developers.

From the perspective of contributors, customers are inclined to go about browsing previous feedback, i.e. reviews prior to their purchases. Helpfulness ratings serve as pre-perceived expertise during other customers' purchase decisions or simply become a practice of values interaction [1].

On the other hand, customer-based texts can constitute a recommendation in a recommender-rich environment like Amazon. No recommender system can be 100% correct or produce systematically novel and haphazard results[1, 2] . To some extent, ratings of products and of reviews are dominant factors presented on a public platform without voir dire, which might offer misleading information about the merchant.

Consequently, crucial analysis in recessive factors like selection of reviews that possess the most valid contents allows e-commerce companies to make improvements and adjustments to proffer products with high quality and subtlety[3].

We place great emphasis on the correlation between customer reviews, ratings and the ratings of reviews to determine period-choice sales patterns and help key features of three products to percolate down the industry for Sunshine company.

Figure 1: Mapping description of the relationship between Amazon reviews and users via Visio.

[1] https://www.amazon.com/gp/help/customer/display.html?nodeId=G3UA5WC5S5UUKB5G

1.2 Related Works

Danescu and Kossinets[4] took Amazon as an example to assess how opinions are evaluated by members of an online community in a large scale. They found out that a review's helpfulness depends not only on the content, but also on the relation of its score to other scores. The result resonates with the findings by Leino et al.[7], which underscore the significance of a recommender system as a whole. Direct recommendations proffer clients a closer scrutiny of their preferences.

Nicolau[5] noted that extreme ratings (positive or negative) coexist with polarized sentiments with asymmetrical effects. A new dimension of reviews called "trustfulness" was developed by Kokkodis[6] to construct models to rank customers, which later to be proved thought-provoking in future application scenarios like whether to put the review contains information exposed only to buyers.

Models above have presented overwhelming standpoints on the impact of local features or recommender system. However, for e-commerce companies like Amazon and e-Bay, comprehensive trade strategies entail a deliberate combination of measures to ensure enterprise's soundness.

We adopt credibility to ascertain the relationship of reviews and ratings of 1st and 2nd order. In our model, a series of word vectors are generated from reviews to execute LightGBM for ideal output to informative reviews. We use a data set originated from Amazon to test our theories to prove efficacy and robustness.

2 Fundamental Assumptions

Since the data of three products are given, we make following assumptions to help us operate modeling.

- We presume that the impact Neutral reviews (3-star reviews) can be neglected due to its trivial portion among all comments.
- For simplicity, we suppose customer reviews are recognizable and logic comments with semantics. Therefore, we can extract useful texts with actual meanings.
- We postulate that rate of helpful votes indicates effectiveness of a review. In other words, the quality of a review is determined by the number of thumb-ups.

3 Notation

Symbols	Definitions
x	the word vector
w	word vector corresponding to review x
y	a binary value like 0 or 1
C	credibility
h	amount of helpful votes in a review
v	total votes in a review
NPS	Net Promoter Score
P	reputation of the product during period Δt
R	reputation of a product
H	weight of a review
A	star rating matrix
B	sentiments score matrix
$Supp(X \bigcap Y)$	Support of item sets X and Y
$Conf(X \leftarrow Y)$	Confidence of X to Y
D	weighted distance between series
Γ_i	the potential vlaue for P_i

4 Rating-based Measure

Before identifying the rating-based measures, the rating distribution of the same amount of sampling products can be visualized as below

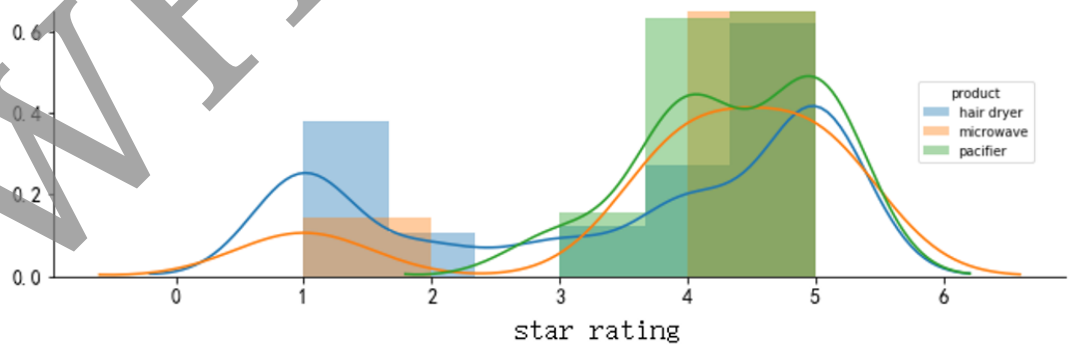


Figure 2: star rating distribution of sampling product.

To measure quantitative and qualitative patterns of ratings, we use the following common metrics, and the calculation results are shown in the table 1.

Skewness - Skewness is a measure of the deflection direction and degree of statistical data distribution. It is a numerical characteristic of the degree of asymmetry in the distribution of statistical data, which reflects intensity of the same rating. It is calculated as $\text{Skew}(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right]$.

Net Promoters Score - As a gauge that represents how the enterprise and employees treat customers, NPS signals the possibility that users recommend products and services to their colleagues and friends. In this sense, customers are classified as recommenders, passives and detractors. NPS helps us evaluate customer satisfaction and loyalty[19]. NPS can be formulated as

$$NPS = (Promoters - Detractors)/Totalratings * 100 \quad (1)$$

where the reviews with rating 1 and 2 are regarded as Detractors and rating 3 denotes Passives while the ones with rating 4 and 5 are Promoters.

Table 1: Metrics of rating in three products.

Product title	Average	Skewness	NPS
andis fold-n-go ionic hair dryer	4.5129	-1.51	73.66
pearl ceramic hair dryer	4.6092	-1.33	75.55
watt tourmaline ceramic hair dryer	4.6252	-1.54	73.41
cu.ft. countertop microwave	4.4476	-0.98	71.54
wmc20005yw countertop microwave	3.8654	-1.41	62.47
sharp microwave drawer oven	3.9240	-1.31	67.58
free contemporary freeflow pacifier	3.5277	-2.58	57.54
free soothie pacifier	4.2666	-2.69	71.15
wubbanub infant pacifier	3.9115	-1.99	64.48

Table 1 merely shows the most-reviewed products' rating measures. Detailed results are available in the appendix 11.

5 Text-based Measure

To quantify the text information in reviews and identify the text-based measures, we execute sentimental analysis to the reviews in this phase.

Based on the measures, we gain the specific quality descriptors that are strongly associated with rating levels.

5.1 Text preprocessing

Text data should be cleaned and encoded into numerical values before operating machine learning models. This process of data cleaning and encoding is defined as Text Preprocessing.

5.1.1 Data Cleaning

We define the reviews with Score 1 and 2 as Negative reviews and let Score 4 and 5 be Positive reviews. The Neutral reviews with Score 3 are removed, which merely account for a small portion of the database. Then the word vectors[8] are obtained after converting all the words to lowercase and removing punctuations. The we utilize the Stemming and Stopwords strategies to simplify the word vectors.

Stemming - Transforming the words into their base words or stem words. This reduces the vector dimension. For instance, we merge homologous words like *'efficiently'*, *'efficient'* into their stem word *'effect'*.

Stopwords - Filtering words without actual semantemes like 'This microwave is so efficiently' become 'microwave', 'efficiently', then stopwords are removed.

5.1.2 Encoding

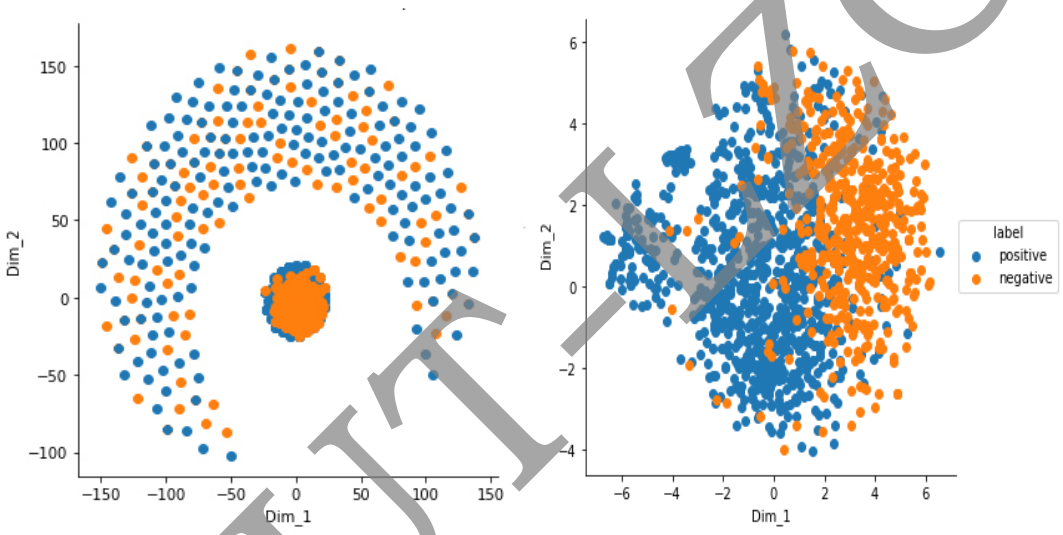
Raw text data should be converted into numerical vectors that can be recognized in machine learning. Thus, we adopt the popular encoding techniques as follows:

- **Bag of Words [8, 9]:** Bag of Words is a method that computes the frequency of the appearing times of each word. The vectors it creates rest with the frequency of words instead of order. Using cluster methods like K-means, we can classify words just like separating things in bags.

- **TF-IDF [10] :** Term Frequency - Inverse Document Frequency guarantees that the less important but given to the most frequent words are also considered as less frequent words. For instance, articles like "a/an" and "the" are considered to be meaningless though they appear ubiquitously in texts.

- **Word2Vec [11, 12, 13]:** Word2Vec actually draws the semantic meaning of the words and their relationships between other words. It extracts all the internal relationships between the words and turn the word into a vector form.

We utilize t-distributed Stochastic Neighborhood Embedding[14] to visualize the high-dimension word vectors to be 2-dimension surface as Figure 3.



(a) TSNE for TF-IDF (b) TSNE for Word2Vec

Figure 3: Vectors that perform well in classification.

Each point in Figure 3 can represented a review in the dataset. The color of plot would be blue when the review is positive, while it would be orange as the review is negative. As the positive and negative points can be separated in general, one could expect that the TF-IDF and Word2Vec are able to separate the class labels in our dataset. Thus, we merge the word vectors in the two model into each other to gain the text features.

5.2 Sentimental Analysis

Before computing the sentimental rating for the each review, we would determine which words that are strongly associated with rating levels and draw the sentimental rating of this words based on the text feature yielded in text preprocessing stage. Then the text-based measure can be gained by summing up the sentimental ratings of the words in each review.

5.2.1 Logistic Regression

We adopt Logistic Regression to execute the binary classification for each review. The core model can be give as

$$y = \frac{1}{1 + e^{-(w^T x + b)}} \tag{2}$$

where x is word vector, y is a binary value. We apply the gradient descent algorithm and define the loss function as $L = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$, then we compare the predictions with the Bayesian algorithm [16] as below

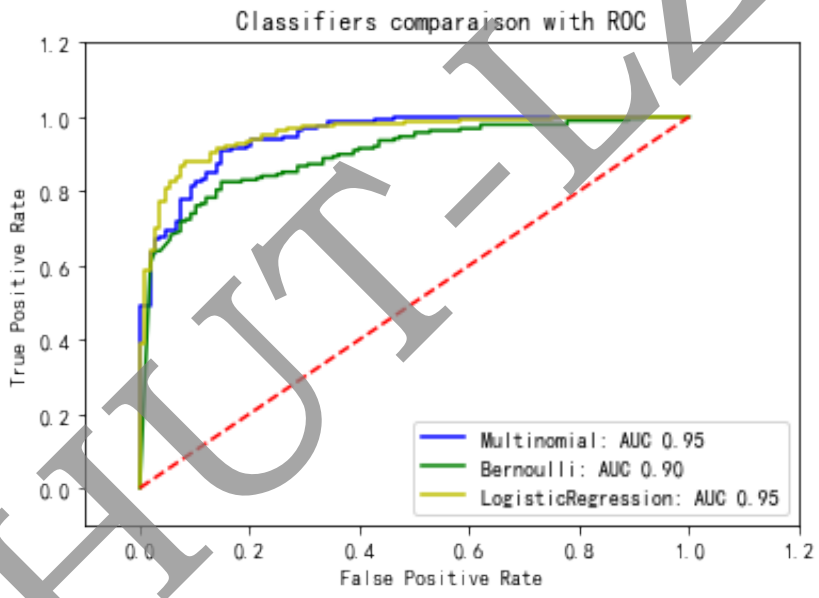


Figure 4: The ROC curve for different algorithms.

Figure 4 reveals that Accuracy is around 95.9%. Top 10 negative/positive words are listed in Table 2. The coefficient reflect the confidence to result, which are positive correlation to the classification accuracy when the word is identified as positive word but negative correlation when negative. As all $|coefficient| > 1$, the classification result can be deemed to be reliable.

Table 2: Logistic regression model on word count.

positive word	coefficient	negative word	coefficient
love	2.020178	stop	-2.655619
excel	1.967720	disappoint	-1.985671
perfect	1.775876	wast	-1.935428
great	1.775876	junk	-1.736096
nice	1.727514	suck	-1.695555
definit	1.698679	fail	-1.679773
awesom	1.633642	defect	-1.612045
best	1.612823	poor	-1.595778
amaz	1.403347	worst	-1.464123
fast	1.259788	broke	-1.567801

The WordNetLemmatizer from NLTK[2] can be applied to determine the morphemes by gathering word class. Horizontally examining 2-gram and 3-gram, one can analyze ratings of different products and extract identical characteristics of the contexts. Focusing on popular single **adjective** rating words, one can gain phrases that occur more than benchmarks of reviews. We take the database of hair dryer as an example, the result could be showed in the cloud map as follow

(a) Negitive words (b) Postive words

Figure 5: Certain descriptors of quality are evidently bound up withs specific ratings.

[2] Library from http://www.nltk.org/nltk_data/

Figure. 5 shows the association strength between each word of the reviews and its corresponding rating. It can be intuitively expressed as the bigger the word, the stronger the association. The detail of result can be checked in appendix.1

5.2.2 Sentimental Rating

The attitude of the review R to the product can be expressed in Sentimental Rating E, which can be formulated as

$$E_i = \sum_{n=1}^{N} e_n, e_n \in R \tag{3}$$

where N denotes the length of the review R. e_n is the coefficients of each word in the review R. The sentimental rating of the reviews can be illustrated in table 3 . Obviously, the higher the rate, the more positive the review is. The lower the rate, the more negative the review is. The detail data can be checked in appendix 11 .

Table 3: Sentimental rating measure for each review

Review id	Review title	Sentimental rating
R1SO9VMCIGZX3U	*Love this!*	24.020178
R2E7N0TVLUHUDR	*total junk - get a munchkin brush instead*	-11.512384
R1A3ZUBR8TSAKY	*Four Stars*	8.748561
...	...	...
RLJNYBK4FGBYX	*Five Stars*	17.712887
R26QCW75C4JDOK	*Kid loves it, so I do to.*	14.191633

6 Time-based Reputation Model

Due to result of Sentimental Analysis that some reviews might show attitude deviating from their star rating, we draw up the credibility of the reviews and ratings. Thus, the model can reflect the change of products' reputation reliably.

6.1 Review Credibility

The credibility C is defined to reflect the effective information contained in the reviews and ratings. It can be simply formulated below

$$C = \frac{h}{t}(t \neq 0) \tag{4}$$

where h denotes the amount of voters that vote the review as helpful and t represents the total of voters. As we assume all the reviews with $t = 0$ have the C equal to 0, the relation matrix of C and star rating of the database of baby pacifier can be give as below

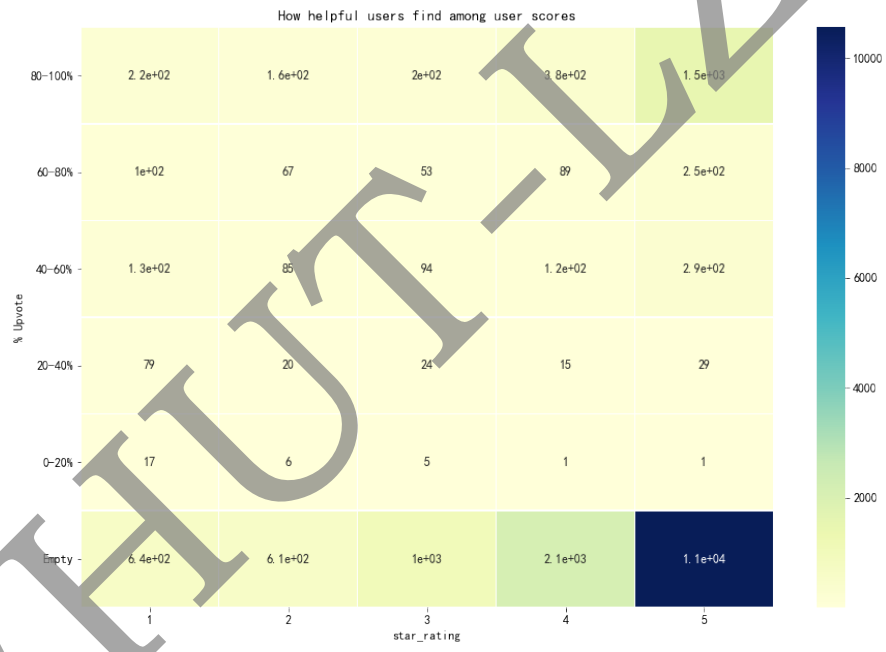


Figure 6: The relationship between score and voting

The elements in the matrix is the quantity of corresponding reviews. It can be analyzed that

a. Most of the reviewers are inclined to give a high rating.

b. More than half of the reviews have no vote, especially the recent ones.

We would utilize the LightGBM algorithm to recompute the credibility C for all reviews in following stage. Thus, we could gain the exact credibility of all reviews even the non-vote ones that are too new to be noticed.

6.2 LightGBM Recomputing

The LightGBM[17, 18] model shows high performance in processing the features with different measure[3] . We take a part of early reviews as training set and output the credibility of later reviews.

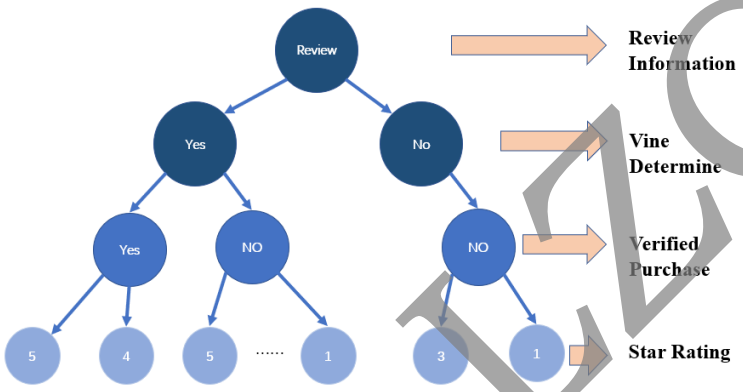


Figure 7: Schematic diagram of various features decided by LightGBM.

In LightGBM model, we recompute the credibility C for each review shown in table.4. Obviously, the credibility of vines' reviews is above average. And the credibility of later reviews is basically the same as the early one.

Table 4: Credibility of microwave ovens' review

Review id	Star rating	Vine	Date	**Credibility**
R3HT7OGKQO8Q0E	5	Y	2010-11-22	0.762627
R3LXL05NPDK7P2	5	N	2010-11-23	0.566363
RWZZ77PTTHCDV	4	Y	2010-11-28	0.928533
R2JK82HHMUGGOU	1	N	2010-12-21	0.089120
...	...	...	...	...
RJ4HTF6UC3JGC	2	N	2015-08-22	0.240850
R1X47WDNBT4OHZ	1	N	2015-08-23	0.843194
R9T1FE2ZX2X04	5	Y	2015-08-31	0.710421

[3] For example, the length of review and whether the reviewer is a vine are both feature of the review but have different measure.

6.3 Reputation model

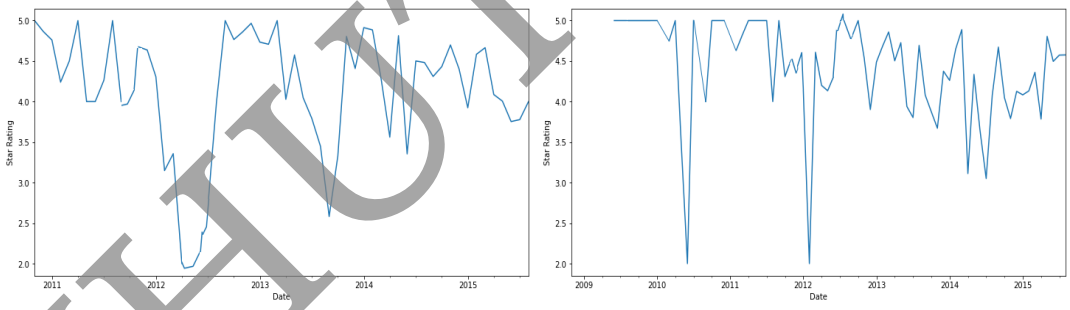
Before showing the fluctuation of the products' reputation, we draw up the model to quantify reputation of the product in a fixed time period ΔT, which can be give as

$$P = \frac{\sum_{n=1}^{N} r \cdot H}{\sum_{n=1}^{N} H} \tag{5}$$

where P reflects the reputation during the time slot ΔT, in which the amount of updated reviews equal to N. r denote the star rating of the review. H is the weight for each review, which can be yielded as

$$H = \begin{cases} C, (V = 0) \\ 10C, (V = 1) \end{cases} \tag{6}$$

where C denote the help-rating, and the index V denoted whether the reviewer is a vine. The V equal to 1 when the reviewer is a vine or 0 when not. We input the database of two products with the most reviews to yield the result as follows



(a) product 1 (b) product 2

Figure 8: Honor of different products over time.

Figure 8 demonstrate that the product-1 has a more stable during the period. One can gain the reputation fluctuation of any products by take their databases into the time-based model. The detail result can be checked in appendix 11.

7 Combinations of Text and Rating

We utilize the Association Rule Analysis to gain the combination of text-measure and rating-measure that can indicate whether a product is potentially successful or failing.

7.1 Support and Confidence

A review can be rewritten as item sets $I = \{i_1, i_2, \cdots, i_m\}$, where the element i represents the word contained in the corresponding review. Thus, the superset of all reviews can be given as item data base $D = \{I\}$. The amount of elements of item set is defined as the length of item set, so that the $k-length$ item set I can be rewritten as I_k. For any two item sets $X \subseteq I_x$ and $Y \subseteq I_y \quad (x, y \in \{1, 2, \cdots, n\})$, the Support of X and Y can be defined as

$$Supp(X \bigcap Y) = \frac{num(X \bigcap Y)}{num(AllSample)} \tag{7}$$

where the $num(X \bigcap Y)$ and $num(AllSample)$ denotes the amount of reviews containing both X and Y and the quantity of all reviews respectively. The $X \bigcap Y$ can be regarded as the frequent item sets(FIS) [20] when $Sup(X, Y) > Sup_{min}$, where the Sup_{min} is the minimum Support threshold. The Confidence of X to Y can be given as

$$Conf(X \leftarrow Y) = \frac{num(XY)}{num(Y)} \tag{8}$$

The FIS X and Y will be deemed as strongly associated[20] if $Conf(X \leftarrow Y) > Conf_{min}$, where $Conf_{min}$ is the minimum Confidence threshold.

7.2 Apriori algorithm

The algorithm is used to find out all the FIS. Obviously,when the word sets X is FIS, its subsets are all FIS[21], which can be fomulated as

$$Supp(X) \geqslant Supp(X \bigcap Y) > Supp_{min} \tag{9}$$

Moreover the supersets of X will be all non-FIS if the X is non-FIS[21], which can be represent as

$$Supp_{min} > Supp(X) \geqslant Supp(X \bigcap Y) \tag{10}$$

Thus, we could gain the process of Apriori algorithm based on this two feature as , the pseudocode in appendix 11.

We input the database of hair dryer to yield the result showed in table 5 . The table 5 shows the frequent item sets including the highest or the lowest star ratings and their corresponding words combinations. Obviously, the products with the review label 'quite', 'new' and 'compact'

are more likely to success while the one with 'loud', 'old' and 'damage' are more likely to fail. We sum up the sentimental rating of the words set in each combination to gain the text-based measures.

Table 5: the frequent item sets

Label	Words Set	Star Rating	Sentimental Rating
successful	[quiet, new, small, compact]	5	7.16581
	[practic, much, pleasant, good]	5	6.24621
	... ...	...	...
	[good, great, nice]	5	5.84687
failing	[prong, needless, unmanag]	1	-4.81354
	... ...	...	...
	[open, total, wast, locat]	1	-5.18134
	[loud, bad, wrong, old , damag]	1	-6.47152

We define a review as the positive feedback while the its star rating is not less than 5 and sentimental rating is not less than 5.84687. The negative one has star rating not more than 1 and sentimental rating not more than -4.81354.

7.3 Potential Analysis

For the product P_i, the potential value Γ_i can be given as

$$\Gamma_i = \frac{(\alpha_i - \beta_i)}{t_i} \times 100 \tag{11}$$

The α_i and β_i denote the amount of positive and negative feedback of P_i . The t_i represent total review of P_i. We demonstrate the potential value of some products at random as below.

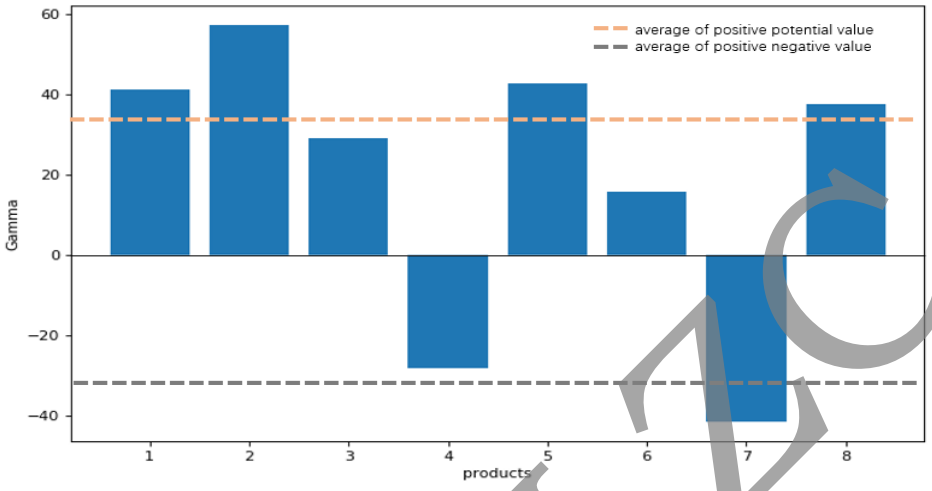


Figure 9: Potential value of each product

The product can be deemed as successful production when the corresponding Γ is higher than the average of positive potential value. On the contrary, the one can be considered as failure when Γ is lower than the average negative value.

8 Following Trend Phenomenon

Based on the measure extracted from text and rating, we would determine whether are customers more likely to write some corresponding reviews after seeing a series of low star ratings by dynamic time warping(DTW) model.

8.1 Dynamic Time Warping

The star rating and sentimental score of each product can be transform into time series as $A = [a_1, a_2, \cdots, a_s]$ and $B = [b_1, b_2, \cdots, b_s]$ where a_i and b_i represent the average rating during t_i to t_{i+1}. The point in time $\{t_i\}$ differ by a constant Δt, which can be expressed as $t_{i+1} - t_i = \Delta t$. For the time series A and B have different dimensions, we would execute dimensionless processing as follows.

$$a_i \leftarrow a_i / a_{max}$$
$$b_i \leftarrow b_i / b_{max}$$

Where a_{max} and b_{max} denote the maximum of star rating and sentimental score. Moreover, we remove $\{b_1, \cdots, b_m\}\,(1 \leqslant m < s)$ from B to identify the specific star ratings would incite more reviews with the similar sentiment in subsequent period, so that the time series B can be rewritten as $B = [b_1, b_2, \cdots, b_t](t < s)$. The distance between a_i and b_j can be given as $\delta(a_i, b_j)$. Thus, the correlation between reviews and star rating can be given as

$$D = \frac{\sum_{n=1}^{N} \delta(a_i, b_j) \cdot W_n}{\sum_{n=1}^{N} W_n} \tag{12}$$

where D is the weighted average of the distance between time series A and B, which has negative correlation with the relevance between the star rating and future reviews. After setting all the weights as 1, the minimum of D can be calculated in the algorithm as follow

Algorithm 1: Procedure of DWT

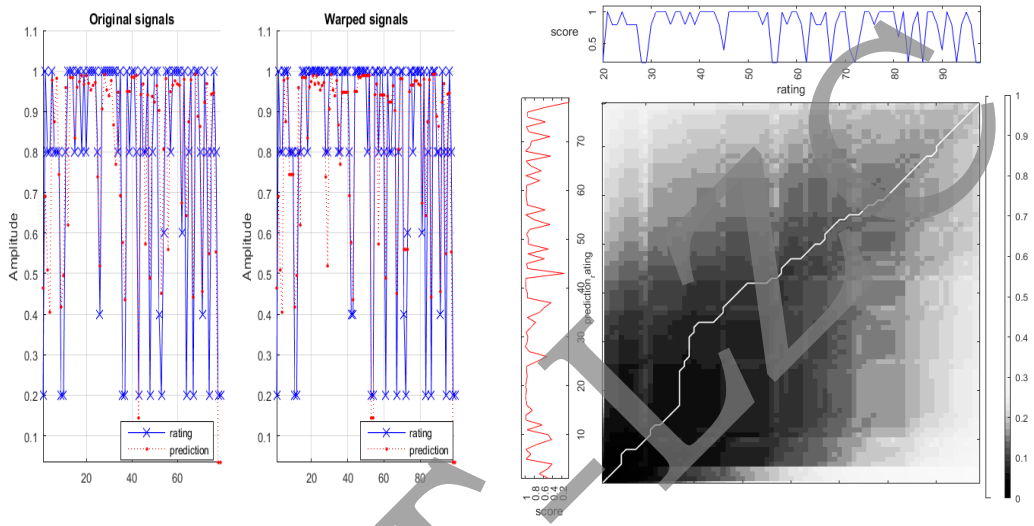
Input: Rating time series: $A = [a_1, a_2, \cdots, a_s]$
Review time series: $B = [b_1, b_2, \cdots, b_t]$

Output: Series correlation D

Let δ be a distance between coordinates of sequence
Let m(S,T) be the matrix of couples(cost,path)
$m[1, 1, 1:2] \leftarrow (\delta(a, b), (0, 0))$
for *i=2 to s* **do**
  $m[i, 1, 1:2] \leftarrow (m[i, 1, 1] + \delta(a_i, b_1), (i-1, 1))$
end
for *j=2 to T* **do**
  $m[1, j, 1:2] \leftarrow (m[1, j-1, 1] + \delta(a_1, b_j), (1, j-1))$
end
for *i=2 to s* **do**
  for *j=2 to T* **do**
    $minimum \leftarrow minVal(m[i-1, j, 1], m[i, j-1, 1], m[i-1, j-1, 1])$
    $m[i, j, 1:2] \leftarrow (first(minimum) + \delta(a_i, b_j), second(minimum))$
  end
end
return $M[S, T]$

We input the data of *Danby 0.7 cu.ft. countertop microwave* into the algorithm to yield an example wrapping path in figure. 10 . The figure. 10 shows that the wrapping degree is slight, which means that the

text-based measures reflect strong similarity with previous star ratings. Thus, it can be deemed that the specific star ratings incite more corresponding reviews.



(a) time series of rating (b) wrapping path

Figure 10: Result yeilded by DWT.

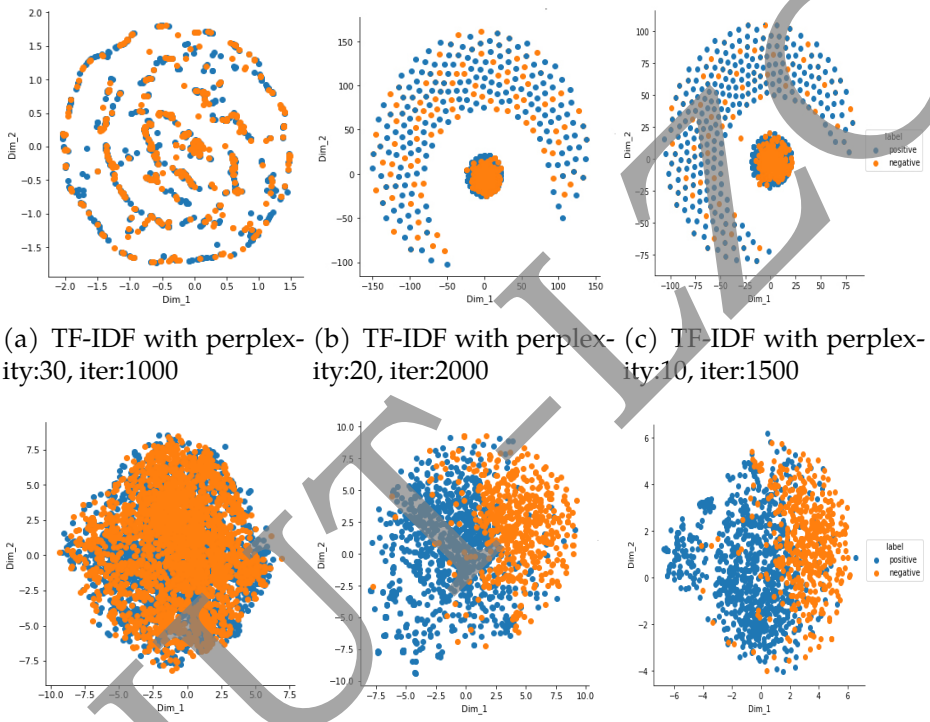
After identifying the specific star ratings incite more corresponding reviews, we compare the inciting rate within the product with the most reviews in table. 6 . It can be discovered that star ratings of the pacifier are more likely to incite corresponding reviews.

Table 6: Inciting rate of the best-sell product.

Product	Product title	Distance
hair dryer	andis 1875-watt fold-n-go ionic hair dryer	1.0255098
	remington salon collection pearl ceramic hair dryer	1.0337255
	conair 1875 watt tourmaline ceramic hair dryer	1.0421478
microwave	danby 0.7 cu.ft. countertop microwave	0.6733481
	whirlpool wmc20005yw countertop microwave	0.82484716
	sharp microwave drawer oven	0.8349739
pacifier	philips avent bpa free contemporary freeflow pacifier	1.403347
	philips avent bpa free soothie pacifier	1.086339
	wubbanub infant pacifier - giraffe	1.101904

9 Sensitivity Analysis

We select proper parameters of Word2Vec and TF-IDF to precisely discriminate positive and negative reviews. Then we adjust the parameter in binary classification model and visualize the results as below



(a) TF-IDF with perplexity:30, iter:1000

(b) TF-IDF with perplexity:20, iter:2000

(c) TF-IDF with perplexity:10, iter:1500

(d) Word2Vec with perplexity:30, iter:1000

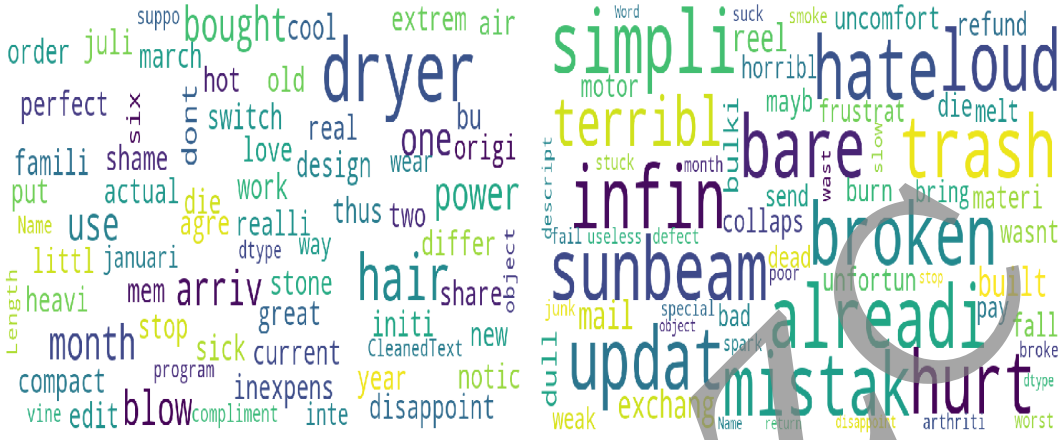
(e) Word2Vec with perplexity:10, iter:1500

(f) Word2Vec with perplexity:20, iter:2000

Figure 11: Visualization of word vectors under different parameters.

Obviously, the TF-IDF with the perplexity and iter equal to 30 and 1000 has the best distinguishing plane, where the best perplexity and iter for Word2Vec are 20 and 2000.

In text preprocessing phase, if we removed the Data Cleaning process, we would gain the result as left of figure 12 . Obviously ,the word such as 'dryer', 'hair' and 'month' would be regarded as the negative words. The result after Data Cleaning is shown in the right of figure 12



(a) Without Data Cleaning (b) After Data Cleaning

Figure 12: Test for Data Cleaning

10 Strengths and Weakness

10.1 Strengths

- Due to the model combination adopted in text analysis, the robustness is so great that the model can adjust to reviews of totally different products.
- Owing to the Gradient Boosting Decision Tree algorithm adopted in time-based reputation model, the data with different dimensions can be processed smoothly and efficiently.
- We introduce the DWT to analysis the association between the rating and reviews innovatively and gain remarkable result.

10.2 Weakness

When performing sentiment analysis, logistic regression can't do enough fine-grained sentiment classification problems, so that Postive words such as *complaint* in the cloud figure have not been separated.

11 Conclusions and Discussion

We propose three major models by effectively combining reviews and ratings. Our models exhibit great potential in drawing conclusions as

following:

- We introduce Sentiment Prediction model, the result of which is a direct and positive answer to the connection question, that is, certain descriptors of quality are indeed evidently connected with rating levels.
- In response to inquiry on internal relationship between ratings and reviews, Dynamic Time Warp model, our second model, compares the similarities of ratings and sentimental scores. Through the best warping path, we find that specific ratings incite more corresponding reviews.
- In terms of time-based measures and patterns, we establish Credibility model to determine the most informative reviews and rating for Sunshine Company to track. These data then can be adapted into time-based reputation measure.
- Due to the dynamic time warping result, we verify that particular star ratings do incite more reviews with the same sentiment. Moreover we find some kinds of products are more likely to incite more reviews.
- By introducing Net Promoter Score and feeding it to time-based procedure, we manage to draw a map that symbolizes the period-rating map. On this foundation, we offer our suggestions to facilitate online activities concerning three commodities.

References

[1] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T., Evaluating Collaborative Filtering Recommender Systems. ACM Trans. on Information Systems, 22, 1, 2004, 5-53.

[2] Mudambi S M, Schuff D. Research note: What makes a helpful online review? A study of customer reviews on Amazon. com. MIS quarterly, 2010: 185-200.

[3] Cheng Z, Ding Y, Zhu L, et al. Aspect-aware latent factor model: Rating prediction with ratings and reviews//Proceedings of the 2018 world wide web conference. 2018: 639-648.

[4] Park S, Nicolau J L. Asymmetric effects of online consumer reviews. Annals of Tourism Research, 2015, 50: 67-83.

[5] Danescu-Niculescu-Mizil, C., Kossinets, G., Kleinberg, J., & Lee, L. (2009). Tourism Research, 50, 67–83.

[6] Kokkodis M. Learning from positive and unlabeled Amazon reviews: Towards identifying trustworthy reviewers//Proceedings of the 21st International Conference on World Wide Web. 2012: 545-546.

[7] Leino, J., & Räihä, K.-J. (2007). Case amazon. Proceedings of the 2007 ACM Conference on Recommender Systems - RecSys '07.

[8] Wallach H M. Topic modeling: beyond bag-of-words//Proceedings of the 23rd international conference on Machine learning. 2006: 977-984.

[9] Zhang Y, Jin R, Zhou Z H. Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics, 2010, 1(1-4): 43-52.

[10] Ramos J. Using tf-idf to determine word relevance in document queries//Proceedings of the first instructional conference on machine learning. 2003, 242: 133-142.

[11] Rong X. word2vec parameter learning explained. arXiv preprint arXiv:1411.2738, 2014.

[12] Goldberg, Yoav, and Omer Levy. "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method." arXiv preprint arXiv:1402.3722 (2014).

[13] Lilleberg J, Zhu Y, Zhang Y. Support vector machines and word2vec for text classification with semantic features//2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC). IEEE, 2015: 136-140.

[14] Linderman G C, Rachh M, Hoskins J G, et al. Efficient algorithms for t-distributed stochastic neighborhood embedding. arXiv preprint arXiv:1712.09005, 2017.

[15] Hosmer Jr D W, Lemeshow S, Sturdivant R X. Applied logistic regression[M]. John Wiley & Sons, 2013.

[16] Juneja P, Ojha U. Casting online votes: to predict offline results using sentiment analysis by machine learning classifiers//2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2017: 1-6.

[17] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree//Advances in neural information processing systems. 2017: 3146-3154.

[18] Prokhorenkova L, Gusev G, Vorobev A, et al. CatBoost: unbiased boosting with categorical features//Advances in neural information processing systems. 2018: 6638-6648.

[19] Grisaffe D B. Questions about the ultimate question: conceptual considerations in evaluating Reichheld's net promoter score (NPS). Journal of Consumer Satisfaction, Dissatisfaction and Complaining Behavior, 2007, 20: 36.

[20] Ye Y, Chiang C C. A parallel apriori algorithm for frequent itemsets mining//Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06). IEEE, 2006: 87-94.

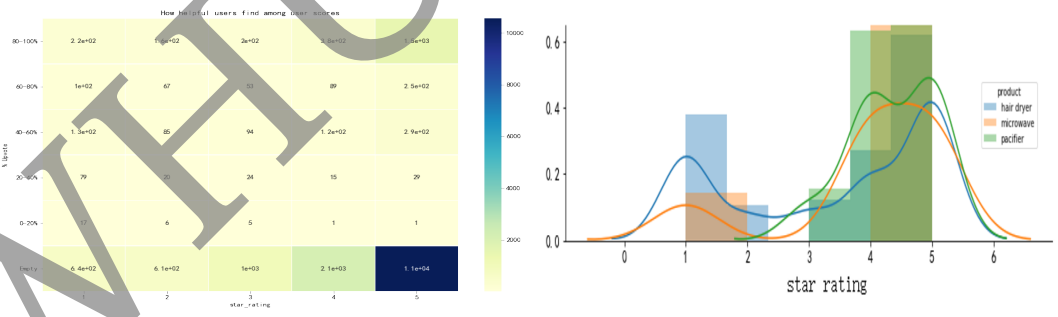
[21] Lenca P, Vaillant B, Meyer P, et al. Association rule interestingness measures: Experimental and theoretical studies//Quality Measures in Data Mining. Springer, Berlin, Heidelberg, 2007: 51-76.

Appendices

Here is **Code and Figures** we used in our paper, where python is the main development language. Since we mainly use **jupyter notebook** for programming, most of the python code will only be represented by pictures in the appendix. Some important model code will be placed here. The detailed working code can be found on ours **github repository**[4].

Appendices A : Exploratory Data Analysis in the paper

Figure 13: Mapping description of the relationship between Amazon reviews and users via Visio.



(a) The relationship between score and voting. (b) The 'star_rating' with the distribution of 'vine'

Figure 14: Exploratory Data Analysis.

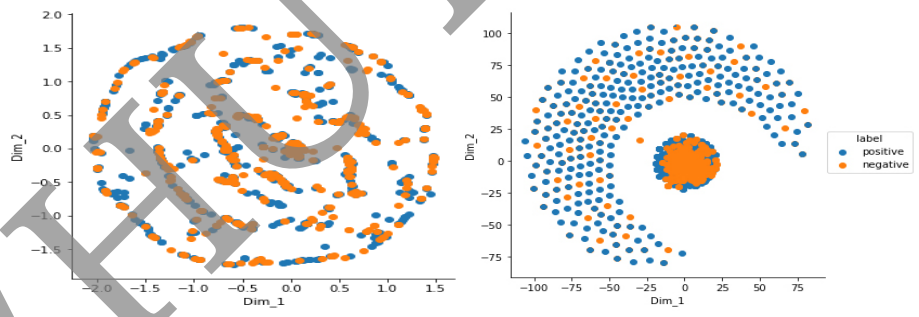
Exploratory Data Analysis

```
df['Helpful %'] = np.where(df['helpful_votes'] > 0,
                  df['helpful_votes'] / df['total_votes'], -1)
df['% Upvote'] = pd.cut(df['Helpful %'],
         bins = [-1, 0, 0.2, 0.4, 0.6, 0.8, 1.0],
```

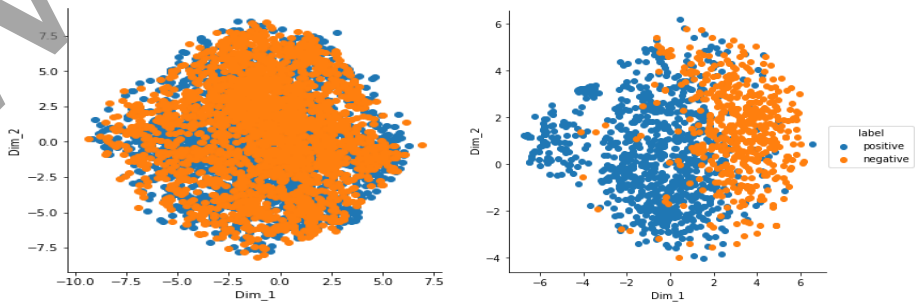
[4] https://github.com/2015970/MCMICM2020

```
        labels = ['Empty', '0-20%', '20-40%',
                  '40-60%', '60-80%', '80-100%']
                      , include_lowest = True)
df.head()
df_s = df.groupby(['star_rating', '% Upvote'])
                  .agg({'review_id': 'count'})
df_s = df_s.unstack()
df_s.columns = df_s.columns.get_level_values(1)
fig = plt.figure(figsize=(15,10))
sns.heatmap(df_s[df_s.columns[::-1]].T,
            cmap = 'YlGnBu', linewidths=.5, annot = True)
plt.yticks(rotation=0)
plt.title('How helpful users find among user scores')
plt.show()
```

Appendices B : Text processing and sentiment analysis for question 1, 2-a and 2-e



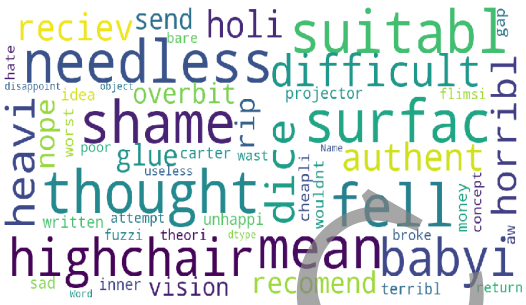
(a) TF-IDF with default parameters (b) TF-IDF with perplexity:10, iter:1500



(c) Word2Vec with default parameters (d) Word2Vec with perplexity:20, iter:2000

Figure 15: Visualization of word vectors under different parameters.

(a) Pacifier's positive reviews

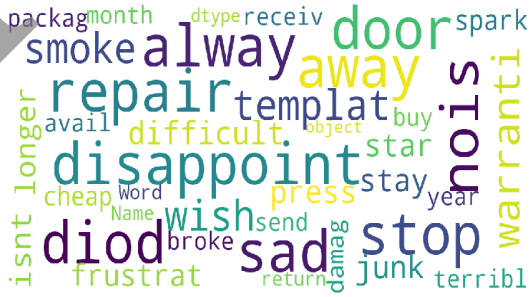


(b) Pacifier's negative reviews

(c) Hair dryer's positive reviews

(d) Hair dryer's negative reviews

(e) Microwave's positive reviews



(f) Microwave's negative reviews

Text processing main code via python

```
import re
# Important steps to clean the text data.
filtered_data = df[df['star_rating'] != 3]
def partition(x):
    if x>3:
        return 'positive'
    return 'negative'

actual_score = filtered_data['star_rating']
positiveNegative = actual_score.map(partition)
filtered_data['Score'] = positiveNegative
```

```
filtered_data.head()

# Defining function to clean html tags
def cleanhtml(sentence):
    cleaner = re.compile('<.*>')
    cleantext = re.sub(cleaner, ' ', sentence)
    return cleantext

# Defining function to remove special symbols
def cleanpunc(sentence):
    cleaned = re.sub
        (r'[?|.|!|*|@|#|\'|"|,|)|(|\|/]', r'', sentence)
    return cleaned
```

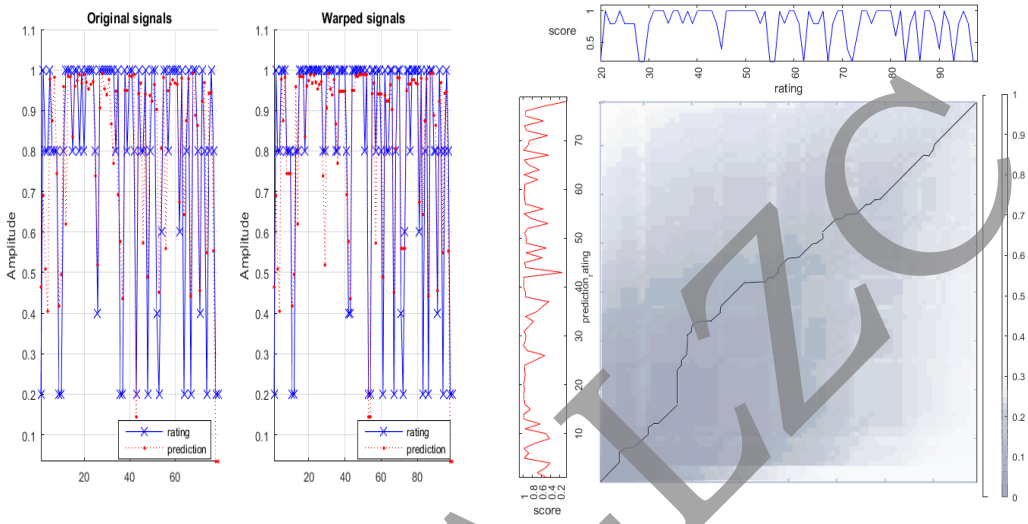
Logistic Regression main code

```
def text_fit(X, y, model, clf_model, coef_show=1):
    X_c = model.fit_transform(X)
    print('# features: {}'.format(X_c.shape[1]))
    X_train, X_test, y_train, y_test = \
        train_test_split(X_c, y, random_state=0)
    print('# train records: {}'.format(X_train.shape[0]))
    print('# test records: {}'.format(X_test.shape[0]))
    clf = clf_model.fit(X_train, y_train)
    acc = clf.score(X_test, y_test)
    print('Model Accuracy: {}'.format(acc))

    if coef_show == 1:
        w = model.get_feature_names()
        coef = clf.coef_.tolist()[0]
        coeff_df = pd.DataFrame({'Word': w, 'Coefficient': coef})
        coeff_df = coeff_df.sort_values\
            (['Coefficient', 'Word'], ascending=[0, 1])
        print('-Top 20 positive-')
        print(coeff_df.head(20).to_string(index=False))
        print('')
        print('-Top 20 negative-')
        print(coeff_df.tail(20).to_string(index=False))
    return coeff_df


coeff_df = text_fit(X, y, c, LogisticRegression())
```

Appendices C : Dynamic Time Warping mian code for question 2-d



(g) time series of rating (h) wraping path

Figure 16: Result yeilded by DWT.

Dynamic Time Warping main code via matlab (The drawing code is too long and omitted)

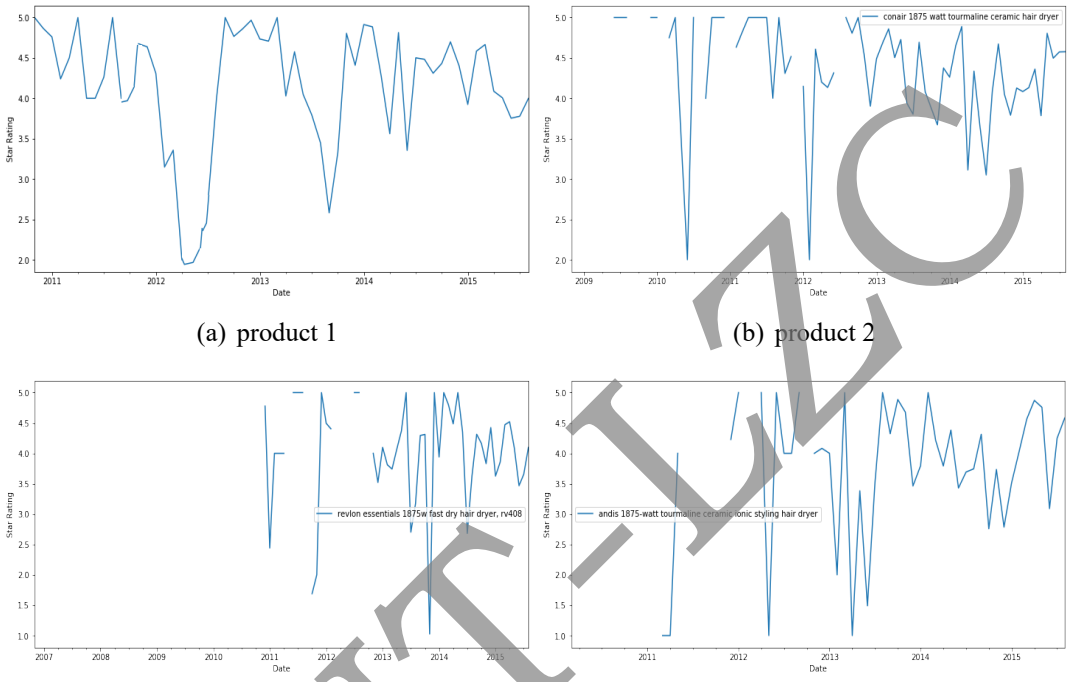
```
function [Dist,D,k,w,rw,tw]=dtw(r,t,pflag)
[row,M]=size(r); if (row > M) M=row; r=r'; end;
[row,N]=size(t); if (row > N) N=row; t=t'; end;
d=sqrt((repmat(r',1,N)-repmat(t,M,1)).^2);
 %this makes clear the above instruction Thanks Pau Mic
D=zeros(size(d));
D(1,1)=d(1,1);
for m=2:M
    D(m,1)=d(m,1)+D(m-1,1);
end
for n=2:N
    D(1,n)=d(1,n)+D(1,n-1);
end
for m=2:M
    for n=2:N
        D(m,n)=d(m,n)+min(D(m-1,n),min(D(m-1,n-1),D(m,n-1)));
    end
end
Dist=D(M,N);n=N;m=M;k=1;w=[M N];
while ((n+m)~=2)
```

```
        if (n-1)==0
            m=m-1;
        elseif (m-1)==0
            n=n-1;
        else
          [values,number]=min([D(m-1,n),D(m,n-1),D(m-1,n-1)]);
          switch number
          case 1
            m=m-1;
          case 2
            n=n-1;
          case 3
            m=m-1;n=n-1;
          end
    end
        k=k+1;
        w=[m n; w]; % this replace the above sentence.
end
% warped waves
rw=r(w(:,1));
tw=t(w(:,2));
end
```

Table 7: Inciting rate of the best-sell product.

Product	Product title	Distance
hair dryer	andis 1875-watt fold-n-go ionic hair dryer	1.0255098
	remington salon collection pearl ceramic hair dryer	1.0337255
	conair 1875 watt tourmaline ceramic hair dryer	1.0421478
microwave	danby 0.7 cu.ft. countertop microwave	0.6733481
	whirlpool wmc20005yw countertop microwave	0.82484716
	sharp microwave drawer oven	0.8349739
pacifier	philips avent bpa free contemporary freeflow pacifier	1.403347
	philips avent bpa free soothie pacifier	1.086339
	wubbanub infant pacifier - giraffe	1.101904

Appendices D : Temporal Reputaion and information for 2-a, 2-b



(a) product 1 (b) product 2

(c) product 3 (d) product 4

Figure 17: Product reputation chart over time.

text-based measure(s) and ratings-based measures.

```
rating_series = pd.DataFrame(kindle.review_date)
dforms=[]
for x in rating_series.review_date:
    dforms.append((pd.to_datetime(x)).value)
# now we have dforms which has dates transformed to numeric values
rating2 = rating_series.assign(date_min = dforms)
rating2.reset_index(inplace=True)
#rating2.set_index('date_min')
#rating2.columns=['timestamp_string','review_count','date_min']
bins = np.linspace(min(rating2.date_min),max(rating2.date_min),num
rating2.hist(column='date_min', bins=20,figsize=(10,6),)
rating2.hist(column='date_min', bins=30,figsize=(10,6))
rating2.hist(column='date_min', bins=50,figsize=(10,6))

def NPS_eval (A):
    score =0
    for x in A[:]:
```

```
            if (x>4) :
                score+=1
            elif (x<3) :
                score-=1
        return 100*score/len(A)
NPS_overtime = kindle[['temp','star_rating']]
NPS_overtime.groupby(by='temp').agg(NPS_eval).plot(figsize=(15,10))

for i in range(8):
    title = final['product_title'].value_counts().index[i]
    XXXX = final[final['product_title']==title]
    month = XXXX.resample('M').sum()
    month['H/P'] = month['H']/month['P']
    month_dates = month['H/P']
    month_dates.sort_index(inplace=True)
    month_dates.plot(figsize=(12,6))
    plt.legend([title])
    plt.ylabel('Star Rating')
    plt.show()
```

Table 8: LightGBM calculated importance(credibility)

Review id	Star rating	Vine	Date	**credibility**
R3HT7OGKQO8Q0E	5	Y	2010-11-22	0.762627
R3LXL05NPDK7P2	5	N	2010-11-23	0.566363
RWZZ77PTTHCDV	4	Y	2010-11-28	0.928533
R2JK82HHMUGGOU	1	N	2010-12-21	0.089120
...	...	...	...	...
RJ4HTF6UC3JGC	2	N	2015-08-22	0.240850
R1X47WDNBT4OHZ	1	N	2015-08-23	0.843194
R9T1FE2ZX2X04	5	Y	2015-08-31	0.710421

Appendices E : Apriori algorithm 2-c

Because the code is too long, it is represented by pseudo code

Algorithm 2: Procedure of Apriori

Input: item data base: D
minimum Support threshold: Sup_{min}
minimum Confidence threshold: $Conf_{min}$
Output: frequent item sets F

Initialize
iteration $t \leftarrow 1$
The candidate FIS:$C_t = \varnothing$
The length of FIS:$length = 1$
for *i=1 to sizeof(D)* **do**
  I_i=D(i)
  n=sizeof(I_i)
  for *j=1 to n* **do**
    if $I_i(j) \notin C_t$ **then**
      $C_t = C_t \cup I_i(j)$
    end
  end
end
$F_t = \{f | f \in C_t, Sup(f) > Sup_{min}\}$
while $F \neq \varnothing$ **do**
  t=t+1
  length=length+1
  $C_t \leftarrow$ all candidate of FIS in F_{t-1}
  $F_t = \{f | f \in C_t, (Sup(f) > Sup_{min}) \bigcap (Comf(f) > Conf_{min})\}$
end
return F_{t-1}

Table 9: the frequent item sets

Label	Words Set	Star Rating	Sentimental Rating
successful	[quiet, new, small, compact]	5	7.16581
	[practic, much, pleasant, good]	5	6.24621
	... ...	...	...
	[good, great, nice]	5	5.84687
failing	[prong, needless, unmanag]	1	-4.81354
	... ...	...	...
	[open, total, wast, locat]	1	-5.18134
	[loud, bad, wrong, old , damag]	1	-6.47152