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

IWOM Analysis in data mining insight

With the development of online shopping in recent years, online shopping platforms, such as Amazon, provide customers with some methods, i.e. staring rating, review and helpful rating, to express their opinions about the product or the store. The evaluation data customers submit could profoundly affect the store and product's reputation, which is known as internet word-of-mouth(IWOM).

To begin with, we preprocessed text data by data cleaning and normalization.Thus, we obtained a set of candidate keywords whose size was largely reduced. Then, before measure identification, we did **Natural Language Processing(NLP)** which is used to extract keywords and process text. We split text data given into three categories: product title, review title and review body. Then, we do keyword extraction on them respectively. For the titles, we utilize **term frequency–inverse document frequency(TF-IDF)** to extract keywords from candidate keywords. However, we find that TF-IDF is not suitable for review body in that it can hardly extract any sentiment information. Instead, we developed a sentiment analysis model for analyzing review body. By utilizing the combination of pretrained fastText word embedding and **support vector machine(SVM)** classifier, we got the sentiment score of each word and a set of keywords. To get a direct view of the keywords, we visualized keywords of the three kinds text using word cloud. In particular, keywords of review body are split into positive or negative category.

After text processing, we developed a **profile** that contains eight data measures, half of which are measures on the product and the rest are on reviews. We considered lots of factors, such as time, the effectiveness and sentiment of reviews and etc. In particular, we developed an **influence factor(IF) model** which is based on information entropy, so that we could quantize the effectiveness of each review.

Then, to measure the trend of a product's reputation and find the relation based on time series between ratings and reviews, we utilized **Autoregressive moving average model(ARMA)** to analyze time series. The model predicted products' reputation and correlation between star ratings and reviews.

To determine the combination of text-based measure and ratings-based measures, we need to obtain a subset of the total set of features, which we constructed based one profile. The subset should contain fewer features while these features have the best performance in reflecting products' potential. Therefore, we transformed the problem into a combinatorial optimization problem by developing a feature selection optimization model. To solve the model, we developed **TOPSIS** model based on data of a long period to evaluate the success of products. Then, we took its results as labels and constructed a total set of feature based on the product's data in the first six months. Afterwards, we utilized **Elastic Net regression**, which is a regularized method for selection features, to solve the optimization model. By analyzing the feature subset that we obtained from the model, we would be able to suggest product's design direction and sales strategy.

To get the relations between specific descriptors and ratings, we did data mining on all of the reviews that each product had. Then, in order to obtain the set of keywords which influence ratings the most, we did Lasso regression on ratings and candidate keywords frequency. The linear weight of one keyword reflects the effect it has on rating.Then we used cross validation to optimize Lasso.

Keywords: NLP, Influence factor, feature selection optimization model

Contents

1 Introduction

1.1 Background

Online shopping has been of great popularity these years. Not only is it convenient and economical for the customers, but also good for companies in that they could easily collect all the data they want, including feedbacks from the customers, to get to know the items they sale and the markets better, instead of spending lots of efforts to do questionnaires or telephone follow-up. Convenient as it is, online shopping, especially the reviews that purchasers submitted, could have deep effect on the company's reputation. A large quantity of reviews could form internet word-of-mouth(IWOM)[1], the negative voice of which might lead to reputation crisis that is fatal to the company. Therefore, companies have to grasp the IWOM at all times to prepare for reputation crisis.

As one of the most remarkable online retailers in the world, Amazon does pretty well in providing ways for purchasers to do rating and review. In particular, the rating is called star ratings with a scale of 1 to 5, corresponding to low satisfaction to high satisfaction. Furthermore, to evaluate the effectiveness of the reviews submitted, Amazon launched helpfulness rating, where customers are able to rate the reviews according to how helpful they think the review is.

1.2 Problem Statement

Sunshine company is about to launch three products, which are microwave oven, baby pacifier and hair dryer. So, for pre-market research, an appropriate online sales strategy and notable design features, they would like to have a model to analyze the data, i.e. star rating, review and helpful rating, of other competing products. In particular, the company is interested in the time-based pattern in the data and the relationships between the data, say, some words in the review may have something to do with the number of rating stars.

2 Assumptions

- **Assumption 1:** Keywords could reflect the main idea of the whole review to some extent.
- **Assumption 2**: The more successful a product is, the more possible it is that the product gets reviews, high ratings and longer life cycle.

- **Assumption 3**:The sentiment tendency of one word used daily would not change in the reviews.

3 Data Preprocessing

In this section, We give an overview of how the data is preprocessed for the preparation of developing our Text Processing Model. In order to describe the whole process more clearly, we made a flowchart for it, as can be seen in Figure 1.

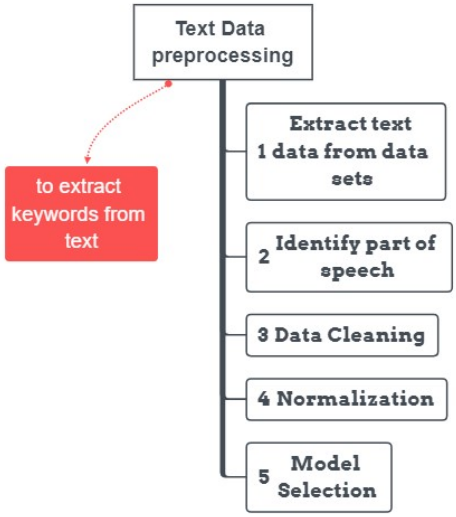

Figure 1. Flowchart of data preprocessing

Due to the fact that reviews contain too much information that makes it hard for us to develop our model, we mainly focus on processing text data. In order to extract key information from text reviews as much as possible, so that we could explore the relationship between ratings and reviews, we adopt supervised methods to do keyword extraction.

3.1 Create dictionary

To begin with, one document could be represented as the vector form

$$D = (w_1, w_2, \dots, w_n) \tag{1}$$

Where w_k represents a word in the document D which contains n words. It's worth noticing that one document contains lots of stop words, phrase delimiters and word delimiters, e.g. a, the, of. These words and delimiters are basically of little use for getting main ideas of the document while there is a large quantity of them in one document. To simplify the words vector and get

more precise keywords, we use the stop words and delimiters to parse the document into a set of candidate keywords[3].

To get a set of candidate keywords, we do the steps as follows:

- **Identify part of speech(POS)**. Adding POS tags to the words in the text.
- **Data cleaning.** By getting rid of stop words, phrase delimiters and word delimiters, we could get a basic set of candidate keywords. In particular, we also erase words that only have 2 or fewer character, for the reason that the stop words might not be erased thoroughly.
- **Normalization.** To Eliminate the influence of different word forms, we change all the capital letters into lower-case ones.

Through the above data processing, a large number of words were reduced. The reduction ratio is listed in chart 1 below.

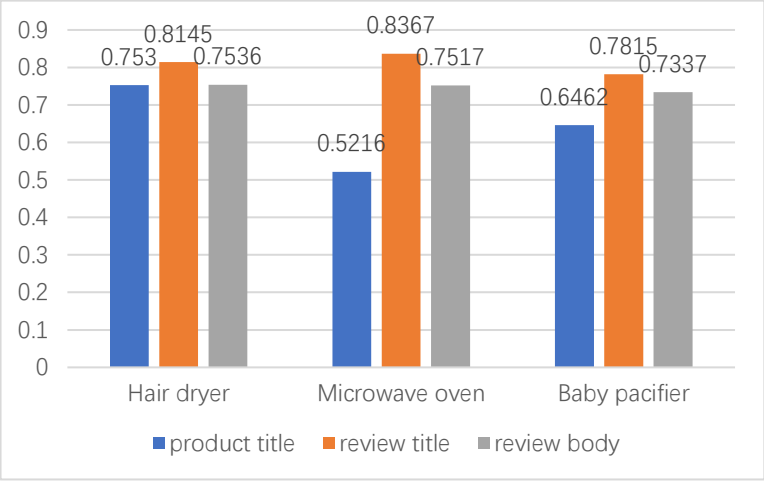

Chart 1. Reduction ratio

The chart shows that each text category of each product gets rid of more than half of the words it has, which means that the extraction of candidate keywords is effective.

Then, in order to extract text features, we choose between bag-of-words model and N-gram model.

3.1.1 Create word bag

Bag-of-words model and N-gram model are ways for extracting features from text. The differences between them lie in the fact that in the former model, text is represented as a bag of words that does not take grammar or word order into account, but keeps the multiplicity of each word in the text, while the latter one based on Markov assumption considers word order so that it is able to

extract phrases. Considering the fact that word order in the reviews is not important because we just need to extract the keywords, we decide to utilize bag-of-words model.

4 Natural Language Processing

In this section, we introduce the Text Processing Model we developed. First, we use TF-IDF to extract keywords from the set of candidate keywords we got earlier. owever, we find that the keywords extracted from review body contain too many meaningless words and do not contain much sentiment information. Thus, we split the text data into three categories: product title, review title and review body. Then, we developed a Sentiment Analysis Model to analyze review body.

4.1 Weight measure of Candidate keywords

In order to select keywords from the set of candidate keywords, we utilize term frequency–inverse document frequency(TF-IDF)[2] to measure the weight of each word in the candidate keywords set.

TF-IDF is better than word frequency in that words that are common in a single or a small group of documents tend to have higher TF-IDF values than common words. Thus, words that mentioned several times in few reviews have low probability to be counted as keywords, which is appropriate because these words usually do not reflect common information.

The overall approach of TF-IDF works as follows.

$$TFIDF(w,d) = TF(w,d) \times IDF(w) \tag{2}$$

$$TF(w,d) = \frac{N(w,d)}{|d|} \tag{3}$$

$$IDF(w) = \log\left(1 + \frac{|D|}{N(w,D)}\right) \tag{4}$$

where N(w,d) is the number of times word w appears in document d, N(w,D) is the number of documents where w appears in D. |d| is the size of document d and |D| is the number of documents.

Thus, we extract the keywords from the reviews. To be more distinct, we visualize the keywords of baby pacifier dataset with word cloud, as can be seen in Figure 2.

Figure 2. word cloud of keywords in titles of baby pacifier dataset

The word cloud in the left contains keywords extracted from the product's titles and the ones in the right are from the reviews' titles of the products. As for the review bodies, we find that after the extraction, all of the negative expressions are gone. It's impossible to find the relations between star ratings and specific words under this circumstance. Therefore, we built a sentiment analysis model to solve this problem.

4.2 Sentiment Analysis Model

We trained a sentiment classifier based on support vector machine to predict the sentiment of words in the reviews. The classifier is trained using pretrained fastText word embedding and a list of words annotated as positive or negative. The main steps of building the classifier are showed as follows:

- **Word embedding.**

  To avoid dimension disaster, we do word embedding by loading pretrained word embedding. Word embeddings convert words into numeric vectors, and since they can capture semantic details of the words, similar words would have similar vectors.

- **Load sentiment lexicon**.

  Sentiment lexicon contains both positive and negative words that are annotated.

- **Embedding sentiment lexicon**

- **Train SVM classifier**.

  We trained a support vector machine(SVM) classifier that could classify word vectors into positive or negative categories.

  The decision function of SVM is:

$$f(\boldsymbol{x}) = \omega \cdot \Phi(\boldsymbol{x}) + b = \sum_i \alpha_i^0 y_i K(\boldsymbol{x_i}, \boldsymbol{x}) + b \quad (5)$$

  where K(·,·) is a kernel function that maps x into a high dimensional space, ω is weight vector and $\omega = \sum_i \alpha_i^0 y_i x_i$, b is a constant.

- **Test SVM classifier.**

  We use confusion matrix to show the result of the test and effectiveness of the SVM classifier we trained, as can be seen in Figure 3.

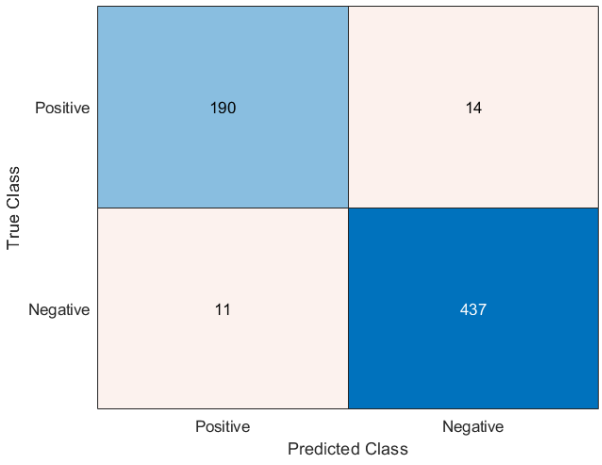

Figure 3. Confusion matrix of test result

From the confusion matrix, we could evaluate the result with F1-score.

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (6)$$

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (7)$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \quad (8)$$

We get a F1 score of 0.938, which is pretty good.

- **Embedding input words**

To better embed the words to be predicted, we utilize continuous bag of words model(CBOW), which could provide better word embedding that is compressed and contains the context.

We visualize the word embedding in Figure 4.

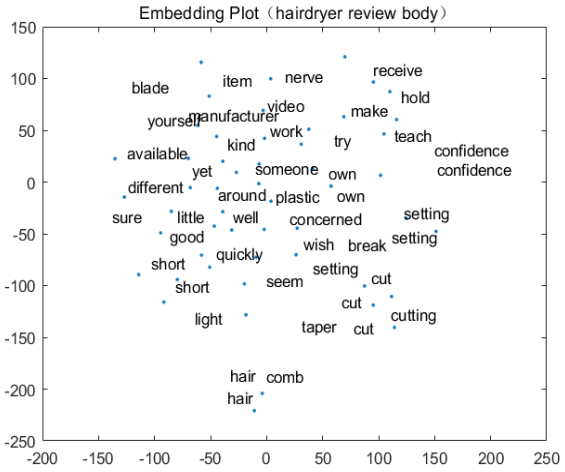

Figure 4. Embedding plot of hair dryer review body

- **Predict text sentiment**

Predict review body's sentiment using SVM and each word would get their sentiment score, which is used to clarify it's label. aving gained the predict results in the test step, we visualize them in word clouds. The word sizes correspond to prediction scores. For example, the sentiment classification result of baby pacifier is visualized as Figure 5.

Figure 5. word cloud of keywords in reviews of baby pacifier dataset

The word cloud in the left contains words with positive sentiment and the one in the right contains words with negative sentiment.

5 Data Measures Identification

In this section, we mainly discuss the data measures that we identified. aving considered every aspect that could reflect the trend of a product's reputation and sales, we identified eight data measures, half of which are measures on the product and the rest are on reviews. To have an overall view of the eight measures, we summarize them in the table 2.

Name	Formula
Detailed seller ratings	$DSR(d) = \frac{\sum RATINGS}{Number\ of\ ratings\ in\ six\ months}$
Monthly active reviews	——
Monthly review increment	——
Internet word of mouth	$IWOM = \sum IF^* \times Score_{review}$
Influence factor	$H_j = -\sum_{i=1}^{n_j} P_{j,i} \times \alpha_{j,i} \times \log_{\alpha_j} P_{j,i}$ $C = \sum H_j + \lambda$
Sentiment score of title	$Score_{title} = \sum Score_{word}$
Sentiment score of body	$Score_{body} = \sum Score_{word}$
Sentiment score of review	$Score_{review} = \frac{M \times Score_{body} + N \times Score_{title}}{M + N}$

Table2. data measures

i. Detailed seller ratings

We adopt Detailed seller ratings to measure the score of one product based on all the ratings it received in the last six months.

Lots of online retailers allow purchasers to rate the store in many aspects, such as delivery service, attitude of service, item description and etc.. For a

more accurate rating, they also allow customers to do the rating in 60 days after the purchase. Then, the ratings would be collected for the platform rating the store. Considering the fact that ratings done long time ago may not evaluate the store accurately, the online platform usually utilize dynamic rating strategy to rate the store, i.e. Detailed seller ratings(DSR).

DSR of date d are calculated in the way as follows:

$$DSR(d) = \frac{\sum RATINGS}{Number\ of\ ratings\ in\ six\ months} \tag{5}$$

$$RATINGS \in A(date)$$

$$A(date) = \{x | x\ is\ rating\ that\ done\ six\ months\ before\ date\ d\}$$

The formula above describes evidently that the DSR of the product on date d is the average rating of the last six months.

ii. Monthly active reviews

We utilize the number of reviews submitted in one month to measure the sales situation of one product in the month.

iii. Monthly review increment

We utilize the number of reviews submitted in one month to measure the trend of sales situation of one product in the month.

iv. Influence factor

To develop a model for evaluating the influence and importance of each review, we take four kinds of data, i.e. helpful_votes, vine, verified_purchase, review as Influence factors(IF), which stand for number of stars, number of helpful votes, being member or not, having purchased or not and review text. The description of the factors could be seen in table 3.

Factor	Values	Number of Values
helpful_votes	{Y,N}	$n_1 = 2$
vine	{Y,N}	$n_2 = 2$
verified_purchase	{Y,N}	$n_3 = 2$
review	——	n_4

Tabel 3. Description of influence factors

As the fact that information entropy could describe the uncertainty of variates, and the more uncertain one variate is, the more information it contains, we utilize it to evaluate the importance of each factor.

The fomulas of the model are listed as follows[6]:

$$H_j = -\sum_{i=1}^{n_j} P_{j,i} \times \alpha_{j,i} \times \log_{n_j} P_{j,i} \quad , \text{j} = 1,2,3,4,5 \tag{6}$$

$$IF = \sum_{j=1}^{5} H_j + \lambda \tag{7}$$

where H_j is the entropy of factor j, $P_{j,i}$ is the probability that factor j's No.i value shows up, $\alpha_{j,i}$ is the weight of factor j's No.i value, λ is a constant value and C is the influence score.

v. Internet word of mouth

We developed a model of Internet word of mouth(IWOM) to measure product's reputation. The formula is as follows:

$$IWOM = \sum IF^* \times Score_{review} \tag{8}$$

where IF^* is the normalized influence factor and $Score_{review}$ is the sentiment score of review. The higher IWOM is, the better the reputation is, as the fact that higher IWOM means the product has more positive reviews.

vi. Sentiment score of title

After exploring the data, we find that titles of reviews usually contain sentiment. Therefore, we use the formula below to evaluate it.

$$Score_{title} = \sum Score_{word} \tag{9}$$

where $Score_{title}$ is the sentiment score of one title and it is the linear sum of sentiment score of the words in the title. In particular, due to the fact that in this case word order does not count, we utilize bag-of-words model to get the sentiment score of words. Only words in the bag have scores.

vii. Sentiment score of body

Similar to sentiment score of title, the formula is as follows:

$$Score_{body} = \sum Score_{word} \tag{10}$$

viii. Sentiment score of review

As for evaluating the sentiment score of review, we take one common sense into account. It's usually the case that people tend to get overall information from the title when the review body is too long, and vice versa. Therefore, we

evaluate the score as the formula:

$$Score_{review} = \frac{M \times Score_{body} + N \times Score_{title}}{M+N} \tag{11}$$

where M and N are weights of sentiment score of review body and title respectively.

6 Time-based measure

In this section, we introduce the model we use to discuss time series.

6.1 ARMA

ARMA stands for Autoregressive moving average model, which is useful to discuss time series. We focus on ARMA(p,q), where p is the number of autoregressive terms and q is the number of moving average terms.

The main process of ARMA(p,q) is showed as follows:[5]

Normalize the data to get sequence with zero mean

$$\{X_t | t = 0, \pm 1, \pm 2 \dots\} \tag{12}$$

Then, we can get

$$x_t - \varphi_1 x_{t-1} - \varphi_2 x_{t-2} - \cdots - \varphi_p x_{t-p} = \varepsilon_t - \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \tag{13}$$

Thus

$$\varphi(B) x_t = \theta(B) \varepsilon_t \tag{14}$$

and

$$x_t = \varphi_1 x_{t-1} + \varepsilon_t \tag{15}$$

$$x_{t+1} = \varphi_1 x_t + \varepsilon_{t+1} \tag{16}$$

Then, do condition least squares estimation on the parameters:

$$s(\varphi) = \sum_{t=p+1}^{n} \varepsilon_t^2 \tag{17}$$

$$= \sum_{t=p+1}^{n} (x_t - \varphi_1 x_{t-1} - \varphi_2 x_{t-2} - \cdots - \varphi_p x_{t-p})^2 \tag{18}$$

To get argmin $s(\varphi)$, let

$$\frac{\alpha s(\varphi)}{\alpha \varphi} = 0 \tag{19}$$

Get

$$(A^T A)\varphi = A^T \alpha \tag{20}$$

Where

$$A = \begin{pmatrix} x_p & x_{p-1} & \cdots & x_1 \\ x_{p+1} & x_p & \cdots & x_2 \\ \cdots & \cdots & \cdots & \cdots \\ x_{n-1} & x_{n-2} & \cdots & x_{n-p} \end{pmatrix} \tag{21}$$

$$\alpha = \begin{pmatrix} x_{p+1} \\ x_{p+2} \\ \cdots \\ x_n \end{pmatrix} \tag{22}$$

Let

$$\widetilde{x_{ij}^{t}} = \frac{1}{n}\sum_{t=p+1}^{n} x_{t-i}x_{t-j} \ , i = 0,1,\dots,p \ , j = 1,2,\dots,p \tag{23}$$

Get

$$\widehat{\varphi_2} = \begin{pmatrix} \dot{\varphi}_1^t \\ \dot{\varphi}_2^t \\ \cdots \\ \dot{\varphi}_p^t \end{pmatrix} = \begin{pmatrix} \widetilde{\gamma_{11}^t} & \widetilde{\gamma_{12}^t} & \cdots & \widetilde{\gamma_{1p}^t} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \widetilde{\gamma_{p1}^t} & \widetilde{\gamma_{p1}^t} & \cdots & \widetilde{\gamma_{pp}^t} \end{pmatrix} \begin{pmatrix} \widetilde{\gamma_{01}^t} \\ \widetilde{\gamma_{02}^t} \\ \cdots \\ \widetilde{\gamma_{0p}^t} \end{pmatrix} \tag{24}$$

6.2 Product Reputation Trend Prediction

We predict products' reputation trend based on ARMA, as can be seen in Algorithm 1.

Algorithm 1

Input: $X_t \leftarrow Influence\ of\ each\ product\ in\ time\ t$

Output: X_{t+1}

1. Construct equation: $\varphi(B)X_t = \theta(B)\varepsilon_t$
2. Do condition least squares estimation on parameters
3. $X_{t+1} = \varphi_1 X_t + \varepsilon_{t+1}$

We run Algorithm 1 on all three products and visualized their reputation trends.

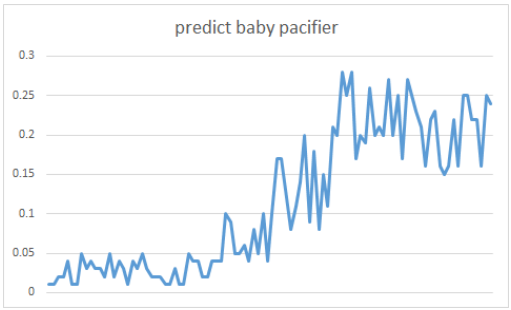

Figure 6. prediction of baby pacifier

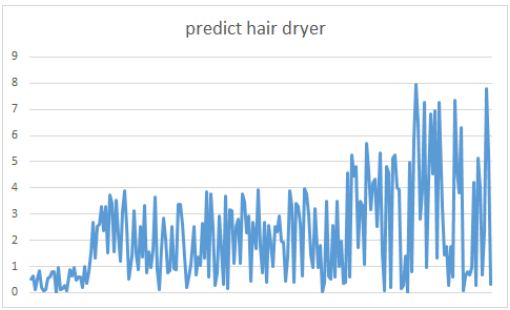

Figure 7. prediction of hair dryer

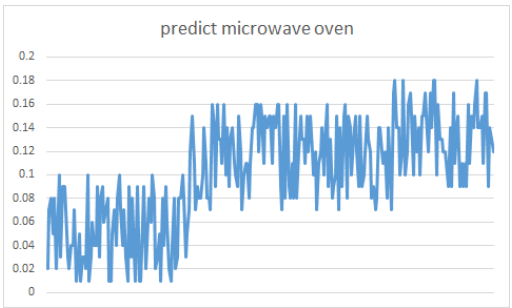

Figure 8. prediction of microwave oven

Figure 9. relationsbetween ratings

and reviews

From the time series prediction figures above, we can see that these three products' reputation have an upward trend with jitters.

6.3 Rating's Influence on review

We discuss rating's influence on review based on ARMA, as can be seen in Algorithm 2. It's worth noticing that the output is the number of reviews received by the product in future time t+1.

Algorithm 2

Input: $X_t \leftarrow each\ star\ rating\ along\ with\ its\ review\ during\ time\ t$

Output: X_{t+1}

1. Construct equation: $\varphi(B)X_t = \theta(B)\varepsilon_t$
2. Do condition least squares estimation on parameters
3. $X_{t+1} = \varphi_1 X_t + \varepsilon_{t+1}$
4. Construct relationship diagram of trend of star rating and review

We run algorithm 2 and get the result as figure 9.

By discussing the correlation in time series, we got the relations between star ratings and reviews as depicted in Figure 9. It's evident that the trend of the relations are not monotonic.

7 Feature Selection optimization model

In this section, we introduce the development of our feature selection optimization model.

By combining text-based measures and rating-based measures identified above, we get too many mearsures that Sunshine company can hardly track them all. It's important to get rid of irrelevant features and redundant features, by which we could keep fewer features but have most of the information. Therefore, we transformed the problem into a combinatorial optimization problem by developing a feature selection optimization model.

To be more distinct, we describe the total features set in Table 4.

Feature	Description
X_1	Average growth rate of score in six months
X_2	Average growth rate of reputation

	in six months
X_3	Evaluated score of one product
$X_4 \sim X_{24}$	number of occurrences of key words in one product title
$X_{25} \sim X_{125}$	number of occurrences of key words in one review

Table 4. Product evaluation feature description

7.1 Optimization model

In order to get best feature sub set, we mainly consider the problem as this way: given a fixed m $\ll$ n, find the m features that could give the smallest expected generalization error.

This problem is formulated as follows.

Given a set of functions

$$y = f(\mathbf{x}, \alpha) \tag{25}$$

We need to find a preprocessing of the data $\mathbf{x} \rightarrow (\mathbf{x} * \sigma), \sigma \in \{0,1\}^n$, and the parameters α such that they could give

$$min_\alpha \tau(\sigma, \alpha) \tag{26}$$

Where[4]

$$\tau(\sigma, \alpha) = \int V(y, f\big((\mathbf{x} * \sigma), \alpha\big)) dP(\mathbf{x}, y) \tag{27}$$
$$s.t. \|\sigma\|_0 = m$$

$P(\mathbf{x}, y)$ is unknown but makes sure completeness of subset, $\mathbf{x} * \sigma = (x_1\sigma_1, x_2\sigma_2, \dots, x_n\sigma_n)$, V(·,·) is a loss function.

To solve this problem and make it applicable for our circumstance, we consider two steps:

i. Based on a long-term (more than one year) consideration of a product, we have developed a model to evaluate the degree of success and failure of a product by using TOPSIS (Technique for Order Preference by Similarity to an Ideal Solution).

ii. Utilize embedding feature selection model to find optional sub set

7.2 Data Selection

To assess the potential, we need to select the products that have been on the market for more than one year. Only these products would have enough data for us to find out what the most succesful product's data is like.

7.3 TOPSIS method

TOPSIS is a multi-criteria decision analysis method which is often used to

evaluate a set of alternatives by comparing them with the ideal alternative that the method learned.

The details of TOPSIS are listed as follows:

- Assumption.

  The criteria of the alternatives are monotonical.

- Normalization.

  Put the original dataset into matrix as follows:

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} \tag{29}$$

  where n is the number of alternatives and m is the number of criteria.

  matrix X is normalised by:

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}} \text{ , i = 1,2,…,n, j = 1,2,…,m} \tag{30}$$

  Finally, we get the normalized matrix Z:

$$Z = \begin{bmatrix} z_{11} & \cdots & z_{1m} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nm} \end{bmatrix} \tag{31}$$

- Determine best and worst alternatives.

$$Z_b = (\max\{z_{11}, z_{21}, \dots, z_{n1}\}, \dots, \max\{z_{1m}, z_{2m}, \dots, z_{nm}\}) \tag{32}$$

$$Z_w = (\min\{z_{11}, z_{21}, \dots, z_{n1}\}, \dots, \min\{z_{1m}, z_{2m}, \dots, z_{nm}\}) \tag{33}$$

- Distance Calculation.

  Calculate the L^2-distance between target alternative a and best ,worst alternative respectively:

$$d_{ab} = \sqrt{\sum_{j=1}^{m} w_j \times (Z_{bj} - z_{ij})^2} \tag{34}$$

$$d_{aw} = \sqrt{\sum_{j=1}^{m} w_j \times (Z_{wj} - z_{ij})^2} \tag{35}$$

- Similarity Calculation

  The similarity between alternative i and the best alternative could be evaluated as follows:

$$Score_i = \frac{d_{aw}}{d_{aw} + d_{ab}}$$

  where $Score_i$ is the score of alternative i. The closer $Score_i$ is to 1, the better alternative i is.

7.4 Elastic Net Regression

Elastic Net regression is a linear regression that is trained with both L_1 and L_2-norm regularization of the weights. It could learn a sparse model while maintaining the regularization properties of Ridge, which is to say, Elastic Net regression combines ridge and Lasso regression and avoids their weaknesses. Thus, by tracking features of the smallest feature subset the regression gets, the company would know if the product has the potential to be successful only in several months.

Elastic Net is of great use when there are lots of features that are correlated with one another. It overcomes the limitations of Lasso in that Lasso just selects one of the features and ignores others, while Elastic Net selects both.

Mathematically, the object function to minimize is :

$$min_w \frac{1}{2n_{samples}} \|Xw - Score\|_2^2 + \alpha\rho\|w\|_1 + \frac{\alpha(1-\rho)}{2}\|w\|_2^2 \quad (28)$$

where ρ is the ratio of L_1 and L_2-norm and the $Score$ is the result from TOPSIS method

7.5 Instantiation

In our case, we choose follow measures as criteria to combine them. It's worth noticing that all of the three are based on the time series of one product's whole sales life cycle.

i. Number of valid reviews
ii. DSR
iii. Selected text-based feature:

	X_4 ~ X_{24}	X_{25} ~ X_{125}
pacifier	Wubban,pilliphs , manufacture , natural	elegant, discolored , versatility, adaptability
microwave	Quality, excellent	Versatility, durable , afford , overcooked, moldy
hair_dryer	Quality, dry	Uncluster, inexpensive,inconvenient,overheats, lightweight/convenient

Through embedding feature selection model, we find the optional sub set which indicate the potential of success above. Analysis result, we can provide product design direction reference, guide product marketing strategy. As for microwave, we suggest the company to focus on quality, product function and Cost performance.

7.6 Find most related words with ratings

Although SVM is good at generlization, it does not work well on this circumstance. We found that it happens that there are words of positive sentiment in low rating reviews. In order to explore the relationship between each product's market review words and the rating of the review, we need to performed specific data minning . Due to the regularization of a norm, the learning is highly sparse, so that we can find out several key words that have the most important influence on star level. We utilize Lasso regression to pick a set of keywords that are most related to ratings. Then, we evaluate the impact each keyword has on ratings by the linear weights learned from the regression.

7.6.1 LASSO

Our model is based on Lasso regression, which does penalized regression on all of the features' weights to compress the weights of less important features to zero so that they could be eliminated from the model.

To be more specific, Lasso stands for Least Absolute Shrinkage and Selection Operator, which is a linear model that estimates sparse coefficients. Mathematically, Lasso regression performs L1 regularization, which adds a penalty that equal to the absolute value of the coefficients. The objective function to minimize is:

$$min_w \frac{1}{2n_{samples}} \|Xw - y\|_2^2 + \alpha \|w\|_1 \tag{36}$$

where X is the frequency of the candidate word in a review, y is the star ratings of the review, $\|w\|_1$ is the L1-norm of the weight vector , α is a constant.

7.6.2 Evaluation

We use cross-validation to evaluate Lasso. The cross-validated Mean Squared Error（MSE） of Lasso fit is showed in Figure 10.

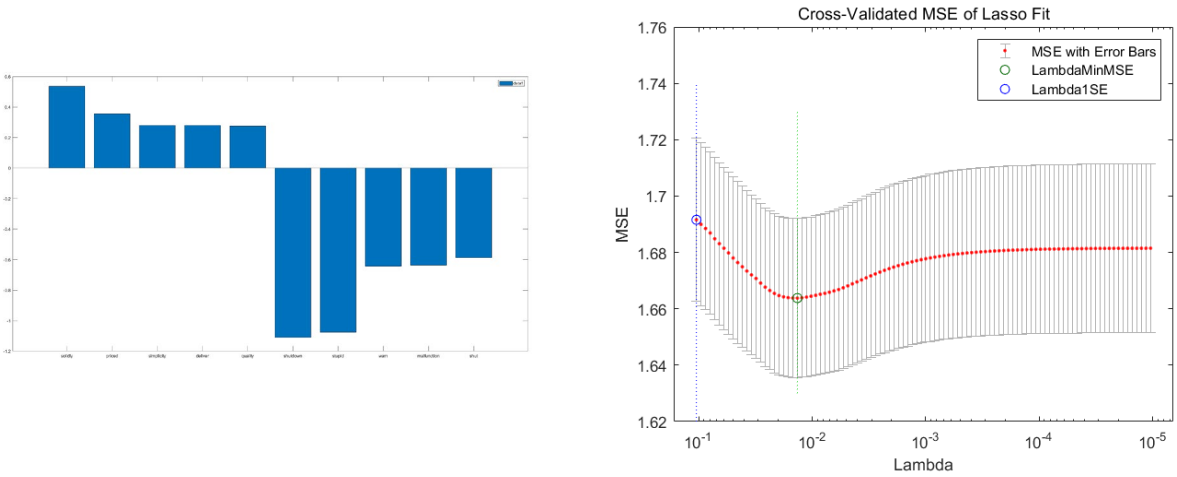

Figure 10. evaluation of Lasso Fit

Using the data in the microwave oven dataset for training, applied to the alpha (36) to cross validation, looking for can obtain minimum mean square error (green line) the alpha value of the trained model . As it can be found in the figure 10, 'solid', 'priced' words have higher positive weights, which means their have a great positive effect on stars ;The words stupid, shutdown, etc., are often used to indicate a low star rating

References

[1] Duan W, Gu B, W hinston A B. The Dynamics of Online Word-of-mouth and Product Sales-An Empirical Investigation of the Movie Industry[J]. Journal of Retailing, 2008, 84(2):233-242.

[2] Ramos, Juan Enrique. "Using TF-IDF to Determine Word Relevance in Document Queries." (2003).

[3] Rose, Stuart J., Engel, David W., Cramer, Nicholas O., & Cowley, Wendy E. Automatic Keyword Extraction from Individual Documents. United States.

[4] Weston, J., S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. "Feature Selection for SVMs." Advances in Neural Information Processing

Systems 13 (NIPS 2000).
[5] Jincheng Fan, Changlin Mei. Data Analysis. Science Press.
[6] aitao Zhang, uiran Zhang, Ping Wei, uizi Yin. Evaluation model of Wechat public accounts influence[J].Library and information service.2019,63(4).

Team #

Dear Sir/Madam

We are honored to inform you the analysis and results of our models.

Please note that the analysis is based on the three products that your company is about to sell online, i.e. baby pacifier, microwave oven and hair dryer. By analyzing the data of past online reviews, we obtained good design features of each product from the analysis of product titles, so that we are able to advise you on product design features. What's more, by extracting and analyzing words with strong sentiment from the reviews, we could provide you with ideas on product design directions.

First, the advice on baby pacifier's design and sales are as follows. By analyzing title of pacifiers which come from different brands, we find that:

- Stores like wubbanub and pilliphs prefer to add their brand name to the product's title, which indicates that their reputation is pretty good among customers. Therefore, the advice is that you could take their design features as references.
- Stores on Amazon prefer to add 'manufacture' and 'natural' to the title, which indicates that being natural and harmless might be a selling point.

Then, by extracting keywords of reviews, we find that customers are concerned about discolorration and more fond of the functionality of pacifier. We suggest that you should notice this.

Then, as for microwave oven, we find stores are likely to stress quality and add words that has positive suggestion, e.g. excellent, to the title. What's more, customers usually leave words like, versatility, durable, afford, overcooked, in the review. These phenomena suggest that :

- customers prefer microwave ovens with good functionality
- most of customers are sensitive to price
- A large quantity of customers don't know how to use it properly

Therefore, we suggest that you should take functionality seriously and consider cost performance. Last but not least, you could consider design an oven with intelligence or promote the proper ways of using the oven to avoid some bad reviews.

As for hair dryer, we find that stores tend to use word like quality, dry, lightweight, to promote the product, and customers are concerned about overheating and convenience. So, you should try to solve the problem of overheating.

Advices provided above are about design features and sales strategy of three products repspectively. Now, we introduce how to evaluate your

products.

First, you should pay attention to the four measures we identified in the profile. They could be used to evaluate sales situation based on reviews so that your company would be able to adjust your sales plans in time.

Then, to evaluate a product's potential, you could consider these measures: monthly average growth ratio of the first six months, DSR, keywords of the review. igher ratio and DSR, specific positive keywords frequently appearing and etc. indicates the potential of the product to be successful.

We find that ratings are strongly related with words like, excellent, solid, dispoint,stupid. Therefore, if you find that these negative words appear in the reviews frequently, you should adjust the sales strategy and service strategy.

We hope that the advices mentioned above will provide you with some useful information.

sincerely
Team #

Appendix

- Data Processing

```
%%table_read must run before this
%%creat word cloud
%analysis title review

% text_title_r=table2cell(file(:,14));
% text=string(text_title_r);

% analysis body

text_body_r=table2cell(file(:,15));
text=string(text_body_r);

%analysis title product

% text_title_p=table2cell(file(:,7));
% text=string(text_title_p);

text=tokenizedDocument(text);
bagUncleaned = bagOfWords(text);
%%data pre_process
clean_text=pre_process(text);
%%creat bag of words
bagCleaned=create_bagOfWords(clean_text);
%%get reduction
num_words_clean=bagCleaned.NumWords;
num_Words_Raw = bagUncleaned.NumWords;
% reduction=1-num_words_clean/num_Words_Raw;
% figure
% subplot(1,2,1)
% wordcloud(bagUncleaned);
% title("Oringin Data")
% subplot(1,2,2)
% wordcloud(bagCleaned);
% title("Cleaned Data")
```

```
% reduction;
% % %N-grammar way
% bag=bagOfNgrams(clean_text,'NgramLengths',3);
% % figure
% % wordcloud(bag);



function cleanedtext=pre_process(text)
%Create Tokenized Documents
cleanedtext=tokenizedDocument(text);
%add part of speech details
cleanedtext = addPartOfSpeechDetails(cleanedtext);
%removeStopWords
cleanedtext = removeStopWords(cleanedtext);
%Lemmatize the words using normalizeWords.
cleanedtext = normalizeWords(cleanedtext,'Style','lemma');
%Erase the punctuation from the file
cleanedtext = erasePunctuation(cleanedtext);
%Remove words with 2 or fewer characters
cleanedtext =  removeShortWords(cleanedtext,2);
end
function bag=create_bagOfWords(text)
%Create a bag-of-words model.
bag=bagOfWords(text);
%Remove words that do not appear more than two times in the bag-of-words model.
bag=removeInfrequentWords(bag,2);
%%Remove empty documents from the bag-of-words model and the corresponding labels
from labels.
bag = removeEmptyDocuments(bag);


end
```

- TF-IDF Analysis

```
%%selected feature number
Num=100;


TF_Matrix_b=tfidf(bagCleaned,clean_text);
```

```
% TF_Matrix_N=tfidf(bag,clean_text);
[index_f,index_l]=sort_way(TF_Matrix_b,Num);
bag_tfidf=removeWords(bagCleaned,index_l);

% [index_f,index_l]=sort_way(TF_Matrix_N,Num);
% bag_tfidfN=removeNgrams(bag,index_l);
%  figure
% % wordcloud(bag_tfidf);
% subplot(1,4,1);
% wordcloud(bag_tfidf);
% title("products heads")

subplot(1,4,2);
wordcloud(bag_tfidf);
title("reviews heads")

figure
wordcloud(bag);
function [index_f,index_l]=sort_way(Matrix,Num)
[M,N]=size(Matrix);
S=sum(Matrix);
[sorted,index]=sort(S,'descend');
index_f=index(1:Num);
index_l=index(101:N);
end
```

- Embedding of feature Selection

```
Xpw=full(Bagposi.Counts);
Xnw=full(Bagnage.Counts);

X=[Xpw,Xnw];
Y=file.star_rating;
[B,FitInfo] = lasso(X,Y,'CV',10);
% lassoPlot(B,FitInfo,'PlotType','CV');

values_em=B(:,(FitInfo.IndexMinMSE));
[~,index_em_1]=sort(values_em*(-1),'descend');
index_em_1=index_em_1(1:1:5);
```

```
[sorted,index_em_2]=sort(values_em,'descend');
index_em_2=index_em_2(1:1:5);
index_em=[index_em_2 ;index_em_1];

for i=1:5
    if (index_em_1(i))>100
        index_em_1(i)=index_em_1(i)-100;
    end

end
for i=1:5
    if (index_em_2(i))>100
        index_em_2(i)=index_em_2(i)-100;
    end

end
text1=Bagnage.Vocabulary(index_em_1);

text2=Bagposi.Vocabulary(index_em_2);
textt=[text2,text1];
CELL=cellstr(textt);
figure

bar(values_em(index_em),'FaceColor','flat');
xticks([1 2 3 4 5 6 7 8 9 10])
xticklabels(CELL)
legend('show') % Show legend
```

- Sentiment Classifier

```
emb=fastTextWordEmbedding;
data=readLexicon;

idx = ~isVocabularyWord(emb,data.Word);
data(idx,:)=[];
%Set aside 10% of the words at random for testing.
numWords = size(data,1);
cvp = cvpartition(numWords,'HoldOut',0.1);
dataTrain = data(training(cvp),:);
```

```
dataTest = data(test(cvp),:);

% Convert the words in the training data to word vectors using word2vec.
wordsTrain = dataTrain.Word;
XTrain = word2vec(emb,wordsTrain);
YTrain = dataTrain.Label;

%Train Sentiment Classifier
mdl = fitcsvm(XTrain,YTrain);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 测试 svm 段代码
% % wordsTest = dataTest.Word;
% % XTest = word2vec(emb,wordsTest);
% % YTest = dataTest.Label;
% % [YPred,scores] = predict(mdl,XTest);
% % figure
% % confusionchart(YTest,YPred);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Remove the words from the documents that do not appear in the word embedding emb.

idx = ~isVocabularyWord(emb,clean_text.Vocabulary);
clean_text = removeWords(clean_text,idx);
words=clean_text.Vocabulary;
clean_text=removeWords(clean_text,ismember(clean_text.Vocabulary,wordsTrain));
words=clean_text.Vocabulary;
vec = word2vec(emb,words);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%词嵌入结果可视化
% XY = tsne(vec);
% figure
% textscatter(XY,words);
% title(" Embedding Plot（hairdryer review body）")

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[YPred,scores] = predict(mdl,vec);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% % figure
% % % subplot(1,4,3)
% % idx2 = YPred == "Positive";
% % wordcloud(words(idx2),scores(idx2,1));
% % title("Predicted Positive Sentiment")
%
% % subplot(1,4,4)
% % wordcloud(words(~idx2),scores(~idx2,2));
% % title("Predicted Negative Sentiment")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% % sentimentScore=zeros(numel(clean_text),1);
% % for i = 1:numel(clean_text)
% %       words = string(clean_text(i));
% %       vec = word2vec(emb,words);
% %       [~,scores] = predict(mdl,vec);
% %       sentimentScore(i) = mean(scores(:,1));
% % end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%sort key_words
indx_po = YPred == "Positive";
lenscore=length(scores);
indexN=(1:1:lenscore)';
score_posi=[scores(indx_po,1), indexN(indx_po)];
score_neg=[scores(~indx_po,2),indexN(~indx_po)];

score_posi=[sortrows(score_posi,1,'descend');score_neg];
score_neg=[sortrows(score_neg,1,'descend');score_posi];

indexSp=score_posi(101:lenscore,2);
indexSn=score_neg(101:lenscore,2);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%extract key words
positveword=words(indexSp);
negeword=words(indexSn);

Bagposi=bagOfWords(removeWords(clean_text,indexSp));
Bagnage=bagOfWords(removeWords(clean_text,indexSn));




function data = readLexicon

% Read positive words
fidPositive = fopen(fullfile('opinion-lexicon-English','positive-words.txt'));
C = textscan(fidPositive,'%s','CommentStyle',';');
wordsPositive = string(C{1});

% Read negative words
fidNegative = fopen(fullfile('opinion-lexicon-English','negative-words.txt'));
C = textscan(fidNegative,'%s','CommentStyle',';');
wordsNegative = string(C{1});
fclose all;

% Create table of labeled words
words = [wordsPositive;wordsNegative];
labels = categorical(nan(numel(words),1));
labels(1:numel(wordsPositive)) = "Positive";
labels(numel(wordsPositive)+1:end) = "Negative";

data = table(words,labels,'VariableNames',{'Word','Label'});

end
```

- LDA Analysis

```
%  bag=bagOfNgrams(clean_text);
% figure
% wordcloud(bag);
```

```
numTopics = 7;
% mdl = fitlda(bag,numTopics,'Verbose',0);
mdl=fitlda(bagCleaned,numTopics,'Verbose',0);
figure;
for topicIdx = 1:4
    subplot(2,2,topicIdx)
    wordcloud(mdl,topicIdx);
    title("Topic " + topicIdx)
end
```