Problem Chosen	2020 MCM/ICM Summary Sheet	Team Control Number
C		2014986

Understanding Online Reviews Through Topic Modeling

Summary

With the sharply increasing number of online reviews, how to extract useful information from massive data and how does it influence both products and customers have become an intriguing but underdeveloped problem.

Entrusted by Sunshine Company,our team first process the given data sets through deleting null values, amending abnormal values and omitting minor variables. For evaluation, we adopt 8 variables as available data in the research process. After that, we delete records from unverified customers but retain reviews from trusted buyers. Whats more, we sort the review data variable in chronological order for easier viewing.

Next, we explore the relationship between rating levels and specific emotional words in reviews with sentiment analysis theory. We find that rating levels are strongly associated with some emotional descriptions in reviews. It is a foundational work for following study.

Then we use topic modeling method to extract latent topics from the reviews. These topics summarize customers frequent appeared words and we apply Topic Score to measure the satisfaction level of consumers on the topic measures. After that, we use bayesain lower bound model to estimate the reviews' real quality. High-quality reviews are beneficial for Summer Company to understand customers feedback and improve their goods, so we form weights as a filter to get most informative reviews. Based on trained Latent dirichlet allocation and weights, we obtain the real value of the topic measures. Then, we observe the topic measures and ratings of three products over the years and suggest weather their reputation is increasing or decreasing in the marketplace. We find that all three products reviews have steadily improved these years and we predict that their reputations are still increasing, though the extent differs. Through review system and rating system, we take the two variables influences and their interactions into consideration to predict which product will succeed and which one is potentially a failure. Then, we use correlation analysis to detect the most correlated combinition of the three topic measures.

Finally, we construct a lag-linear model to simulate and estimate the process of customer ratings. We discover that, sometimes negative ratings and reviews will cause more sub-reviews. Final results confirm our guess.

Keywords: Topic Modeling;Data Mining;Bayesian Inference

Contents

1 Introduction

1.1 Problem Statement

Amazon online mall provides customers with an opportunity to rate and review the products after they purchase them. Customers can express their satisfaction with the product by submitting star ratings from 1 (lowest rating) to 5 (highest rating). In addition, customers can submit reviews to express their opinions and further feedback about products. Other customers can also express their attitudes towards these reviews by submitting helpfulness ratings on them to represent their own buying tendency. Through these data, companies can have an in-depth understanding of the market they participate in, the timing of their participation, and the potential success of product function design choices.

In view of the plan of sunshine company to launch and sell three new products in the online market, we will analyze the potential important design features of the products to attract customers, the interaction between time and products through the historical rating and comments provided by customers, so as to help the company create successful products and accurate online sales strategy.

1.2 Our Goals

We set goals below after discussion and analysis of this problem.

(i) Choose helpful data for Summer Company in market competetion and product survey.

- Identify decisive quantitative or qualitative patterns among given data which will benefit the Sunshine Company in product offerings.
- Build a model to find informative data for Sunshine Company to track their product sales in the online markets.

(ii) Summarize the interaction between reviews and products at different times.

- According to the current customer reviews, build a system of evaluation to predict three products' future reputation.
- Consider measures of text, ratings and their interaction which best suggest products' success.

(iii) Determine the relevance of star ratings and customer behavior.

- Find specific relationship between current reviews and customer follwoing behaviors in rating. Discuss when seeing a series of similar star ratings, weather customers tend to give reviews in the same position or not.
- Explore the link between rating levels and reviews with certain emotional words.

(iv) Based on the evaluations and established models, summary our results and give comprehensive reasons for recommending it to Summer Company's Marketing Director.

1.3 Our Thinking

This is a typical text mining problem, so we handle it from the point of view of statistical text mining. Here is our thinking.

Then we plot the time-based measures to see the trend of product reputation. In section 4.5, we perform correlation analysis to find the maximum-correlation combinition. In section 4.6, we use a lag-linear model to describe the relationship between the number of total reviews and the number of emotinal reviews. We discover that, negative ratings or reviews can always cause more reviews for pacifier and microwave. For hairdryer, positive reviews or rating will cause more.

First, we preprocess the given data sets, including filling in blank values, amending abnormal values and omitting minor variables. Based on our definitions, we choose some major review information as analytical material and refine them. We delete records from unverified consumers but retain reviews from trusted buyers. At last, we sort the review data variable in chronological order for easier viewing.

Second, we test the problem 2.e first as a premise of our model. We perform sentiment analysis on these data, to get a polarity value for each review, testing their emotions. Then we compare the values with ratings, and it shows that star ratings are closely related to reviews.

Based on this,we use topic modeling to extract latent topics. These topics summarize the products common features, which can help Summer Company track their products in time. and use bayesian lower bounds model to calculate real quality of each review, to get the most informative data.Based on the topics and weights, we form three topic measures for each product. Then we plot the time-based measures which displays the changes of comprehensive comments over time, to see the trend of product reputation. Thus,we can suggest weather the reputation of these three products is increasing in the online market or not. Next,we perform a correlation analysis to find the maximum-correlation combinition.

Finally, we constuct a lag-linear model to simulate and estimate the process of review increasing. Using this model, we can get the patterns of the negative review influence and postive review influence.

2 Assumptions and Notations

2.1 Assumptions

Our model is based on following assumptions:

- All reviews only consists three main topics.
- Stars rated greater than or equal to three are seen as positive emotions, and less than three are seen as negative emotions.

2.2 Notations

- Helpful votes are denoted as HV_i.
- Total votes are denoted as TV_i.
- Not-helpful votes are denoted as NV_i.

3 Data Preprocessing

For the problem of data analysis, there are usually abnormal missing value or too redundant data in the initial large amount of data, which may affect the efficiency of modeling and the accuracy of results. Therefore, data preprocessing is very important.

3.1 Default Value Processing

In the original data, there are a few fields with vacancy values, which will affect the arrangement of the following fields when importing the data. Therefore, we choose to fill in the vacancy with null values to ensure that all the data are in good order.

3.2 Redundant Data Processing

In the original data table, there are many fields, such as marketplace, review id, etc., which are too complex and not helpful in the data analysis process. Therefore, we choose to delete all fields from marketplace to product category in the original table to simplify the data.

3.3 Data Filtering

In the original table, there are two fields that provide the key information about whether the customer's evaluation is true, they are vine and verified purchase. Among them, vine indicates that the user has high credit and can obtain the free copies of products provided by the supplier, so we keep all the lines with the value of vine field as Y. In the rest of the data, verified purchase indicates whether the commenter has purchased the product normally, so we choose to delete all the lines with the value of verified purchase field as N. The data left behind by the screening indicates that the reviews of these customers are based on the actual products, rather than the fake comments without the actual products.

4 Model Construction

4.1 Premise for Our Model: Sentiment Analysis

In the following parts of our model construction, we always assume that the ratings can represent product reputation tendency, so we brought Problem E about to test the premise that, ratings and reviews are closely related. In this section, we first use traditional seniment analysis, to evaluate reviews and predict their emotion tendency. Then we compare our review emotion scores with ratings and calculate the correlation value between them.

After importing the three preprocessed tables in Python, we use textblob package to analyze the comments in the tables. For example, the 397th comment in the hair dryer table: This dries my hair foster that bigger, more powerful models. 0.267 is the emotional score of the review.

Therefore, the comments in the three tables can be analyzed in turn. Here, the emotional values of the comments in hair dryer, microwave and pacifier are named sentiment1, sentiment2 and sentiment3. In order to distinguish emotion trend clearly, zero is the dividing point, positive value means positive emotion and negative value means negative emotion.

Correspondingly, the columns of star rating in the above three tables are successively named star rating, star rating 2, star rating 3. If the score of star rating is 1 or 2, it is determined as negative emotion; if the score is greater than or equal to 3, it is determined as positive emotion.

Next, we compare the similarity between sentiment and star rating in three tables. The

results show that the consistency of comments and star rating in the hair dryer, microwave and pacifier tables is 0.866, 0.842 and 0.884, respectively, with high values. It can be concluded that the description characteristics of text-based reviews are consistent with the star ratings.

4.2 Topic Modeling for Extractinging Topics and Building Indexes

4.2.1 Topic Modeling

Topic modeling is a statistical learning model for observing latent topics in documentions.[2] It is kind of like the principle component analysis and factor analysis in statistics, but topic modeling is mainly for text mining. In PCA and factor analysis, we compute the factor matrix, to discover latent indexes among already-known indexes. Here, using topic modeling, we can discover the main topics for all these reviews. For instance, if some review says that the hair dryer works great and consumes less power, then the main topic for this review is 'power'. [1]

The topic modeling method we use in this paper is latent dirichlet allocation. Latent dirichlet allocation can detect latent preferences based on bayes inference. Below is a common graph for explaining LDA.

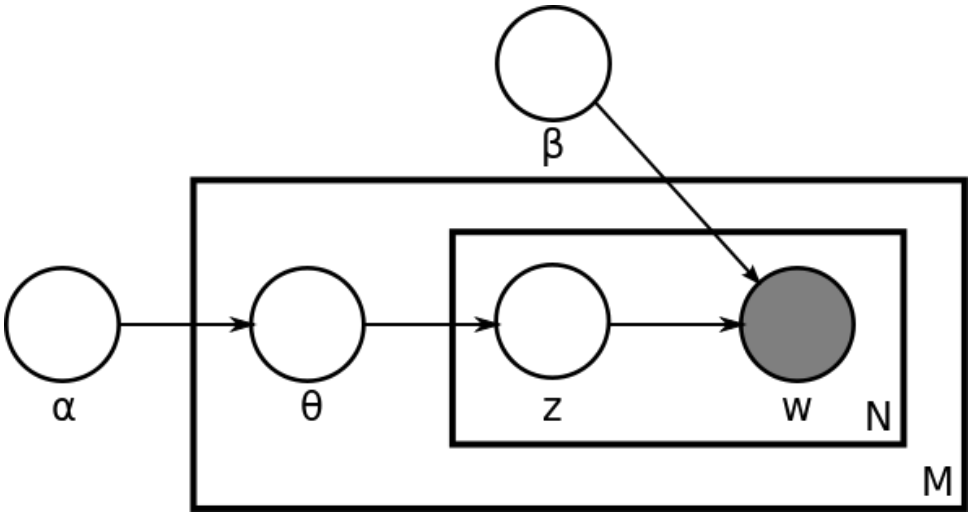

Figure 1: LDA model

Suppose the dataset is $D = w_{i\,i=1}^{M}$, and we have a mixture of topics $\theta \sim Dir(\alpha)$(α is a parameter in Dir). For each of the words(N) w_n, we suppose:

$$z_n \sim Multinomial(\theta), w_n|z_n, \beta \sim p(w_n|z_n, \beta)$$

And using bayesian inference, we have:

$$p(D|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) d\theta_d \tag{1}$$

To describe this process in a more simple way, LDA model for extracting topics is just like the way in which we write an article. First we sample from a dirichlet allocation, $Dir(\alpha)$, to form the topic distribution θ_i for $document_i$, and $\theta_i \sim Multinomial$. Then we can sample from the multinomial distribution as the topic for $word_{i,j}$ in $document_i$, the jth word, denoted

as $z_{i,j}$. So we get the topics for each word. Then we use another dirichlet distribution $Dir(\beta)$ and sample from it, to form the word distribution $\phi(z_{i,j})$ subject to topic $z_{i,j}$. Finally, we sample from the multinomial distribution $\phi_{z_{i,j}}$ to form word $w_{i,j}$. Using the dataset we get to estimate the parameters by Gibbs sampling, we obtain the topic distribution.

Based on assumption 1, in this section, we extract three topics from reviews of each product. Specific results are stated below.

4.2.2 Scores for Reviews on Topics

After generalizing the topics, we can compute each comment's probability of concerning each topic. For example, 'This hair dryer works really great and is easy to carry when you are travelling.', according to the LDA model we build, this review's probabilities of concerning $topic_1$, $topic_2$, $topic_3$ are 0.7, 0.2, 0.1. Here we view the probabilities as topic scores, for the probabilities shows how much of a review is concerned with a certain topic. Based on this result, we can form three indexes for each product. The indexes are generated from customer reviews' probability of concerning the topics. For each product, we can obtain the specific value by using the LDA model. To conclude user emotions, indicating the topic is positive or negative, stars can be used as a judging criteria. In this paper, we simply take stars less than three reviews as negative and others are. If the the raing is positive, then we take the positive value of the score. If the rating is negative, then we take the negative value of the score.

After this process, we can get the specific scores for each review on each topic.

4.2.3 Results

The topics extracted from reviews of hair dryer are:

Table 1: Topics in Hair Dryer Reviews

Topic1	Topic2	Topic3
light	product	like
blow	blow	recommend
power	heat	buy
buy	set	travel
purchas	like	cord
year	time	time
time	cord	high

According to the words contained in each topic, we can infer the content of the topics. In topic1, 'light', are stressed, which shows that topic1 mainly refers to volume and design of the hair dryer. In topic2, 'heat' ,'time'is stressed, which shows that topic2 may refer to the power of the hair dryer. In topic3, 'recommend', 'buy', 'travel' are stressed, which shows that topic3 may refer to the propaganda of the hair dryer.

The topics extracted from reviews of microwaves are:

Table 2: Topics in Microwave Reviews

Topic1	Topic2	Topic3
cook	small	door
time	easy	like
good	button	unit
need	look	product
oven	like	instal
look	kitchen	replac
perfect	space	size

It is shown in the table that, the main reason people like it is the small size of the microwave, which makes it easy to carry. In topic1, 'cook', 'time', 'oven' are stressed, which shows that topic1 mainly refers to the microwave's cooking function. In topic2, 'small', 'easy', 'look' are stressed, indicating that the size of this product is rather small, which shows the design of this microwave. In topic3, 'instal', 'unit' are stressed, which shows that the topic mainly refers to the installation and functions of the microwave.

The topics extracted from reviews of pacifiers are:

Table 3: Topics in Pacifier Reviews

Topic1	Topic2	Topic3
little	cute	month
great	great	daughter
good	look	great
mouth	hold	easi
cute	think	recommend
time	wubbanub	little
buy	product	mouth
month	daughter	buy

In topic1, 'cook','little','mouth' indicates that this topic mainly tells about product functions. In topic2, 'cute','look','wubbanub' are stressed, showing that topic2 mainly focuses on product appearence. In topic3, 'daughter', 'recommend', 'buy' are stressed, showing that this topic may be correlated with product propaganda.

After specifying all the topics, we use the trained-LDA model to compute reviews' scores on these topics.

To summarize, in this section, we explore the latent topics in reviews, and form three indexes based on the topics for each product. We have found that, users mainly focus on the appearence , size, and power for the hair dryer. For the microwave, its size, cooking function are valued. For the pacifier, its appearence ,comfort, and publicity mechanism are relatively important.

4.3 The Bayesian Lower Bounds Information System

To select informative reviews in problem 2.a, we form an algorithm to represent the reviews' real quality, defined as information. Popular ways to calculate information contains in online

reviews including the readability(also known as reading levels or writing style) analysis, sentence length analysis or so on. Here, we use the helpful votes and total votes to estimate the review's information and quality, based on Jying-Nan Wang's research[3].

The reason we use this model is that, helpful votes cannot simply represent a review's quality. Actually, we have discovered that most of the reviews are of relatively high quality, but they didn't all recieve helpful votes. The voting system is a self-motivated system, the reviews which recieve more helpful votes are always on the top of website and can be seen by more people than other reviews. Thus, only 10% reviews recieved votes and this is why the quality of reviews cannot be simply defined by helpful reviews.

4.3.1 Bayesian Lower Bounds Model

Recall the notations we set before, HV_t^i stands for helpful votes of $review_i$ at time t, NV_t^i stands for not-helpful votes and TV_t^i stands for total votes. We suppose that all review-host firms has a prior belief of the reviews' true quality, q_i. q_i follows a symmetric Beta distribution as follows:

$$B(x;\beta_H;\beta_N) = \frac{\Gamma(\beta_H+\beta_N)}{\Gamma(\beta_H)\Gamma(\beta_N)} x^{\beta_H-1}(1-x)^{\beta_N-1}, 0 \le x \le 1 \tag{2}$$

We set $\beta_H = \beta_N$ in this paper, to make the expected quality equal to 0.5

The parameters β_H, β_N represents the firm's confidence on its prior beliefs, here we set $\beta_H = 5, \beta_N = 5$ to put on emphasis on likelihood function.

Suppose we observe the specific number for HV_t^i and NV_t^i, we obtain the conjugate posterior disribution for the review quality:

$$B_t^i(x;\beta_H^*,\beta_N^*) = B(x;\beta_H + HV_t^i, \beta_N + NV_t^i) \tag{3}$$

Based on this posterior belief distribuiton, we define

$$\Phi(review_t^i) = Q(B_t^i(x;\beta_H^*,\beta_N^*), 0.05) \tag{4}$$

The $Q(B_t^i(x;\beta_H^*,\beta_N^*), 0.05)$ is the 5% quantile of the posterior belief distribuiton. Here we define this as the information measure of reviews.

We use the information measure as a weight on reviews' topic scores. The weight represents the importance of each review in the voting system. We also defined another weight decided by vine qualification. If the customer is a vine customer, then we added 0.5, the average quality to his or her review. Thus, we obtain the weighted topic scores.

4.4 Time-based Analysis of Topics and Ratings

We explore the time-based topic scores and ratings mainly by comparing the trend. First, we compute the weighted mean value of each day's topic scores and ratings. For better showing the reputation time trend, we compute the weighted mean value of each year's topic scores and ratings, because data will be densed if the scores are calculated in day intervals. Then, we plot the time series graph to analyze this problem.

4.4.1 For Hair Dryer

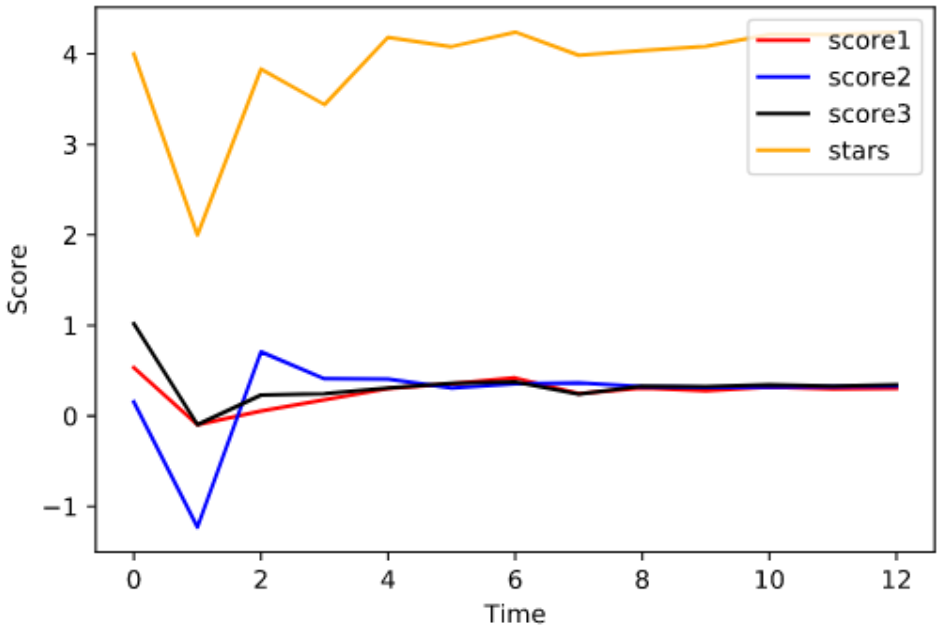

Figure 2: Time Series Scores for Hair Dryer

It can be seen from the figure that, the beginning score for this hair dryer product is really low, and the topic2 average score even drops to -1. This means that the product was not a good product and recieved a lot of complaints. However, after recieving so many negative reviews, the company seemed to update their product in 2003, and the score rises quickly. After that, the reputation kept going up and reached a balance after 2008. To summarize, the main trend for the hair dryer is going up.

4.4.2 For Microwave

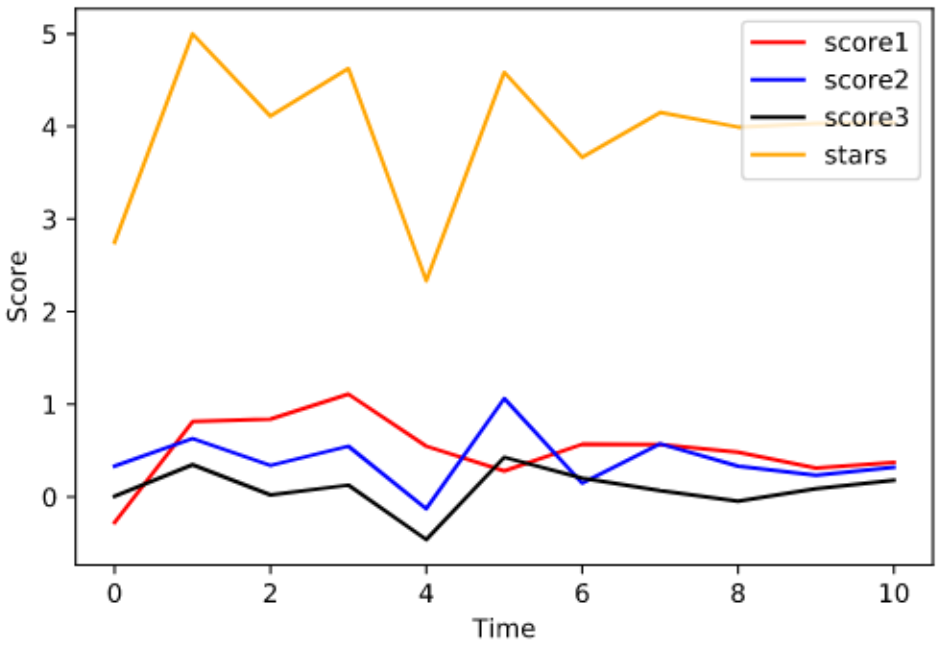

Figure 3: Time Series Scores for Microwave

The figure shows that, the scores for the microwave are not so stable as the hair dryer. The beginning point was also relatively low. Then the company might have updated their products, and the score kept going up quickly. However, in 2009, the product experienced a significant drop in reputation. But this didn't have a serious influence on the product, and the reputation were back to normal in 2010. Then, the microwave's reputation reached a balance. To summarize, the reputation for microwave experienced a significant drop but it still reached a balance at last. And according to the figure, the future reputation is predicted to go up.

4.4.3 For Pacifier

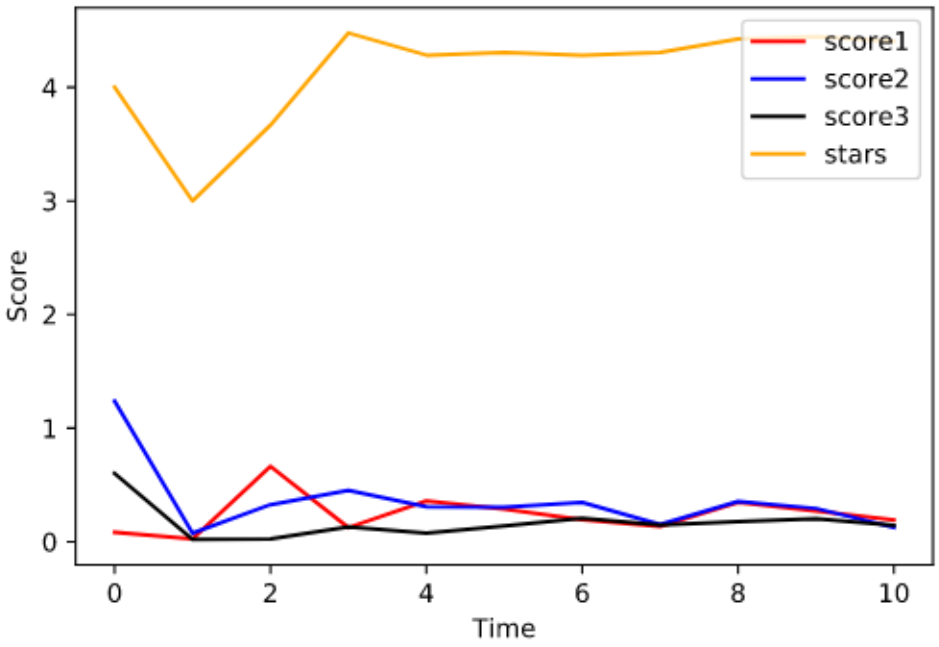

Figure 4: Time Series Scores for Pacifier

Different from the other two, figure, the apparent change in this figure is the drop in 2006. After 2006, the ratings and scores goes up, and reached a balance in 2008. Then, there's no significant change in the reputation, the reputation keeps in a high level. Generally, this is a successful product.

4.5 Correlation Analysis for Topics and Reputations

In this section, to determine the combinations of text-based measures that best indicate a potentially successful or failing product, we perform correlation analysis on the topic scores and stars. We suppose that ratings can represent the reputation, based on our premise tested in problem 2.e, and reputation can directly tell whether a product is a successful product or a failing product.

The data used for correlation analysis is the weighted single review, containing topic scores. To determine the combinations, we calculate the separate correlation value between scores and ratings. The combinitions refer to the single scores: $score_1, score_2, score_3$, and the weighted average scores:

$$\begin{aligned} score_4 &= \frac{score_1 + score_2 + score_3}{3} \\ score_5 &= \frac{score_1 + score_2}{2} \\ score_6 &= \frac{score_1 + score_3}{2} \\ score_7 &= \frac{score_3 + score_2}{2} \end{aligned} \tag{5}$$

We test the correlation between the scores and the ratings, and take the maximum as the best combinition. The results are as below.

Table 4: Correlation Coefficient between Scores and Ratings

Products	score1	score2	score3	score4	score5	score6	score7
Hair Dryer	0.430	0.488	0.445	0.842	0.659	0.649	0.687
Microwave	0.492	0.558	0.496	0.861	0.723	0.697	0.727
Pacifier	0.442	0.434	0.443	0.837	0.643	0.667	0.642

Obviously, the maximum correlation value is always the average of the three topics. So, the best combition is the average of the three topic scores.

4.6 Using Lag-Linear Model to Describe Reviews

To help get a better observation of the data, we draw the distribution plot for each product, discribing the correlation between time and star rating.

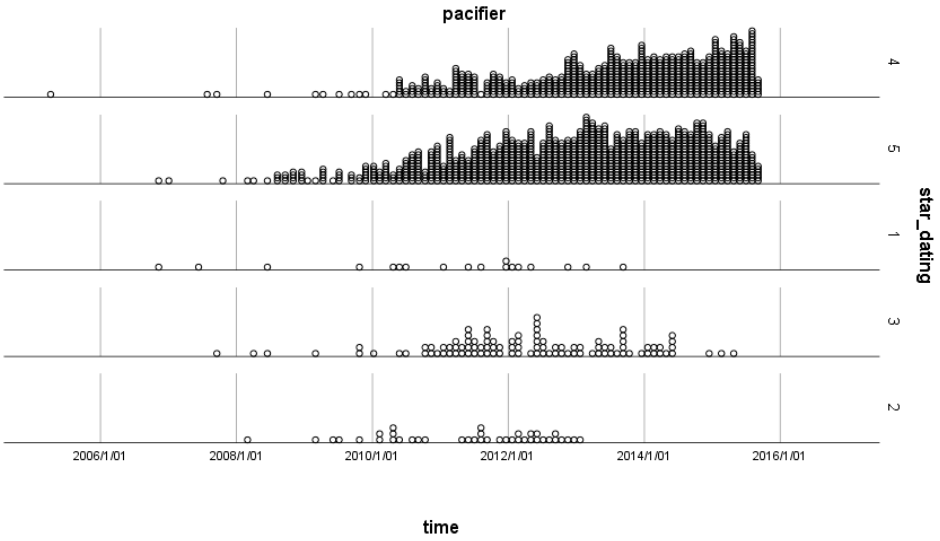

Figure 5: Time-Rating Correlation for Pacifier

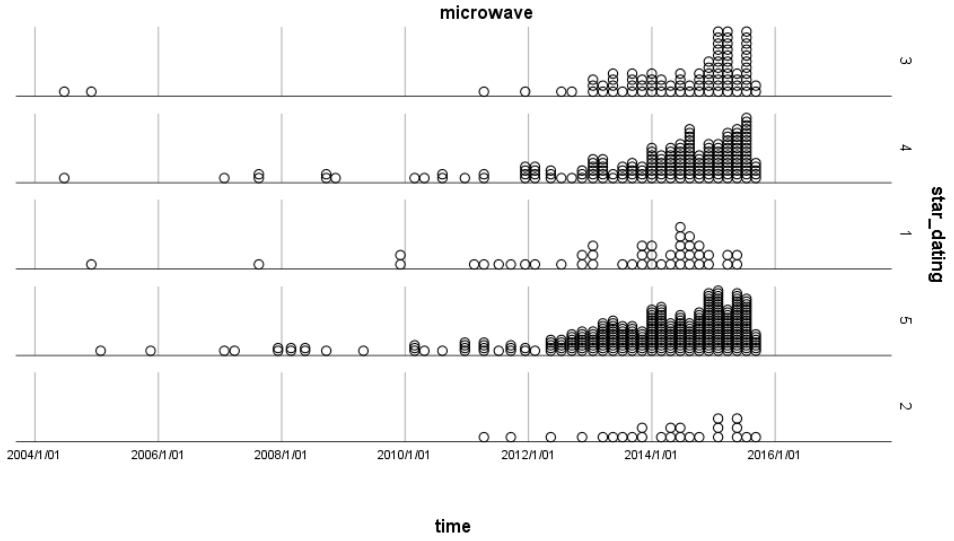

Figure 6: Time-Rating Correlation for Microwave oven

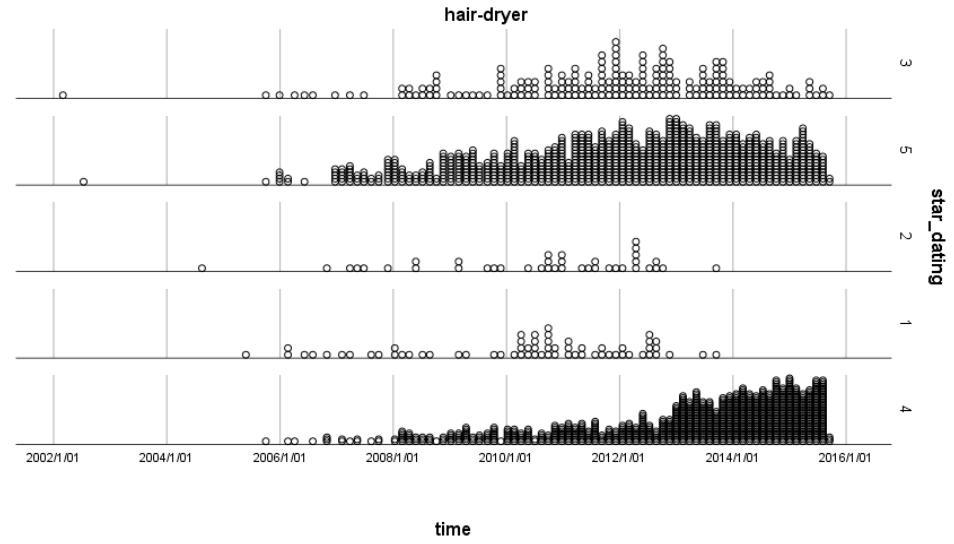

Figure 7: Time-Rating Correlation for Hair Dryer

It can be seen from the above figures that, ratings increase following some pattern. The ratings will encourage customers to post review and ratings, just like a self-motivated system.

For problem 2.d, we form a lag linear model to describe the relationship between the number of reviews and specific star ratings. We suppose that people will react to both postive and negative ratings, but not neutral ratings, which means that ratings score like 3 will not affect people. We also suppose that people will notive ratings proposed in three days. Then, we define positive ratings as ratings greater than three, and negative ratings as ratings less than three. Based on this assumption, we form a lag-linear model to test people's reactions to specific star ratings.

We set the lag operator is -1, which means that people will take three days to realize the change of ratings. We generalized the number of positive ratings pr, number of negative ratings

nr, and the total ratings tr. The model is

$$tr(-1) = \alpha pr + \beta nr + c \tag{6}$$

The estimation for pacifier is as below.

Table 5: Estimation Result for Pacifier

Variable	Coefficient	Prob	Prob(F-statistic)
C	2.473	0.00***	-
nr	1.762	0.00***	-
pr	0.782	0.00***	-
total	-	-	0.00***

The model is significant in statistics, and so are the variables. The model is

$$tr(-1)_{paci} = 2.473 + 1.762nr_{paci} + 0.782nr_{paci} \tag{7}$$

The fun fact we can seen from this model is that actually negative ratings and reviews can bring about more reviews than positive ratings. The other two results are as below.

Table 6: Estimation Result for Microwave

Variable	Coefficient	Prob	Prob(F-statistic)
C	1.218	0.00***	-
nr	0.704	0.00***	-
pr	0.610	0.00***	-
total	-	-	0.00***

Table 7: Estimation Result for Hair Dryer

Variable	Coefficient	Prob	Prob(F-statistic)
C	2.123	0.00***	-
nr	0.436	0.02**	-
pr	0.955	0.00***	-
total	-	-	0.00***

The estimated equations are as below:

$$tr(-1)_{mw} = 1.218 + 0.704nr_{mw} + 0.610pr_{mw} \tag{8}$$

$$tr(-1)_{hd} = 2.123 + 0.436nr_{hd} + 0.955pr_{hd} \tag{9}$$

To summarize, the reviews of microwave and pacifier are tend to become more when negative ratings or reviews take place, but the reviews of hair dryer are tend to become more when positive ratings or reviews take place.

5 Conclusions

We are asked to analyze products data on microwave oven, pacifier and hair dryer for Sunshine Company to promote their new products. After performing data analysis and modeling, we have finished our task successfully. First, we use topic modeling to detect latent topics in the reviews. We find out that for the three products, the appearence are always an important topic. Small-sized products are always more favored by customers of hair dryer. But for microwave oven, small-size can be a problem. Also, the specific abilities and propaganda are valued too.

Secondly, considering the special nature of the voting system, we use bayesian lower bounds model to estimate the real quality of each review, for selecting most informative reviews. We use the estimated quality as weights for the topic measures. Obtaining weighted topic measures, we perform a time-based analysis, and we discover that, the pacifier is a stable product, and also is the hair dryer. But the microwave oven market is a little unstable. Then, we perform correlation analysis to detect the most correlated combinition with the product reputations. We found that the average of the topics is always the best combinition.

Finally, we construct a lag-linear model to simulate and estimate the relationship between star ratings ans the number of reviews. We discover that, negative reviews always cause more reviews for pacifier and microwave oven, but for hair dryer, positive reviews always cause more.

6 Strengths and Weaknesses

6.1 Strengths

- Using Topic Modeling Method.

  Topic modeling method can help us detect latent topics in documents, and this can help companies to detect the main area where a product needs to be developed.

- Using Bayesian Lower Bounds to eliminate the influence the sorting

  The reviews we can notice on the internet are always sorted by the website. Thus, some of the valuable comments may not be seen and the helpful votes may lose their ability to distinct valuable reviews from normal reviews. But using bayesian lower bounds, we can get a more precise estimate of the reviews' real quality.

- Wide application.

  Review is a common and important reference point in nowadays commercial activities. As a result, our model can be applied in many ways, including online retails, big data recommendation and so on.

6.2 Weaknesses

- Subjectivity of some definitions.

  The definition of some qualitative parameters are subjective and it can cause deviation from evaluations of customer reviews.

- Insufficiency of variable consideration.

  We give up some variables as research object to simplify the data processing. If more variables are taken into consideration, our model will become more detailed and comprehensive.

- Simplification of assumptions.

  To construct the model conveniently, we neglect the relationship between time and helpful votes, which can influence model results to some degree.

7 Letter

From: Team 2014986 , MCM 2020

To: Marketing Director of Sunshine Company

Data: March 9, 2020

Subject: Summary of our teams analysis and results

Dear Director,

We are honored to inform you our achievements after analyzing product data and constructing models.

First of all, after using the method of emotional analysis to analyze the reviews and star datings, we found that the positive and negative attitude of emotional words in the comments was basically consistent with the datings, which also laid a foundation for the accuracy of our subsequent conclusions. Secondly, in addition to analyzing datings, we also extract high-frequency keyword directions from the reviews to evaluate customer satisfaction with product performance. According to this model, we found that all three products have strong product characteristics. The key topics of the hair dryer are design, power and propaganda; the microwave oven is cooking ability, design and installation; and the baby pacifier is product functions (comfort), appearance and propaganda. Based on these aspects, your company can know the general directions of enhancing product appeal.

All three products of your company have been for sale for more than ten years, so we can see the relationship between reputation and time in the online market from a large number of reviews and star datings left by customers. Generally speaking, the growth rate of reviews under the three products is relatively stable. After the initial shock of the rating stars, they have been stable at about 4 in recent years. This shows that the number of customers visiting every year is relatively stable, and their overall evaluation of the three products is also very high. This means the reputation of your company's products is guaranteed and tends to be maintained at a high level as time passing.

According to the data analysis of our group, observing the star dating of each product and the change of emotion and attitude of the decisive topic, we can see that the change trend of different product evaluation over time is different. In 2004-2006, shortly after the start of sales, there was an obvious decline in the score of hair dryer. The total score at the bottom was only 2, indicating that the negative attitude of customers was very obvious. However, since 2007, the score has been stable at about 4, and the emotional words in customer reviews are basically positive words. The star rating and topic scores of microwave fluctuated greatly, especially in 2011, when the score dropped to the bottom, with an average of only 3.66. The rest of the time was basically stable, but the fluctuation range was still larger than that of the other two products. Pacifier's rating trend over time is similar to that of hair dryer, and it also encountered a crisis in 2006-2007. But the difference is that during this period, the gap of star dating is not as big as that of hair dryer, and the score of topic scores is also perfectly maintained above 0 in all time periods.

In a word, pacifier has the highest evaluation among the three products. Customers generally think that the product is comfortable and cute, and they are willing to recommend others to buy your company's baby pacifier. Therefore, you can rest assured of the quality of pacifier and choose to appropriately expand the output of the product. The second is hair dryer. In recent years, although the star rating is slightly lower than pacifier, it is also very stable. Many customers will use the positive words such as product portability and high power to describe the product. Therefore, if your company can keep the quality of the hair dryer stable, the sales volume will continue to be good. Finally, there's microwave. In recent years, the star rating of microwave is not low, and it is basically maintained at 4. But the disadvantage is that the fluctuation of star dating and topic scores is too large, which reflects that the product is not very stable to attract customers. In addition, in the key topics we summed up, there is a negative aspect, that is, the installation is cumbersome. Therefore, if your company intends to continue to improve the product, simplifying the installation steps will be a very good direction.

Do specific star ratings incite more reviews? To solve this problem, we use a lag linear model to describe the relationship between the number of total reviews and the number of emotional reviews. We found that for baby pacifier and microwave oven, negative reviews tend to lead to more comments, while for hair dryers, positive reviews usually attract more attention.

The above is the summary of our team's analysis and results. Wish our conclusions can inspire you at some points.

Please contact us if you have any problems.

Yours sincerely,

Team 2014986

References

[1] Shawn Mankad, Hyunjeong Spring Han, Joel Goh, Srinagesh Gavirneni, Understanding Online Hotel Reviews Through Automated Text Analysis, Service Science, 2016.

[2] Blei D, Lafferty J(2009), Topic models. Srivastava AN, Sahami M, eds. Text Mining: Classification, Clustering, and Applications(CRC Press, Boca Raton, FL), 71-94.

[3] Jying-Nan Wang, Jiangze Du, Ya-Ling Chiu, Can online user reviews be more helpful? Evaluating and improving ranking approaches, Information & Management, 2020.

Appendices

Appendix A Python codes

Below are our python codes to impletement this model. **Input Python source:**

```
#MCM2020
# %%
from IPython import get_ipython

# %% [markdown]
# # MCM2020 Problem C
# %% [markdown]
# ## Import Necessary Packages

# %%
import pandas as pd
import gensim

from gensim import corpora, models
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import STOPWORDS

from nltk.stem import WordNetLemmatizer, SnowballStemmer
from nltk.stem.porter import *

from pprint import pprint
import numpy as np

import nltk

# %% [markdown]
# ## Import datasets

# %%
hair_dryer = pd.read_csv('hair_dryer.tsv', sep = '\t')
microwave = pd.read_csv('microwave.tsv', sep = '\t')
pacifier = pd.read_csv('pacifier.tsv', sep = '\t')


# %%
document1 = hair_dryer.dropna(subset=['review_body'])
```

```
document2 = microwave.dropna(subset = ['review_body'])
document3 = pacifier.dropna(subset = ['review_body'])


# %%
print(len(pacifier['review_body']), len(pacifier['review_body']))

# %% [markdown]
# ## Take a look at datasets
# %% [markdown]
# Key Values:
# marketplace, customer_id, review_id, product_id, product_parent, product_tile, product

# %%
print('Rows:' , len(hair_dryer['marketplace']))
print('Categories:', hair_dryer['product_category'].nunique())
print('Numbers of product:', hair_dryer['product_id'].nunique())
print('Numbers of parent:', hair_dryer['product_parent'].nunique())

# %% [markdown]
# ## Data Cleaning
# %% [markdown]
# — Delete all `Verified_purchase` = 'N' and save `vine` = 'Y'

# %%
# delete all No—purchase comments
clean1 = hair_dryer[(hair_dryer['verified_purchase'] == 'Y') | (hair_dryer['vine'] == 'Y


# %%
clean2 = microwave[(microwave['verified_purchase'] == 'Y') | (microwave['vine'] == 'Y')]
comment2 = clean2.review_body.values.tolist()


# %%
clean3 = pacifier[(pacifier['verified_purchase'] == 'Y') | (pacifier['vine'] == 'Y')]
precomment3 = clean3.review_body.fillna('').astype(str)
comment3 = precomment3.to_list()


# %%
comment1 = clean1.review_body.values.tolist()

# %% [markdown]
# ## Problem E
# %% [markdown]
# Main point here, is to use `snownlp` and `textblob` to analyze.

# %%
sent1 = comment1[1]
from textblob import TextBlob
blob = TextBlob(sent1)


# %%
item1 = blob.sentences[0]
print(len(blob),item1)
item1.sentiment.polarity
```

```
# %%
def sentimentana(dataset):
    sentiscore = []
    for sent in dataset:
        single_score = 0
        blobs = TextBlob(sent)
        for blob in blobs.sentences :
            single_score = single_score + blob.sentiment.polarity
        sentiscore.append(single_score)
    return(sentiscore)


# %%
sentiment1 = sentimentana(comment1)
sentiment2 = sentimentana(comment2)
sentiment3 = sentimentana(comment3)


# %%
star_rating = clean1.star_rating.values.tolist()
star_rating2 = clean2.star_rating.values.tolist()
star_rating3 = clean3.star_rating.values.tolist()


# %%
print(np.corrcoef(star_rating, sentiment1))
print(np.corrcoef(star_rating2, sentiment2))
print(np.corrcoef(star_rating3, sentiment3))


# %%
attitude1 = []
for i in sentiment1:
    if i >= 0:
        attitude1.append('positive')
    else:
        attitude1.append('negative')

attitude2 = []
for i in star_rating:
    if i >= 3:
        attitude2.append('positive')
    else:
        attitude2.append('negative')


# %%
distin = [attitude1[i] == attitude2[i] for i in range(len(attitude1)) ]
distin = pd.DataFrame(distin)


# %%
distin_ratio = len(distin[distin[0] == False])/len(distin)
print(1-distin_ratio)
```

```
# %%
attitude3 = []
for i in sentiment2:
    if i >= 0:
        attitude3.append('positive')
    else:
        attitude3.append('negative')

attitude4 = []
for i in star_rating2:
    if i >= 3:
        attitude4.append('positive')
    else:
        attitude4.append('negative')

distin = [attitude3[i] == attitude4[i] for i in range(len(attitude3)) ]
distin = pd.DataFrame(distin)

distin_ratio = len(distin[distin[0] == False])/len(distin)
print(1-distin_ratio)


# %%
attitude5 = []
for i in sentiment3:
    if i >= 0:
        attitude5.append('positive')
    else:
        attitude5.append('negative')

attitude6 = []
for i in star_rating3:
    if i >= 3:
        attitude6.append('positive')
    else:
        attitude6.append('negative')

distin = [attitude5[i] == attitude6[i] for i in range(len(attitude3)) ]
distin = pd.DataFrame(distin)

distin_ratio = len(distin[distin[0] == False])/len(distin)
print(1-distin_ratio)

# %% [markdown]
# ## Problem A
# %% [markdown]
# ### Data Cleaning

# %%
#tokenizer
stemmer = SnowballStemmer('english')

def lemmatize_stemming(text):
    return stemmer.stem(WordNetLemmatizer().lemmatize(text, pos = 'v'))

def preprocess(text):
    result = []
    for token in gensim.utils.simple_preprocess(text):
```

```
        if token not in gensim.parsing.preprocessing.STOPWORDS and len(token) > 3:
            result.append(lemmatize_stemming(token))
    return result

processed_docs = [preprocess(sent) for sent in comment1]


# %%
processed_docs


# %%
processed_docs_2 = [preprocess(sent) for sent in comment2]
processed_docs_3 = [preprocess(sent) for sent in comment3]


# %%
def preprocess2(text):
    result = []
    for senten in text:
        if len(senten) > 3 :
            result.append(senten)
        else:
            pass
    return result

processed_docs2 = preprocess2(processed_docs)


# %%
print('total reviews:',len(processed_docs))
print('total reviews:',len(processed_docs_2))
print('total reviews:',len(processed_docs_3))


# %%
dictionary = gensim.corpora.Dictionary(processed_docs)


# %%
dictionary2 = gensim.corpora.Dictionary(processed_docs_2)
dictionary3 = gensim.corpora.Dictionary(processed_docs_3)


# %%
len(dictionary)
len(dictionary2)
len(dictionary3)


# %%
dictionary.filter_extremes(no_above = 0.2)

len(dictionary)


# %%
dictionary2.filter_extremes(no_above = 0.2)
```

```
len(dictionary2)


# %%
dictionary3.filter_extremes(no_above = 0.2)
len(dictionary3)


# %%
for i in range(10):
    print(i,dictionary3[i])


# %%
bow_corpus = [dictionary.doc2bow(doc) for doc in processed_docs]
print(processed_docs[1])
print(bow_corpus[1])


# %%
bow_corpus_2 = [dictionary2.doc2bow(doc) for doc in processed_docs_2]

bow_corpus_3 = [dictionary3.doc2bow(doc) for doc in processed_docs_3]


# %%
tfidf = models.TfidfModel(bow_corpus)
corpus_tfidf = tfidf[bow_corpus]
for doc in corpus_tfidf:
    pprint(doc)
    break


# %%
tfidf2 = models.TfidfModel(bow_corpus_2)
corpus_tfidf2 = tfidf[bow_corpus_2]
for doc in corpus_tfidf2:
    pprint(doc)
    break


# %%
tfidf3 = models.TfidfModel(bow_corpus_3)
corpus_tfidf3 = tfidf[bow_corpus_3]
for doc in corpus_tfidf3:
    pprint(doc)
    break


# %%
lda_model = gensim.models.LdaMulticore(bow_corpus, num_topics=3, id2word=dictionary, mir


# %%
lda_model2 = gensim.models.LdaMulticore(bow_corpus_2, num_topics=3, id2word=dictionary2,


# %%
```

```
lda_model3 = gensim.models.LdaMulticore(bow_corpus_3, num_topics=3, id2word=dictionary3,


# %%
for idx, topic in lda_model.print_topics(-1):
    print('Topic: {} \nWords: {}'.format(idx, topic))


# %%
for idx, topic in lda_model2.print_topics(-1):
    print('Topic: {} \nWords: {}'.format(idx, topic))


# %%
for idx, topic in lda_model3.print_topics(-1):
    print('Topic: {} \nWords: {}'.format(idx, topic))


# %%
docID = 279
print(comment1[docID])
print(processed_docs[docID])
for index,score in sorted(lda_model[bow_corpus[docID]], key = lambda tup: -1*tup[1]):
    print("\nScore: {}\t \nTopic{}: {}".format(score,index,lda_model.print_topic(index,


# %%
docID = 279
print(comment2[docID])
print(processed_docs_2[docID])
for index,score in sorted(lda_model2[bow_corpus_2[docID]], key = lambda tup: -1*tup[1]):
    print("\nScore: {}\t \nTopic{}: {}".format(score,index,lda_model2.print_topic(index,


# %%
docID = 279
print(comment3[docID])
print(processed_docs_3[docID])
for index,score in sorted(lda_model3[bow_corpus_3[docID]], key = lambda tup: -1*tup[1]):
    print("\nScore: {}\t \nTopic{}: {}".format(score,index,lda_model3.print_topic(index,

# %% [markdown]
# ## Problem B
# %% [markdown]
# - Use time as index

# %%
hairdry = clean1[['star_rating','helpful_votes','total_votes','vine','review_body','revi
hairdry['review_date']  = pd.to_datetime(hairdry['review_date'])
hairdry = hairdry.set_index('review_date')


# %%
micro = clean2[['star_rating','helpful_votes','total_votes','vine','review_body','review
micro['review_date']  = pd.to_datetime(micro['review_date'])
micro = micro.set_index('review_date')
```

```
# %%
paci =  clean3[['star_rating','helpful_votes','total_votes','vine','review_body','review
paci['review_date']  = pd.to_datetime(paci['review_date'])
paci = paci.set_index('review_date')

# %% [markdown]
# — Calculate topic values

# %%
def caltopic(topic, corpus):
    result = []
    for corp in corpus:
        value = lda_model[corp][topic][1]
        result.append(value)
    return result


# %%
topic0 = caltopic(0, bow_corpus)
topic1 = caltopic(1, bow_corpus)
topic2 = caltopic(2, bow_corpus)


# %%
hairdry['topic0'] = topic0
hairdry['topic1'] = topic1
hairdry['topic2'] = topic2


# %%
topic3 = caltopic(0, bow_corpus_2)
topic4 = caltopic(1, bow_corpus_2)
topic5 = caltopic(2, bow_corpus_2)


# %%
micro['topic0'] = topic3
micro['topic1'] = topic4
micro['topic2'] = topic5


# %%
topic6 = caltopic(0, bow_corpus_3)
topic7 = caltopic(1, bow_corpus_3)
topic8 = caltopic(2, bow_corpus_3)


# %%
paci['topic0'] = topic6
paci['topic1'] = topic7
paci['topic2'] = topic8


# %%
paci.to_csv('paci2.csv')
micro.to_csv('micro2.csv')
hairdry.to_csv('hairdry2.csv')
```

```
# %% [markdown]
# ## Problem B and C
# %% [markdown]
# — Calculate weights

# %%
import scipy.stats as st


# %%
# calculate d1 d2
hairdry['no_helpful'] = hairdry['total_votes'] — hairdry['helpful_votes']
beta_H = 5
beta_N = 5
hairdry['helpful_votes'] = hairdry['helpful_votes'] + beta_H
hairdry['no_helpful'] = hairdry['no_helpful'] + beta_N

beta1 = hairdry['helpful_votes'].to_list()
beta2 = hairdry['no_helpful'].to_list()

informa1 = []
for i in range(len(beta1)):
    beta_1 = beta1[i]
    beta_2 = beta2[i]
    informa1.append(st.beta.ppf(0.05, beta_1, beta_2))

informa1 = np.array(informa1)
hairweight1 = informa1 + 1

vine = hairdry['vine']

def calvine(x):
    if x == 'Y':
        return 0.5
    else:
        return 0

vine1 = hairdry.vine.apply(calvine).to_list()
hairweight2 = np.array(vine1)

totalhairweight = hairweight1 + hairweight2

def distvalue(x):
    if x >= 3:
        return 1
    else:
        return —1

attitude1 = hairdry.star_rating.apply(distvalue).to_list()
attitude1 = np.array(attitude1)

hairtop0 = np.array(hairdry['topic0'])
hairtop1 = np.array(hairdry['topic1'])
hairtop2 = np.array(hairdry['topic2'])
hairtop0 = hairtop0*attitude1
hairtop1 = hairtop1*attitude1
hairtop2 = hairtop2*attitude1
top0score = pd.DataFrame(totalhairweight * hairtop0)
```

```
top1score = pd.DataFrame(totalhairweight * hairtop1)
top2score = pd.DataFrame(totalhairweight * hairtop2)

hairscore = pd.concat([top0score, top1score, top2score], axis = 1)
hairscore.columns = ['score1' , 'score2' , 'score3']
hairscore  = hairscore.set_index(hairdry.index)

hairscoremean = hairscore.resample('1AS').mean()
stars = hairdry['star_rating']
hairstarmean = stars.resample('1AS').mean()


# %%
hairscoremean = hairscoremean.dropna()
hairstarmean = hairstarmean.dropna()


# %%
micro['no_helpful'] = micro['total_votes'] - micro['helpful_votes']
beta_H = 5
beta_N = 5
micro['helpful_votes'] = micro['helpful_votes'] + beta_H
micro['no_helpful'] = micro['no_helpful'] + beta_N

beta1 = micro['helpful_votes'].to_list()
beta2 = micro['no_helpful'].to_list()

informa2 = []
for i in range(len(beta1)):
    beta_1 = beta1[i]
    beta_2 = beta2[i]
    informa2.append(st.beta.ppf(0.05, beta_1, beta_2))

informa2 = np.array(informa2)
microweight1 = informa2 + 1

vine = micro['vine']

vine2 = micro.vine.apply(calvine).to_list()
microweight2 = np.array(vine2)

totalmicroweight = microweight1 + microweight2


attitude1 = micro.star_rating.apply(distvalue).to_list()
attitude1 = np.array(attitude1)

microtop0 = np.array(micro['topic0'])
microtop1 = np.array(micro['topic1'])
microtop2 = np.array(micro['topic2'])
microtop0 = microtop0*attitude1
microtop1 = microtop1*attitude1
microtop2 = microtop2*attitude1
mtop0score = pd.DataFrame(totalmicroweight * microtop0)
mtop1score = pd.DataFrame(totalmicroweight * microtop1)
mtop2score = pd.DataFrame(totalmicroweight * microtop2)

microscore = pd.concat([mtop0score, mtop1score, mtop2score], axis = 1)
```

```
microscore.columns = ['score1' , 'score2' , 'score3']
microscore = microscore.set_index(micro.index)

microscoremean = microscore.resample('1AS').mean()
mstars = micro['star_rating']
microstarmean = mstars.resample('1AS').mean()


# %%
paci['no_helpful'] = paci['total_votes'] - paci['helpful_votes']
beta_H = 5
beta_N = 5
paci['helpful_votes'] = paci['helpful_votes'] + beta_H
paci['no_helpful'] = paci['no_helpful'] + beta_N

beta1 = paci['helpful_votes'].to_list()
beta2 = paci['no_helpful'].to_list()

informa3 = []
for i in range(len(beta1)):
    beta_1 = beta1[i]
    beta_2 = beta2[i]
    informa3.append(st.beta.ppf(0.05, beta_1, beta_2))

informa3 = np.array(informa3)
paciweight1 = informa3 + 1

vine = paci['vine']

vine2 = paci.vine.apply(calvine).to_list()
paciweight2 = np.array(vine2)

totalpaciweight = paciweight1 + paciweight2


attitude1 = paci.star_rating.apply(distvalue).to_list()
attitude1 = np.array(attitude1)

pacitop0 = np.array(paci['topic0'])
pacitop1 = np.array(paci['topic1'])
pacitop2 = np.array(paci['topic2'])
pacitop0 = pacitop0*attitude1
pacitop1 = pacitop1*attitude1
pacitop2 = pacitop2*attitude1
ptop0score = pd.DataFrame(totalpaciweight * pacitop0)
ptop1score = pd.DataFrame(totalpaciweight * pacitop1)
ptop2score = pd.DataFrame(totalpaciweight * pacitop2)


paciscore = pd.concat([ptop0score, ptop1score, ptop2score], axis = 1)
paciscore.columns = ['score1' , 'score2' , 'score3']
paciscore  = paciscore.set_index(paci.index)

paciscoremean = paciscore.resample('1AS').mean()
pstars = paci['star_rating']
pacistarmean = pstars.resample('1AS').mean()
```

```
# %%
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')


# %%
def trannum(data):
    score1 = np.array(data['score1'])
    score2 = np.array(data['score2'])
    score3 = np.array(data['score3'])

    return score1, score2, score3

score1, score2, score3 = trannum(hairscoremean)

def tranplot(data1, data2, data3, data4):
    plt.plot(data1, color = 'red', label = 'score1')
    plt.plot(data2, color = 'blue', label = 'score2')
    plt.plot(data3, color = 'black', label = 'score3')
    plt.plot(data4, color = 'orange', label = 'stars')

    plt.xlabel('Time')
    plt.ylabel('Score')
    plt.legend(loc = 'upper right')

stars = np.array(hairstarmean)
tranplot(score1, score2, score3, stars)


# %%
score1, score2, score3 = trannum(microscoremean.dropna())
stars = np.array(microstarmean.dropna())
tranplot(score1, score2, score3, stars)


# %%
score1, score2, score3 = trannum(paciscoremean.dropna())
stars = np.array(pacistarmean.dropna())
tranplot(score1, score2, score3, stars)

# %% [markdown]
# — Correlation Analysis
# %% [markdown]
# Suppose that `stars` represents the tendency.

# %%
score1 = np.array(hairscore.score1)
score2 = np.array(hairscore.score2)
score3 = np.array(hairscore.score3)


# %%
stars = np.array(hairdry.star_rating)


# %%
coef4 = np.corrcoef((score3+score2+score1)/2, stars)[0][1]
coef4
```

```
# %% [markdown]
# Now the microwave:

# %%
def analyzescore(scores, stars):
    score1 = np.array(scores.score1)
    score2 = np.array(scores.score2)
    score3 = np.array(scores.score3)

    coef1 = np.corrcoef(score1, stars)[0][1]
    coef2 = np.corrcoef(score2, stars)[0][1]
    coef3 = np.corrcoef(score3, stars)[0][1]
    coef4 = np.corrcoef((score1+score2+score3)/3, stars)[0][1]
    coef5 = np.corrcoef((score1+score2)/2, stars)[0][1]
    coef6 = np.corrcoef((score1+score3)/2, stars)[0][1]
    coef7 = np.corrcoef((score3+score2)/2, stars)[0][1]

    coefs = (coef1,coef2,coef3,coef4,coef5,coef6,coef7)
    return(coefs)


# %%
analyzescore(hairscore, stars = stars)


# %%
analyzescore(microscore, stars = mstars)


# %%
analyzescore(paciscore, stars = pstars)


# %%
hstars = hairdry.groupby('review_date').star_rating.mean()


# %%
hstars.to_csv('hstars.csv')


# %%
hd = pd.concat([hdcount,hdstars], axis=1)


# %%
hd.to_csv('hd.csv')


# %%
def countvalue(x):
    x = x.tolist()
    return x.count(5) + x.count(1)


# %%
def countvalue5(x):
```

```
    x = x.tolist()
    return x.count(5)+x.count(4)


# %%
def countvalue1(x):
    x = x.tolist()
    return x.count(1)+x.count(2)


# %%
a = [1,2,3,4]
type(a)


# %%
num = 0
pastars1 = paci.resample('3D').star_rating.apply(countvalue)


# %%
pastars1


# %%
pastarsp = paci.resample('3D').star_rating.apply(countvalue5).tolist()
pastarsn = paci.resample('3D').star_rating.apply(countvalue1).tolist()
pacount = paci.resample('3D').star_rating.count().tolist()

pastarsp = pd.DataFrame(pastarsp)
pastarsn = pd.DataFrame(pastarsn)
pacount = pd.DataFrame(pacount)

pc = pd.concat([pacount,pastarsp,pastarsn], axis=1)
pc.columns = ['count','starsp','starsn']
pc = pc[pc['count'] != 0]

pc.to_csv('pc.csv')


# %%
mwstarsp =micro.resample('3D').star_rating.apply(countvalue5).tolist()
mwstarsn = micro.resample('3D').star_rating.apply(countvalue1).tolist()
mwcount =micro.resample('3D').star_rating.count().tolist()

mwstarsp = pd.DataFrame(mwstarsp)
mwstarsn = pd.DataFrame(mwstarsn)
mwcount = pd.DataFrame(mwcount)

mw = pd.concat([mwcount,mwstarsp,mwstarsn], axis=1)
mw.columns = ['count','starsp','starsn']
mw = mw[mw['count'] != 0]

mw.to_csv('mw.csv')


# %%
hdstarsp =hairdry.resample('3D').star_rating.apply(countvalue5).tolist()
```

```
hdstarsn = hairdry.resample('3D').star_rating.apply(countvalue1).tolist()
hdcount =hairdry.resample('3D').star_rating.count().tolist()

hdstarsp = pd.DataFrame(hdstarsp)
hdstarsn = pd.DataFrame(hdstarsn)
hdcount = pd.DataFrame(hdcount)

hd = pd.concat([hdcount,hdstarsp,hdstarsn], axis=1)
hd.columns = ['count','starsp','starsn']
hd = hd[hd['count'] != 0]

hd.to_csv('hd.csv')


# %%
pastars1 = paci.resample('3D').star_rating.apply(countvalue).tolist()
pacount = paci.resample('3D').star_rating.count().tolist()

pastars = pd.DataFrame(pastars)
pacount = pd.DataFrame(pacount)

pc = pd.concat([pacount,pastars], axis=1)
pc.columns = ['count','stars']
pc = pc[pc['count'] != 0]

pc.to_csv('pc.csv')


# %%
pc = pd.concat([pacount,pastars], axis=1)
pc.to_csv('pc.csv')
```