

E-commerce begins to spread across our vision. Based on the past data, we gradually analyze the marketing strategy and design scheme. Traditional schemes focus on static evaluation, which is difficult to reflect the real-time dynamic changes of enterprise marketing strategies. Here are three products to sale: pacifier, hair dryer and microwave.

We first build FTN, a framework with fuzzy judgement by AHP analysis, time series analysis, NLP semantic parsing method. We construct the fuzzy judgement model and evaluation index system with four layers, analyzing the problem with fuzzy matrix and comprehensive weight by judgement matrix. We get the proportion of sales in the five levels of three products. For this ratio, we combine rating, review and helpfulness to analyze. In order to analyze the emotional color of the comments, we used NLTK to extract all the words. For each product and three kind of users: Vine, verified and normal, we took out the top 50 representative adjectives and each word was assigned a score of 1-9. Rating, review and helpfulness. Combine them to get a reference value. It can reflect the sales value. Then we use time series analysis, and analyze its reference value to determine whether it is a stationary series. We also compared the correlation between specific words and star, words with the reference value, and obtained many reference values for the changing of star over time. If it is not stationary, we consider using model ARIMA to fit it. In this way, we can get a trend of its sales over time through the model. This way we know whether the product is potentially successful or failing product.

Having done the basics above, we use the variance characteristic model of portfolio to consider the optimal ratio of investment between three products. And we analyze these data, the company's sales strategy and sales plan to give appropriate recommendations.

Because the fuzzy matrix is subjective and arbitrary, we conduct sensitivity analysis to observe and present a memo of our work to the Sunshine company.

Contents

Table of Contents

Table of Contents	1
Letter to Sunshine Company.....	1
1 Introduction.....	3
2 Assumptions.....	5
3 Parameter and Definitions.....	6
4 FTN: fuzzy judge, time series analysis, and NLP of wealth profile	7
4.1 Why FTN	7
4.2 Wealth Profile Analysis.....	8
4.3 NLP Processing.....	8
▲ NTLK Processing.....	9
4.4 Fuzzy Judge	11
4.5 Time series analysis	17
4.6 NLP semantic parsing method	23
4.7 Designs and page	23
4.8 Sensitivity Analysis.....	24
5 Conclusion	26
6 Appendix	27
➤ Code	27
➤ Imagies	38

Letter to Sunshine Company

Dear to Sunshine Company:

We have analyzed the data of three products: a **microwave oven**, a **baby pacifier**, and a **hair dryer**. The data are from 2002 to 2015. The data for the first five years are relatively few, so we mainly analyzed the data from 2008 to 2015. We used mathematical quantitative and qualitative methods to obtain some useful information for you. We will use the following index to reflect the status of the company's products.

Online sales strategy:

We used fuzzy judgement for each product. The secondary index goes for : star ratings,

reviews and helpfulness rating. We give it a weighing 0.3, 0.6, 0.1 in terms of their importance.

The reference of sales is divided into 5 levels. And the product of each level is listed in the table below.

The level of sales	1	2	3	4	5
Microwave oven	0.09278	0.10817	0.10690	0.31013	0.38202
Baby pacifier	0.12041	0.08179	0.16811	0.42468	0.20501
Hair dryer	0.11343	0.08076	0.12339	0.38076	0.30166

We know the main level of three products fall in the area of level 4 and 5. The sales level of all three products is not bad, but one thing we cannot neglect:

Microwave :The weighing of star ratings $O=(0.2508, 0.0695, 0.0828, 0.1845, 0.4125)$

The proportion of the first level is extremely higher than other products. It does enjoy a good reputation in the market.

So what you need do is to enhance product quality and next we will analyze how to improve product quality.

Through the customer return rate and the dependence of the product, we find that the product return rate is higher and higher. You need to develop some marketing tools to attract those old customers.

Using the variance characteristic model of portfolio in finance.

We define a reference value: $S_i = m_i \times y_1 + \bar{S}_r \times y_2 + z_i \times y_3$

Three kinds of products' reference value of each product are S_1, S_2, S_3 as Microwave oven, baby pacifier and hair dryer. The mean value is $\bar{r}_1, \bar{r}_2, \bar{r}_3$, weight is set to $\omega_1, \omega_2, \omega_3$, the average of the reference values for this combination is $\bar{r}_p = \omega_1 \bar{r}_1 + \omega_2 \bar{r}_2 + \omega_3 \bar{r}_3$, the variance of the combination is $\sigma_p^2 = \omega_1^2 \sigma_1^2 + \omega_2^2 \sigma_2^2 + \omega_3^2 \sigma_3^2$,

First order condition:

$$\frac{\partial \zeta}{\partial \omega_i} = 0 \rightarrow 2\sigma_i^2 \omega_i + \lambda \bar{r}_i + \mu = 0 \quad i=1,2,3$$

The final result of optimal result is : $(\bar{\omega}_1, \bar{\omega}_2, \bar{\omega}_3) = (0.34, 0.41, 0.25)$

That means if you're going to invest S in three products. The Microwave oven should be 0.34S, the baby pacifier should be 0.41S, the hair dryer should be 0.25S. That's the sales strategy to allocate the money.

To conclude:

The microwave and hair dryer are potential successful products, accounting for nearly 70% of sales in level 4 and 5. But microwave is the most controversial products, so if the sunshine

company enter this market, the company will face the biggest challenge. For the hair dryer, the overall sales level will be in the middle of the position and the high level probability is not high.

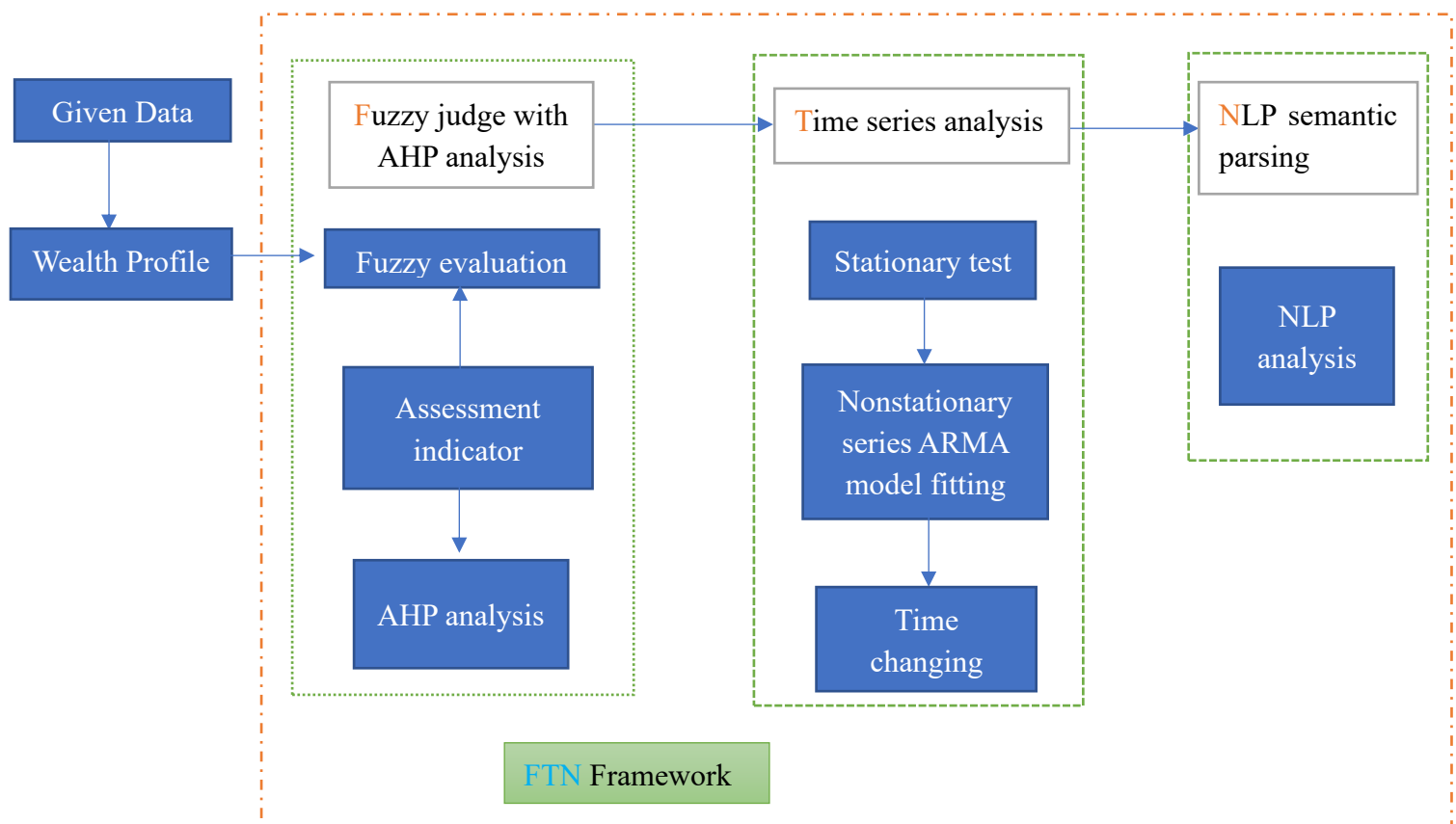
Sincerely team#2011744

1 Introduction

E-commerce has gradually become one of the mainstream consumption methods and it compete with traditional offline consumption constantly. As the world's largest e-commerce platform, amazon has its own unique evaluation system. It gives customers who buy or spend a chance to review and rate stars and vote on their comments. E-commerce companies can use these data to find the shortcomings of their products and the response in the market, so as to improve their products and decide to adapt to the market better.

Our task is to establish a model suitable for sunshine company to deal with the big data of previous feedback, and provide a marketing strategy including product relevance, customer return rate and so on from the qualitative and quantitative perspective, so as to complete the analysis of different products of sunshine company.

In this work, we propose a framework, named **FTN** (Fuzzy judge with AHP analysis, Time series analysis, NLP semantic parsing method), to fill in the gap above. The framework of FTN is shown below.



Framework of FTN of wealth profile can be summarized as the following steps:

- Fuzzy judge with AHP analysis
 - ✧ By constructing the evaluation index system & analyzing the data of rating and review, we can establish goal criteria and objects.
 - ✧ AHP: We construct the judgment matrix with the scale method of 1-9, then obtain the characteristic vector of the judgment matrix. Furthermore, we make the consistency test through the hierarchical single sorting and total sorting. So, we can get the weight of the index.
 - ✧ Establish an evaluation set and construct an evaluation matrix.
 - ✧ Construct the fuzzy comprehensive evaluation function, determine the weights of ratings and reviews for all levels of sales in question a, and analysis sales in the last season.

The above solves *question a & c*.

- Time series analysis
 - ✧ To better reflect the trend based on time, we introduce the time series analysis. First, according to the data obtained in *question a*, we get a reference by the subset of star ratings and review, then find the monthly average.

Sales										
Review									Star	Hel_
Vine			Verified			Normal				Helpful
Pos_	Neg_	Neu_	Pos_	Neg_	Neu_	Pos_	Neg_	Neu_		Total

- ✧ We import the data into MATLAB[®], let X-axis be time, Y-axis be a reference of Sales. Then consider the smooth consequence (wide and smooth). If the consequence is not smooth, then use ARIMA to construct model. After data fitting, we then are able to tell if a product's reputation is increasing or decreasing.
- ✧ Third, we count separately from one star to five stars, then plot these into the axis with an abscissa of time and an ordinate of one to five stars. Meanwhile, counting the average review for a day gives us a graph of time-based review. After that, as the trend changes over time, we look for the change of the reference index, and see if it exists a clear trend of change before and after a certain specific rating.

Till here, *question b & d* should be gracefully addressed if everything goes as expected.

- NLP semantic parsing method
 - ✧ In practice, review affects customer even more than the star rating. Therefore, we use

the *NLTK* ^{*}(Natural Language Toolkit) to process the reviews by extracting the 50 most frequent words and the 50 most frequent adjectives.

2 Assumptions

First and foremost, we make some fundamental assumptions and explain the rationality.

- **Assumption 1:** The sample reflects the assumption of all customers.

This assumption is the premise of our work as we are dealing with a limited amount of data. If we recognize the randomness of the distribution, we can easily get all customers' preferences and other factors.

- **Assumption 2:** All the assessments given by customers are in line with their own true wishes.

There are some bad purchases like click farming and malicious bad review because of some reasons like the competitiveness, but these people make up a very small part of the population, so we neglect the impact of the bad behavior.

- **Assumption 3:** The company and the market are in the opening and continuous running situation.

The qualitative and quantitative measures are based on the predictability of the time so that time can be fitted through the model. And we need continuity in the market.

- **Assumption 4:** The market is perfect and all companies are in a fully competitive market.

This kind of market can achieve the allocation of resources and pareto optimization. Both producers and consumers have accurate information. Only when we admit that there are no other factors like monopoly, we can consider the conclusion to be universal and the conclusion has the very good reference significance.

- **Assumption 5:** The parallel evaluation index system constructed in fuzzy evaluation can be compared in pairs.

Like the axiom of choice, the qualitative of the review is obscure, and we can take the comparability like two reviews for granted. And we can get the eigenvector by AHP method and figure out the weight.

- **Assumption 6:** All evaluation factors sets can be divided into several levels.

^{*} NLTK is a leading platform for building Python programs to work with human language data. See more at <https://www.nltk.org>

For example, sales can be divided into four levels, we put the evaluation factors and evaluation levels to form a judgement matrix according to the combination of qualitative and quantitative. We make this assumption to guarantee a valid solution.

➤ **Assumption 7:** Longer reviews and more helpful votes have a positive effect on sales. When people see longer reviews and more helpful votes, they will have a great interest in the product. They have a certain understanding of the product, which increases the likelihood of purchase.

3 Parameter and Definitions

For compactness, we define a series of parameters for notions concerning wealth profile in following tables.

Symbol	referent
Hel_	Helpfulness_rating
Pos_	Positive review
Neg_	Negative review
Neu_	Neutral review
Star	Star ratings
Sales	Sales value
Helpful	Helpful_votes
Total	Total votes
Verified	Verified purchase
Normal	Exception of Vine and Verified
Review	Review value

Table 1: Symbol description of relevant terms

Parameter	Referent	Parameter
S_i	Reference value of sales	A the weight vector of star
r_i	The mean value of sales	B the weight vector of review
ω_i	The weight of rating, review, hel_	C the weight vector of hel_
σ_i	The variance of rating, review, hel_	Z the weight vector of rating, review, hel_
λ	Lagrange multipliers	W the fuzzy evaluation matrix of sales

μ	Lagrange multipliers	F the weight vector of sales
A_i	Primary index of fuzzy judge	
B_i	Secondary index of fuzzy judge	
C_{ij}	Third index of fuzzy judge	
D_{ij}	Fourth index of fuzzy judge	
l_i	The ratio of Vector A	
m_i	The ratio of Vector B	
n_i	The ratio of vector C	
f_i	The ratio of vector F	
$k_{i,j}$	The score of pos_, neu_ and neg_ words.	

Table2: parameter evaluation

4 FTN: fuzzy judge, time series analysis, and NLP of wealth profile

4.1 Why FTN

There are many requirements given in the title. It is difficult for a single model to answer the question perfect. We use FTN to solve the problems, which combined fuzzy requirement, time series and natural language processing. Through the given data, we use the NLTK package in python to make the machine recognize human feelings, grab text and select the "attitude words" (words that indicate whether customer like or not) , And we quantified the emotional words with the fuzzy judgment, which we can get the useful data for our next step. Then using time series can extend historical data and infer product changes in the next few years. Fuzzy judgment can quantify the emotional vocabulary of text in order to make better use of computers to calculate reasoning.

In this section, we firstly create the wealth according to provided data. Then we will give a model: FTN, to fuzzy judge, give time series analysis and use NLP semantic parsing method, based on the provided data set comprised of 45 variables from 2002 to 2015.

4.2 Wealth Profile Analysis

After we got the data of the attached file, we have deeply analyzed and we found it exists redundant data, and the entire feedback for Sunshine occurs in US, so the data of marketplace is useless. And the product parent is same for the same product. The review title and review are bundled. We can also ignore these data. At the same time, we also find correlation or logical relationship between some data. In addition, we used mathematical quantitative and qualitative methods to obtain some useful information for Sunshine. We will use the following index to reflect the status of the company's products.

We can see that customer satisfaction

Product relevance:

We all know the sales cases of beer and diapers, and we can't help but consider whether it exists the relevance of three products of Sunshine. We get the Figure

We establish the three -level index with the data of Verified, Vine, Normal, Total, and Helpful votes. The three-level index is recorded as C1, C2, C3, C4, C5. We set the sale as the main index, set Review, Rating and Helpfulness votes as the sub-index , set Vine, Verified purchase, Normal review as the sub-index of the review, and set Positive Negative Neutral as the forth-level index to comfort the different types of the review. Then wo use fuzzy evaluation.

After getting the corresponding weights, we first analyze the ratings and reviews.

Then we can get the trend of Sunshine product, ally successful or failing product.

In *question b*, we multiplied rating and review and whether they were vine or verified and multiplied their respective weights. Then we took a reference value, took out the data for each month as the average value, and then imported it into MATLAB to observe the change of sales reference value. First, do the stationary test. If the sequence is stable, we think that the reputation will not change significantly. If the sequence is not stable, consider using the AMRA model to fit. Forecast changes in reference values.

Then we find the top fifty words, and we observe the rating level corresponding to each word. The details are explained in the next model.

4.3 NLP Processing

In this section, we firstly create the wealth according to provided data. Then we will give a model: FTN, to fuzzy judge, give time series analysis and use NLP semantic parsing method,

based on the provided data set comprised of 45 variables from 2002 to 2015.

▲ NTLK Processing

Find the two titles in data: review headline and review body.

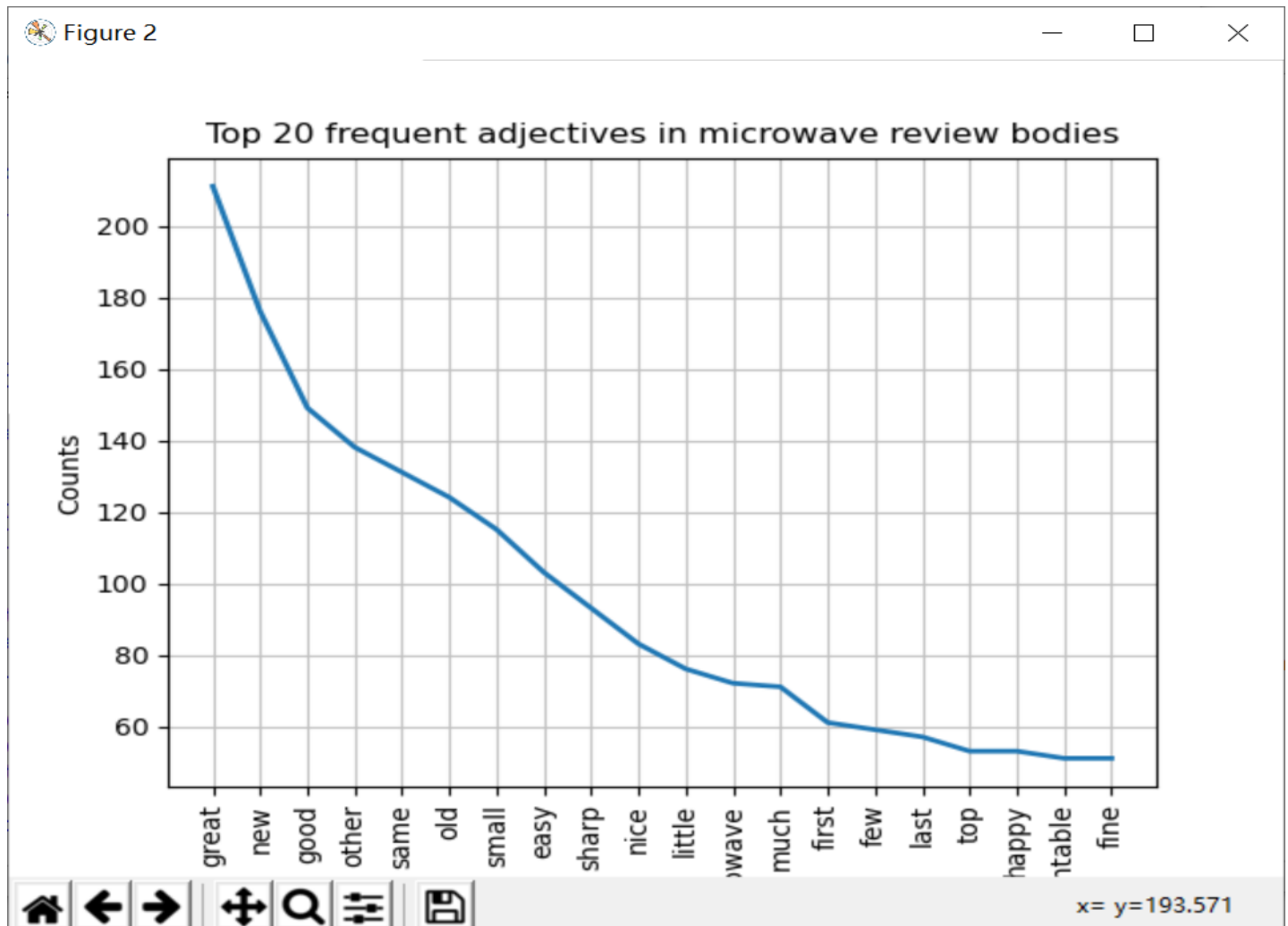
Review headline and review body is found with every 2 words.

We first remove some words like 'a', 'the', the words that come up a lot time but we cannot figure out the emotion of these people.

We choose the most emotional adjectives and the first fifty words are arranged in order.

We use 1-9 scale method of fuzzy matrix.

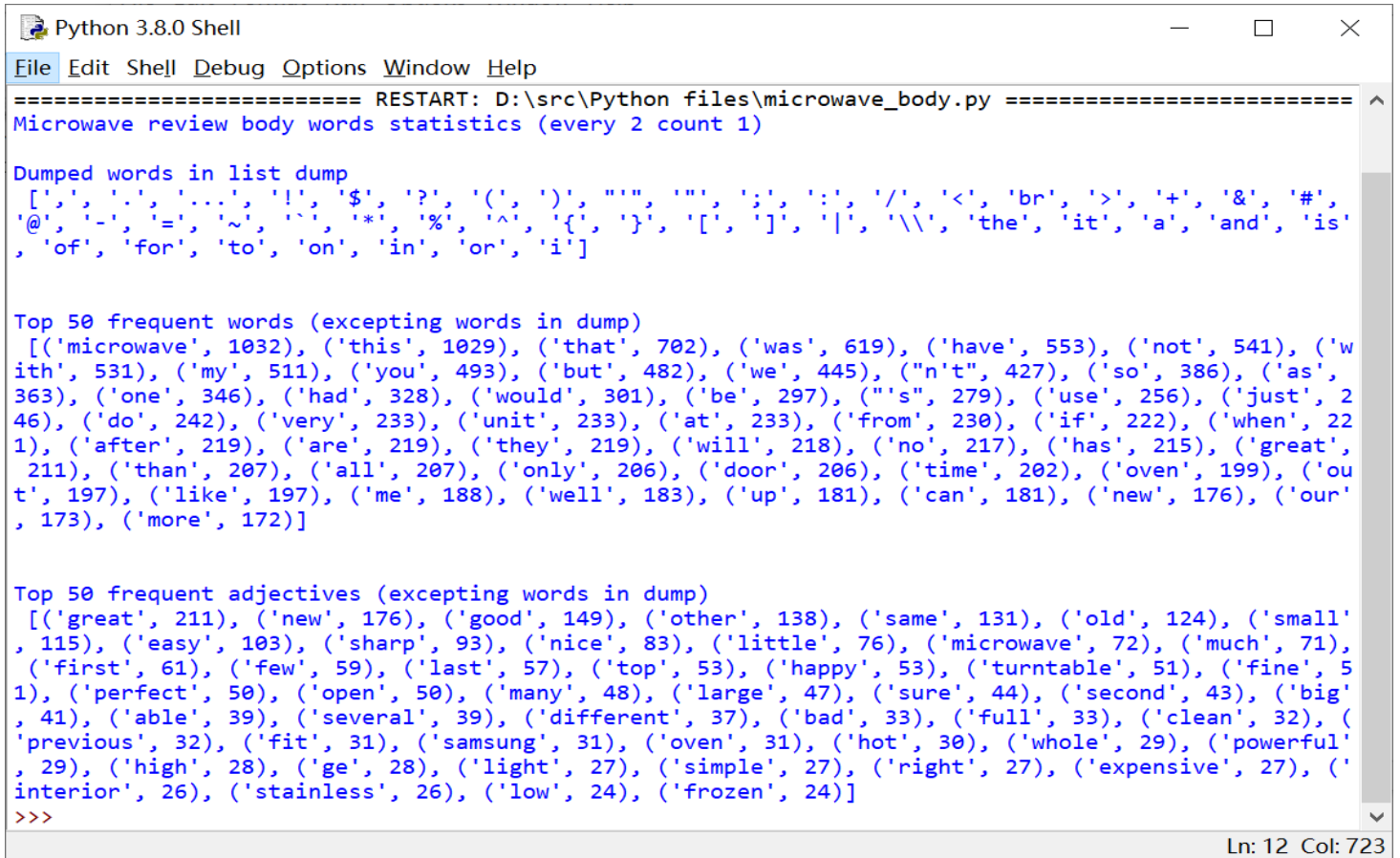
We assigned the positive words with 7-9 scores;
the neutral words with 4-6 scores;
the negative words with 1-3 scores.



Here are top 20 words (**all** processed with the form of **every 2 count 1**) in the figure, see appendix for hair dryer and pacifier.

Here are top 50 frequent words in microwave body for example. And the customers are

verified customer, they don't enjoy the discount, and make up the majority of all customers.



```

Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
===== RESTART: D:\src\Python files\microwave_body.py =====
Microwave review body words statistics (every 2 count 1)

Dumped words in list dump
['.', ',', '!', '$', '?', '(', ')', '"', "'", ':', ';', '/', '<', 'br', '>', '+', '&', '#', '@', '-', '=', '~', '^', '*', '%', '{', '}', '[', ']', '|', '\\', 'the', 'it', 'a', 'and', 'is', 'of', 'for', 'to', 'on', 'in', 'on', 'i']

Top 50 frequent words (excepting words in dump)
[('microwave', 1032), ('this', 1029), ('that', 702), ('was', 619), ('have', 553), ('not', 541), ('with', 531), ('my', 511), ('you', 493), ('but', 482), ('we', 445), ('n't', 427), ('so', 386), ('as', 363), ('one', 346), ('had', 328), ('would', 301), ('be', 297), ('s', 279), ('use', 256), ('just', 246), ('do', 242), ('very', 233), ('unit', 233), ('at', 233), ('from', 230), ('if', 222), ('when', 221), ('after', 219), ('are', 219), ('they', 219), ('will', 218), ('no', 217), ('has', 215), ('great', 211), ('than', 207), ('all', 207), ('only', 206), ('door', 206), ('time', 202), ('oven', 199), ('out', 197), ('like', 197), ('me', 188), ('well', 183), ('up', 181), ('can', 181), ('new', 176), ('our', 173), ('more', 172)]

Top 50 frequent adjectives (excepting words in dump)
[('great', 211), ('new', 176), ('good', 149), ('other', 138), ('same', 131), ('old', 124), ('small', 115), ('easy', 103), ('sharp', 93), ('nice', 83), ('little', 76), ('microwave', 72), ('much', 71), ('first', 61), ('few', 59), ('last', 57), ('top', 53), ('happy', 53), ('turntable', 51), ('fine', 51), ('perfect', 50), ('open', 50), ('many', 48), ('large', 47), ('sure', 44), ('second', 43), ('big', 41), ('able', 39), ('several', 39), ('different', 37), ('bad', 33), ('full', 33), ('clean', 32), ('previous', 32), ('fit', 31), ('samsung', 31), ('oven', 31), ('hot', 30), ('whole', 29), ('powerful', 29), ('high', 28), ('ge', 28), ('light', 27), ('simple', 27), ('right', 27), ('expensive', 27), ('interior', 26), ('stainless', 26), ('low', 24), ('frozen', 24)]
>>>
Ln: 12 Col: 723

```

Here is an example of how we process the data in Microwave_review_body.

word	number	category	score	Total score	The reason
great	211	Positive words	8	1688	It's a positive adjective and can describe the great feeling of the customer.
sharp	93	Neutral words	4	372	We look it up in the comments. Comments like sharp warranty is negative but it is also used to compare with Some reputed companies which has a name on it. So, it's a neutral word.
old	124	Negative words	2	248	It's used to describe the microwave. It's too old to use. So it's a negative word.

Take the microwave review body for example.

The positive words in 50 typical words score a total of 9448 points. And it occupies 20 words.

The neutral words in 50 typical words score a total of 3963 points. And it occupies 22 words.

The negative words in 50 typical words score a total of 1130 points. And it occupies 8 words.

4.4 Fuzzy Judge

Based on the analytic hierarchy process core, we find three kinds of adjective words via NLTK package in python and use the 1-9 scale method.

The kind of the customer	vine	verified purchase
Vine	Y	N
Verified	N	Y
Normal	N	N

To distinguish between the kind of the customer.

Here is a brief point of view.

Vine customer are often at the top of the list of reviews and attract the attention of customers. They play an important role in company's credibility.

Verified customer don't enjoy the discount, but there will be no brushing behavior, their purchase is more trusty than normal customer.

Normal customer usually get a discount. They may buy the product because of the promotion, but it is not their original intention. So it is not valuable for reference.

Next we give the **normal customer** for example and analyze its fuzzy evaluation result.

These indicators are different for different products, microwave is represented for the fuzzy judge, and we will analyze the statistic in two other products.

$$\text{Firstly, there is a judgment matrix } p_{i,j} = \begin{pmatrix} 1 & \frac{k_{11}}{k_{12}} & \frac{k_{11}}{k_{13}} \\ \frac{k_{12}}{k_{11}} & 1 & \frac{k_{12}}{k_{13}} \\ \frac{k_{13}}{k_{11}} & \frac{k_{13}}{k_{12}} & 1 \end{pmatrix} \quad (i, j=1,2,3)$$

$k_{i,j}$ means the total score of the adjective words. (This applies to normal customers.)

We know that $k_{11} = 9448$, $k_{12} = 1130$, $k_{13} = 3963$;

$$\text{So } p_{i,j} = \begin{pmatrix} 1 & a_{12} & a_{13} \\ a_{21} & 1 & a_{23} \\ a_{31} & a_{32} & 1 \end{pmatrix} \quad p_{i,j} = \begin{pmatrix} 1 & \frac{9448}{1130} & \frac{9448}{3963} \\ \frac{1130}{9448} & 1 & \frac{1130}{3963} \\ \frac{3963}{9448} & \frac{3963}{1130} & 1 \end{pmatrix}$$

Then we use the square root method to solve the eigenvectors.

$$\text{We know that } M_i = \prod_{j=1}^n p_{ij} \quad \omega_i = \sqrt[n]{M_i}$$

After normalization as follows, ω_i can be converted to eigenvectors.

$$W'_i = \frac{\overline{\omega_i}}{\sum_{i=1}^n \overline{\omega_i}} \quad W' = (W'_1, W'_2, \dots, W'_n)$$

$$\rightarrow (0.649, 0.077, 0.242)$$

In order to judge the accuracy of the matrix, we need to carry out consistency test.

Divided sales into five levels: 1, 2, 3, 4, 5

The membership degree is calculated from relative priority matrix to form fuzzy evaluation matrix.

$$\text{So there is matrix } L_{i,j} = \begin{pmatrix} a_{11} & \dots & a_{15} \\ \vdots & \ddots & \vdots \\ a_{31} & \dots & a_{35} \end{pmatrix} \quad (I=1,2,3; j=1,2,3,4,5)$$

Adjective words are divided into 9 levels, and we need to match the adjective words to the 5 levels of sales.

We make a table for the corresponding.

Level of sales	Level of adjective words
1	1,2
2	3,4
3	5
4	6,7
5	8,9

The positive words range from 7-9

So it can only be judged from 4 to 5.

The number of 7 level of positive words equals to $k_{14} = 7 * k$, and the vector of the fuzzy

judgement will be $\left(0, 0, 0, \frac{k_{14}}{k_{11}}, \frac{k_{11} - k_{14}}{k_{11}}\right)$

Just like the way above, we can easily get the vector of positive, negative and neutral words

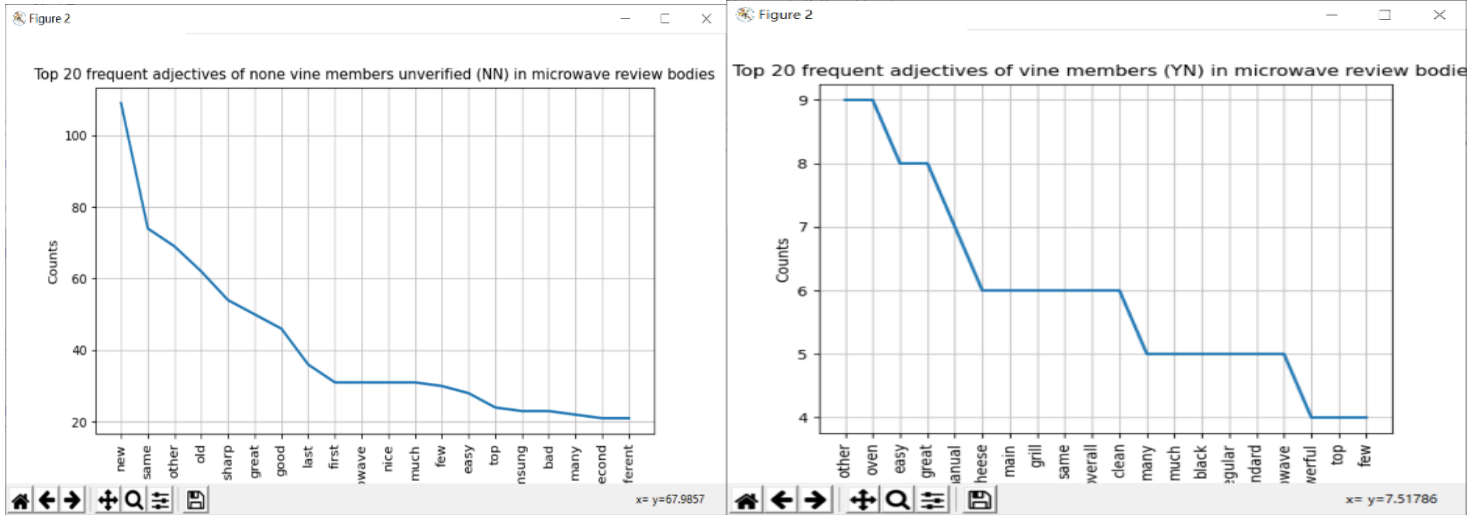
Mark them as $X_{c_1}, X_{c_2}, X_{c_3}$, we just combine them as

$$R = \begin{pmatrix} X_{c_1} \\ X_{c_2} \\ X_{c_3} \end{pmatrix} \rightarrow R = \begin{pmatrix} 0 & 0 & 0 & 0.31 & 0.69 \\ 0 & 0.09 & 0.71 & 0.20 & 0 \\ 0.33 & 0.67 & 0 & 0 & 0 \end{pmatrix}$$

$$X_2 = W' \times R = (0.01, 0.10, 0.27, 0.40, 0.22)$$

This applies to all users and we identify it as normal matrix.

Next, we find the vine customers and extract their key words, the same as above.



The vine users have the fuzzy matrix as R_2 and W_2'

The normal customers have the fuzzy matrix as R_3 and W_3'

$$X_1 = W_2' \times R_2 = (0.02, 0.13, 0.24, 0.43, 0.18)$$

$$X_3 = W_3' \times R_3 = (0.02, 0.12, 0.26, 0.45, 0.15)$$

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 0.02 & 0.13 & 0.24 & 0.43 & 0.18 \\ 0.01 & 0.10 & 0.27 & 0.40 & 0.22 \\ 0.02 & 0.12 & 0.26 & 0.45 & 0.15 \end{pmatrix}$$

Next, we will calculate the weights of vine, verified and normal customers.

First, we will have the kind of customers divided into 9 levels as above.

Here is the table.

The kind of the customer	Vine	Verified	Normal
The level	9	5	1

and the fuzzy matrix goes for $q_{i,j} = \begin{pmatrix} 1 & \frac{9}{5} & 9 \\ \frac{5}{9} & 1 & 5 \\ \frac{1}{9} & \frac{1}{5} & 1 \end{pmatrix}$ let's normalize it and figure out the

weights.

$$M_i = \prod_{j=1}^n q_{i,j} \quad \omega_i = \sqrt[n]{M_i}, W'_i = \frac{\overline{\omega_i}}{\sum_{i=1}^n \overline{\omega_i}}, \quad W' = (W'_1, W'_2, \dots, W'_n)$$

$$\begin{pmatrix} 1 & \frac{9}{5} & 9 \\ \frac{5}{9} & 1 & 5 \\ \frac{1}{9} & \frac{1}{5} & 1 \end{pmatrix} \rightarrow Z = (0.61, 0.32, 0.07)$$

So there is a fuzzy comprehensive evaluation matrix X of the secondary index of rating and the weight distribution matrix Z of the primary index. The effective value of the final result is expressed as $F = Z \times X = (0.0168, 0.1197, 0.0251, 0.4218, 0.1907)$

Next steps we will go for B2: Star

We're going to weight each level of these stars, and there is a table for the corresponding.

The level of star ting	1	2	3	4	5
The level of sale	1	2	3	4	5

We need to figure out the ratio of each grade of star from 1 to 5.

The amount of the specific rating level defines as u_i ($i=1, 2, 3, 4, 5$)

$$\text{And the vector goes for } O = \left(\frac{u_1}{\sum_{i=1}^n u_i}, \frac{u_2}{\sum_{i=1}^n u_i}, \frac{u_3}{\sum_{i=1}^n u_i}, \frac{u_4}{\sum_{i=1}^n u_i}, \frac{u_5}{\sum_{i=1}^n u_i} \right)$$

$$\rightarrow O = (0.2508, 0.0695, 0.0828, 0.1845, 0.4125)$$

Now we analyze B3: help_, it can be divided into C31 :helpful_ and C32: total_.

The difference is not distinct, so we assign the weight vector between the total and helpful votes as the same.

To analyze the problem, we have to admit that B3_help have relevance to B1: review and B2: star ratings. When we give it a reference value, we should consider that there will be much votes but most of them are zero. According to the theory of utility in economics, the diminishing marginal utility means the more votes there are, the less marginal utility there is.

There might be the vote against the review, so we compare the data and multiply it with (-1).

So we define the formula mode as follows:

$$\text{ScoreB3} = 0.1 \times m_1 \times z_1 + \overline{S}_r$$

m_1 : the rating in line with the number of helpful votes

z_1 : the reference of the hel_

$$z_0 = \begin{cases} z_o, z_o \leq 5 \\ 5 + 0.1(z_o - 5), 5 < z_o \leq 50 \\ 9.5 + 0.01(z_o - 50), z_o > 50 \end{cases}$$

$$z_1 = z_0 - 1.0 \times (m_2 - m_3)$$

\overline{S}_r : the average score of reviews (It is different from the 1-9scale method above, we take positive words with a positive number and no more than three points, while the negative words with a minus and not less than minus two.)

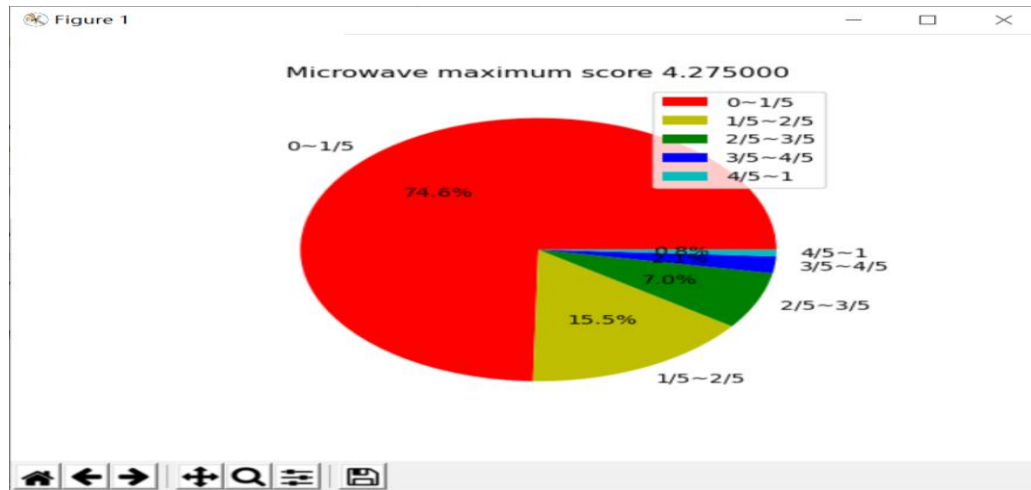
m_2 : the amount of total words

m_3 : the amount of helpful words

The highest number is $S_{B_3} = 3.775$. From 0 to S_{B_3} , divide it to 5 levels.

There will be a vector $V_{B_3} = (0.746, 0.155, 0.07, 0.023, 0.008)$

And here is a histogram to see how it weighs.



$$F_{B_3} = V_{B_3} = (0.746, 0.155, 0.07, 0.023, 0.008)$$

To sum it up, the weigh between the factors should be:

B1: review = 0.6 B2: Star = 0.3 B3: Hel_ = 0.1

$$F = (0.0168, 0.1197, 0.251, 0.4218, 0.1907)$$

$$O = (0.2508, 0.0695, 0.0828, 0.1845, 0.4125)$$

$$F_{B_3} = (0.746, 0.155, 0.07, 0.023, 0.008)$$

So the final result of the vector will be

$$J_1 = \frac{3}{5} \cdot F + \frac{3}{10} \cdot O + \frac{1}{10} \cdot F_{B_3} = (0.15992, 0.10817, 0.0469, 0.31073, 0.37428)$$

This is microwave's weigh between the level of sales.

Do as above, we can easily get the level of pacifier and hair dryer

$$\text{Pacifier: } F = (0.0193, 0.1005, 0.2071, 0.3725, 0.3006)$$

$$O = (0.0621, 0.0493, 0.0756, 0.1427, 0.6704)$$

$$F_{B_3} = (0.902, 0.067, 0.02, 0.0104, 0.0006)$$

$$\text{Hairdryer: } F = (0.0039, 0.1053, 0.1542, 0.5536, 0.1830)$$

$$O = (0.0883, 0.0554, 0.0869, 0.1820, 0.5874)$$

$$F_{B_3} = (0.846, 0.096, 0.048, 0.008, 0.002)$$

$$\text{Mark them as } J_2 = (0.12041, 0.08179, 0.16811, 0.42468, 0.20501)$$

$$J_3 = (0.11343, 0.08076, 0.12339, 0.38076, 0.30166)$$

From the vector above:

Pacifier: The level of sales accounts for 63% between 4 and 5. It's slightly lower than the other two. Though from the average rating star O, we can see that its level of 5 is higher than other two products. And the F: the distribution of the review, we can see it's going to focus on 4 and 5. If we just look at the star rating and reviews, we will find it a very good product.

It does enjoy a good reputation in the market. But when we look at the F_{B_3} : the helpfulness rating, it's quite different from other two products. Few customers give helpful votes. Therefore, pacifiers has a disadvantage in this secondary indicator. But this is not bad, the company can do more product innovation to attract the attention of customers. They may give their helpful votes in turn. So we think the pacifier has the greatest potential to be a successful product.

The microwave and hairdryer are potential successful products, accounting for nearly 70% of sales in level 4 and 5.

Microwave: If we just look at it now, the microwave has the highest composite ratio of level 4 and 5. Also, the proportion of level 5 is significantly higher than the other two products. However, it cannot be ignored that the microwave also has the largest proportion at level 1.

We look up the negative words by NLTK, and we find many negative words like frozen, bad.

But there are a lot of helpful votes on the market. According to the assumption 7, it attracts a lot of attention. We conclude that it is the most controversial products, so if the sunshine company enter this market, the company will face the biggest challenge. But as long as the company do a good job in product quality and reputation, they won't have a big problem.

Hairdryer: its sales' level is above the average, but the level of 5 accounts for not large proportion, its overall sales level is relatively stable. If the company enters the sales market, it needn't to do much marketing strategy, but the overall sales level will be in the middle of the position and the high-level probability is not high.

4.5 Time series analysis

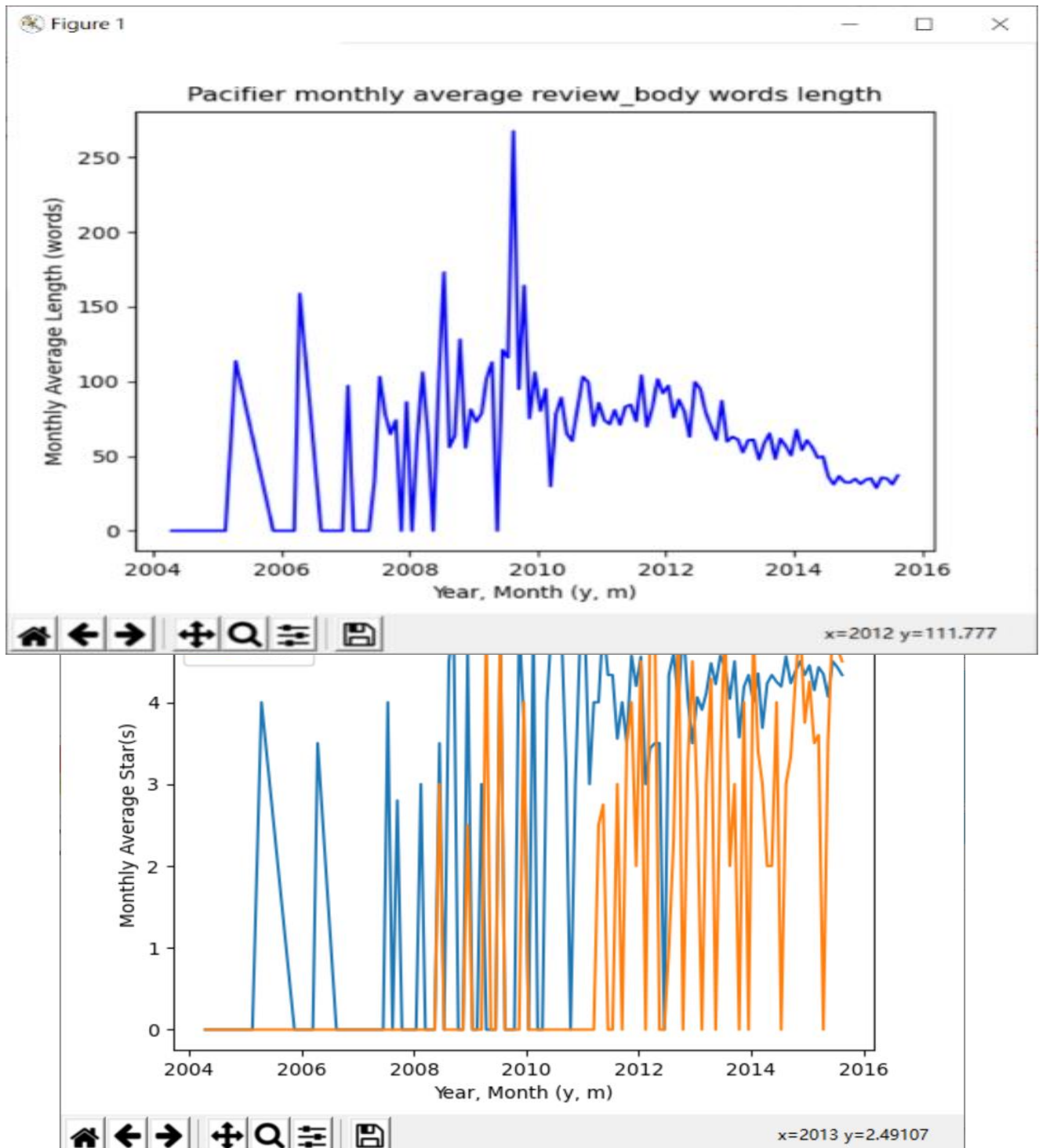
- ▲ We first find 50 words in every product. All words are in common, so we

Select 2 words (1 positive word and 1 negative word): *good, bad*

Select one word, and find the rating level of it, average it in one month. So, we get a point a month, and we could plot it with python in terms of time.

▲ Pacifier:

At the same time, we value the average length of the review monthly. Plot it in MATLAB. We will get 3 curves and below is one of them.



✧ Select the level of rating from 1 to 5, each rating we average it monthly and plot it with the changing of time from 2008 to 2015. We will get 5 curves c_1, c_2, c_3, c_4, c_5 (c_i means the level i of sales) and three figures.

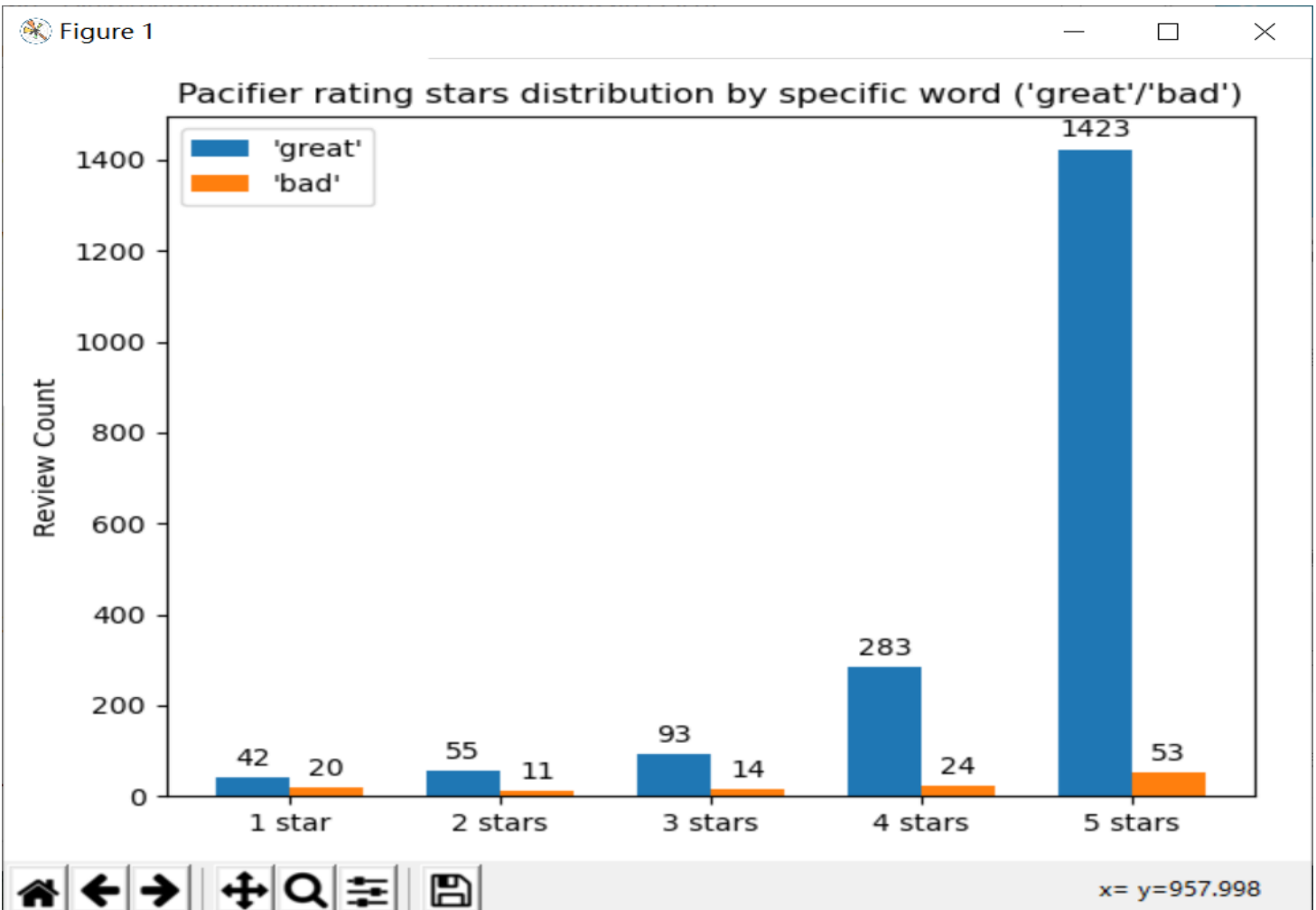
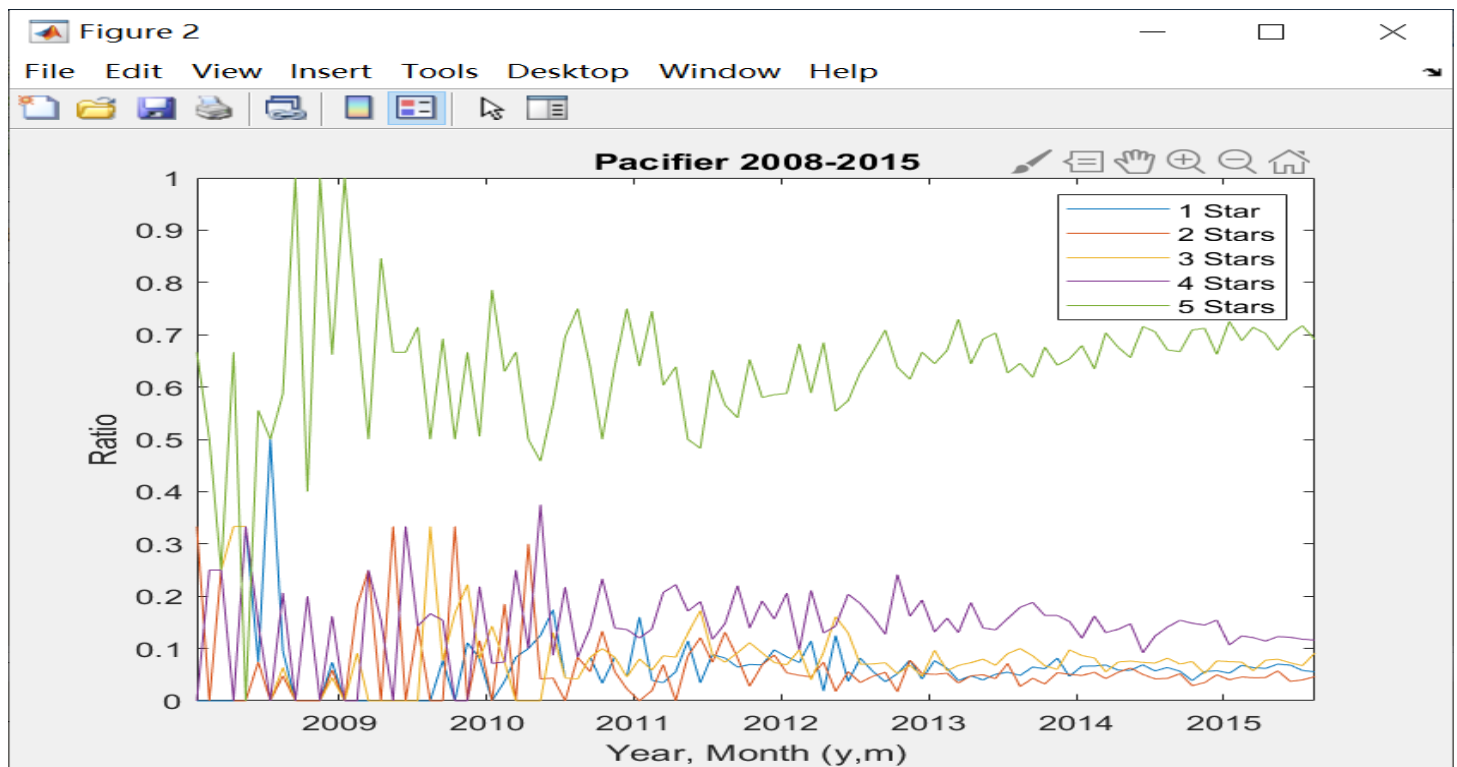
Next step, a combination of 1 and 3 will indicate the text-based reviews are associated with the ratings.

Pacifier: The curve c_5 and the figure of good have the same figure. So, we can deduce that the word like 'good' are strongly associated with the high level especially the level of 5. From the fig as follow, we find that the word 'bad' are strongly associated with level 1 and 2.

We find that the specific words have commonalities, and each product has a certain relevance.

Here we give a rating stars distribution by two words(great and bad) according to the data of microwave, the distribution of great is mainly focused on the level of 5 stars. The word 'bad' mainly falls in level of 1 star. In terms of the phrases 'not bad', and the

sample size is not large. The word 'bad' has not strong reference value.



Also , a combination of 2 and 3 tell us people tend to write more reviews after seeing a certain level. Pacifier and microwave are similar, people tend to write more reviews after seeing a large series of high ratings like 5 stars. And people tend to write fewer words after seeing less series of low star ratings. And they are relevant either at the bottom or at the peak of the 5 curves.

Pacifier: The average length of the pacifier and the proportion of the 5 stars change in tends to be the same. Or we can say that the length of the comment usually varies with the ratio of star.

The star ratio peaked in early 2009, and the average length of reviews peaked in late 2009. This indicates people tend to write more reviews after seeing a large series of high star levels.

Microwave: there is also similar correlation.

The average length peaked in late 2011 after the time when the star peaked in early 2011.

Hairdryer:

The situation is a little bit different.

The length of comments has a downward trend in the year of 2012-2015, but gradually leveled off. The same trend can be seen in the star of 3,4 and 5. And it's clear that 5 Stars have an upward trend over the years. So as the number 3,4,5 of stars decreases, people tend to write fewer words. People become more satisfied with the hair_dryer in the late years.

To analyze problem b, we have a combination of rating, review, help_ , and the weight vector is (0.3,0.6,0.1)

B1 ,B2, B3 have a weigh vector (y_1, y_2, y_3)

Here we define a function:

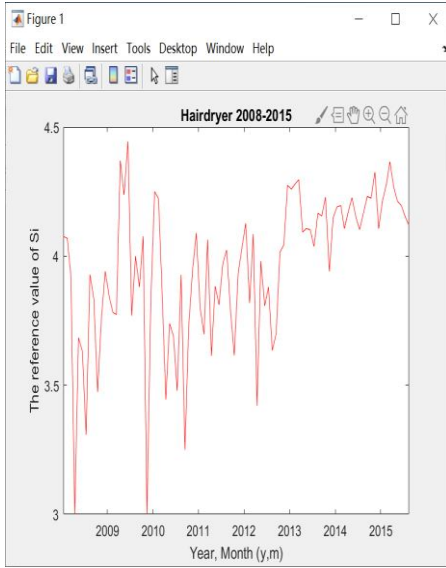
$$S_i = m_1 \times y_1 + \overline{S}_r \times y_2 + z_1 \times y_3, \quad (i=1,2,3)$$

m_1 : the star rating

\overline{S}_r : the average score of reviews (Unlike the scale of 1-9, if there are two positive words, just average them, positive words' value is no more than 3 and the negative words are assigned no less than minus 2)

z_1 : the amount number of help votes

Each id has a reference value S_i at a time. Sum the reference values of a month and average them. Draw the reference time series with time as horizontal axis in fig7.



Time series: $(S_{t_1}, S_{t_2}, \dots, S_{t_n})$, $\forall t_1, t_2, t_n \in T$

We analyze the time series with steps as follows.

1. Observe whether the sequence value is stable.

$$\rightarrow (1) \forall t \in T, EX_t^2 < \infty$$

$$(2) \forall t \in T, EX_t = \mu$$

$$(3)$$

$$\forall t, s, k \in T, k+s-t \in T, \gamma(t, s) = \gamma(k, k+s-t);$$

$$\gamma(t, s) = E(X_t - \mu_t)(X_s - \mu_s)$$

The fig.8 shows the series:

The pacifier satisfies the above conditions of the system, and we consider it to be a stationary sequence. So we can predict it will hover around the reference value 4.3, we know it's a potentially successful product.

We can get that the sequence of hair dryer and microwave does not satisfy with the stationary condition.

Considering the decomposition of unstable factors, we use ARIMA model to fit the curve.

$ARIMA(p, d, q)$ satisfy the equation set:

Each product has the different parameter of p, d, q

$$\begin{cases} \varphi(B)\Delta^d x_t = \theta(B)\varepsilon_t \\ E(\varepsilon_t) = 0, Var(\varepsilon_t) = \sigma_t^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \\ Ex_s \varepsilon_t = 0, \forall s < t \end{cases}$$

$$\Delta^d = (1-B)^d, \varphi(B) = 1 - \phi_1 B - \dots - \phi_p B^p, \theta_B = 1 - \theta_1 B - \dots - \theta_q B^q$$

\rightarrow the figure 8 is the time series S_t , and the figure 9 is the fitted curve

The microwave fits with (0.24,1,0) and (0,1,1)

After the curve fitting, we can predict that the microwave's reference will have fluctuations, but a gradual upward trend. Hair_dryer will have a sharp decline in the short term, but the reference value will rise again after a period of time, but the market outlook is uncertain. Pacifier's reference value is stable and maintains a good level.

The pacifier's market is pretty good, and the microwave is a potential successful product.

The hair_dryer's outlook is uncertain.

4.6 NLP semantic parsing method

We find 50 most frequent adjective words for each product, some words come up all the time. Here we list 5 words to see the rating level distribution of each, there are three positive words great, happy, perfect and two negative words: old, bad.

We analyze the data with each product and draw up the histogram. Here is the distribution of great and bad, we find that positive words have a higher rating level and negative words have a lower histogram.

From the 1-9 scale method for the fuzzy analysis above, we can have a corresponding table.

The level of adjective words	1-2	3-4	5	6-7	8-9
The level of star_rating	1	2	3	4	5
The variance of star_rating	0.11	0.39	0.82	0.40	0.26

Here is an example for word 'great' and word 'bad' distribution of three products. According to the data of microwave, the distribution of great is mainly focused on the level of 5 stars.

The word 'bad' mainly falls in level of 1 star. In terms of the phrases 'not bad', and the sample size is not large. The word 'bad' has not strong reference value.

4.7 Designs and page

From the time analysis above, specific ratings incite more reviews, for example, the word 'bad' may in a high frequency of occurrence after seeing a series of 1 or 2 ratings. People tend to look at the comments and feel a certain of empathy.

Also, people are likely to write more reviews after seeing a large number of specific ratings. According to assumption7, longer reviews stand for stronger purchasing power.

Measures1: company needs to enhance product quality and service level to ensure high star_rating, then the positive words will appear in large numbers accordingly. We use the fuzzy evaluation matrix to solve the problem, there is weight assigned between the two. We know positive words and higher star_ratings get a higher sales reference.

Measures2: Companies can offer discounts to customers who write longer reviews. Longer reviews stand for stronger purchasing power. This will stimulate the interest and desire of other customers.

we know that the reputation of product a,b,c is increasing or decreasing in a short time. The long-term trend is shown in the figure. For the period of decline in reputation, we need to

do something to stimulate the consumption. The influencing factors are rating, review and help_.

Measures3: more discounts means more 'No' in verified purchase. The reference value of help_ will increasing.

Measures4: Improve product quality in the early stage of product sales. Invite professionals to become vine members of amazon and invest some production cost in the early stage. The subsequent sales impact on the brand in permanent. Vine reviews will be at the top of the review page, and the professional information and good reviews will make the review more credible and persuasive. This builds the credibility and reputation of the product.

Measures5: In the decline reputation of the certain stage, the company could put a lot of energy into that reputable product.

Using the variance characteristic model of portfolio in finance.

Reference value of each product S_1, S_2, S_3 mean value $\bar{r}_1, \bar{r}_2, \bar{r}_3$, weight is set to $\omega_1, \omega_2, \omega_3$, the average of the reference values for this combination is $\bar{r}_p = \omega_1 \bar{r}_1 + \omega_2 \bar{r}_2 + \omega_3 \bar{r}_3$, the variance of the combination is $\sigma_p^2 = \omega_1^2 \sigma_1^2 + \omega_2^2 \sigma_2^2 + \omega_3^2 \sigma_3^2$,

So here's what we need to satisfy:

$$\min_{\omega_1, \omega_2, \omega_3} \omega_1^2 \sigma_1^2 + \omega_2^2 \sigma_2^2 + \omega_3^2 \sigma_3^2$$

$$\text{s.t. } \omega_1 \bar{r}_1 + \omega_2 \bar{r}_2 + \omega_3 \bar{r}_3 = \bar{r}$$

$$\omega_1 + \omega_2 + \omega_3 = 1$$

Set up the Lagrange function

$$\zeta = \omega_1^2 \sigma_1^2 + \omega_2^2 \sigma_2^2 + \omega_3^2 \sigma_3^2 + \lambda [\omega_1 \bar{r}_1 + \omega_2 \bar{r}_2 + \omega_3 \bar{r}_3 - \bar{r}] + \mu [\omega_1 + \omega_2 + \omega_3 - 1]$$

First order condition:

$$\frac{\partial \zeta}{\partial \omega_i} = 0 \rightarrow 2\sigma_i^2 \omega_i + \lambda \bar{r}_i + \mu = 0 \quad i=1,2,3$$

We can get the optimal selection $\bar{\omega}_1, \bar{\omega}_2, \bar{\omega}_3 = (0.34, 0.41, 0.25)$

4.8 Sensitivity Analysis

We assign 0.6 weight to review, 0.3 weight to star rating, 0.1 weight to helpfulness votes, while processing the data. But its mixed with some subjective emotion. We do the sensitivity analysis to determine the affect of review, rating and helpfulness votes for sales.

In order to show more clearly the effect of three factors for sales ,we specific the small variable to 0.01 . Here are three situations

First

review =0.601 B2: Star=0.3 B3: Hel_ =0.1

Do as above, we can get :

$$J_1' = (0.159937, 0.10809, 0.182661, 0.311152, 0.239161)$$

$$J_2' = (0.120429, 0.081891, 0.149016, 0.267493, 0.381811)$$

$$J_3' = (0.113434, 0.089505, 0.123544, 0.388114, 0.286403)$$

Second

review =0.6 B2: Star=0.301 B3: Hel_ =0.1

$$J_1'' = (0.160171, 0.10804, 0.182493, 0.310915, 0.239383)$$

$$J_2'' = (0.120472, 0.081839, 0.149016, 0.267493, 0.38218)$$

$$J_3'' = (0.113518, 0.089455, 0.123477, 0.387742, 0.286807)$$

Third

review =0.6 B2: Star=0.3 B3: Hel_ =0.101

$$J_1''' = (0.160666, 0.108123, 0.18248, 0.310753, 0.238978)$$

$$J_2''' = (0.121312, 0.081857, 0.14896, 0.26736, 0.381511)$$

$$J_3''' = (0.114276, 0.089496, 0.123438, 0.387568, 0.286222)$$

Comparing the four types of data before and after, we can see that increasing the weight of review can influence more than others for microwave. And for pacifier and hair-dryer, review still is the most important parameter. But changing the weight has little effect on the entire model. The weights we use are reasonable.

7.Strengths and Weaknesses

Strengths:

1) solve the requirements of the question better

It is difficult to solve all the problems with a single model. We choose to combine fuzzy judge, time series analysis, and NLP of wealth profile to form FTN. Through the connection of the data before and after, we can solve the requirements in the question one by one

2) fit better

Instead of a single model, we chose a combination of models. For each requirement, we choose the most suitable model to solve it. And, through the data connection between requirements , we make the logic more meticulous and reduce the workload.

3) More similar with the reality

In model processing, we are as close as possible to reality. In addition to some necessary

idealization assumptions, we take into account customer psychology, customer behavior, customer loyalty, etc.

4) Using NLP

One of the difficulties in the requirements is the processing of text . Here we use NLP to capture text data and quantify the key words in the text to enable the computer to calculate. Using NLP, we don't need to spend a lot of time reading the text

5) Handle data more rigorously

The use of data in our problem is continuous. For data that does not come from historical data, we verify it again . For example, in order to ensure the rationality of weight distribution, we also take the sensitivity analysis.

Weakness:

There are several weaknesses we could find for the model.

- (a) A person who says negative words doesn't necessarily mean they hate the product. 'not bad' may indicate he is fond of the product. But when we search for negative words and categorize them as negative emotions.
- (b) We only searched for the first fifty words and assigned them with scores according to the emotion. But many words are not common, but they express strong feelings. In this way we omit some necessary words.
- (c) Each individual is different, assigning the same value to one words doesn't fully reflects its emotional factors.
- (d) In the time series analysis of 4.5, we omit the data from 2002 to 2007, because the data is limited. But it cannot be denied that they are also of reference significance. There will be deviation in fitting the ARIMA time series model.
- (e) The 1-9 scaling method is not exactly correct. We have a certain subjectivity in assigning values to adjectives.
- (f) z_0 's definition is not good, the ScoreB3, we cannot easily figure out the relationship between the reference of the helpfulness and average scores of reviews.
- (g) In terms of corresponding from helpfulness to level of sales, the level 1 is the majority and it is over 80%.
- (h) The specific assignment of rate, review and helpfulness is subjective, and the influence of specific change on the weight vector is described in the sensitivity analysis.

5 Conclusion

In this paper, we first create a wealth profile by selecting and aggregating the variables in the provided data. Then we propose a framework FTN (a framework with fuzzy judgement by


```
freq_words = nltk.FreqDist(all_words)
freq_words_50 = freq_words.most_common(50)
print('Top 50 frequent words (excepting words in dump)\n',freq_words_50,'\n\n')
plt.figure(1)
plt.title('Top 20 frequent words in ' + product_name + ' review bodies')
freq_words.plot(20, cumulative=False) # close the plot window to continue execution

# use tag to find only adj
tagged_words = nltk.pos_tag(all_words)
adj_words = [word for (word, tag) in tagged_words if tag == 'JJ']
freq_adj = nltk.FreqDist(adj_words)
freq_adj_50 = freq_adj.most_common(50)
print('Top 50 frequent adjectives (excepting words in dump)\n',freq_adj_50)
plt.figure(2)
plt.title('Top 20 frequent adjectives in ' + product_name + ' review bodies')
freq_adj.plot(20, cumulative=False)

## score_pie.py
import nltk
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

product_name = "microwave"
#product_name = "pacifier"
#product_name = "hair_dryer"
file_name = product_name + ".tsv"

df = pd.read_csv(file_name, sep="\t")
scope_list = list(range(0,len(df.review_body),2)) ## use half data
#scope_list = list(range(0,len(df))) ## use all data
body = df.review_body[scope_list]
score_list = []

grade_words = ['great','happy','perfect','old','bad']

def word_to_grade(word):
    switcher = {
        'great':3 ,
        'perfect':3,
```

```
        'happy':2,
        'old':-1,
        'bad':-2
    }
    return switcher.get(word, 0)

print(product_name.title() + ' cumstomer scores statistics (every 2 count 1)\n')

def clac_star_score(i):
    if df.total_votes[i] <= 5:
        votes = df.total_votes[i]
    elif df.total_votes[i] <= 50:
        votes = 5 + 0.1 * (df.total_votes[i] - 5)
    else:
        votes = 5 + 0.1 * 45 + 0.01 * (df.total_votes[i] - 50)

    votes -= 1.0*(df.total_votes[i]-df.helpful_votes[i])
    if votes < 0:
        votes = 0

    score = 0.1 * df.star_rating[i] * votes
    return score

for i in scope_list:
    score = clac_star_score(i)
    #score = 0
    if not pd.isna(body[i]):
        tokens = [w.lower() for w in nltk.word_tokenize(body[i]) if w.lower() in grade_words]

    review_score = 0
    if tokens:
        for w in tokens:
            review_score += word_to_grade(w)
        if review_score < 0:
            review_score = 0
        review_score /= len(tokens)

    score += review_score
    score_list.append(score)
```

```
max_score = max(score_list)
print(product_name.title() + ' maximum score is: ',max_score)

partition = np.linspace(0,max_score,6)
partition_no = [0, 0, 0, 0, 0]

for i in range(5):
    for score in score_list:
        if partition[i] <= score and score < partition[i+1]:
            partition_no[i] += 1
            score_list.remove(score)

# plot pie chart
intervals = ['0~1/5', '1/5~2/5', '2/5~3/5', '3/5~4/5', '4/5~1'] # labels
colors = ['r', 'y', 'g', 'b', 'c'] # color for each label
plt.pie(partition_no, labels = intervals, colors=colors, autopct = '%1.1f%%')

plt.title(product_name.title() + ' maximum score %f' %max_score)
plt.legend()
plt.show()

## monthly_specific_word.py
import nltk
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
from datetime import datetime

#product_name = "microwave"
#product_name = "pacifier"
product_name = "hair_dryer"
file_name = product_name + ".tsv"

df = pd.read_csv(file_name, sep="\t")
scope_list = list(range(0,len(df.review_body),2)) ## use half data
#scope_list = list(range(0,len(df))) ## use all data
body = df.review_body[scope_list]

print(product_name.title() + " monthly average rating stars whose review words contain specific word 'good'/'bad' (every
2 count 1)\n")
```

```

monthly_total_stars = [0, 0]                # monthly number of rating stars that have word 'good'/'bad'
monthly_day_no = [0, 0]                    # monthly number of reviews that have word 'good'/'bad'
monthly_star_list = [[], []]
monthly_date_list = []
date_0 = datetime.strptime(df.review_date[0], '%m/%d/%Y')
for i in scope_list:
    if not pd.isna(body[i]):
        good = [w for w in nltk.word_tokenize(body[i]) if w.lower() == 'good']
        bad = [w for w in nltk.word_tokenize(body[i]) if w.lower() == 'bad']

    date = datetime.strptime(df.review_date[i], '%m/%d/%Y')
    if date.month == date_0.month and date.year == date_0.year:
        if good:
            monthly_total_stars[0] += df.star_rating[i]
            monthly_day_no[0] += 1
        if bad:
            monthly_total_stars[1] += df.star_rating[i]
            monthly_day_no[1] += 1
    else:
        for k in range(2):
            if monthly_day_no[k]:
                monthly_star_list[k].append(monthly_total_stars[k]/monthly_day_no[k])
            else:
                monthly_star_list[k].append(0)
        monthly_date_list.append(datetime(date_0.year, date_0.month, 15))
        monthly_total_stars = [0, 0]
        monthly_day_no = [0, 0]

    date_0 = date

plt.plot(monthly_date_list, monthly_star_list[0], label = "good")
plt.plot(monthly_date_list, monthly_star_list[1], label = "bad")
plt.title(product_name.title() + " reviews that contain specific word 'good'/'bad'")
plt.legend()
plt.xlabel("Year, Month (y, m)")
plt.ylabel("Monthly Average Star(s)")
plt.show()

## vine.py
import nltk
import pandas as pd
import matplotlib

```



```

import matplotlib.pyplot as plt

# 3-by-3 combinations

product_name = "microwave"
#product_name = "pacifier"
#product_name = "hair_dryer"
file_name = product_name + ".tsv"

df = pd.read_csv(file_name, sep="\t")
scope_list = list(range(0, len(df.review_body), 2))  ## use half data
#scope_list = list(range(0, len(df)))  ## use all data
body = df.review_body[scope_list]

#vine_type = "YN"
vine_type = "NY"
#vine_type = "NN"

if vine_type == "YN":
    vine_type_scope_list = [i for i in scope_list if df.vine[i] == 'Y' and df.verified_purchase[i] == 'N']
    vine_type_str = "vine members (YN)"
elif vine_type == "NY":
    vine_type_scope_list = [i for i in scope_list if df.vine[i] == 'N' and df.verified_purchase[i] == 'Y']
    vine_type_str = "none vine members verified (NY)"
else:
    vine_type_scope_list = [i for i in scope_list if df.vine[i] == 'N' and df.verified_purchase[i] == 'N']
    vine_type_str = "none vine members unverified (NN)"

print(product_name.title() + ' review body words of ' + vine_type_str + ' statistics (every 2 count 1)\n\n')

all_words = []
dump = [',', '!', '!', '!', '!', '!', '$', '?', '(', ')', '"', '"', '"', '"', '"', '"', '"', '<', 'br', '>',
        '+', '&', '#', '@', '-', '=', '~', '!', '*', '%', '^', '{', '}', '[', ']', '|', '\\']
dump_words = ['the', 'it', 'a', 'and', 'is', 'of', 'for', 'to', 'on', 'in', 'or', 'i']
dump += dump_words
print('Dumped words in list dump\n', dump, '\n\n')

for i in vine_type_scope_list:
    # may use ps.stem()
    if not pd.isna(body[i]):
        all_words += [w.lower() for w in nltk.word_tokenize(body[i]) if w.lower() not in dump]

```

```
#sorted_words = sorted(all_words,key=all_words.count,reverse=True)

freq_words = nltk.FreqDist(all_words)
freq_words_50 = freq_words.most_common(50)
print("Top 50 frequent words (excepting words in dump)\n',freq_words_50,'\n\n')

plt.figure(1)
plt.title("Top 20 frequent words of ' + vine_type_str + ' in ' + product_name + ' review bodies')
freq_words.plot(20, cumulative=False) # close the plot window to continue execution

# use tag to find only adj
tagged_words = nltk.pos_tag(all_words)
adj_words = [word for (word, tag) in tagged_words if tag == 'JJ']
freq_adj = nltk.FreqDist(adj_words)
freq_adj_50 = freq_adj.most_common(50)
print("Top 50 frequent adjectives (excepting words in dump)\n',freq_adj_50)

plt.figure(2)
plt.title("Top 20 frequent adjectives of ' + vine_type_str + ' in ' + product_name + ' review bodies')
freq_adj.plot(20, cumulative=False)

## import nltk
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

product_name = "microwave"
#product_name = "pacifier"
#product_name = "hair_dryer"
file_name = product_name + ".tsv"

df = pd.read_csv(file_name, sep="\t")
scope_list = list(range(0,len(df.review_body),2)) ## use half data
#scope_list = list(range(0,len(df))) ## use all data
body = df.review_body[scope_list]

print(product_name.title() + " rating stars distribution by specific word ('great'/'bad') statistics (every 2 count 1)\n\n")

# two list for storing star numbers for 'great', 'bad'
star_list = [[0, 0, 0, 0, 0],[0, 0, 0, 0, 0]]
for i in scope_list:
```

```
if not pd.isna(body[i]):
    great = [w for w in nltk.word_tokenize(body[i]) if w.lower() == 'great']
    bad = [w for w in nltk.word_tokenize(body[i]) if w.lower() == 'bad']

if great:
    for j in range(5):
        if df.star_rating[i] == j+1:
            star_list[0][j] += 1
            break
if bad:
    for j in range(5):
        if df.star_rating[i] == j+1:
            star_list[1][j] += 1
            break

## borrowed from <matplotlib.org>
labels = ['1 star', '2 stars', '3 stars', '4 stars', '5 stars']

x = np.arange(len(labels)) # the label locations
width = 0.35 # the width of the bars

fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, star_list[0], width, label="great")
rects2 = ax.bar(x + width/2, star_list[1], width, label="bad")

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Review Count')
ax.set_title(product_name.title() + " rating stars distribution by specific word ('great'/'bad')")
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()

def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3), # 3 points vertical offset
                    textcoords="offset points",
```

```

        ha='center', va='bottom')

autolabel(rects1)
autolabel(rects2)

fig.tight_layout()

plt.show()

import nltk
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
from datetime import datetime

product_name = "microwave"
#product_name = "pacifier"
#product_name = "hair_dryer"
file_name = product_name + ".tsv"

df = pd.read_csv(file_name, sep="\t")
scope_list = list(range(0, len(df.review_body), 2))    ## use half data
#scope_list = list(range(0, len(df)))                ## use all data
body = df.review_body[scope_list]

punc = ['!', '!', '!', '!', '!', '$', '?', '(', ')', '"', "'", ':', ';', ',', '/', '<', 'br', '>',
        '+', '&', '#', '@', '-', '=', '~', '^', '*', '%', '^', '{', '}', '[', ']', '|', '\\']

print(product_name.title() + ' monthly average review length statistics (every 2 count 1)\n')

#length_list = []          # review words length of each item in scope_list
monthly_length_list = []   # monthly average review words length
monthly_date_list = []
monthly_total_length = 0
monthly_day_no = 0
date_0 = datetime.strptime(df.review_date[0], '%m/%d/%Y')
for i in scope_list:
    words_no = 0
    if not pd.isna(body[i]):
        words_no = len([w for w in nltk.word_tokenize(body[i]) if w not in punc])

```

```

date = datetime.strptime(df.review_date[i], '%m/%d/%Y')
if date.month == date_0.month and date.year == date_0.year:
    monthly_total_length += words_no
    monthly_day_no += 1
else:
    monthly_length_list.append(monthly_total_length/monthly_day_no)
    monthly_date_list.append(datetime(date_0.year, date_0.month, 15))
    monthly_total_length = 0
    monthly_day_no = 1

date_0 = date

plt.plot(monthly_date_list, monthly_length_list, '-b')
plt.title(product_name.title() + ' monthly average review_body words length')
plt.xlabel('Year, Month (y, m)')
plt.ylabel('Monthly Average Length (words)')
plt.show()

load('hair_dryer.mat', 'hairdryer');
%load('microwave.mat', 'microwave');
%load('pacifier.mat', 'pacifier');
%product_name = pacifier; product_str = 'Pacifier';
%product_name = microwave; product_str = 'Microwave';
product_name = hairdryer; product_str = 'Hairdryer';

% only use date after 2008
idx_since_2008 = find(product_name.review_date.Year >= 2008);
product_name = product_name(idx_since_2008,:);

product_name_no = size(product_name, 1);
stars_no = zeros(5, 1);
s_1 = find(product_name.star_rating == 1); s_1_ratio = length(s_1)/product_name_no
s_2 = find(product_name.star_rating == 2); s_2_ratio = length(s_2)/product_name_no
s_3 = find(product_name.star_rating == 3); s_3_ratio = length(s_3)/product_name_no
s_4 = find(product_name.star_rating == 4); s_4_ratio = length(s_4)/product_name_no
s_5 = find(product_name.star_rating == 5); s_5_ratio = length(s_5)/product_name_no
vine_no = length(find(product_name.vine == 'Y' | product_name.vine == 'y'));
vine_ratio = vine_no/product_name_no
verified_no = length(find(product_name.verified_purchase == 'Y' | product_name.verified_purchase == 'y'));
verified_ratio = verified_no/product_name_no

```

```

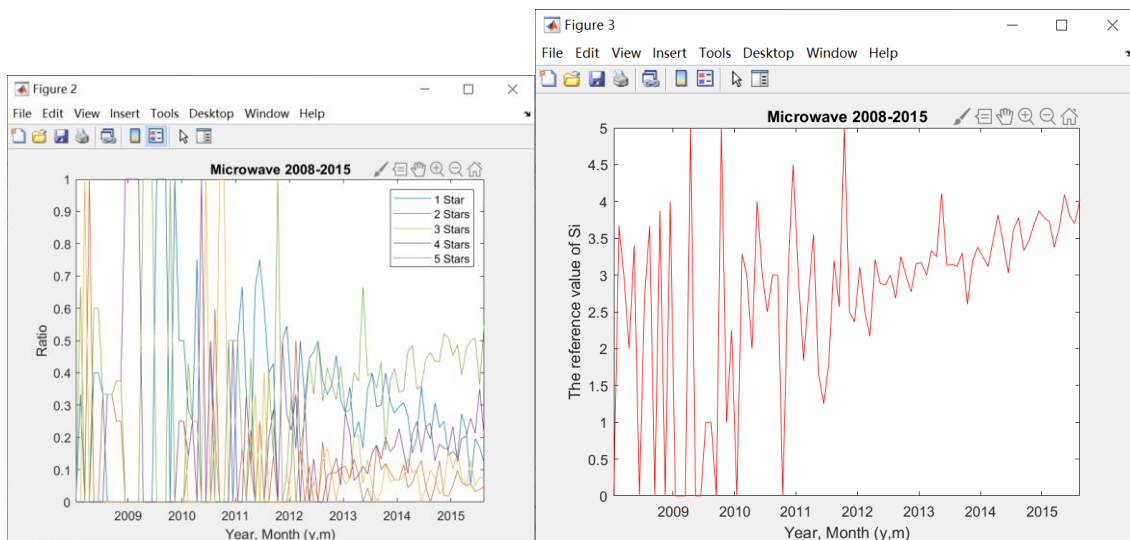
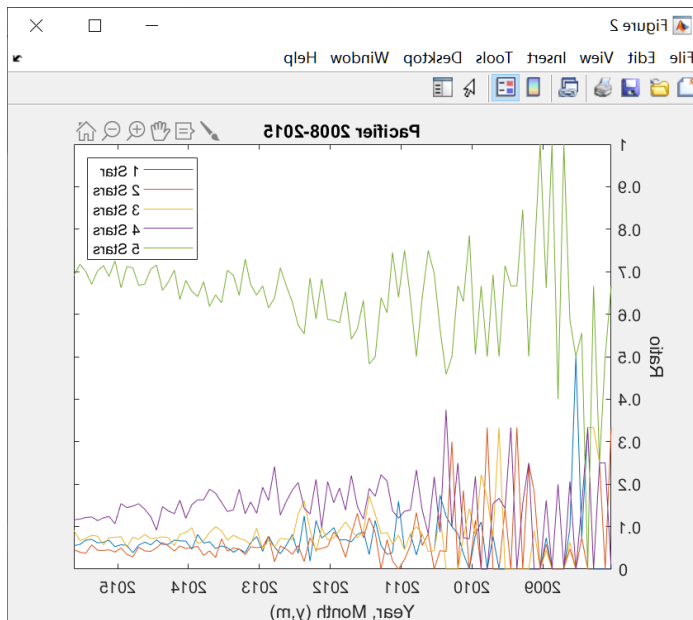
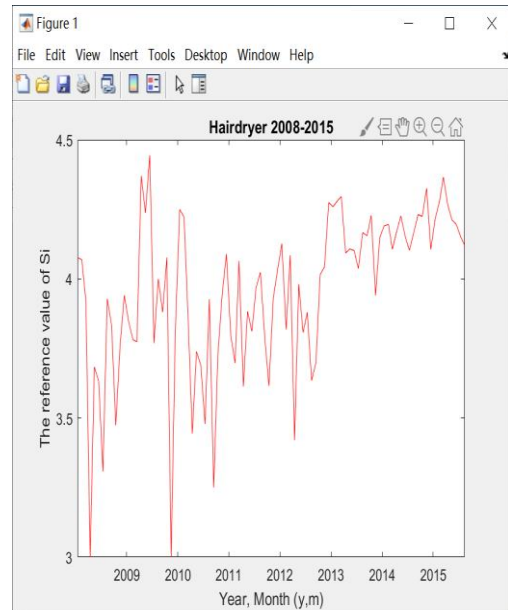
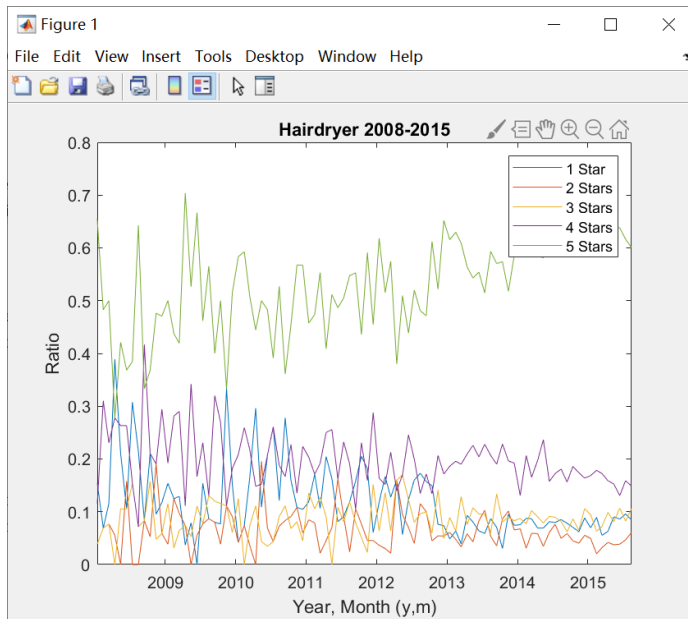
% do monthly stars rating statistic
monthNumber=(2015-2008+1)*12-4;
monthArray=product_name.review_date(1:monthNumber);
aveMatrix=zeros(monthNumber,5);
i = 1;
r=zeros(5,1);
ave=zeros(1,5);
s=zeros(5,1);
for y=2008: 2015
    iy=find(product_name.review_date.Year==y);
    if y==2015
        n_m=8;
    else
        n_m=12;
    end
    for m=1:n_m
        im= find(product_name.review_date.Month(iy)==m)+iy(1)-1; % index of month
        if ~isempty(im)
            for k=1:5
                imk=find(product_name.star_rating(im)==k)+im(1)-1; % index of monthly rating of k
                ave(k)=length(imk)/length(im);
            end
        end
        monthArray(i)=datetime(y,m,15);
        aveMatrix(i,:)=ave;
        i = i + 1;
    end
end
figure
plot(monthArray,aveMatrix)
legend('1 Star','2 Stars','3 Stars','4 Stars','5 Stars')
xlabel('Year, Month (y,m)')
ylabel('Ratio')
title([product_str, ' 2008-2015'])

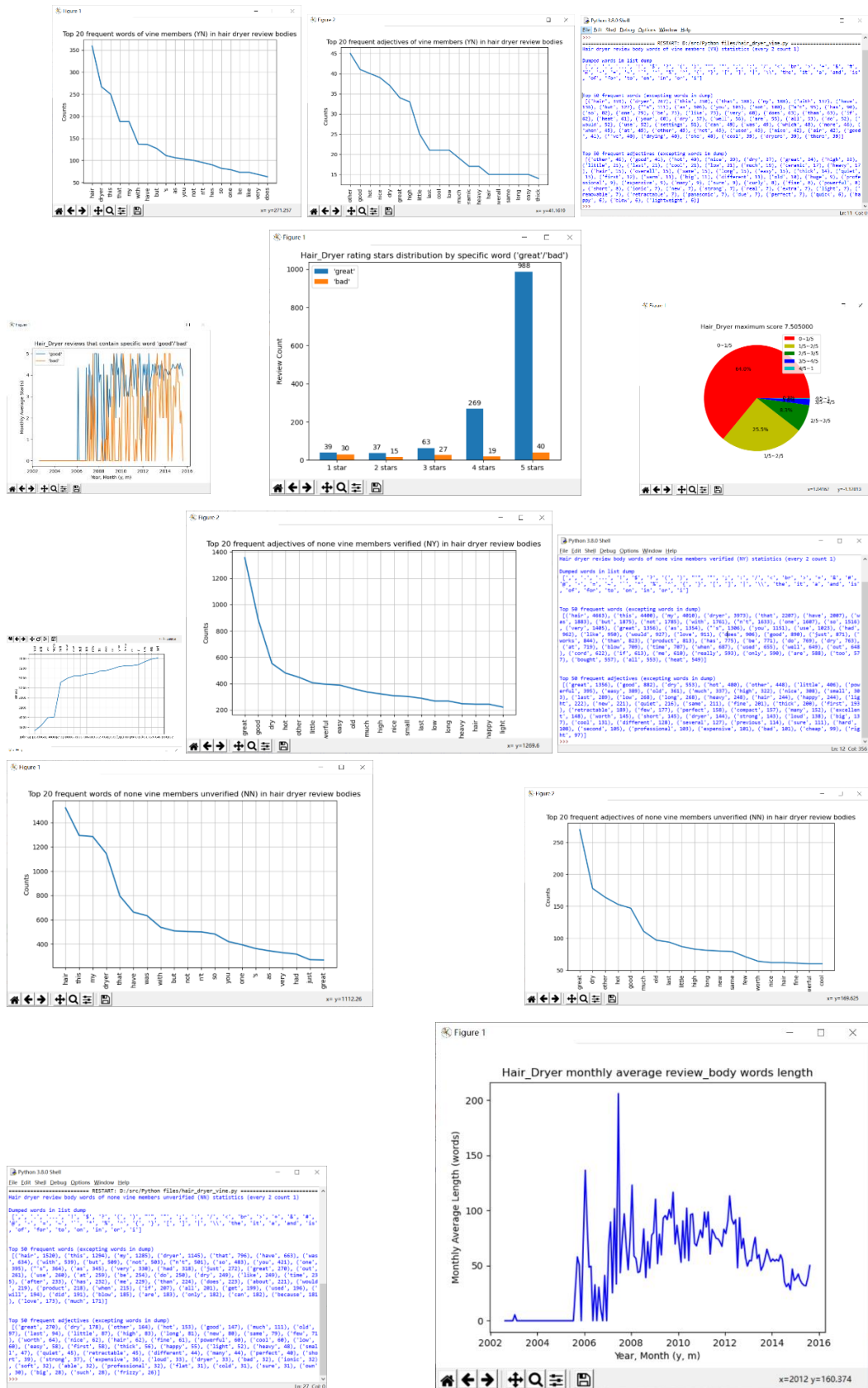
%load('hair_dryer.mat','hairdryer');
load('microwave.mat', 'microwave');
%load('pacifier.mat','pacifier');
%product_name = pacifier; product_str = 'Pacifier';
product_name = microwave; product_str = 'Microwave';

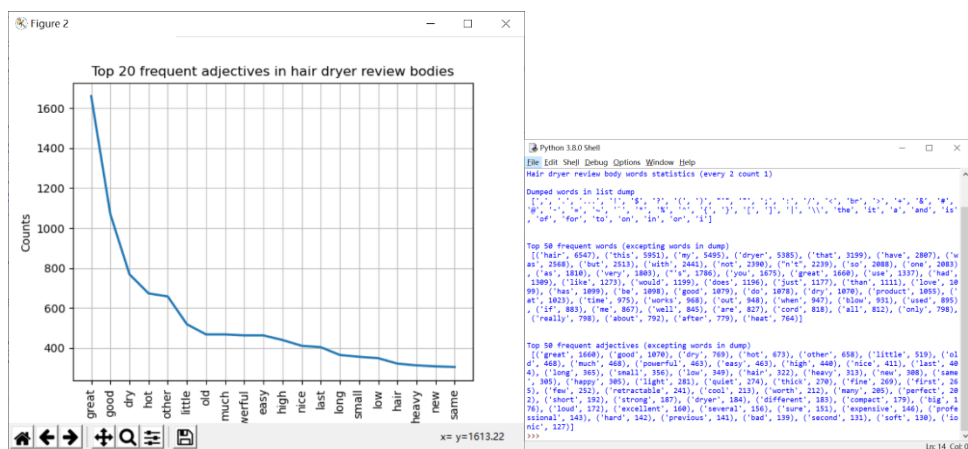
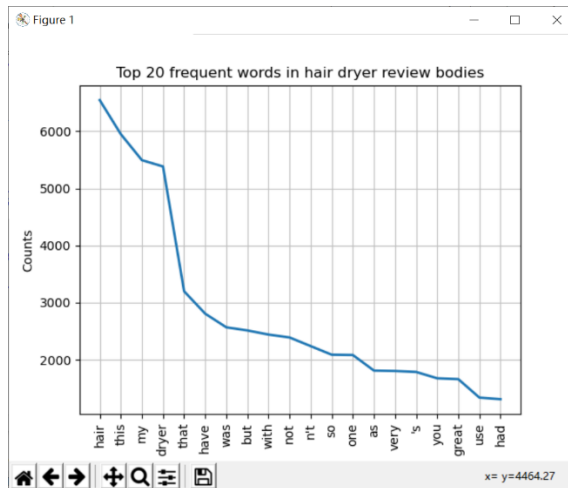
```

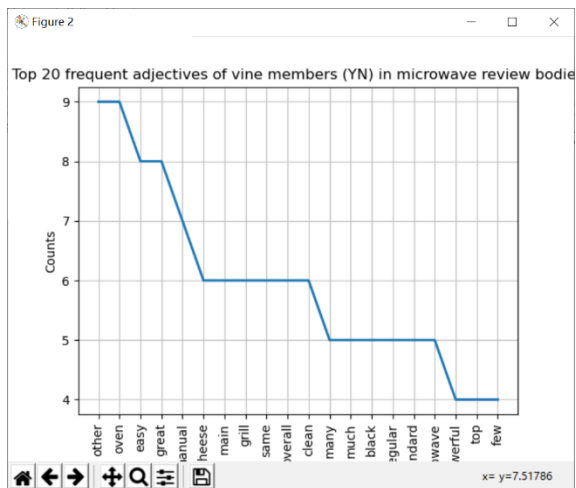
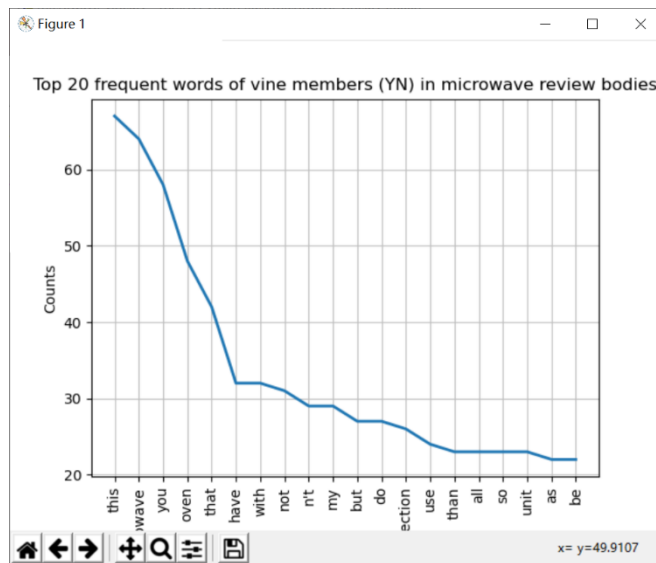
```
%product_name = hairdryer; product_str = 'Hairdryer';

monthNumber=(2015-2008+1)*12-4;
monthArray=product_name.review_date(1:monthNumber);
aveArray=zeros(monthNumber,1);
i = 1;
for y=2008: 2015
    iy=find(product_name.review_date.Year==y);
    if y==2015
        n_m=8;
    else
        n_m=12;
    end
    for m=1:n_m
        im=find(product_name.review_date.Month(iy)==m);
        s=sum(product_name.star_rating(iy(1)+im-1));
        n = length(im);
        if n>=1
            ave=s/n;
        else
            ave = 0;
        end
        monthArray(i)=datetime(y,m,15);
        aveArray(i)=ave;
        i = i + 1;
    end
end
end
figure
plot(monthArray,aveArray,'-r')
xlabel('Year, Month (y,m)')
ylabel('The reference value of Si')
title([product_str, ' 2008-2015'])
```









Python 3.8.0 Shell

File Edit Shell Debug Options Window Help

```
>>>
===== RESTART: D:/src/Python files/microwave_vine.py =====
Microwave review body words of vine members (YN) statistics (every 2 count 1)

Dumped words in list dump
[...]
```

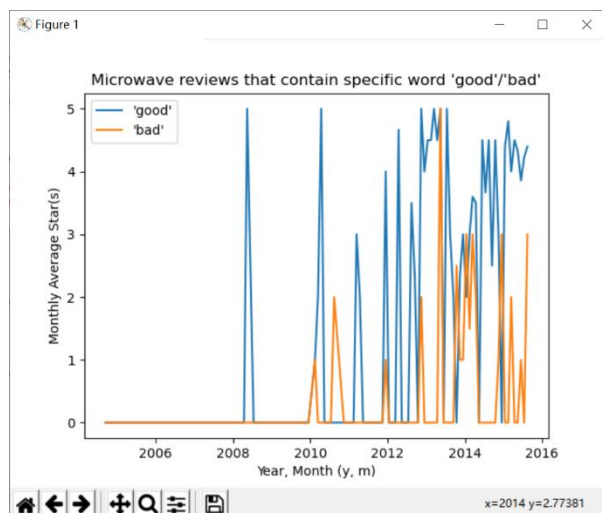
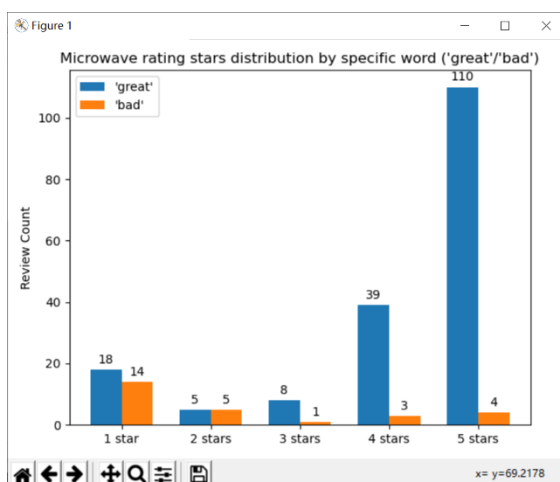
Top 50 frequent words (excepting words in dump)

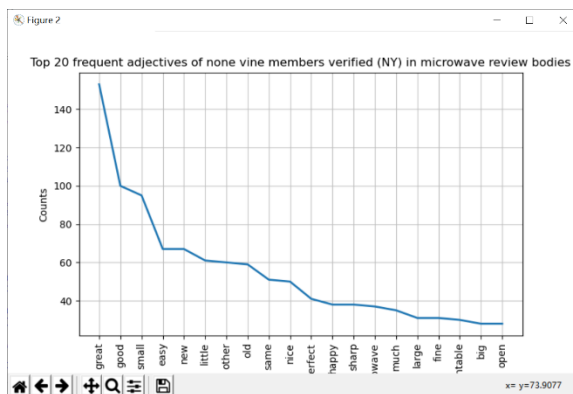
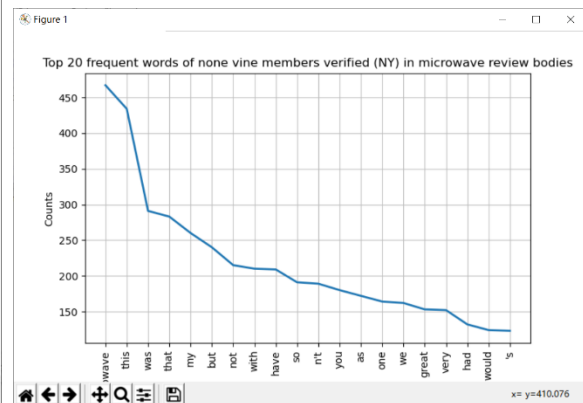
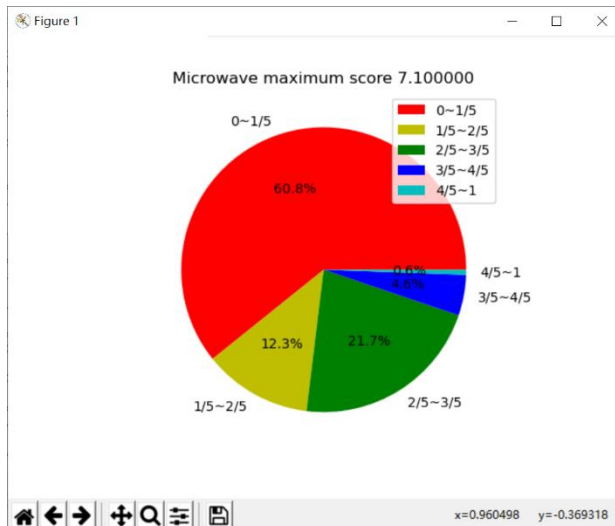
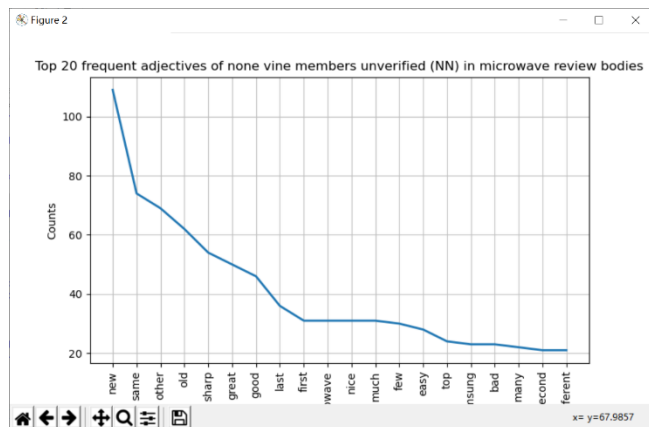
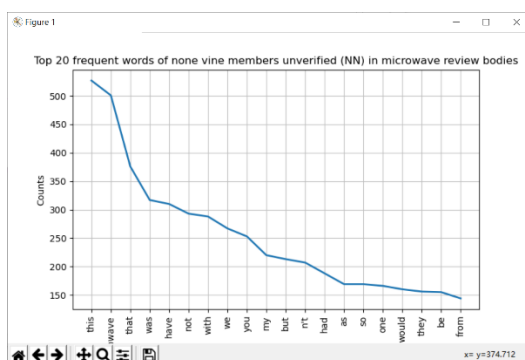
```
[('this', 189), ('microwave', 186), ('you', 97), ('that', 75), ('oven', 66), ('but', 51), ('not', 49), ('width', 48), ('have', 48), ('as', 45), ('top', 43), ('my', 42), ('n't', 41), ('use', 40), ('cook', 36), ('are', 33), ('than', 33), ('s', 32), ('do', 31), ('be', 30), ('does', 30), ('time', 30), ('we', 30), ('grill', 29), ('can', 29), ('all', 29), ('convection', 29), ('unit', 28), ('when', 26), ('samsung', 25), ('like', 24), ('up', 23), ('more', 23), ('has', 22), ('one', 22), ('if', 21), ('use d', 21), ('would', 21), ('at', 20), ('there', 20), ('was', 20), ('very', 20), ('am', 19), ('well', 19), ('function', 19), ('frozen', 19), ('cooking', 19), ('will', 19), ('using', 18), ('just', 18)]
```

Top 50 frequent adjectives (excepting words in dump)

```
[('frozen', 16), ('great', 14), ('manual', 12), ('other', 11), ('nice', 10), ('good', 10), ('easy', 10), ('much', 10), ('grill', 9), ('overall', 9), ('oven', 9), ('microwave', 8), ('main', 8), ('regu lar', 7), ('first', 7), ('same', 7), ('slim', 7), ('high', 6), ('standard', 6), ('simple', 6), ('che ese', 6), ('clean', 6), ('hot', 6), ('large', 5), ('sure', 5), ('perfect', 5), ('powerful', 5), ('ma ny', 5), ('black', 5), ('able', 5), ('little', 5), ('fine', 4), ('cold', 4), ('popcorn', 4), ('stain less', 4), ('element', 4), ('top', 4), ('few', 4), ('outside', 4), ('open', 4), ('true', 4), ('frenc h', 4), ('ready', 4), ('plastic', 3), ('real', 3), ('past', 3), ('such', 3), ('cubic', 3), ('conveni ent', 3), ('old', 3)]
```

Ln: 24 Col: 0



[illegible]

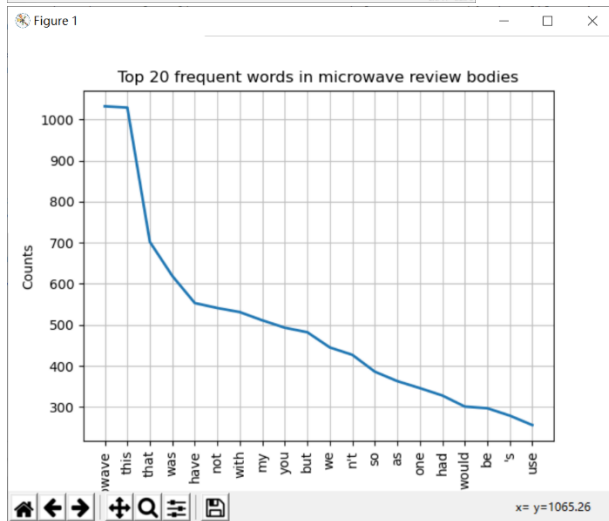
```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
===== RESTART: D:\src\Python files\microwave_vine.py =====
Microwave review body words of none vine members unrefined (M) statistics (every 2 count 1)

Dumped words in list dump
['the', 'is', 'a', 'an', 'and', 'or', 'of', 'for', 'to', 'on', 'in', 'on', 'i']

Top 50 frequent words (excepting words in dump)
[('this', 327), ('microwave', 301), ('that', 276), ('was', 317), ('have', 318), ('not', 293), ('it', 288), ('we', 267), ('you', 253), ('my', 238), ('but', 213), ('n't', 207), ('had', 188), ('as', 169), ('so', 169), ('one', 166), ('would', 160), ('they', 156), ('we', 155), ('from', 144), ('at', 135), ('n', 135), ('just', 134), ('do', 131), ('after', 129), ('will', 129), ('unit', 127), ('if', 123), ('no', 121), ('has', 120), ('use', 117), ('years', 116), ('out', 116), ('me', 115), ('when', 114), ('service', 113), ('are', 111), ('p', 111), ('new', 108), ('s', 108), ('all', 103), ('oven', 98), ('about', 99), ('door', 97), ('only', 96), ('time', 96), ('samsung', 96), ('our', 93), ('buy', 91), ('can', 89)]

Top 50 frequent adjectives (excepting words in dump)
[('new', 189), ('same', 74), ('other', 69), ('old', 62), ('sharp', 54), ('great', 50), ('good', 46), ('last', 36), ('first', 31), ('microwave', 31), ('nice', 31), ('much', 31), ('few', 30), ('easy', 28), ('top', 24), ('samsung', 23), ('bad', 23), ('many', 22), ('second', 21), ('different', 21), ('g', 20), ('several', 20), ('open', 20), ('fine', 19), ('able', 18), ('sure', 18), ('turntable', 18), ('small', 18), ('expensive', 17), ('whole', 17), ('frozen', 15), ('large', 15), ('high', 15), ('loc', 14), ('oven', 14), ('short', 14), ('popcorn', 14), ('whirlpool', 14), ('hot', 14), ('loud', 14), ('light', 13), ('bottom', 13), ('little', 13), ('previous', 12), ('happy', 12), ('huge', 12), ('available', 12), ('big', 12), ('full', 12), ('front', 12)]

Ln: 17 Col: 0
```



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
===== RESTART: D:\src\Python files\microwave_body.py =====
Microwave review body words statistics (every 2 count 1)

Dumped words in list dump
['the', 'is', 'a', 'an', 'and', 'or', 'of', 'for', 'to', 'on', 'in', 'on', 'i']

Top 50 frequent words (excepting words in dump)
[('microwave', 1032), ('this', 1029), ('that', 702), ('was', 619), ('have', 553), ('not', 541), ('with', 531), ('my', 511), ('you', 493), ('but', 482), ('we', 445), ('n't', 427), ('so', 386), ('as', 363), ('one', 346), ('had', 328), ('would', 301), ('be', 297), ('s', 279), ('use', 256), ('just', 246), ('do', 242), ('very', 233), ('unit', 233), ('at', 233), ('from', 230), ('if', 222), ('when', 221), ('after', 219), ('are', 219), ('they', 219), ('will', 218), ('no', 217), ('has', 215), ('great', 211), ('than', 207), ('all', 207), ('only', 206), ('door', 206), ('time', 202), ('oven', 199), ('out', 197), ('like', 197), ('me', 188), ('well', 183), ('up', 181), ('can', 181), ('new', 176), ('our', 175), ('more', 172)]

Top 50 frequent adjectives (excepting words in dump)
[('great', 211), ('new', 176), ('good', 149), ('other', 138), ('same', 131), ('old', 124), ('small', 115), ('easy', 103), ('sharp', 93), ('nice', 83), ('little', 76), ('microwave', 72), ('much', 71), ('first', 61), ('few', 59), ('last', 57), ('top', 53), ('happy', 53), ('turntable', 51), ('fine', 51), ('perfect', 50), ('open', 50), ('many', 48), ('large', 47), ('sure', 44), ('second', 43), ('big', 41), ('table', 39), ('several', 39), ('different', 37), ('bad', 33), ('full', 33), ('clean', 32), ('previous', 32), ('fit', 31), ('samsung', 31), ('oven', 31), ('hot', 30), ('whole', 29), ('powerful', 29), ('high', 28), ('ge', 28), ('light', 27), ('simple', 27), ('right', 27), ('expensive', 27), ('interior', 26), ('stainless', 26), ('low', 24), ('frozen', 24)]

>>>

Ln: 12 Col: 723
```

