Problem Chosen	2020 MCM/ICM Summary Sheet	Team Control Number
C		**2002600**

PPA: A Marketing Strategy Framework For Optimizing Product Scores Based on Reviews and Ratings

Summary

With the expansion of online shopping market, star ratings and reviews are taking account for a larger portion determining the consumer's purchasing decision. These changes encourage companies to attach greater attention to these informative data on the market while making business decisions. We aimed to help Sunshine Company hold a better understanding of market information, and enhance product desirability through some efficient online sales strategy.

To address the problem, we build a **Marketing Strategy Framework PPA** for optimizing product scores based on reviews and ratings. In the first part, we analyze the relationship between and within star ratings, reviews and helpfulness ratings using **Spearman Correlation Coefficient**. We apply the Natural Language API to calculate the sentiment score based on the textual user-review. We also purpose an important indicator: **Rating Inconsistency** to reflect the specific pattern within rating behaviour. These measures contributed to our insight into one reasonable combinations: User-based Product Score. The score numerically reflect the evaluation of a single product in the Pre-purchasing Evaluation stage.

In the Post-purchasing Evaluation stage, we make a **time-series analysis** on user behavior and we define the reputation of a given product. We regard reviews as point processes and construct a **Reactive Marked Point Process Model** to evaluate the arrivals of reviews under the influences of star ratings and helpfulness rates.

In the Analysis of Market Performance stage, we further discuss how to devise the best strategy with all the indicators and methods we have purposed. We integrate and refine them all into a Comprehensive Score through **Entropy Model**. We formulate a method to find the Boom Point and Bust Point, which are reviewing events where strong effects act on the User-based Product Score. We made an entity recognition on textual reviews, and provided strategies optimizing scores and maximizing sales for the companies.

Finally, we change the numerical value of θ_0 to examine the sensitivity of our Comprehensive Score model with well interpretation, we also carry out some meaningful instructions for marketing decision and discuss strengths and weaknesses of our model.

Keywords: Sentiment Analysis, Data Mining, Hawkes Process, Entity Recognition

PPA: A Marketing Strategy Framework For Optimizing Product Scores Based on Reviews and Ratings

March 10, 2020

Table of Contents

1 Introduction

1.1 Problem Statement

With the rapid growth of online shopping platform and the internet, the rating system that let users post reviews of products has taken on a larger role in consumers' Web-based purchasing decisions. Not only are more consumers contributing their ideas, but potential buyers are also increasing relying on the information provided by others in these places[1].

As a marketing manager, the ability to understand and track users' preferences and consumption patterns is a necessity for attaining sales victory over the opponents on the market[2]. Currently, the world's largest e-commerce platform Amazon has provided three indicators to collect consumers' feedback, creating a perfect opportunity and platform for managers to process market information dynamic and in the individual level. We aim to design a framework that gives sales-driven marketing strategies for the companies to craft successful and popular products on the market.

1.2 Planned Approach

Since the product sale is an undeniable goal for a company[3], our objective is to set out the best marketing strategy that could maximize the product sale in the online marketplaces. We propose a novel framework named PPA, which is shown in Figure 1. PPA includes three main stages:

1. Pre-purchasing Evaluation: We create a consumer profile for each consumer or potential buyer, and then we define a user-based product score to show the numerical evaluation of a product based on its ratings and reviews.

2. Post-purchasing Evaluation: We utilize the time-series regression analysis on product sales, then we construct a Reactive Marked Point Process Model for analysis.

3. Analysis of Market Performance: We formulate Comprehensive Score for any product concerning the factors from reviews and market change. We make an entity recognition on textual reviews, and provide strategies optimizing scores and maximizing sales for the companies.

With the help of PPA Framework, we solve tasks listed as follows:

Task 1:

We analyze the relativity among helpfulness ratings, star ratings and reviews.The star ratings are studied in the form of rating inconsistency factor. And the reviews are divided into the word length and the sentiment scores. Details in Section 4.1

Task 2:

a. We build a User-Based Product Score based on user recognition. Details in Section 4.3.1

b. We define the Product Reputation Model. Details in Section 4.3.1

c. We integrate all the indicators from task 2.a, 2.b and 2.d into one Comprehensive Score, which can best describe and predict the future product sales. Details in Section 4.4.3

d. We construct a newly-defined reactive market point process to describe the influence process that the star ratings have an impact on the arrival of the review event. Details in Section 4.4.1

e. We apply entity analysis to the words that best illustrate the information conveyed by a review. Details in Section 4.4.3

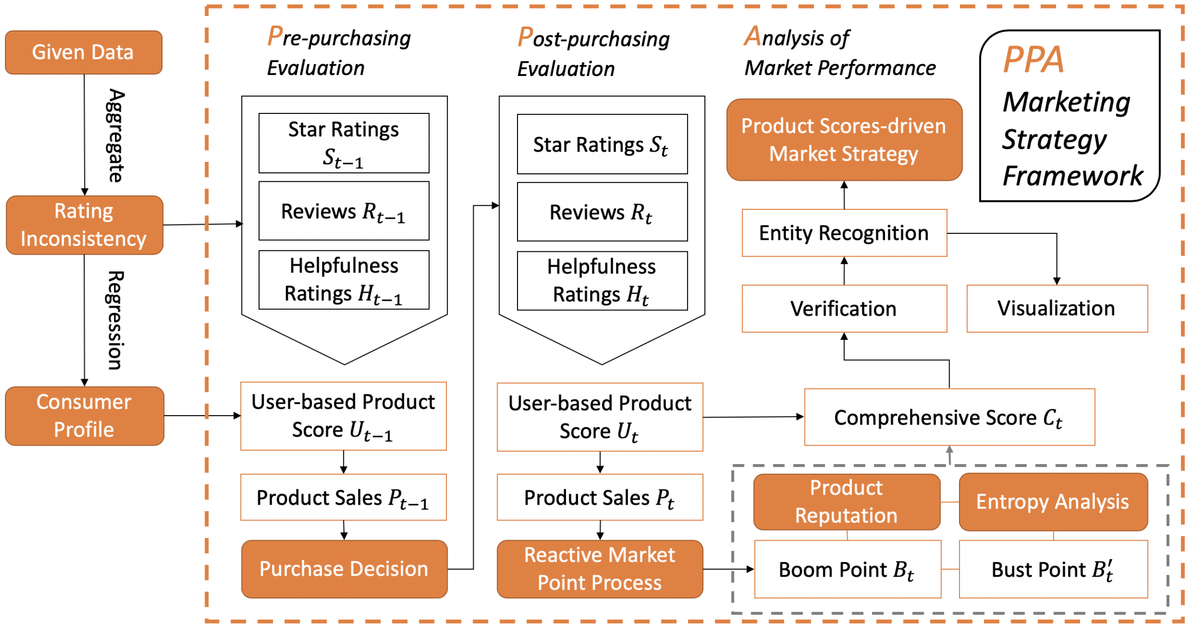


Figure 1: Flow Chart Demonstration of the Model

2 Assumptions

First and foremost, we make some basic assumptions and explain their rationales.

Assumption 1. *Product Sales are proportional to the number of Reviews of that specific product.*

Since we don't have sales data, this assumption is the premise of our work to measure the market performance of the product, and then our purposed targets and strategies will make sense.

Assumption 2. *When consumers have a demand to buy, they will search and compare several options for reference, and buy the final products after a self-evaluation.*

Every customer might think differently regarding the same product, but better products eventually are much more favored and enjoy the higher star ratings.

Assumption 3. *The provided data is reliable and realistic that only reflects the customers' experience with the product independent from their emotional status.*

By saying the data is reliable and realistic, we mean that all the data should be brought out from the rational people whose purchasing decisions will not be manipulated by product-irrelevant issues.

3 Abbreviations and Definitions

For compactness, we define a series of abbreviations for some notions concerning customer profile in Table 1.

Abbreviations	Definitions	Abbreviations	Definitions
B	Boom Point	HV	Helpful Votes
S	Star Ratings	SR	Sentiment Score
C	Comprehensive Score	TV	Total Votes
D	Decaying Effect	RI	Rating Inconsistency
H	Helpfulness Ratings	$MAGR$	Market Average Growth Rate
N	Number of Reviews	$PAGR$	Product Average Growth Rate
P	Product Sales		
U	User-based Product Score		

4 PPA: A Marketing Strategy Framework

4.1 Patterns and Relationships Study within Three Indicators

We investigate the relationships between and within the star ratings, reviews and helpfulness ratings. We will introduce the patterns of either combination separately. The justification would lay a solid foundation for the construction of comprehensive score C model.

4.1.1 Reviews

Since reviews are text-based data which contain more information than regular numeric data, we make efforts to mining the hidden data of various aspects. Our approach is to divide the reviews into three parts. The first is classifying customer sentiment. The second is assessing quality and reliability of reviews, and the third is recognizing useful content and feedback of the products.

4.1.2 Star Ratings and Reviews

As we have inferred above, the information of the reviews we focus on in the study of star ratings and reviews is the sentiment conveyed by the customers. To quantify the sentiment in the textual reviews, we adopt Natural Language API using the Google Cloud SDK, which allows us to apply the sentiment analysis on our product reviews, and identifies the prevailing emotional opinion within the text, especially to determine a review's attitude as positive, negative, or neutral[1].

In each of the three different data-sets (microwave oven, baby pacifier, and hair dryer), we randomly selected two sample sets, with 300 groups of data for every sample. We receive outputs in the form of sentiment score numbers for the corresponding text between -1 and 1.

Since the star ratings are measured in an one to five scale, we want to normalize these attributes to the common scale the same as our sentiment scores. We define the normalized star rating as follows.

$$S' = 2\,\frac{S - \min S}{\max S - \min S} - 1 \in [-1, 1] \tag{1}$$

Next, we compare the relation between the normalized star ratings and sentiment scores.

In Figure 2, the x axis represents normalized star ratings and the y axis represents sentiment scores. We notice that they might be positively correlated, which sparks us to use Spearman's rank order correlation coefficient to validate the relation. Specifically, assume that we measure two traits X and Y on each of n subjects. Let x_i be the rank of the measurement of X taken on the ith individual, y_i being defined similarly. Identical values are assigned a rank equal to the average of their positions in the ascending order of the values. Then average ranks $\bar{x}$ and $\bar{y}$ are equal to $(n + 1)/2$ and

[1]https://cloud.google.com/natural-language

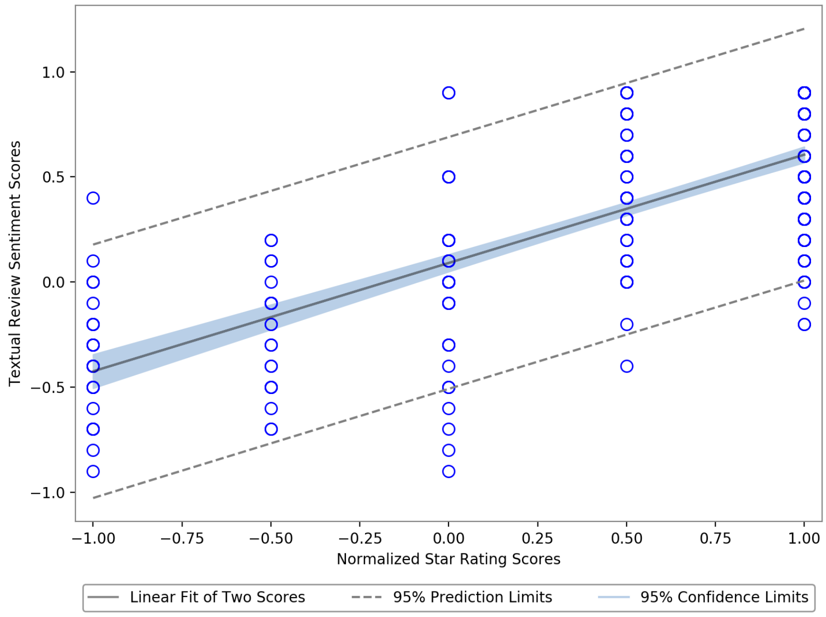


Figure 2: Linear Fit of Sentiment Scores and Normalized Star Ratings

$$r_{\mathrm{S}}=\frac{\sum_{i=1}^{n}\left\{\left(x_{i}-\bar{x}\right)\left(y_{i}-\bar{y}\right)\right\}}{\sqrt{\sum_{i=1}^{n}\left(x_{i}-\bar{x}\right)^{2}} \sqrt{\sum_{i=1}^{n}\left(y_{i}-\bar{y}\right)^{2}}} \tag{2}$$

This formulation will yield a value $-1 \leq r_S \leq 1$. The larger value suggests a stronger association between X and Y, whereas negative values. The positive values suggest a positive correlation, where negative values suggest that higher values of one are associated with lower values of the other. The null hypothesis is that the Spearman correlation coefficient is 0. The hypothesis means that the ranks of one variable do not covary with the ranks of the other variable.

Our data are illustrated in the table below.

Product	Hair dryer		Microwave		Pacifier	
Sample	1	2	1	2	1	2
Spearman's r_s	0.70654	0.74069	0.71158	0.70628	0.74115	0.71223
p value	1.63E-46	3.64E-95	1.38E-89	6.99E-97	2.67E-94	1.5E-60

Table 1: Correlation Analysis of Review's Sentiment Score and Star Rating

We can find from the table that the Spearman correlation coefficient is positive and all larger than 0.7. The p values are far smaller than 0.05. So we have strong evidence to reject the null hypothesis and we can figure out that the star ratings have a strong positive correlation between the sentiment score of the reviews. Therefore, with the mathematical evidence shown above, we can treat the star ratings as a reflection of the sentiment in the corresponding reviews.

4.1.3 Helpfulness Ratings, Star Ratings and Reviews

Online consumer review, a form of electronic word of mouth (eWOM), draws particular attention because of its effect on the purchasing decision of consumers[4]. And the most important factor in eWOM adoption is information credibility[5]. Therefore, an online retail market should provide credible consumer reviews to achieve continued success. In our model, the information credibility of review is indicated by the helpfulness ratings. The helpfulness can be studied in different aspects including the review word length, the star ratings and the sentiment of the review text[4].

First we give a formal definition of helpfulness ratings:

$$H_{ij} = \frac{HV_{ij}}{TV_{ij}} \tag{3}$$

For a product i and a review j, the helpfulness rating H is the ratio of helpful votes and total votes. That is the proportion of the helpful votes among the total votes Then we define a new variable Rating Inconsistency to show the bias of one specific star rating according to the mean of the rating number during one time period

$$RI_{ij} = \frac{S_{ij\tau} - \overline{S_{i\tau}}}{\overline{S_{i\tau}}} \tag{4}$$

$S_{ij\tau}$ is the star ratings for review j and product i during during time period τ, and $\overline{S_{i\tau}}$ is the mean value of the star rating for product i during a time period τ. In the investigation, we use the data set of hair dryer from 2002 to 2015. The data of which the total vote for the review helpfulness is less than 5 are dropped to keep the helpfulness rate more informative and reliable. We directly plot their relationships and apply a polynomial fit in red line, just for better illustrating the trend in the data, and not for investigating any numeric results. Figure 3 shows the relationship of helpfulness ratings with the word length and with the rating inconsistency.

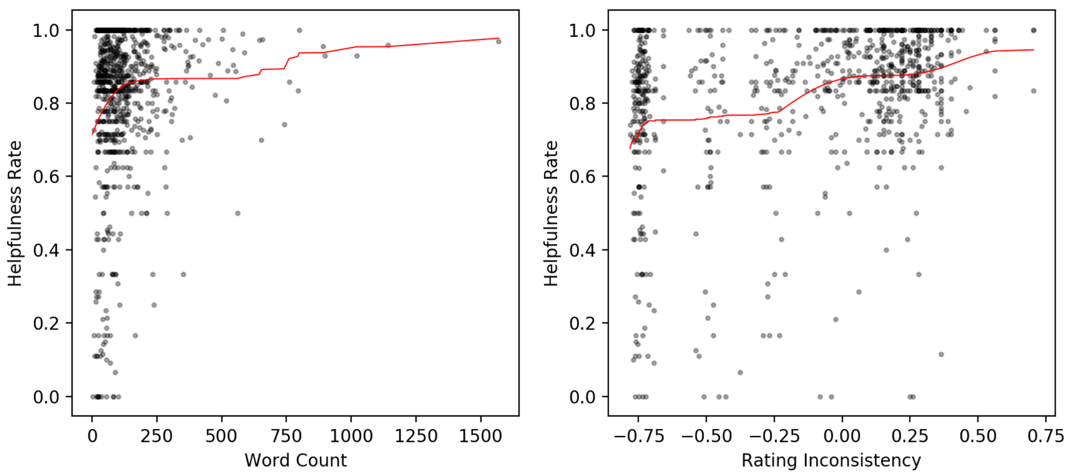


Figure 3: Helpfulness Ratings Relationship plot

We can conclude some quantitative results from the plot that:

1. There exists a nonlinear relationship between word count and helpfulness. Helpfulness increases when the number of words increases if number of words is comparatively small.

When the number of words of a review reaches about 1,000 to 1,500 words, helpfulness increases little. There exists a logarithmic relationship between word count and helpfulness.

2. When the rating inconsistency is positive, the helpfulness rates tend to decrease when the star rating from a review is leaving the average.

3. When the rating inconsistency is negative, the helpfulness rates increase with the star rating of a review approaching the average ratings.

4.2 Pre-purchasing Evaluation

4.2.1 Purchase Decision Model

A challenge in providing the information of a product based on its ratings and reviews is the highly plausible endogenous relationship between the two. That is, a product may show higher ratings not necessarily because of the positive reviews appealing to consumers, but rather due to some other objective factors. In this section, the Pre-purchasing Evaluation depicts a formula that combines the star ratings and sentiment scores of reviews. The two factors are extracted from much of the possible factors and are evaluated as User-based Product Score of the Pre-purchasing Evaluation, which works for the Consumer Purchase Decision Model.

In order to provide enough granularity to quantify the User-based Product Score, we separate the time into 20 periods. To help readers understand the following formulas better, index abbreviations together with their according definitions is provided in Table 2. In each time periods, we

Index Abbreviation	Definitions
τ	Time Period Index
i	Product ID Index
j	Comment Index

Table 2: The clarification of the usage of subscript

want to build a function $U_{i\tau}$ that combines star ratings score of product ID i within time period τ and the score of the product reviews i within time period τ, i.e,

$$U_{i\tau} = f(S_{i\tau}, R_{i\tau}) \tag{5}$$

In view of the score of star ratings of product i, we initially design a simple averaging method:

$$S_{i\tau} = \frac{\sum(S_{ij\tau} - \mu_S)}{N_{i\tau}} \tag{6}$$

where μ_s is the mean value of star ratings.

The case becomes more complex when we want to quantify the review score. The Sentiments of a review and how a review is influential can be two variables that helps a consumer to judge whether he/she will buy the product. When considering the influence of a review, we use total Votes TV and Helpful Votes HV to build a amplification factor function:

$$\gamma_{ij\tau} = \frac{HV_{ij\tau}}{1 + TV_{ij\tau}} \cdot \ln(2 + TV_{ij\tau}) \tag{7}$$

where $\gamma_{ij\tau}$ is the $\gamma_{ij\tau}$ amplification factor of the j^{th} review of product ID i within time period τ, λ is a constant.

The amplification function has larger value when the help votes account for a larger slice of total votes. Besides, the total number of votes has a more significant influence on consumers' purchasing decision.

Combining with the sentiment score done in previous work, we get an overall review score by multiplying sentiment score done in previous work with the :

$$R_{i\mathcal{T}} = \frac{\sum SR_{ij\mathcal{T}} \cdot \gamma_{ij\mathcal{T}}}{N_{i\mathcal{T}}} \tag{8}$$

We defined a weighted function for constructing the function f, which weights Star Rating Score and review score and gives a final User Based Score based on user recognition:

$$U_{i\mathcal{T}} = \theta_0 S_{i\mathcal{T}} + (1 - \theta_0) R_{i\mathcal{T}} \tag{9}$$

4.2.2 Justification of User-based Product Score

User-based Product Score provides a quantitative evaluation of a product based on reviews and star ratings based on the users. By Calculating User-based Product Score, we get the scores of each time period τ of specific product. However, we need to justify our User-based Product Score model. That is, we need to figure out whether there is a connection between Star Rating Score (Equation.5) and Review Score (Equation.7).

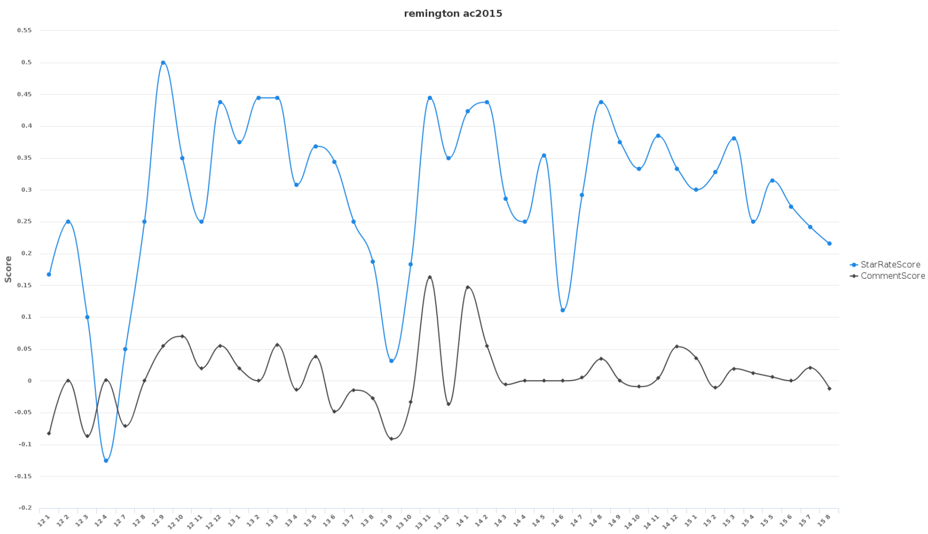


Figure 4: Comparison between Review Score and Star Rate Score of the top selling dryer product

In Figure 4, we choose the top-selling hair dryer product on Amazon from 2012-01 to 2015-08, with product title Remington ac2015, to show the comparison between Review Score and Star Rate Score. We can figure out the two scores have good consistency, while some outliers may occur. In this case, the outliers are the points at 2012-4 and 2013-9. This provided us with a strong timeline for analyzing the influence of specific reviews on product sales later. Besides, the occurrence of outliers usually indicates a series of boom or bust effect. Owing to spatial confined, more product cases are put in Appendix.

4.3 Post-purchasing Evaluation

4.3.1 Product Reputation

We propose to calculate the product reputation by integrating the variation of user ratings based on the reviews and the relative performance of the product regarding the average level of the market.

We first plot the average star ratings of all three genres of products between 2007 to 2015 to show the market condition, and denote the average star rating of product on the market within time period τ as $\overline{S_{m\tau}}$.

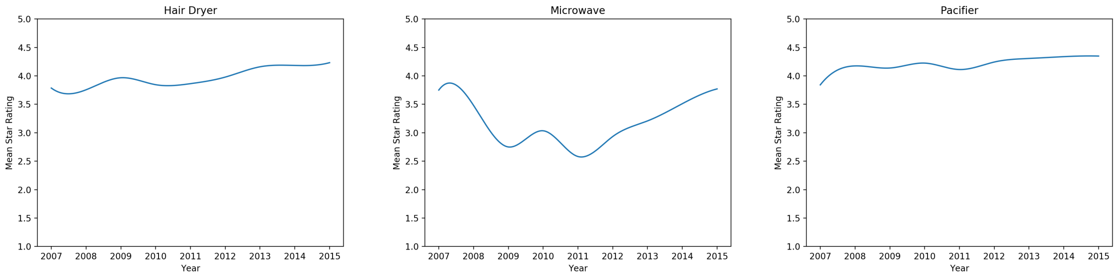


Figure 5: The Average Star Rating of Products on the market

We further denote the Market Average Growth Rate(MAGR) of one category of product on the market in time period τ as:

$$\text{MAGR}_{\tau} = \sum_{\tau_{\text{start}}}^{\tau_{\text{end}}} \frac{GR_{\tau_i}}{N} \tag{10}$$

where GR_{τ_i} represents the growth rate in $i-th$ time period of τ, and N represents the length of time period τ. Similarly, a specific Product Average Growth Rate(PAGR) can be represented as:

$$\text{PAGR}_{\text{j}\tau} = \sum_{\tau_{\text{start}}}^{\tau_{\text{end}}} \frac{GR_{i\tau_i}}{N} \tag{11}$$

For $i-th$ product, its star rating set S_i can be denoted as $S_i = \{s_{i1}, s_{i2}, \cdots s_{iN}\}$, and its average star rating within time period τ can be denoted as $\overline{S_{i\tau}}$. Then we can derive the variance of star rating of $i-th$ product as:

$$s_{i\tau}^2 = \frac{(\overline{S_{i\tau}} - s_{i1})^2 + (\overline{S_{i\tau}} - s_{i2})^2 + \cdots + (\overline{S_{i\tau}} - s_{iN})^2}{N} \tag{12}$$

As we have concluded on Rating Inconsistency in Sec. 4.1.3, products with sparser distribution of star ratings have a relatively lower helpfulness ratings compared to others, which leads to negative impact on perceived value of the product. Concerning this effect, we design a decaying function $D(x)$ to formulate the loss in an acceptable range:

$$D(x) = \alpha e^{-\delta x} \tag{13}$$

where the coefficients α and δ can be determined and optimized using real-time sales data to verify.

Combined the market condition and the effect of ratings and reviews, we define the reputation of $i-th$ product within $\tau-th$ time period in Equation 14:

$$rep_{i\tau} = \theta_1 \frac{\text{PAGR}_{i\tau}}{\text{MAGR}_{\tau}} + \theta_2(\overline{S_{i\tau}} - \overline{S_{m\tau}}) + D(s_{i\tau}^2) \tag{14}$$

where θ_1 and θ_2 are two weighted parameters for a product's relative performance to the market and user-based ratings.

4.3.2 Influence of Star Ratings

We want to find out how star ratings influence the consumer decision-making process in making purchases and writing reviews. To do so, we analyze the average star ratings and sales of hair dryer and pacifier between 2014 and 2015.

From Figure 6, we notice that more stars do lead to more sales. It seems fairly intuitive that higher ratings would lead to higher sales. However, we find that it happens only up to a point. The purchase likelihood typically peaks at ratings in the approximately 4.2 - 4.6 range, and then begins to decrease as ratings approaching 5.0.

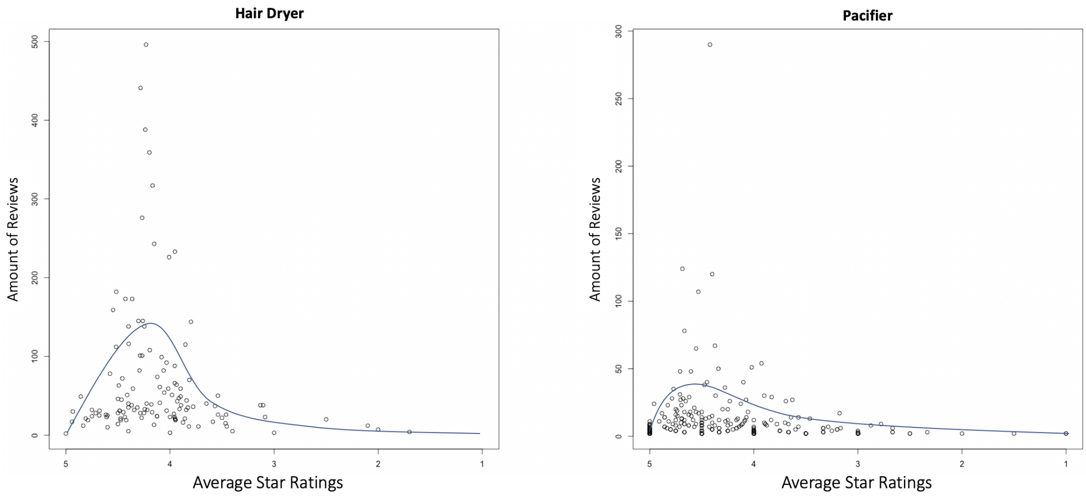


Figure 6: Average Star Rating and Number of Reviews

After detecting a nonlinear relationship between star ratings and the amount of reviews, some may conclude the effect of star rating on review directly. However, we carry out the results under a stable situation where the time is not considered a variable. Therefore, based on time series, we construct a point process model to gain a boarder view of the change in amount of reviews. Then we can answer the question about the relative influence of star rating on reviews more comprehensively.

4.4 Analysis of Market Performance

4.4.1 Reactive Marked Point Process Model

Point processes are collections of random points falling in some space, like time and space.[6] We regard posting a review as a discrete inter-dependent events over continuous time. The information conveyed by a review could influence the purchasing decisions of future customers. In terms of the point processes, the event would increases or decreases the likelihood of observing events in the near future. To reveal the dependency of event arrival rate on past events, we apply self-exciting processes for simulation and parameter estimation.

We construct a Reactive Marked Point Process Model(RMPP), which is a transformation of Hawkes Processes combining the magnitude of review influence, and both the exciting and inhibiting effects of events.[7]

We first define the number of events up to time t:

$$N_t := \sum_{i \geq 1} \Bbbk_{\{t \geq T_i\}} \tag{15}$$

Here T_i is the arrival time for event i. $\Bbbk$ is a indicator function that takes value 1 when the condition is true and 0 otherwise. Then we can represent the point process by $\{N_t\}_{t>0}$ with associated history $\mathcal{H}_t, t \geq 0.$. Define *event intensity* to characterize its conditional intensity functions.

$$\lambda\left(t|\mathcal{H}_t\right) = \lim_{h \to 0} \frac{\mathbb{P}\left\{N_{t+h} - N_t = 1|\mathcal{H}_t\right\}}{h} \tag{16}$$

In our RMPP, the conditional intensity function $\lambda\left(t|\mathcal{H}_t\right)$ takes the form:

$$\lambda(t) = \lambda_0(t) + \sum_{T_i < t} \phi_{m_i}\left(t - T_i\right) \tag{17}$$

$\lambda_0(t)$ is the arrival rate of immigrants events into the system. We construct a power-law kernel ϕ_{m_i} with mark m and influence factor RI

$$\phi_{m_i} = \begin{cases} \kappa(1 - RI)m^{\beta}(\tau + c)^{-(1+\theta)} \; RI >= 0 \\ \kappa RI m^{\beta}(\tau + c)^{-(1+\theta)} \; RI < 0 \end{cases} \tag{18}$$

$\kappa(1 - RI)$ or κRI describes the quality of the review content. β introduces a warping effect for review influences. Mark m models the customer influence for each event, here we use the number of total vote to illustrate the influence range of a review. And $1 + \theta(\theta > 0)$ is the power-law exponent, describing how fast an event is forgotten , parameter $c > 0$ is a temporal shift term to keep $\phi_m(\tau)$ bounded when $\tau \simeq 0.$.

As we have proved in Section 5.1, the Rating Inconsistency RI can reflect the helpfulness rate, which is the quality or reliability of the review. And when RI is positive and goes higher, RI indicates a decaying excitement on future event. When RI is negative and goes lower, it holds an increasing inhibiting effect on review event. And a higher helpfulness rating is suggested excite the So the difference between $1 - RI$ and RI can reflect that pattern in the model.

Together with the total vote number of a review which indicates the quantitative influence, we can say that the $\kappa(1 - RI)m^{\beta}$ or $\kappa RI m^{\beta}$ shows the strength of influence. And the power-law kernel $(\tau + c)^{-(1+\theta)}$ models the memory over time.

We use the maximum likelihood estimation technique to estimate the four parameters in our model κ, β, c, θ. We can calculate the log-likelihood formula for the kernel function:

$$\mathcal{L}(\kappa, \beta, c, \theta) = \sum_{i=2}^{n} \log \kappa + \sum_{i=2}^{n} \log \left(\sum_{t_j < t_i} \frac{(m_j)^{\beta}}{(t_i - t_j + c)^{1+\theta}} \right) \tag{19}$$

We use $n^{\star} < 1$ as a non-linear constraint. We process the estimation for one product in hair dryer (remington ac2015) to show the model's efficiency. We fit the parameter using the 496 events observed during 2013 to 2015. The result is $\{\kappa = 1.000, \beta = 1.015, c = 255.563, \theta = 1.43\}$ with a corresponding $n^{\star} = 0.94$ The figures show the time series of the number of reviews in 2015. We can find N_t increases by one unit at each event time T_i.

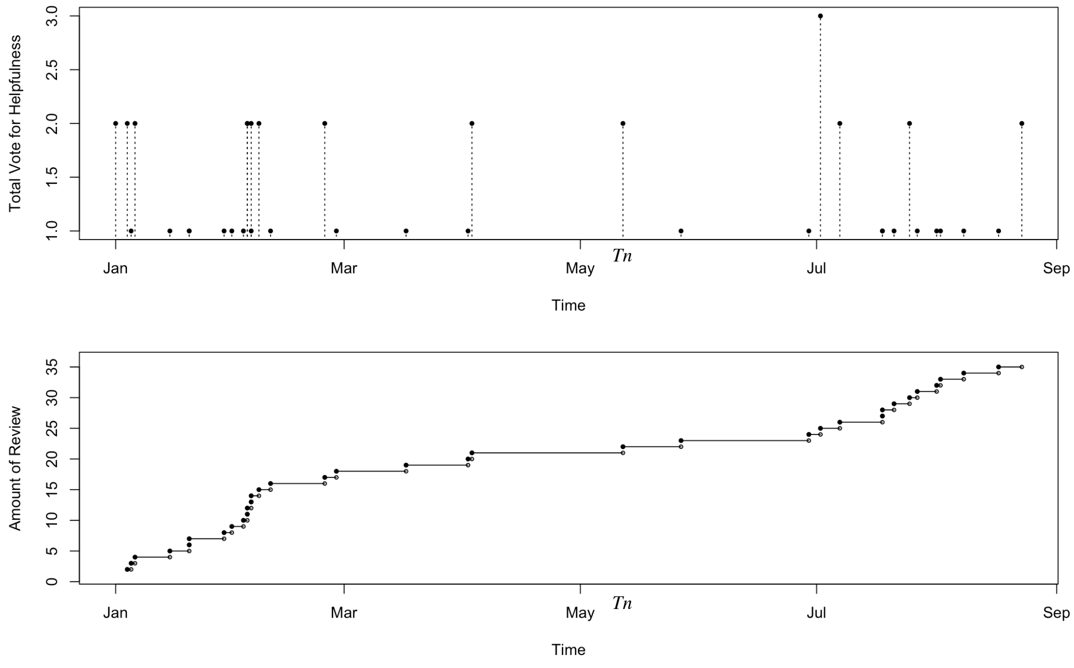


Figure 7: Review Influence and Review Amount Time Series Plot

And therefore, we approach the response to the Task 2d about whether specific star ratings incite more reviews. This task is equivalent to an understanding of the factors influencing conditional intensity function $\lambda(t)$.

The intensity function depends explicitly on all previously occurred events. In our RMPP model, the kernel ϕ_m is composed by mainly two parts, strength of influence for an event and the model's memory over time. There are two factors that count for the strength of influence, where one is the magnitude and another is about property. We use the amount of total votes to represent the magnitude influence. And we use the ratio of the difference between a specific star rating with the average rating during the close time period, which we have defined as RI. So far, it is clear how the star ratings influence the arrival of reviews.

For example, when an abnormal low rating emerge, the model obtain a negative RI with large absolute value. The inhibition effect would restrain the next review to arrive.

4.4.2 Comprehensive Score

Previous work has constructed quantitative descriptions of User Based Product Score, Product Reputation, Star Ratings Influences that help us evaluated how the market data interacts with each other. We want to get a Comprehensive Score based on our previous descriptions that best indicates a potentially successful or failing product by forecasting its sales.

Before we built the Comprehensive Score formula, we have another factor to take into consideration. In the justification part of User Based Product Score, we find that the occurrence of outliers usually indicates a series of boom or bust effect.We define a specific point of time Boom Point B to represent the starting point of the dramatic change on intensity function.

$$B_{i\tau} = \prod_{k=0}^{m-1} \frac{S_{i(\tau-m)} - S_{i(\tau-m-1)}}{S_{i(\tau-m-1)} - S_{i(\tau-m-2)}} \delta(\frac{S_{i\tau} - S_{i(\tau-1)}}{S_{i(\tau-1)} - S_{i(\tau-2)}}) \tag{20}$$

In Equation 19, m means the recent m time period will have influence on the Burst effect. The δ function shows a continuous trend, when the yield rate stop to increase, the burst function shrinks.

Now, we use Entropy Model to build our Comprehensive Score. There exists 3 factors and q time periods, forming the original indicator matrix:

$$X = (X_{ij})_{p\times q} = \begin{bmatrix} U_{i\tau_1} & U_{i\tau_2} & \cdots & U_{i\tau_n} \\ rep_{i\tau_1} & rep_{i\tau_2} & \cdots & rep_{i\tau_n} \\ B_{i\tau_1} & B_{i\tau_2} & \cdots & B_{i\tau_n} \end{bmatrix} \tag{21}$$

However, since the three factors are evaluated on different scales, we have to normalize the matrix:

$$C = (c_{ij})_{p\times q} \tag{22}$$

where r_{ij} is calculated by the equation below:

$$c_{ij} = \frac{(X_{ij} - \min_j x_{i\tau_j})}{\Gamma_i} \tag{23}$$

where Γ_i is calculated as:

$$\Gamma_i = \max_j x_{i\tau_j} - \min_j x_{i\tau_j} \tag{24}$$

Also, a way of calculation of information entropy is established. The relative entropy of i^{th} indicator is defined as:

$$e_i = -\frac{1}{\ln q} \sum_{j=1}^{q} \frac{c_{ij}}{\sum_{j=1}^{q} c_{ij}} \ln \frac{c_{ij}}{\sum_{j=1}^{q} c_{ij}} \tag{25}$$

Finally, the weight of entropy w_i is determined as:

$$w_i = \frac{1 - e_i}{3 - \sum_{i=1}^{3} e_i} \tag{26}$$

where $0 \leq w_i \leq 1$, $\sum_{i=1}^{3} e_i = 1$.

Justification of Comprehensive Score

In order to apply our finalized score model to the given data, we select the top 10 sales product from each genre and set parameter θ_0 as 0.3. Solving for θ_1 and θ_2 from rep$_{i\tau}$ using Equation (12) requires additional assumption that the product's performance will act accordingly to the market conditions. The detailed strategy on the parameters selection would be interesting to investigate in the future study, here we choose $\theta_1 = 0.4$ and $\theta_2 = 0.6$.

Then the Comprehensive Score of each product at time period q can be calculated as:

$$C_i = \sum_{j=1}^{q} C_{ij} \tag{27}$$

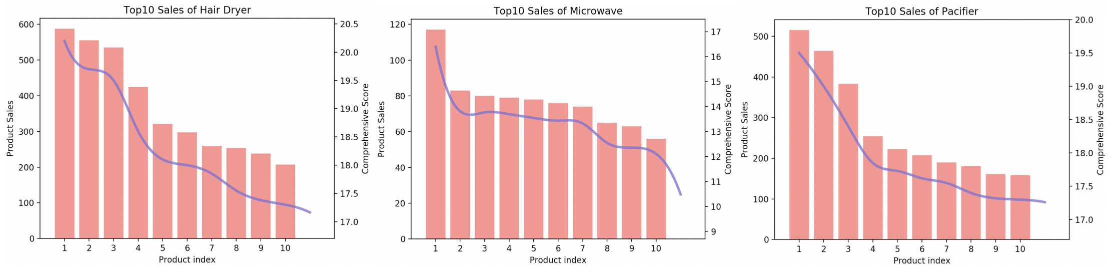


Figure 8: Relationship between Product Sales and Comprehensive Score

From Figure 8, we can find that Comprehensive Score has a strong positive correlation with the product sale. This is to say, we can use Comprehensive Score of a product in one time period to predict the Sales of the product in next time period, expressed as:

$$P_{i\tau} = \alpha_i C_{i(\tau-1)} \tag{28}$$

where α is the unique coefficient of a product, which can be calculated based on history sales amount with recent m time periods. i.e:

$$\alpha_i = \frac{\sum_{k=0}^{m-1} \frac{P_{i(\tau-k)}}{C_{i(\tau-1-k)}}}{m} \tag{29}$$

4.4.3 Entity Analysis

The entity analysis mainly includes three parts. The text processing is the pre-processing of the text content. We figure out the numeric data, the date type process, people's name and the part of speech. After defining all the words in the text, we can attach feature extraction on the data. The feature words and their weight, the keywords summary and the specific information extraction are the main subsections of the extraction. We carry out some important words in this part. And then we apply classification, clustering and filtering on the keywords we got to finally realize the entity analysis.

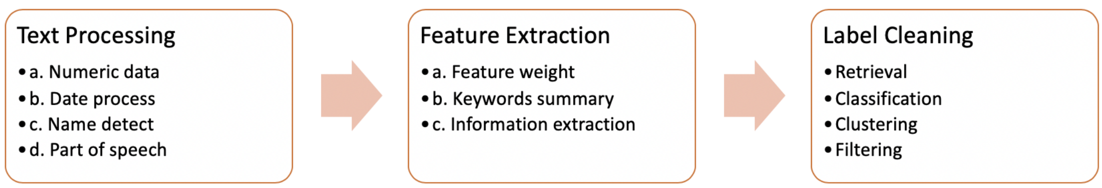


Figure 9: Flow chart for Entity Analysis

Our model pick up some keywords from different rating levels. To illustrate that the result is reliable, we compare the keywords and their emergency counts in the review text. To avoid pollution from unnecessary words, we filter out some irrelevant words in the beginning.

For comparation, we first choose the whole set of review text of hair dryer with average rating higher than 4.0 in the year of 2013 as our input. Table 3 shows the most frequent words within its reviews.

Index	Specific Words	Counts
1	hot	24
2	quiet	22
3	perfect	18
4	good	15
5	worth	13
6	quick	7
7	worth	3
8	cheap	2

Table 3: Most Frequent words appear in Higher Rating Reviews

Index	Specific Words	Counts
1	bad	29
2	hot	25
3	good	22
4	quick	18
5	disappointed	10
6	expensive	7
7	cheap	5
8	annoy	3

Table 4: Most Frequent words appear in Lower Rating Reviews

Next, we choose the whole set of review text of hair dryer with average rating between 2.0 and 4.0 in the year of 2013 as our input. Table 4 shows the most frequent words within its reviews.

We can find from the table that all the counts is large, which means our scheme of selecting words is reasonable. The Entity Analysis is consistent with some simple but powerful ways of selecting the most common words.

Then we perform word clouds corresponding to the Entity Analysis results on the data with low rating levels and high rating levels. The wordclouds present a visual solution of the word results we obtained from the Entity Analysis.

Figure 10: Wordcloud for Higher Rating Reviews

Figure 11: Wordcloud for Lower Rating Reviews

5 Model Analysis

5.1 Sensitivity Analysis

We probe into the sensitivity of some parameters in our PPA Framework. As is shown in Figure 12, when we change θ_0 from 0.3 to 0.7, the overall comprehensive score of the product fluctuates and slightly moves downward. In addition, the scores of the first couple of months show a greater fluctuation when θ_0 changes.

The reason is that our PPA model calculates the comprehensive score by adding weighted sentiment scores and average star ratings. If a product just enter into the market, polarized reviews will have strong effects on its overall ratings. This is very similar to the well-known cold start effect[2]. Moreover, we see the peak of the scores become even higher as θ_0 becomes smaller. This is because larger weight of star ratings generate greater impact on the sentiment scores given by reviews, which means that star rating has a greater impact on consumer's purchasing decision over user's review.

5.2 Strengths and Weaknesses

5.2.1 Strengths

1. **High Generalizability.** We refine three indicators into one numeric measures, and integrate common user behavior and market performance into a comprehensive evaluation.

2. **Point Process.** We regard review as point processes and construct a generalized Hawkes model to evaluated the arrival of reviews under the influences of star ratings and helpfulness rates.

3. **Cross Validation.** In our model, we define some numeric indicators to describe the performance of a given product on the market. We can apply different indicators quantifying the hypothesised relationships between variables, and find they share some common features.

4. **Flexible and Extendable.** We set quantified goals strictly based on optimization theory, and thus our model can be suitable in various market setting, by appropriately altering the

[2]https://en.wikipedia.org/wiki/Cold_start_(computing)

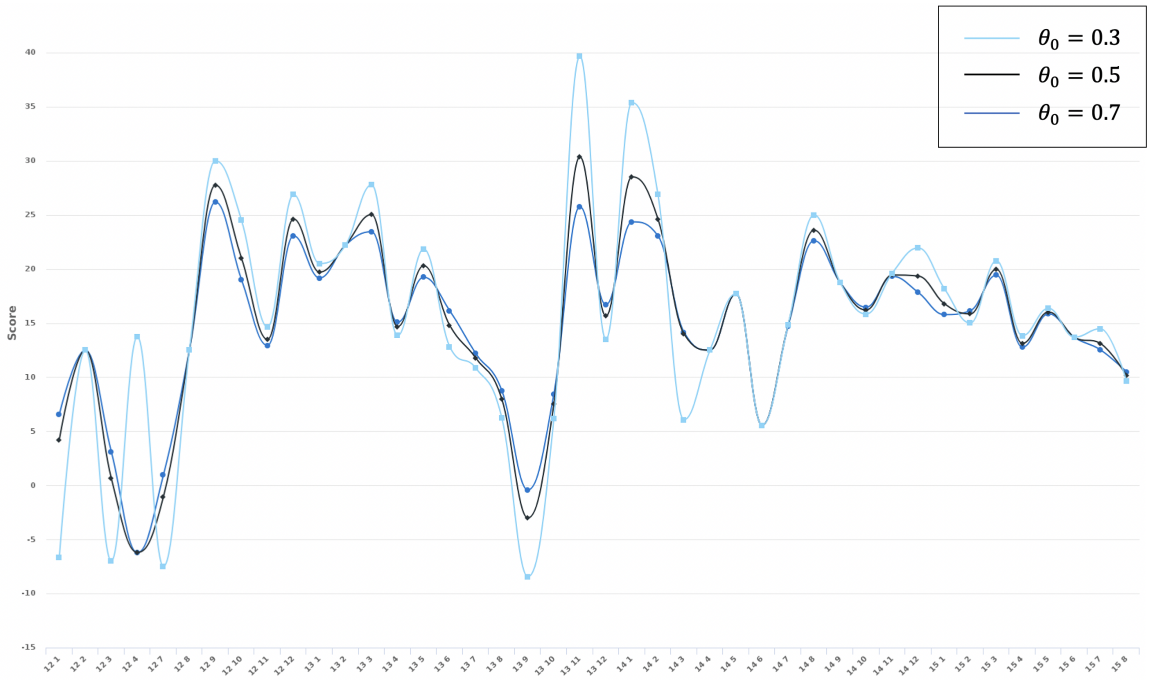


Figure 12: Sensitivity of θ_0

weights of parameters.

5.2.2 Weaknesses

1. **No Verification of Raw Data.** We set our targets based on authentic user behavior, but we have no guarantee of the accuracy of given data.
2. **No Involvement of Brand Benefit.** Brand benefit highly influences people's purchasing decision, but it relates to several factors that cannot be easily quantified, such as self-actualization, love and esteem. Our model only consider some statistics and basic economic principle.
3. **No Consideration of Casual Inference.** We only focus on the correlation between and within variables while omitting the possibility that manipulation of the cause changes the effects.

6 Conclusion

In this paper, we propose a marketing strategy framework called PPA to analyze the patterns and relationships among reviews, star ratings and helpfulness ratings. We define several indicators and refine them all into one comprehensive score that can be optimized for any product on the market.

Since the reviews are textual data, we first apply sentiment analysis and apply Spearman Correlation Analysis between the sentiment scores and corresponding star ratings. We create a consumer profile for all consumer and potential buyer by selecting and aggregating the variables in the provided data. We further purpose the concept of rating inconsistency to reveal the relationship between helpfulness and ratings.

At Pre-purchasing Evaluation stage, we integrate the sentiment score and quantified influence of a review into account, and purpose the User-based Product Score for our Consumer Purchase Decision Model.

At Post-purchasing Evaluation stage, we define the product reputation as the weighted sum of the average star rating growth rate, rating level difference and the variance of ratings. We investigate the relative relation between star ratings and the amount of reviews it receives.

At Analysis of Market Performance stage, we construct a reactive marked point process model, which is a transformation of Hawkes Process combining the magnitude of review influence and inhibiting effect of events. We propose the Comprehensive Score employing all the data indicators we purpose to better predict the product's performance on the market. Based on the score, we pick top sales and make entity analysis to gather the key attributes of their success.

Finally, we conduct sensitivity analysis of some parameters, and discuss strengths and weaknesses of our PPA Framework.

MEMORANDUM

To: Marketing Director of Sunshine Company

From: Team #2002600

Date: March 9^{th}, 2020

Subject: The Smart Review-based Rating Strategy in Online Marketplace

---

Dear Sir/Madam,

We are writing to you regarding the sales strategy of the three new products you plan to introduce and sell in the online marketplace. Our team has made a comprehensive study on market condition based on historical data of star ratings and user reviews. We have created a marketing strategy framework that attach a deep understanding of the market and how it works.
Based on our model's Analysis of Market Performance, We will illustrate how to utilize the results. First look at the Comprehensive scores of the goods in the same product line. Since this score is a combination of user-related product score and quantified relative performance towards the market condition, it can evaluate a product in all directions. Higher score implies a strong competitor in the market while the lower score implies some problems in that product. The Comprehensive Score can be regarded as a standard for selecting outstanding competitors in the market.
After obtaining the object to study, the Boom or Bust Point provide the information of the most important time spots in the history of that product. The point reflects the time when huge changes are aroused in consumer's purchasing behavior. The study of the process will contribute to the decision making and proposal for your new product.
Finally, we can provide you the reason behind the success or failure of a product. Through Entity Recognition based on the helpful reviews sent by the real consumers, our model concludes the precise and important information among thousands of reviews. It shows the right way to further improve the product according to the feedback from the users. Company will definitely benefit from our Entity Recognition in order to enhance their user satisfactory and attract more customers. Specifically, after studying the top-selling product, we found an increasing pattern of Comprehensive Scores. For example, the Entity Analysis for hair dryer shows that keywords like "quiet", "cheap" will contribute to the success of a product.
The above is the summary of our study. We sincerely hope that it will provide you with useful information.

Best regards.

7 Reference

[1] Moe, Wendy W., and Michael Trusov. “The Value of Social Dynamics in Online Product Ratings Forums.” *Journal of Marketing Research*, vol. 48, no. 3, June 2011, pp. 444–456, doi:10.1509/jmkr.48.3.444.

[2] L. Yanbin and Y. Ping, "Processing Online Market Information Based on Users’ Online Information Behavior," *2010 Third International Symposium on Information Processing,* Qingdao, 2010, pp. 58-62. doi: 10.1109/ISIP.2010.84

[3] D. Yang and S. Fujimura, "What Will Influence Customer’s Engagement the Strategies and Goals of Tweet," *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM),* Macao, Macao, 2019, pp. 364-368.

[4] Hyunmi Baek , JoongHo Ahn Youngseok Choi (2012) Helpfulness of Online Consumer Reviews: Readers’ Objectives and Review Cues, International Journal of Electronic Commerce, 17:2, 99-126.

[5] McKnight, D.H., and Kacmar, C. Factors of information credibility for an Internet advice site. In R.H. Sprague Jr. (ed.), Proceedings of the 39th Hawaii Annual International Conference on System Sciences. Los Alamitos, CA: IEEE Computer Society Press, 2006. doi: 10.1109/IEEM44572.2019.8978769

[6] Rizoiu, Marian-Andrei Lee, Young Mishra, Swapnil. (2017). A Tutorial on Hawkes Processes for Events in Social Media.

[7] Ertekin, Şeyda; Rudin, Cynthia; McCormick, Tyler H. Reactive point processes: A new approach to predicting power failures in underground electrical systems. Ann. Appl. Stat. 9 (2015), no. 1, 122–144.

8 Appendices

Program for Sentiment Analysis (Golang)

```go
import (
        language "cloud.google.com/go/language/apiv1"
        languagepb "google.golang.org/genproto/googleapis/cloud/language/v1"
)

func GetScore(id string, str string){
        ctx := context.Background()

        client, err := language.NewClient(ctx)
        if err != nil {
                log.Fatal(err)
        }

        // Sets the text to analyze.
        text := str

        // Detects the sentiment of the text.
        sentiment, err := client.AnalyzeSentiment(ctx, &languagepb.AnalyzeSentimentRequest{
                Document: &languagepb.Document{
                        Source: &languagepb.Document_Content{
                                Content: text,
                        },
                        Type: languagepb.Document_PLAIN_TEXT,
                },
                EncodingType: languagepb.EncodingType_UTF8,
        })
        if err != nil {
                log.Fatalf("Failed to analyze text: %v", err)
        }

        fmt.Println(id, ",", sentiment.DocumentSentiment.Score)
}
```

Program for User Based Score Algorithm (Golang)

```go
func SortCommentNum(ProductContentArr *[]ProductContent) *[]IndexAndNum {
        var arr []IndexAndNum
        for i, productContent := range *ProductContentArr {
                arr = append(arr, IndexAndNum{
                        Index: i,
                        Num: len(productContent.AllComment),
                })
        }
        sort.SliceStable(arr, func(i, j int) bool {
                return arr[i].Num > arr[j].Num
        })
        return &arr
}

func UserBasedScore(theta float32,SenData *SentimentData, Data *GoodsData,
    j int, month int, year int) float32 {
        var StarRateScore float32
        var CommentScore float32
        var StarArr []float32
        var SentimentArr []float32
        var IdArr []string
        var HVArr []int
        var TVArr []int
        // get StarArr amd IdArr
        for _, comment := range Data.ProductContentArr[j].AllComment {
                if comment.Month == month && comment.Year == year {
                        StarArr = append(StarArr, float32(comment.StarRating-3)/4)
                        IdArr = append(IdArr, comment.ReviewID)
                        HVArr = append(HVArr, comment.HV)
                        TVArr = append(TVArr, comment.TV)
                }
        }

        // get SentimentArr []float32
        for _, id := range IdArr {
                var find = false
                for _, SentiContent := range SenData.AllSentimentContent {
                        if id == strings.TrimSpace(SentiContent.ReviewID) {
                                SentimentArr = append(SentimentArr, SentiContent.SentimentScore)
```

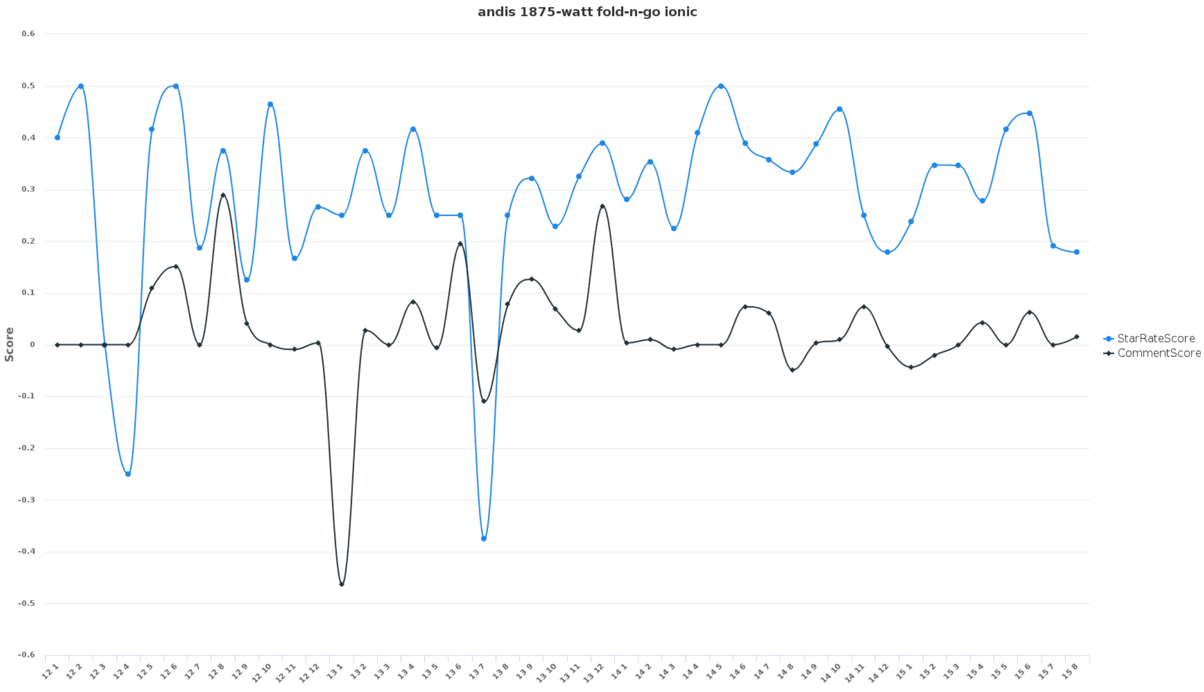
```
                                find = true
                        }
                }
                if find == false {
                        SentimentArr = append(SentimentArr, 0)
                }
        }

        // Calculate StarRateScore
        StarRateScore = GetSum(&StarArr)/float32(len(StarArr))

        // Calculate CommentScore
        for i, _ := range SentimentArr {
                a := float32(HVArr[i])/float32(TVArr[i]+1)
                b := float32(math.Log(float64(2+TVArr[i])))
                c:= SentimentArr[i]
                CommentScore += a*b*c
        }
        CommentScore = CommentScore/float32(len(SentimentArr))

        //calculate UserBasedScore
        UBScore := 100 * ((1-theta) * StarRateScore + theta * CommentScore)

        return  UBScore
}
```

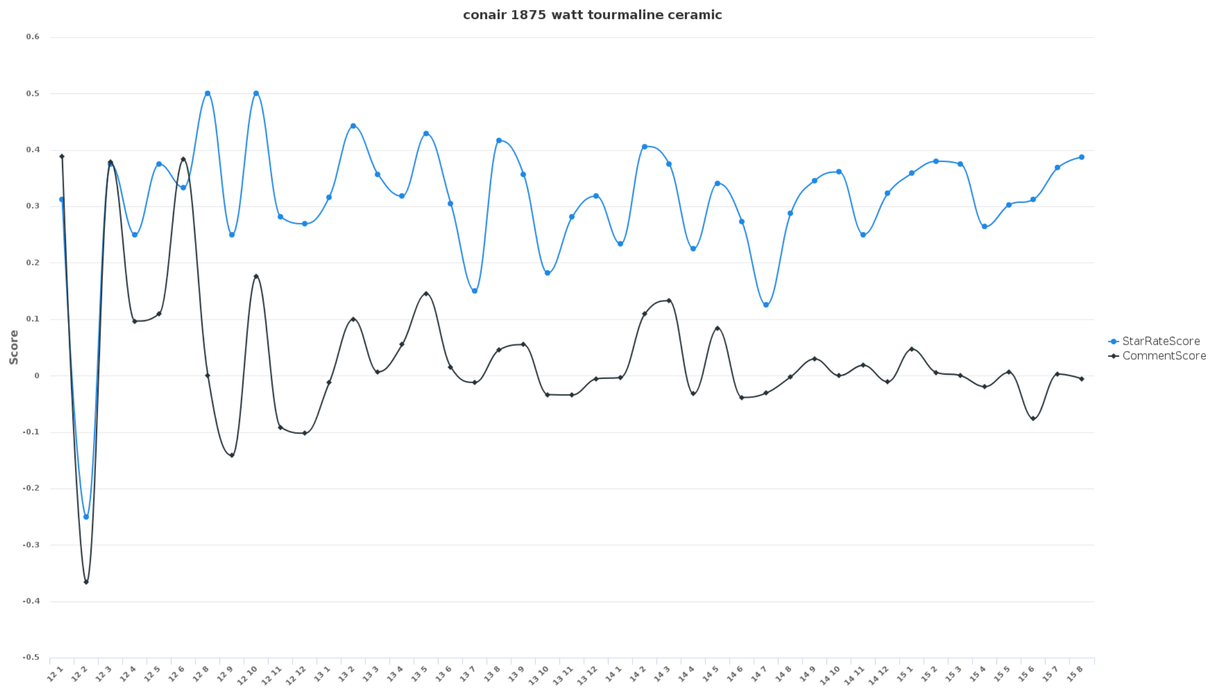
Program for Data Initialize (Golang)

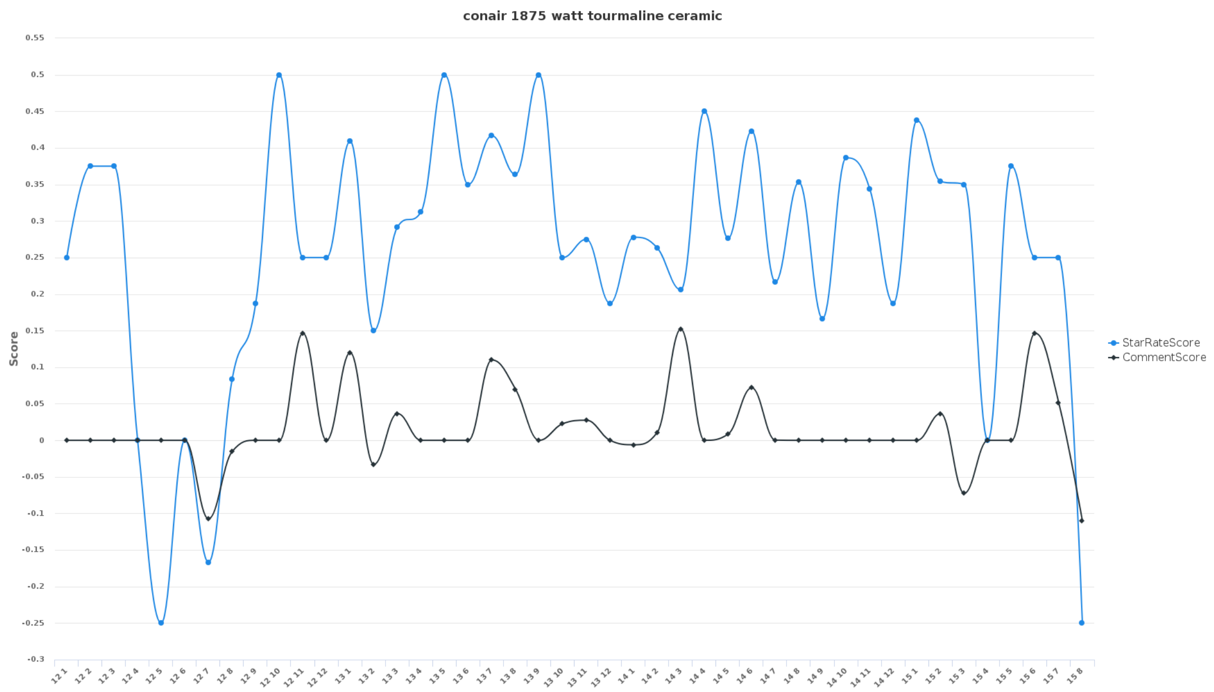
```
type Comment struct {
        StarRating int
        HV int
        TV int
        Year int
        Month int
        ReviewID string
        ReviewBody string
}

type ProductContent struct {
        ProductID string
        ProductTitle string
        AllComment []Comment
}

type GoodsData struct {
        GoodsName string
        ProductContentArr []ProductContent
}

type SentimentContent struct {
        ReviewID string
        SentimentScore float32
}

type SentimentData struct {
        AllSentimentContent []SentimentContent
}


func GetProductIndex(product string, arr *[]ProductContent) int{
        for i, Product := range *arr{
                if product == Product.ProductID {
                        return  i
                }
        }
        return -1
}

func InitializeData( table *csvMgr.CsvTable) *GoodsData {
        var ProductArr []ProductContent

        for _, record := range table.Records[1:] {
                id := record.GetString("product_id")
                title := record.GetString("product_title")
                ProductIndex := GetProductIndex(id, &ProductArr)
                if  ProductIndex == -1 {
                        ProductArr = append(ProductArr, ProductContent{
                                ProductID:          id,
                                ProductTitle:         title,
                        })
                        ProductIndex = len(ProductArr)-1
                }
                ProductArr[ProductIndex].AllComment =
                append(ProductArr[ProductIndex].AllComment, Comment{
```

```
                        StarRating:         record.GetInt("star_rating"),
                        HV:                 record.GetInt("helpful_votes"),
                        TV:                 record.GetInt("total_votes"),
                        Year:               GetYear(record.GetString("review_date")),
                        Month:              GetMonth(record.GetString("review_date")),
                        ReviewID:           record.GetString("review_id"),
                        ReviewBody:         record.GetString("review_body"),
                })
        }

        return &GoodsData{
                GoodsName:                      table.Filename,
                ProductContentArr:              ProductArr,
        }
}


func InitializeSentiment(table *csvMgr.CsvTable) *SentimentData {
        var  AllSentiment         []SentimentContent
        for _, record := range table.Records[1:] {
                id := record.GetString("review_id")
                score := record.GetFloat32("sentiment")
                AllSentiment = append(AllSentiment, SentimentContent{
                        ReviewID:               id,
                        SentimentScore:         score,
                })
        }
        return &SentimentData{AllSentiment}
}
```

Plots of other top 3 sales product on Star Rating Score V.S. Comment Score




conair 1875 watt tourmaline ceramic
Score
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
StarRateScore
CommentScore


conair 1875 watt tourmaline ceramic
Score
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
StarRateScore
CommentScore