Problem Chosen	2020 MCM/ICM Summary Sheet	Team Control Number
C		2007971

Summary

With the fast development of online marketplace, it is essential for companies to hear the comments from customers and respond timely. Sunshine Company plans to use ratings and reviews of other competing products to analyze sales strategy and design features.

In the text mining, to begin with, we preprocess the given data in two stages - noise removal and normalization. Then we use the n-gram model to find most frequently-used words, 2-word and 3-word phrases in high-star reviews, low-star reviews and high-quality reviews respectively. In this way we can get the key features that customers care the most, and the quality descriptors closely related to rating levels. Thus, we come to these conclusions:

(1) For hair dryers, some features need to be considered such as retractable cords and light weight;

(2) For microwave ovens, additional features include stainless steel interior and an instruction & cooking guide.

(3) For pacifiers, they need to be easy to use, easy to find, easy to clean and easy to carry along with diaper bags.

(4) Some descriptors are linked to high-star reviews: wonderful, excellent, awesome, perfect, etc. , while some are linked to low-star reviews: garbage, horrible, junk, awful...

Time-based measures are needed to judge whether specific star ratings incite more reviews. If we calculate the relation between the increased frequency of *k*-star reviews and time, then we will see there are no obvious occasions when specific star rating attracts reviews.

To predict a potentially successful or failing product quantitatively, we introduce the dual linear regressive equation. Since the number of total reviews (TR) can be roughly assumed to be affected by 2 variables - the number of 5-star ratings (FS) and the total of helpful votes (HP), the relation is as follows:

$$TR = a + b \cdot FS + c \cdot HP + d \cdot FS \cdot HP + e \cdot FS^2 + f \cdot HP^2$$

Use TR as a symbol of success. Combine this method with text-based measures with the help of key features and quality descriptors, it can indicate a product's future.

Keywords: Text Mining; Natural Language Processing; N-gram; Correlation Coefficient

Contents

1 Introduction

1.1 Restatement of the Problem

With the fast development of online marketplace, it is essential for companies to hear the comments from customers and respond in a timely manner. To let customers freely deliver their opinions, Amazon has provided three channels:

- Star ratings: Customers can express their level of satisfaction using a scale of 1 to 5;
- Reviews: Customers can submit text-based messages that express further information;
- Helpfulness rating: Customers can submit ratings on the reviews as being helpful or not.

Companies use these data to gain insights into the markets in which they participate , the timing of that participation and the potential success of product design feature choices.

Sunshine Company is planning to introduce and sell three new products in the online marketplace: a microwave oven, a baby pacifier and a hair dryer. The company has collected large quantities of data. These data represent customer-supplied ratings and reviews for other competing products. Hence, Sunshine Company attempts to use these data to 1) inform sales strategy and 2) identify potentially important features that would enhance product desirability.

Particularly, the company's Marketing Director has following questions and requests:

a. Based on ratings and reviews, identify data measures that can help the company track the reputation and the focus of customers' feedback, once the three products are on sale.

b. Identify time-based measures and patterns within each data set that might suggest that a product's reputation is increasing or decreasing.

c. Combine text-based measures and rating-based measures to indicate a potentially successful or failing product.

d. Do specific star ratings trigger more reviews?

e. Are specific quality descriptors of text-based reviews strongly associated with rating levels?

1.2 Literature Review

Conventional credit scoring models are based on individual classifiers or a simple combination of those classifiers which tend to show moderate performance.[1] In recent years, how to quickly and accurately extract the subjective tendency of consumers from a mass of evaluation text has become a research hotspot and contributed to new kinds of scoring system. Some researchers combine TF-IDF (Term Frequency-Inverse Document Frequency) with SVM (Support Vector Machine) algorithm to develop a scoring system based on reviews.[2] Besides, the N-gram model is one of the most commonly used language models in natural language processing[3], so scientists often carry out text mining with N-gram variables and upgrade the model in the process.[4,5]

As the Internet provides everyone with opportunities to leave comments, some difficulties arise. For instance, we sometimes see that the sellers or buyers in bad faith make maliciously unfair score for the goods. Worse still, this kind of users unite and become organizations to disrupt the normal rating system.[6] Besides, the inconsistency between review content and score could bring troubles to the process of text mining. So some scientists have devised algorithms to detect the irregular reviews in time through simulation and verification.[7]

In this paper, our analysis can be divided into 3 parts:

(1) Analyze the headline and body of the reviews and find some useful words or phrases in 5-star reviews, 1-star reviews and reviews of high quality.

(2) Analyze time-based patterns in the "ratings" data and discover whether they interact in ways that will help the company craft successful products.

(3) Combine the (1) and (2) measures to predict a potentially successful or failing product and eventually solve the questions proposed by the Marketing Director.

1.3 Assumptions

Our model is based on the following assumptions:

(1) There is no invalid or unfair reviews in "high-quality reviews".

In our model, we define "high-quality reviews" as the reviews that satisfy

$$vine = "Y" \cup \left(total\_votes > 3 \cap \left(helpful\_votes / total\_votes \geq 50\%\right)\right).$$

Note that we redefine 0/0 = 0 in this formula.

When we browse through the data, we find that there might be inconsistency between review content and star rating. Besides, some customers leave completely the same review over and over again, especially low-star reviews. However, if we select the reviews using the condition formula above (all the reviews that were left by Amazon Vine Voices or a lot of people find helpful), the quality of reviews has improved greatly.

(2) The purchasers who write reviews are a representative sample of all purchasers.

Nowadays, customers can leave their star ratings without necessarily writing text-based messages. However, the star ratings accompanied with text are more convincing.

(3) According to the actual Amazon website, listed on the website are

- overall ratings based on Amazon's exclusive machine learning;
- the percentage of reviews with different numbers of stars;
- the keywords that Amazon has automatically selected in the reviews;
- the content of reviews ranked in "Top Reviews" and "Most Recent".

The website can only show one product's information at a time. The company can use these information to gain insights into the products they are selling.

(4) Indicated by the **Question d**, the customers' reviews might be incited by a series of high or low star ratings listed on the **"Most Recent"** ranking. We do not consider the "Top Reviews" ranking in this part.

(5) We do not consider the cause of price and specifications in our model because the data sets do not cover that information, but we can draw conclusions related to these using analysis of the model.

2 The Model

2.1 Notations

Listed in the Table 1 are the notations that will be used.

Table 1 Notations

Notation	Meaning
$w_i (i = 1,2,...,n)$	A single word

S	A sentence or phrase: $S=(w_1, w_2, ..., w_n)$
$p(S)$	Probability of appearing the sentence or phrase S
$p(w_n)$	Probability of appearing the word w_n
$p(w_n \mid w_{n-1})$	Probability of appearing the word w_n under the condition that w_{n-1} has appeared
$p(w_n \mid w_{n-1}w_{n-2})$	Probability of appearing the word w_n under the condition that $w_{n-1}w_{n-2}$ has appeared
$C(w_i)$	The count of the word w_i
$C(S)$	The count of the sentence or phrase S
C	The count of all reviews (of one type of product)
$C_k(w)$, k=1,...5	The count of all k-star reviews that contain word w
ρ	Pearson Correlation Coefficient
$F_k(year)$	The appearance frequency of k-star reviews in $year$
$C_k(year)$	The count of k-star reviews in $year$ (only appears in Section 2.3)
$I_k(year)$	The increased frequency of k-star reviews in $year$

2.2 Text Mining

We are interested in finding the most common words and phrases in **high-star reviews, low-star reviews and high-quality reviews** to

- check whether there are some quality descriptors strongly associated with rating levels;
- find the features that customers most care about and discuss about.

We define high-star reviews as 5-star reviews. We define low-star reviews as 1-star reviews and 2-star reviews.

We use Python in this process. To start with, we load the data sets and select the high-star reviews, low-star reviews and high-quality reviews respectively in every data set.

2.2.1 Preprocessing

Before we go into the text mining, we should clean the data. We divide the preprocessing period into two stages:

(1) Noise Removal

- Of all data labels, we only extract the text in **"review_headline" and "review_body"**.
- Remove all the records that **do not contain the string "hair dyer", "microwave" and "pacifier" in the "product_title" label.** Since we find shampoo reviews in "hair_dryer.tsv" and toy reviews in "pacifier.tsv", we think it necessary to delete the records that are irrelevant to the given 3 products.
- **Remover redundant text components** such as punctuation, tags, URLs, special digits or characters and "stopwords". The "nltk" package in Python provides a list of "stopwords",

which contain prepositions, pronouns and conjunctions. We need to delete these "stopwords" because we prefer informative nouns, adjectives and adverbs.

(2) Normalization

- Convert the upper case to lower case.
- Convert the plural nouns to singular.
- **Stemming and Lemmatization:** The goal of both methods is to reduce inflectional forms and sometimes derivationally related forms of a word to a base form with the help of "PorterStemmer" and "WordNetLemmatizer" in Python.

Through the two stages, we have cleaned the data. To give a straightforward impression, we use the "wordcloud" in Python to generate the word cloud map. The more frequency a word appears, the bigger its font is. [8]

Take the word cloud of hair dryer's **high-star reviews** and **low-star reviews** for example.

Figure 1 Word cloud for hair dryer's high-star reviews

Figure 2 Word cloud for hair dryer's low-star reviews

From the single word, we cannot draw the conclusion of words most frequently used in good or bad reviews. We need to analyze bi-gram or tri-gram to get further information.

2.2.2 N-gram Model

When we count the frequency of uni-grams(words), we use the "bag of words" model. The sentence is a cluster of words, so the appearance of every word is independent of other words, that is, a word's context will not affect the appearance of it.

However, when it comes to analyzing the occurrence of bi-grams and tri-grams, we introduce the Markov Assumption[9]: The appearance of a word is only associated with several words before it, that is ,

$$p(S) = p(w_1 \cdots w_n) = \prod p(w_i \mid w_{i-1} \cdots w_1) \approx \prod p(w_i \mid w_{i-1} \cdots w_{i-N+1}).$$

If the appearance of a word only depends on the word before it, then these 2 words are called **bi-gram,** that is,

$$p(S) = p(w_1 w_2 \cdots w_n) = p(w_1)p(w_2 \mid w_1)\cdots p(w_n \mid w_{n-1}).$$

If the appearance of a word only depends on the 2 words before it, then these 3 words are called **tri-gram,** that is,

$$p(S) = p(w_1 w_2 \cdots w_n) = p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_2 w_1)\cdots p(w_n \mid w_{n-1} w_{n-2}).$$

The conditional probability can be calculated using Maximum Likelihood Estimation:

$$p(w_n \mid w_{n-1}) = \frac{C(w_{n-1} w_n)}{C(w_{n-1})},\ p(w_n \mid w_{n-1} w_{n-2}) = \frac{C(w_{n-2} w_{n-1} w_n)}{C(w_{n-2} w_{n-1})}.$$

We use CountVectorizer in Python to implement the model and get 20 most-frequently-used uni-grams, bi-grams and tri-grams in high-star reviews, low-star reviews and high-quality reviews respectively in every product.

Take the quality reviews for example. We want to know the features that customers most talk about and care about. Table 2 gives 20 most-frequently-used n-grams in high-quality reviews of 3 products (ranked from the highest frequency to the lowest). Also, we can draw bar charts of the appearance frequency of n-grams.

Table 2 20 most-frequently-used n-grams in high-quality reviews

Hair dryer (1220 high-quality reviews)			**Microwave Oven (436 high-quality reviews)**			**Pacifier (515 high-quality reviews)**		
Uni	Bi	Tri	Uni	Bi	Tri	Uni	Bi	Tri
hair	dry hair	**dry hair quickly**	one	customer service	pm rafael vazquez	baby	month old	**color may vary**
dryer	blow dryer	**cool shot button**	unit	**stainless steel**	ge profile spacemaker	one	**diaper bag**	month old daughter
one	**heat setting**	**dry hair faster**	time	convectio n oven	**oven owner instruction**	like	little one	**baby shower gift**
dry	drying hair	**three heat setting**	oven	**work great**	**owner instruction cooking**	love	**easy clean**	put back mouth
use	hair quickly	**dry hair fast**	use	pm michelle	**instruction cooking guide**	mouth	keep mouth	**last long time**
like	curly hair	get job done	year	**easy use**	profile spacemaker ii	month	baby mouth	could keep mouth
setting	cool shot	blow dryer ever	ge	look like	still going strong	great	first year	bought month old
great	hair dry	best blow dryer	like	stopped working	457 pm michelle	use	make sure	hawaii medical gumdrop
blow	thick hair	blow dry hair	door	**work well**	**work great look**	product	**baby like**	month old still
time	**retractable cord**	long thick hair	work	cook time	whirpool customer service	little	week old	love love love
get	long hair	blow drying hair	product	30 second	another ge appliance	time	great product	make sure wash

product	work well	dryer dry hair	great	year ago	456 pm michelle	get	**easy use**	help keep mouth
heat	highly recommend	first time used	service	old one	501 pm michelle	nipple	one piece	month old son
used	**shot button**	**worth every penny**	appliance	rafael vazquez	three year ago	daughter	shower gift	make sure check
good	old dryer	hair half time	good	year old	**stainless steel interior**	son	**daughter love**	orange newborn gumdrop
well	work great	time dry hair	cook	ge profile	18 month old	take	put mouth	one piece easy
work	air flow	dyer ever used	well	pm rafael	samsung counter top	paci	**baby love**	**worth every penny**
hot	get hot	heat setting two	problem	**look great**	30 second button	also	work well	replace frog turtle
year	make hair	thick curly hair	new	**counter top**	wide range food	really	baby shower	frog turtle recently
really	blow dry	love blow dryer	model	service call	press start button	first	**son love**	turtle recently 12

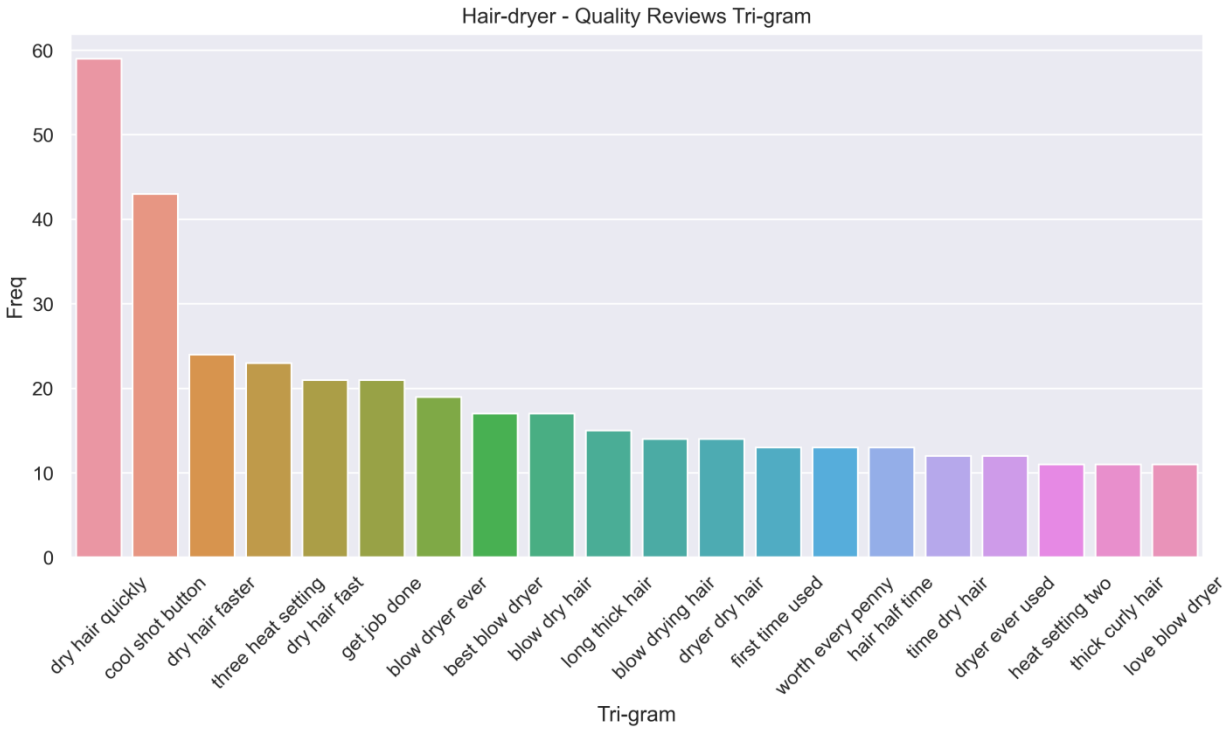

Figure 3 20 most-frequently used tri-grams in high-quality reviews of hair dryers

From the **bold keywords** in Table 2, we can draw preliminary conclusions about potentially important design features that would enhance product desirability:

For hair dryers:

(1) Basic functional needs: dry hair quickly, three heat settings;

(2) Features discussed about: cool shot button, retractable cord, worth every penny.

For microwave ovens:

(1) Basic functional needs: work well, easy to use;

(2) Features discussed about: stainless steel interior, look great, Microwave Oven Owner's

Instructions & Cooking Guide, (place at) counter top.

For pacifiers:

(1) Basic functional needs: baby shower gift, easy to use, keep it in mouth;

(2) Features discussed about: (easy to find or carry in) diaper bags, baby loves it, easy to clean, last for a long time, color may vary, worth every penny.

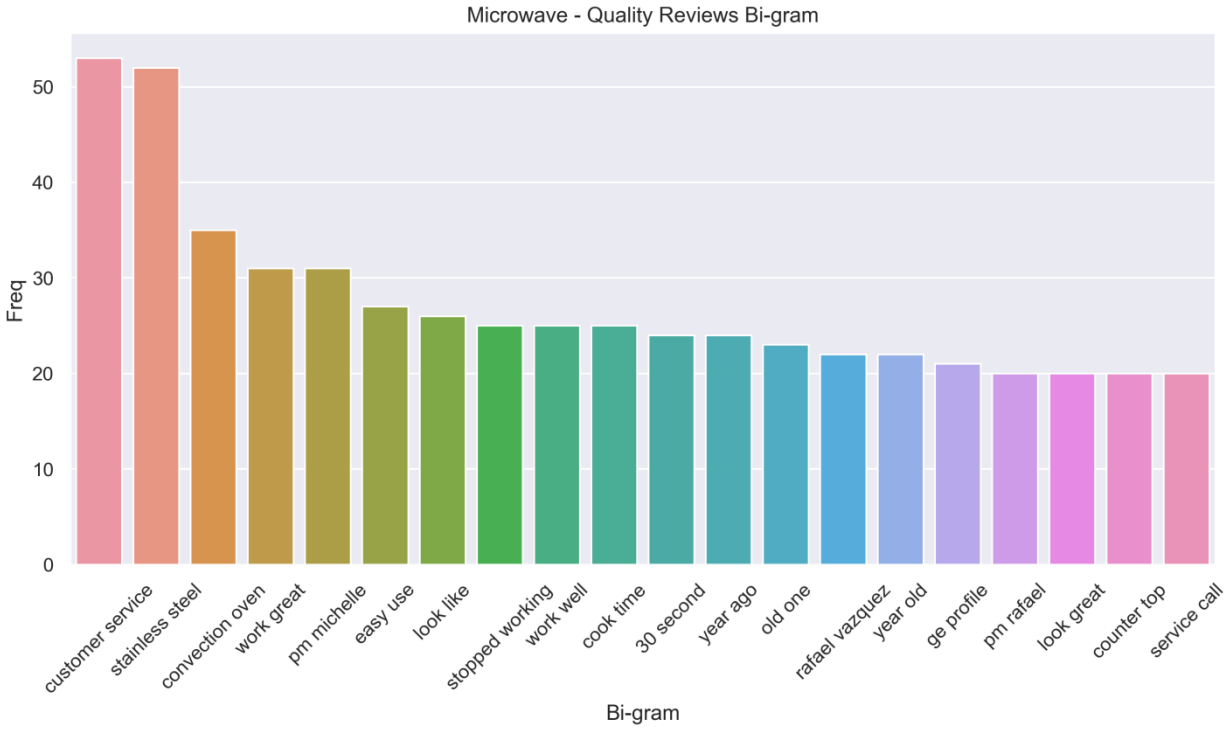

Figure 4 20 most-frequently used bi-grams in high-quality reviews of microwave ovens

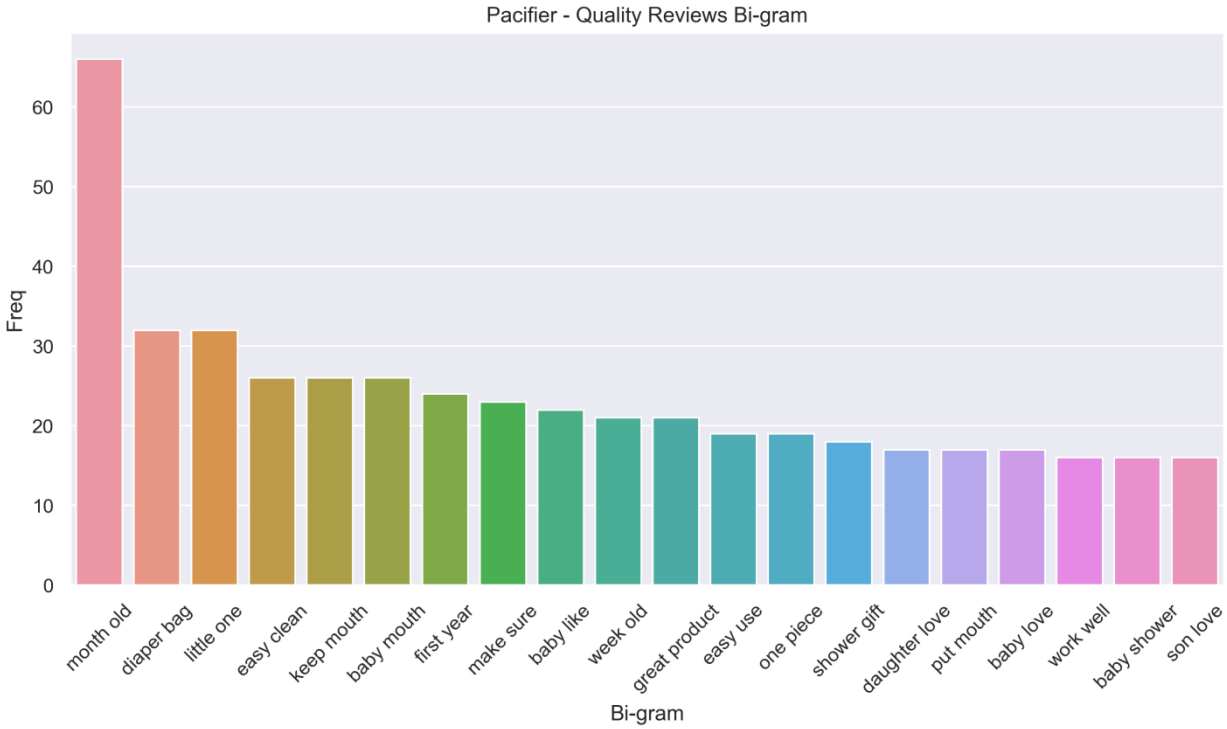

Figure 5 20 most-frequently-used bi-grams in high-quality reviews of pacifiers

Later we will add more features that customers discuss about in high-star reviews and low-star reviews.

2.2.3 Further Discussion

To know more about customers' demand, we need to refer to the most extreme reviews. We have got 20 most-frequently-used n-grams in high-star reviews and low-star reviews (Table 3 and Appendix A). We can get a better idea of what customers love and hate. Also, as indicated in **Question e,** there might be quality descriptors that are associated with rating levels.

Table 3 20 most-frequently-used n-grams in high-star reviews

Hair dryer (6509 high-star reviews)			**Microwave Oven (663 high-star reviews)**			**Pacifier (7812 high-star reviews)**		
Uni	Bi	Tri	Uni	Bi	Tri	Uni	Bi	Tri
hair	five star	dry hair quickly	great	five star	**old spacemaker ii**	love	five star	baby shower gift
dryer	dry hair	dry hair fast	one	work great	little counter space	baby	baby love	love love love
great	blow dryer	best blow dryer	work	work well	**limited counter space**	one	month old	**put back mouth**
one	work great	get job done	use	easy use	**go profile spacemaker**	great	son love	month old love
love	great product	blow dry hair	love	counter space	**30 second button**	star	daughter love	little one love
dry	work well	love blow dryer	space	look great	love love love	five	great product	month old son
work	hair quickly	cut drying time	small	great product	**still going strong**	like	baby shower	much easier find
product	highly recommend	blow dryer ever	fit	small kitchen	look great work	cute	little one	**find middle night**
good	great dryer	love love love	star	dinner plate	**profile spacemaker ii**	son	**easy find**	great baby shower
use	**thick hair**	**long thick hair**	well	stainless steel	best ever owned	daughter	**stuffed animal**	**let air dry**
blow	retractable cord	best dryer ever	good	great price	**add 30 second**	mouth	easy clean	**love glow dark**
like	heat setting	great blow dryer	five	easy install	look work great	little	diaper bag	help keep mouth
star	**light weight**	hair half time	kitchen	**30 second**	**look brand new**	month	love love	daughter love wubbanub
five	hair dry	cool shot button	easy	fit perfectly	size dinner plate	product	**glow dark**	make easy find
time	great price	**love retractable cord**	look	old one	much counter space	easy	shower gift	easy find night
well	**curly hair**	worth every penny	perfect	easy clean	press start button	get	love wubbanub	**throw washing machine**
used	best ever	dry hair faster	oven	convection oven	**work like new**	use	**super cute**	ca live without
year	**long hair**	best ever owned	product	perfect size	using speed cook	find	back mouth	leave home without

setting	drying hair	dry hair half	time	spacemaker ii	work great look	best	keep mouth	best thing ever
really	love dryer	drying time half	like	look like	great value money	good	highly recommend	worth every penny

From the **bold keywords** in Table 3 and Appendix A, we can add more details about important design features to the preliminary conclusions in Section 2.2.2:

For hair dryers:

(1) The basic need is to dry hair quickly with three heat settings;

(2) Retractable cord is controversial. Some love it and some think it goes wrong easily;

(3) Light weight is preferred;

(4) Suitable for many kinds of hair, especially long, thick hair;

(5) Avoid catching fire.

For microwave ovens:

(1) The basic need is to work well, easy to use and to be placed in limited counter places;

(2) Additional features include stainless steel interior, great look, an instruction & cooking guide for microwave oven;

(3) Customer service should be timely and effective.

For pacifiers:

(1) Always used as baby shower gifts;

(2) The basic need is for babies to keep it in mouth;

(3) Easy to use, easy to find, easy to clean, easy to carry along with diaper bags;

(4) Additional features include that baby loves it, last for a long time, color may vary (and the customer can choose the color), super cute, stuffed animals.

2.3 Time-based Ratings

For time-based models, we define $F_k(year)$ as the appearance frequency of *k*-star reviews in this *year* and define $C_k(year)$ as the count of *k*-star reviews in this *year*. It is obvious that

$$F_k(year) = \frac{C_k(year)}{\sum_{k=1}^{5}(year)}.$$

Now we define the increased frequency of *k*-star reviews in this *year*:

$$I_k(year) = F_k(year) - F_k(year-1).$$

If we want to determine whether there is association between the increased frequency and time, then use *year* as x-axis, use $I_k(year)$ as y-axis and do linear fitting. The process will be extensively discussed in Section 3.4.

3 Analysis of Questions

3.1 Question a: Tracking the Products

For companies, once their products are on sale, it is essential to get as much information as they could to keep track of how the products are selling.

Based on the information given on the website, to get an overall impression of the product, there are 3 ways the company can do:

(1) **Use the overall star ratings shown on the Amazon website:** Amazon's calculation of star ratings is not simply an average. It is based on most recent reviews, high-quality reviews, verified purchase and other factors with deep learning involved. Therefore, it is a quick way to get recent updates about the product.

(2) **Use the distribution of star ratings shown on the Amazon website:** Usually there are 4 typical distributions: "F" distribution indicates successful products, "P" distribution indicates mediocre products with a little flaw, "B" distribution means unsuccessful products in need of improvements and "L" distribution means failing products with almost all bad reviews.

Figure 6 The "F", "P", "B", "L" distribution of star ratings

(3) **Use the keywords that Amazon automatically extracts from the reviews:** Compare the keywords with the features suggested in Section 2.2.3 and click on the keywords to find details. Respond to the reviews, especially low-star reviews, about key features timely.

Now we discuss the factors quantitatively that the company should track with an example:

Define word frequency as $\frac{\sum_{k=1}^{5} C_k(w)}{C}$.

For hair dryers, we randomly select 1000 reviews to simulate the current reviews on the website. Analyze the current reviews with the help of the Python codes in the Appendix, and we will get 20 uni-grams, bi-grams and tri-grams with largest word frequency. Ranked from high to low, we select 15 words or phrases in Table 5.

Table 5 Example of how to track the reviews

Word or phrase	Word Frequency	Meaning or features
love	0.212	emotional word
quick	0.088	dry fast / broken quickly
easy	0.075	easy to use/light weight
year	0.132	no particular meaning
month	0.071	no particular meaning
hot	0.147	no particular meaning
cold	0.025	no particular meaning
money	0.040	save money
stop	0.032	stop working
fast	0.091	dry fast
thick hair	0.032	kinds of hair
light weight	0.023	light weight

back	0.630	no particular meaning
working	0.704	stop working
lightweight	0.257	light weight

Categorize the 15 words or phrases using the features in Section 2.2.3. Then we can track what aspects of features the customers are discussing about. For example, in our randomly selected data set, the factors from most important to less important: dry fast (0.179), light weight (0.129), stop working (0.075), kinds of hair (0.032).

The basic purpose of tracking the product updates is to demonstrate the features that customers care about the most. Therefore, with the overall and specific measures listed above, the Sunshine Company can follow the comments in a timely manner.

3.2 Question b: Reputation Change

The easiest way to determine the reputation change is to track the overall star ratings on Amazon. Also, it is efficient to calculate the average star ratings every month or year to see the fluctuations.

We believe that the number of reviews is positively associated with the number of purchasers. The more reviews, the more selling. Take the pacifiers for example, rank the products in the data set based on the average number of reviews per year. The number of reviews in different years (different colors in Figure 7) display fluctuations. If the review number per year decreases, then it might indicate the reputation is dropping.

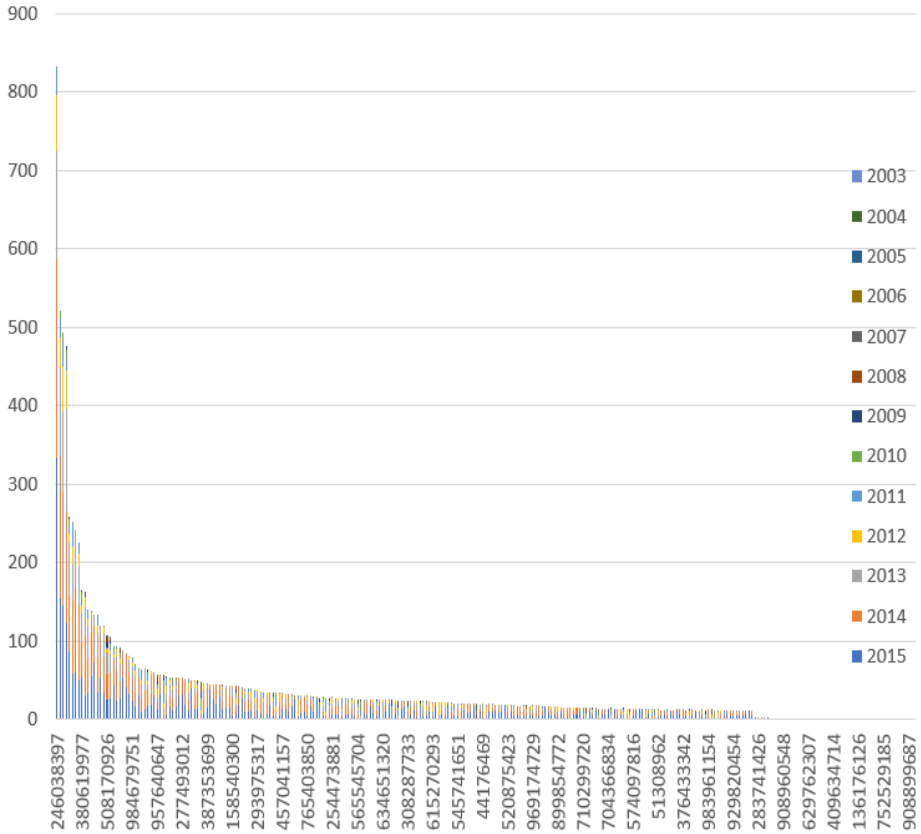

Figure 7 The composition of all reviews for every pacifier

As the reputation grows, the proportion of verified purchase increases. People are more willing to buy the product full-price. Take the pacifiers for example. The proportion kept a trend of increasing since 2006.

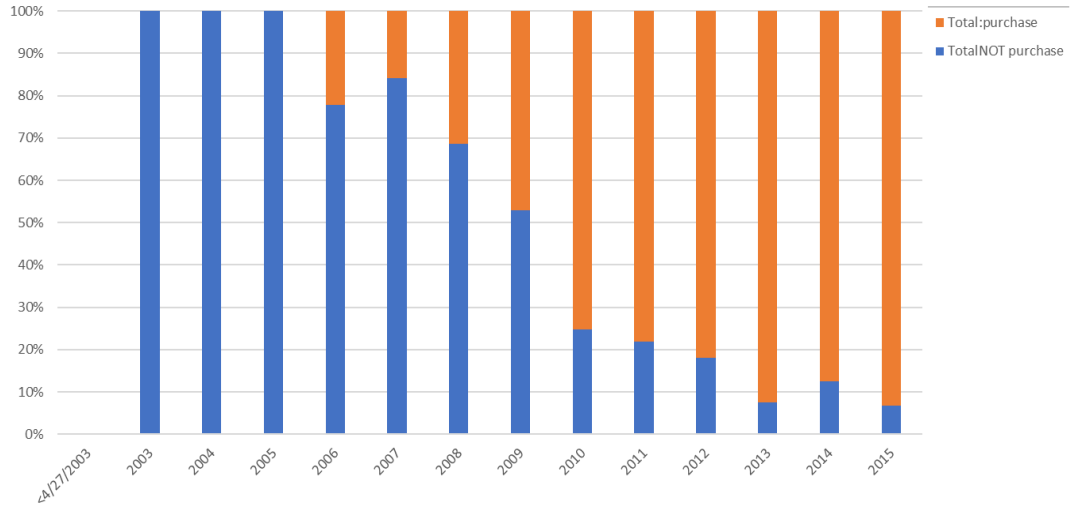

Figure 8 The proportion of verified purchase (orange part)

3.3 Question c: Successful or Failing?

Use quantitative measure to indicate a successful or failing product: With the help of dual linear regressive equation, use number of total review (TR) as a symbol of success. The variables are the number of five-star ratings (FS) and the total of helpful votes (HP), then fit the below equation:

$$TR = a + b \cdot FS + c \cdot HP + d \cdot FS \cdot HP + e \cdot FS^2 + f \cdot HP^2$$

Take all the pacifiers as an example, the results are as follows:

Table 6 The result of pacifiers' dual linear regressive equation

Multiple R	0.932498
R Square	0.869553
Adjusted R Square	0.867342
n	301
Significance F	3.7E-128
a	-1.62158
b	1.697466
c	0.036
d	-0.00094
e	0.005312
f	2.11E-05

Multiple R is very high, which means that there is strong relation among TR, FS and HP. This indicates that we can use this equation to predict the product's future.

Also, with text-based measures mentioned in Section 2.2, the special features and many quality descriptors which will be covered in Section 3.5 are useful in predicting whether the product is high-rating or low-rating.

3.4 Question d: Star Ratings Incite Reviews?

As we have explained in the assumptions, we only consider the situation where the reviews are triggered by the most recent reviews.

From every data set, we select an item that has a lot of reviews, and use recent 5-year selling history to do linear fitting in Section 2.3.

For pacifiers, we select a total of 183 pacifier products with the most reveiews:

Table 7 Linear fitting pacifier

Star(Pacifier)	1	2	3	4	5
Average	-0.00892	-0.00404	-0.00259	-0.00227	0.017817
Standard Deviation	0.047788	0.023874	0.034007	0.057124	0.072306
Correlation Coefficient	0.186729	0.169115	0.076136	0.039679	0.246405

For microwave ovens, we select a total of 49 microwave products with the most reviews:

Table 8 Linear fitting of microwave

Star(Microwave)	1	2	3	4	5
Average	-0.02510527	-0.0053	0.0144628	0.0127032	0.0032387
Standard Deviation	0.13066806	0.0719612	0.0412358	0.0903924	0.1651319
Correlation Coefficient	0.1921301	0.0736439	0.3507336	0.1405342	0.0196131

For hair dryers, we select a total of 231 hair dryer products with the most reviewes:

Table 9 Linear fitting of hair dryer

Star(Hair dryer)	1	2	3	4	5
Average	0.00685	0.009451	0.007668	0.020812	-0.04478
Standard Deviation	0.039877	0.021824	0.031053	0.050842	0.063476
Correlation Coefficient	0.17178	0.433044	0.246924	0.409354	0.705476

As we can see from the Table 7, 8, 9, the association between the year and increased frequency is greater in the high-star reviews and low-star reviews. Also, as we cannot calculate for all the products, we have to say this association is not stable. It may work for some products, but may not work for others.

Therefore, we draw the preliminary conclusion that there are not obvious occasions when specific star rating may attract more reviews.

3.5 Question e: Words linked with Ratings

There might be some quality descriptors w that are closely related with star ratings.

Let event A_k = "w appears in the review of k-star", event B = "w appears in the review". Of the 5 conditional probability $p(A_k \mid B)$ (k =1,...5), the highest probability suggests that if a review contains the word w, then it is most likely to be a k-star review.

Due to $p(A_k \mid B) = p(A_k B)/p(B)$ and $p(B)$ is constant in the above comparison, the conditional probability $p(A_k \mid B)$ depends on $p(A_k B)$, that is

$$p(A_k B) = p(B \mid A_k) \cdot p(A_k) = p(A_k) = \frac{C_k(w)}{C}.$$

The Section 2.1 has defined $C_k(w)$ as the count of all k-star reviews that contain word w and define C as the total number of reviews.

Since C is constant, we only need to compare $C_k(w)$. If $C_1(w)+C_2(w) > C_5(w)$, then the word w is more likely to appear in a low-star review. If not, then it is more likely to be in a high-star review. **This conclusion will be useful in our rankings of data.**

Define these 3 variables:

Table 10 Definition of X, Y, Z

	0	1
X	The review does not contain w.	The review contains w.
Y	The review is not a low-star review.	The review is a low-star review.
Z	The review is not a high-star review.	The review is a high-star review.

Now we introduce Pearson Correlation Coefficient ρ to evaluate the association between the words and the ratings:

$$\rho(X,Y) = \frac{E(XY) - E(X)E(Y)}{\sqrt{D(X)D(Y)}}.$$

Of all the words in the reviews of **hair dryers:**

Using the conclusion that we only need to focus on $C_k(w)$, we rank all the words with the help of $C_1(w)+C_2(w)-C_5(w)$. And we select the smallest 10 and the largest 10 words and check their $\rho(X,Y)$ and $\rho(X,Z)$.

Table 11 Hair dryers: the words likely to be linked with high/low-star reviews

The Smallest 10 words (Likely to be linked with high-star)	$\rho(X,Y)$	$\rho(X,Z)$	The largest 10 words (Likely to be linked with low-star)	$\rho(X,Y)$	$\rho(X,Z)$
great	-4E-05	1.8E-05	return	6.3E-05	-0.0001
hair	-8E-06	3.7E-06	stopped	5.9E-05	-0.0001
dryer	4E-06	-3E-06	month	4.9E-05	-6E-05
love	-1E-04	2.2E-05	waste	6.9E-05	-0.0002
dry	-1E-05	4.8E-06	disappointed	5E-05	-9E-05
work	-9E-06	2.9E-06	fire	6.4E-05	-0.0002
one	1.1E-05	-6E-06	spark	6.3E-05	-0.0002
star	-2E-05	6.9E-06	junk	7.2E-05	-0.0002
good	-2E-05	5.2E-06	returned	5.3E-05	-9E-05
five	-0.0001	2.2E-05	burned	5.8E-05	-1E-04

Table 12 Hair dryers: the words with the largest or smallest $\rho(X,Y)$ and $\rho(X,Z)$

$\rho(X,Y)$	$\rho(X,Z)$

Largest 10 **(Likely to be linked with low-star)**	Smallest 10 **(Likely to be linked with high-star)**	Largest 10 **(Likely to be linked with high-star)**	Smallest 10 **(Likely to be linked with low-star)**
garbage	five	five	sudden
october	comfortable	love	junk
potentially	smoother	excellent	awful
warned	silky	efficient	sparked
humming	solid	lol	painful
scam	excellent	highly	sending
unreliable	efficient	bonus	waste
exchanged	love	great	began
recalled	quicker	perfect	july
az	awesome	met	meh

From the Table 11 and Table 12, we can see some quality descriptors of hair dryers:

(1) High-star reviews: great, love, good, perfect, bonus, excellent, efficient, comfortable, smoother, quicker, awesome;

(2) Low-star reviews: waste, disappointed, fire, spark, junk, burned, garbage, unreliable, awful, painful.

Of all the words in the reviews of **microwave ovens:**

Table 13 Microwave ovens: the words likely to be linked with high/low-star reviews

$\rho(X,Y)$		$\rho(X,Z)$		$C_1(w)+C_2(w)-C_5(w)$	
Largest 10 **(Likely to be linked with low-star)**	Smallest 10 **(Likely to be linked with high-star)**	Largest 10 **(Likely to be linked with high-star)**	Smallest 10 **(Likely to be linked with low-star)**	Largest 10 **(Likely to be linked with low-star)**	Smallest 10 **(Likely to be linked with high-star)**
lemon	perfect	perfect	charge	warranty	great
worst	grill	five	repairman	service	work
recall	smaller	saved	worse	buy	star
smoke	sleek	love	horrible	year	easy
garbage	powerful	whistle	junk	month	good
authorized	foot	kid	error	repair	well
melted	pizza	efficient	contact	another	five
march	solid	plenty	defective	customer	fit
recalled	love	great	dangerous	stopped	perfect
suck	easy	easy	response	working	love

Likewise, we can get the Table 13 and see some quality descriptors of microwave ovens:

(1) High-star reviews: perfect, smaller, grill, sleek, powerful, love, efficient, great, easy, good, well;

(2) Low-star reviews: worst, garbage, suck, horrible, junk, defective, dangerous.

Of all the words in the reviews of **pacifiers:**

Table 14 Pacifiers: the words likely to be linked with high/low-star reviews

$\rho(X,Y)$		$\rho(X,Z)$		$C_1(w)+C_2(w)-C_5(w)$	
Largest 10 **(Likely to be linked with low-star)**	Smallest 10 **(Likely to be linked with high-star)**	Largest 10 **(Likely to be linked with high-star)**	Smallest 10 **(Likely to be linked with low-star)**	Largest 10 **(Likely to be linked with low-star)**	Smallest 10 **(Likely to be linked with high-star)**
refund	five	five	returned	disappointed	love
advertising	wonderful	love	poorly	waste	baby
unsafe	handy	cutest	disappointing	poor	great
suffocated	grandson	washable	logo	disappointing	star
pouring	lifesaver	delivery	misleading	useless	five
suffocation	excellent	bounce	waste	return	one
jolly	soothing	lovey	trunk	returned	like
symbol	lose	penny	claim	opened	cute
pointless	awesome	quiet	fake	vary	little
roof	perfect	outfit	garbage	worst	daughter

Likewise, we can get the Table 14 and see some quality descriptors of pacifiers:

(1) High-star reviews: wonderful, handy, excellent, soothing, awesome, perfect, love, cute, washable, great;

(2) Low-star reviews: unsafe, pointless, poor, disappointing, misleading, waste, fake, garbage, disappointed, useless, worst.

4 Sensitivity Analysis

In the text-mining process, we use all the content in the high-star reviews or low-star reviews, including review titles and review bodies. However, what if we only select the high-quality reviews in these two kinds of reviews? Will the research results be the same?

Take the hair dryer for example. Compare the Figure 9 and the Table 3, there are 13 phrases in common out of the top 20. So the model is still stable even with some of the words unconsidered.

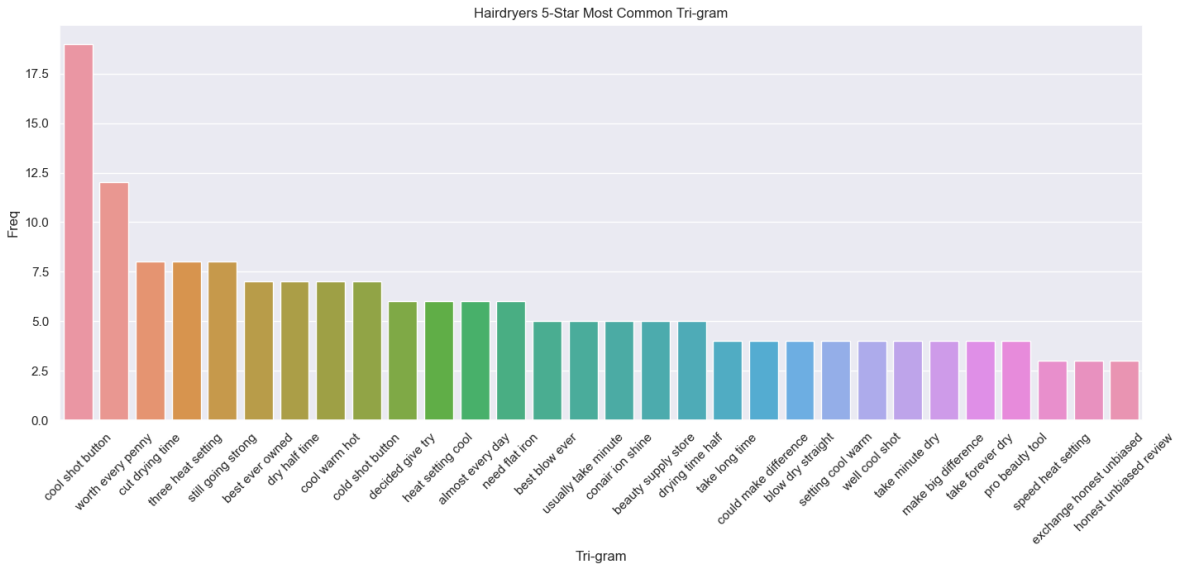

Figure 9 Most common tri-grams in the 5-star high-quality reviews

5 Conclusions

(1) Through **text mining,** we have got the most-frequently-used uni-grams, bi-grams and tri-grams in high-star reviews, low-star reviews and high-quality reviews respectively in every product.

(2) Using the keywords we extract, we find out some **important features** that customers talk about the most. Also we use 3 indicators to select potentially descriptors and finally find some **quality words** such as “garbage” and “excellent” that are strongly associated with low-star reviews and high-star reviews respectively.

(3) Based on the reality Amazon website, we have devised **a method of tracking the customers’ reviews** once the new products are on sale. Use the information on the website to get an overall knowledge and use the Python codes in Appendix B to extract keywords for market researching.

(4) Using **a time-based model,** we come to the conclusion that there are situations when high-star ratings bring about high-star ratings, low-star ratings bring about low-star ratings. However, it is uncertain that every product follows the same rule.

(5) We present **3 ways to evaluate the reputation change of a product:** the Amazon overall star ratings, the number of reviews per year and the proportion of verified purchases.

(6) We use **a dual linear regressive fitting** to indicate a potentially successful or failing product quantitatively. Then **combine this method with the key features from text mining** to predict.

6 The Letter

Dear Marketing Director of Sunshine Company,

Our research is coming to an end. Thank you for your confidence in our team.

In the first stage of our research, we preprocess your given data in two stages - noise removal and normalization. With the implementation of n-gram model, we finally get 20 most-frequently-used words and phrases in high-star reviews, low-star reviews and high-quality reviews respectively for 3 products. Later, considering your particular interest to time-based patterns, we devise a time-based model to find whether star ratings will affect reviews, whether the reputation is increasing or decreasing and to predict a potentially successful or failing product.

During the research process, we have come to conclusions regarding your questions:

First, the quality of products is the most fundamental. Without a well-designed product, any marketing strategy will be useless. We have collected some potentially important design features that would enhance product desirability:

For hair dryers:

(1)The basic need is to dry hair quickly with three heat settings;

(2)Retractable cord is controversial. Some love it and some think it goes wrong easily;

(3)Light weight is preferred;

(4)Suitable for many kinds of hair, especially long, thick hair;

(5)Avoid catching fire.

For microwave ovens:

(1)The basic need is to work well, easy to use and to be placed in limited counter places;

(2)Additional features include stainless steel interior, great look, an instruction & cooking guide for microwave oven;

(3)Customer service should be timely and effective.

For pacifiers:

(1)Always used as baby shower gifts;

(2)The basic need is for babies to keep it in mouth;

(3)Easy to use, easy to find, easy to clean, easy to carry along with diaper bags;

(4)Additional features include that baby loves it, last for a long time, color may vary (and the customer can choose the color), super cute, stuffed animals.

The features listed above can be used in both designing and promoting, which will stand out in the products of the same kind.

About your Question a, you can get an overall impression of how the product is selling by simply clicking on the Amazon website. The overall ratings, the distribution of stars and the keywords Amazon automatically selects for you can be of great help. To make further investigations, you can use the Python codes attached to this letter to analyze the content of reviews, categorize the keywords based on the feature listed above and know what improvements can be made and what features customers are discussing about.

For your Question b, the increase or decrease of a product over time can be tackled from

3 aspects: the overall rating tendency on Amazon website, the number of reviews per year ou per month and the proportion of verified purchases.

The Question c suggests a combination of text-based measures and rating-based measures. We also do this prediction both quantitatively and qualitatively. Since the number of total reviews rely on two variables - the number of five-star ratings and the total of helpful votes, we can use dual linear regressive equation to solve the associations among them. Also, with popular features and high-star or low-star descriptors, it is easy to predict the product's future from the words or phrases frequently used in its reviews.

For Question d, by analyzing the association between the increased frequency of *k*-star reviews in *year* and the time, we can see subtle linear relation through correlation coefficient. There are occasions when high-star ratings attract high-star ratings and low-star ratings attract low-star ratings. However, the rule does not apply to every product in the data set.

In Question e, of all the words in the reviews of one kind of product, we rank them with 3 different indicators:

- the correlation coefficient between a word's appearance and a low-star review;
- the correlation coefficient between a word's appearance and a high-star review;
- the difference between the frequency of a low-star review containing the word and a high-star review containing this word.

In this way, we find some quality descriptors closely related to high-star reviews and low-star reviews, to name a few:

(1) High-star reviews: wonderful, excellent, awesome, perfect, love, great...

(2) Low-star reviews: worst, garbage, horrible, junk, dangerous, awful...

In all, quality is the king. There might be some methods to manipulate the ratings, such as writing good reviews to attract possible good reviews. The company also needs to reply to customers' doubts or complaints timely. However, it is always better to make preparations sufficiently and develop a great design.

Our team hope our suggestions and conclusions can be helpful. Wish you success in your undertaking and wish Sunshine Company success in the launch of the three products!

Yours Sincerely,

Team # 2007971

References

[1] Tripathi D, Edla DR, Cheruku R, et al. Hybrid credit scoring model using neighborhood rough set and multi-layer ensemble classification. *Journal of Intelligent & Fuzzy Systems*, 2018, 34(3): 1543-1549.

[2] Jiacheng Zhang, Yanhui Zhu. SVM-based Commodity Scoring System. *Computer Knowledge and Technology (Academic Exchange)*, 2018, 10: 223-225.

[3] Chen Yin, Min Wu. Survey on N-gram Model. *Computer System & Applications*, 2018, 27(10): 33-38.

[4] Schonlau M, Guenther N, Sucholutsky I. Text mining with n-gram variables. *Stata Journal*, 2017, 17(4): 866-881.

[5] Xudong Wang, Jing Duan, et al. Improvement and Application of N-Gram Algorithm Based on Similar Duplicate Records. *Modern Computer*, 2018, 17: 78-82.

[6] Chaojun Liu. Group Attack Simulation and Detection in Rating Systems. *Microcomputer Applications*, 2015, 31(11): 69-71.

[7] Weijun Wang, Yanqiu Song, et al. A Correction Method for Score Deviation of Online Product Reviews Based on Discourse Markers Theory. *Journal of The China Society for Scientific and Technical Information*, 2016, 35(4): 358-368.

[8] Heimerl, F. , Lohmann, S. , Lange, S. , & Ertl, T. . Word cloud explorer: Text analytics based on word clouds. *Hawaii International Conference on System Sciences* IEEE,2014.

[9] Kalbfleisch J D , Lawless J F . The Analysis of Panel Data under a Markov Assumption. *Journal of the American Statistical Association*, 1985, 80(392): 863-871.

Appendices

Appendix A:20 most-frequently-used n-grams in low-star reviews

Hair dryer (1611 low-star reviews)			**Microwave Oven (512 low-star reviews)**			**Pacifier (1046 low-star reviews)**		
Uni	Bi	Tri	Uni	Bi	Tri	Uni	Bi	Tri
dryer	dry hair	get hot enough	year	**customer service**	pm rafael vazquez	baby	one star	color may vary
hair	blow dyer	stopped working month	one	stopped working	never buy another	one	baby like	say color may
one	stopped working	**cool shot button**	ge	**service call**	stopped working year	like	month old	month old son
month	waste money	first time used	service	year ago	stopped working month	mouth	two star	one month old
product	one star	blow cold air	product	never buy	buy new one	product	baby mouth	**hawaii medical gumdrop**
use	**retractable cord**	dryer stopped working	buy	new one	three year ago	son	waste money	stay baby mouth
time	get hot	**take long time**	time	pm michelle	**circuit board mounting**	month	may vary	10 month old

year	two star	blowing hot air	door	le year	**board mounting bracket**	star	keep mouth	baby didn't like
get	hot enough	trying dry hair	unit	stainless steel	stand behind product	use	color may	month old baby
hot	month ago	time dry hair	month	stay away	another ge appliance	ripple	**stuffed animal**	total waste money
like	new one	stopped blowing hot	samsung	buy another	ge profile spacemaker	get	**glow dark**	baby keep mouth
work	drying hair	dryer get hot	new	two year	457 pm michelle	color	diaper bag	baby like paci
blow	hot air	long time dry	problem	open door	se error code	really	look like	dr brown bottle
dry	first time	enough dry hair	**warranty**	worked fine	**customer service rep**	time	**poor quality**	baby never liked
bought	first one	dry hair quickly	use	rafael vazquez	**wish read review**	good	son like	put back mouth
buy	worked great	long dry hair	repair	year old	whirlpool customer service	two	03 month	**hard open close**
used	**caught fire**	month stop working	appliance	se error	le two year	bought	great idea	ca keep mouth
back	le year	take forever dry	work	control panel	bought year ago	take	little one	baby shower gift
cord	time used	blow hot air	get	pm rafael	buy piece junk	got	**hawaii medical**	hard time finding
working	wall mount	first one lasted	oven	one star	give zero star	daughter	baby take	hawaii medical one

Appendix B: Python Codes used in Text Mining

(1) DataParser.py:

```
import os
import csv
import html
from datetime import datetime

BASEDIR = os.path.abspath(os.path.dirname(__file__))


def read_tsv(tsv_filename="microwave.tsv"):
    ret = []
    with open(os.path.join(BASEDIR, tsv_filename), "r", encoding="utf-8") as
tsv:
        reader = csv.reader(tsv, dialect="excel-tab")
        for idx, line in enumerate(reader):
            ret.append(RecordWrap(line) if idx != 0 else line)
    return ret


def mdY2date(c: str):
    if '/' not in c:
```

```
        return c
    return datetime.strptime(c, "%m/%d/%Y")


class RecordWrap():

    def __init__(self, record_list, *args, **kwargs):
        super(RecordWrap, self).__init__(*args, **kwargs)
        self._data = record_list
        self._review_date = mdY2date(self._data[14])
        pass

    @property
    def star_rating(self):
        return self.__util_to_int__(self._data[7])

    @property
    def helpful_votes(self):
        return self.__util_to_int__(self._data[8])

    @property
    def total_votes(self):
        return self.__util_to_int__(self._data[9])

    @property
    def helpful_votes_rate(self):
        return self.helpful_votes / self.total_votes

    @property
    def is_vine(self):
        d = self._data[10].lower()
        if d == "n":
            return False
        elif d == "y":
            return True
        else:
            return d

    @property
    def review_headline(self):
        return self._data[12]

    @property
    def review_body(self):
        return text_normalize(self._data[13])

    @property
    def review_date(self):
        return self._review_date

    def __repr__(self):
        return repr(self._data)

    @staticmethod
    def __util_to_int__(data):
        try:
            return int(data)
        except:
            return data


class _ProductsData:
```

```
    __name__ = "ProductsData"
    hair_dryer = read_tsv("hair_dryer.tsv")
    microwave = read_tsv("microwave.tsv")
    pacifier = read_tsv("pacifier.tsv")


def text_normalize(text: str):
    """
     HTML unescape
     <br /> -> \n
    """
    text: str = html.unescape(text)
    text = text.replace("<br />", "\n")
    return text

#     marketplace  customer_id  review_id    product_id   product_parent
#     product_title      product_category   star_rating  helpful_votes
#     total_votes  vine   verified_purchase  review_headline    review_body
      review_date


ProductsData = _ProductsData()

if __name__ == "__main__":
    print(ProductsData)
    pass
```

(2) NL.py

```
from nltk.corpus import stopwords
from nltk.corpus.reader import reviews
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.stem.porter import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
from nltk import pos_tag, FreqDist

from wordcloud import WordCloud

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

import re

from DataParser import ProductsData, RecordWrap

_stopwords = stopwords.words('english')
_stopwords += ["would", "microwave"]

print("_stopwords", _stopwords)

pd.set_option("display.max_colwidth", 200)

nonalpha = r"[^A-Za-z'\s]"
re_nonalpha = re.compile(nonalpha)


# function to plot most frequent terms
def freq_words_diag(words: list, terms=10, title=""):
    fdist = FreqDist(words)
```

```
    # print({'word': list(fdist.keys()), 'count': list(fdist.values())})
    words_df    =    pd.DataFrame({'word':    list(fdist.keys()),    'count':
list(fdist.values())})

    # selecting top 20 most frequent words
    d = words_df.nlargest(columns="count", n=terms)
    plt.figure(figsize=(20, 5))
    ax = sns.barplot(data=d, x="word", y="count")
    ax.set(ylabel='Count')
    ax.set_title(title)
    plt.show()


def wordscloud_diag(fdist):
    wordcloud = WordCloud(
        width=1024,
        height=768,
        background_color='white',
        max_words=100,
        max_font_size=80,
        random_state=42
    ).generate_from_frequencies(fdist)

    fig = plt.figure()
    plt.imshow(wordcloud)
    plt.axis('off')
    plt.show()


def get_words_list(words: str):
    # remove some symbols
    words = re.sub(re_nonalpha, '', words)
    # to lower case
    words = words.lower()
    # split
    words = words.split()

    # Remove
    # words = [word for word in words
    #          if word]

    # Remove stopwords
    words = [word for word in words if word not in _stopwords]

    b = words
    # # Stemming
    # ps = PorterStemmer()
    # Lemmatisation
    lem = WordNetLemmatizer()
    words = [lem.lemmatize(word) for word in words]

    # Remove stopwords
    words = [word for word in words if word not in _stopwords]


    # for k, v in enumerate(b):
    #     if b[k] != words[k]:
    #         print(f"${b[k]} -> ${words[k]}")
    return words
```

```
def get_quality_review(records: list, head=True):

    ret = []
    if head:
        ret += records[0]
    r = records[1:] if head else records
    for record in r:
        if is_quality_review(record):
            ret += record
    return ret

def is_quality_review(record: RecordWrap):
    isgood = record.is_vine or \
             (record.total_votes >= 4 and
              record.helpful_votes_rate >= 0.5)
    return isgood

if __name__ == '__main__':
    # all_reviews = "\n".join(
    #     record.review_date for record in ProductsData.microwave[1:60])

    # all_reviews = ProductsData.microwave[2].review_body

    # print(pos_tag(word_tokenize(all_reviews)))

    stat = {
        "all": [],
        "quality": [],
        "star1": [],
        "star2": [],
        "star3": [],
        "star4": [],
        "star5": [],
    }

    for record in ProductsData.microwave[1:]:
        words_list = get_words_list(record.review_body)
        assert "wa" not in words_list

        stat["all"] += words_list
        if is_quality_review(record):
            stat["quality"] += words_list
        stat[f"star{record.star_rating}"] += words_list


    # print(all_reviews)
    freq_words_diag(stat["all"], 40, "All Reviews")
    freq_words_diag(stat["quality"], 40, "Quality Reviews")
    freq_words_diag(stat["star1"], 40, "Star-1 Reviews")
    freq_words_diag(stat["star5"], 40, "Star-5 Reviews")
```