## 3 The Model

We believe that it is important to tell if a customer’s review is ‘reliable’, which means whether the customer is a trustworthy person who actually purchased the merchandise and gave an objective review, or a person who gave false review with nasty intention, or maybe even a bot made to deceive customers. We designed an evaluation system weighting every review entry to find out if the review entry, or the customer himself, is trustworthy.

Our model contains two main systems and some subsystems. The two main systems respectively give the score customer evaluate the merchandise and the weight of the review entry. The subsystems process the data given by main systems to enhance the readability of the data.

The scoring system is the main system that aims at making the mechanism more accurate in reflecting customer’s rating. The star-rating system is discrete and having only 5 levels for customers to choose which struggles in judging ‘how much’ do the customers love/hate the merchandise. We apply an NLP instance to analyze the sentiment of the review content as a complement to the star rated which is converted to a float number between 0 and 1. Due to the accuracy of the NLP instance, if the difference of the two scores are larger than 0.3, the ‘sentiment score’ will not be used, otherwise the final score will be the average of the sentiment score and the star score.

The weighting system is the main system that aims at judging if the review or the customer is reliable. The output of the system is the weight of each review entry while the overall score of a product is the weighted average of each entry’s ‘final score’.

The weighting system will judge the weight in 2 aspects: the review itself and the customer. The review’s weight will only be applied to itself, while the customer’s weight will be applied to every review he wrote. The system holds a ‘dictionary’ to record the customer’s weight. The system uses several ‘weight multipliers’ to adjust the weight as following:

First, the system gives the customer’s weight calculated by the current review entry:

1. If the customer is marked as ‘vine’, then he should be awarded for the trust he earned. A ‘vine’ customer’s weight will be directly set to 3.0, which is the largest weight of the system. The ‘vine’ customer’s review will use no other multipliers.
2. If the review has not less than 5 votes, we will take the votes into consideration and enables ‘vote multiplier’. We define ‘helpful rate’ as the percentage of helpful votes among all votes. If ‘helpful rate’ is more than 0.9 or less than 0.3, we will consider the customer is ‘remarkably’ supported or opposed by others, which indicates the customer is trustworthy or suspicious. The multiplier will be set at 0.2-0.8 (helpful rate under 0.3) or 1.2-1.5 (helpful rate over 0.9).
3. Meanwhile, if there are more votes, the ‘helpful rate’ is more convincing. The system uses an amplifier which the value will increase 0.02 for every vote more than the base 5 votes with its maximum value limited at 2.0. The final output of the vote multiplier will be the ‘amplifier value’ power of ‘multiplier value’.
4. We noticed that some reviews are long, but the whole review are talking nonsense, some of them are even not English (or any other language). If the review’s length reaches the average length while number of its keywords fails to reach half of the average, the weight of the customer will be reduced by half.
5. If the customer’s weight is modified, the system will write the customer and his weight into the dictionary. If the customer already has a weight, the two weights will get geometric averaged.

Second, the system gives the review’s own weight:

1. The system will use the same ‘vote multiplier’ and amplifier for the reviews whose ‘helpful rate’ is more than 0.3 and less than 0.9. The multiplier will be set at 0.8-1.0 (helpful rate under 0.7) or 1.0-1.2 (helpful rate over 0.7), since the helpful rate cannot be considered as ‘remarkable’, the multiplier will be applied to the review itself only.
2. The ‘length multiplier’ will measure the length of the review and modify the weight. If the length of review is less the 4/5 of the average, the multiplier will be reduced to the minimum of 0.9, otherwise the multiplier will be enlarged to the maximum pf 1.5.
3. Some customers may abuse the refund policy and intentionally add negative reviews for the product, and some customers may receive free products from the reseller to help ‘boost’ the score of the product, so if the purchase is not verified, the review’s weight will be reduced to half.

The final weight that will be applied to calculating the average is the multiply of the two weights.

The visualizing subsystem calculates the ‘real-time’ and ‘accumulated’ reputation score of the select period and product and outputs time-based patterns. The ‘real-time’ pattern is the weighted average of a certain period, for example, 30 days, and the ‘accumulated’ pattern is the weighted average of all the scores before the time stamp. This subsystem will help analysts to have an overall view of the product’s reputation and identify the inflection points.

The keyword subsystem collects keywords that required by clients. Based on the NLP instance, the subsystem can run a statistic on all keywords and found keywords in strong relationship with certain emotion, and provide the ‘feature keywords’ of a certain type of review for user to predict the trend of product reputation or frequently mentioned keywords along with sentiments used to improve marketing strategy.

为了解决上面提及的五个问题，我们建立了一个打分（Mark）模型，将用户的评论转化成具体分值。

Firstly，得出有关情感的keywords，以及相应的frequency。这一过程是基于自然语言处理的。具体的实现过程是先取出评论中frequency超过3000的词，作为样本训练，找出能区分好评和差评的词语，即为关键字。

Secondly，为关键字赋不同的权重。不同的关键字，根据情感强弱，赋不同的权重。例如“good”，“excellent”，我们觉得后者积极情感更强，所以权重要更大一些。

Thirdly，得出文本评论的score，以及综合数字星级与文本评论两项参数的得分point。我们借助Azure的API做情感分析，以keywords、权重作为输入，得出每一个文字评论的score。Score与星级各占50%比重，得出point。

Lastly，计算加权，根据helpfulness ratings的高低，我们给每一个产品point加权，得出最终打分。

即每一个产品的point=（score\*0.5+star\*0.5）\*Weight\_2

## 4 Solutions

4.1 Mask model

对于第一个问题，我们建立了一个Mask model，以star ratings, reviews, and helpfulness ratings三项参数为输入，来定量衡量每一个产品。

4.2 time-based

一方面，我们会分析情感keywords随时间的走势；另一方面，我们绘出每一款产品平均mask随时间变化的曲线；综合这两方面来判断产品’s reputation is increasing or decreasing in the online marketplace.

4.3

4.4 从众心理

“Do specific star ratings incite more reviews? ”,我们的观点是客户在看到之前有很多低星评价会更有可能给出差评。相反，如果他看到之前大部分是五星评价，会更倾向于做出好评。从心理学的角度上看，这是一种从众心理。对于商家来说，有一定的滚雪球效应，即如果出现一定量低星评价，用户差评占比会越来愈多。这时应立即采取措施应对，否则会造成严重损失。

我们的证据是文本评论score与随时间变化的曲线，以及数字星级随时间变化的曲线，两者的走势大致相同。

4.5 emotion analysis

“Are specific quality descriptors of text-based reviews such as ‘enthusiastic’, ‘disappointed’, and others, strongly associated with rating levels? ”我们分析筛选出的关键词发现两者相关性很强，尤其负面评论，与星级关联很紧密。比如，在

pacifier.tsv 中，waste在一星与五星的占比比值是78.6.还有useless在在二星与五星的占比比值是75.5。更多的见下表

但是，有学者研究发现“Text-Rating-Inconsistency”（TRI）现象，【4】。比如说，The comment expresses strong negative sentiment, but is associated with a 5-star rating.Removing these TRI reviews also helps us reduce the noise in the dataset, yielding better performance in later analysis.

**A Letter**

Dear the Marketing Director of Sunshine Company:

It’s our honor to be hired as consultants to identify key patterns, relationships, measures, and parameters in past customer-supplied ratings and reviews associated with other competing products. We are writing this letter to report our latest breakthroughs and findings.

We use natural language processing to analyze the data of three product reviews and derive the semantic and emotional keywords along with the sentiment of the reviews.

With the support of our NLP instance, our keywords subsystem is able to collect keywords that match certain requirements. For example, it could show all the keywords that are most frequently used by the customers who like/dislike the product, which can be an evidence indicating that the reputation of the product is going to rise/drop, your team may as a result update the marketing strategy to increase sales or fix the reputation of the product.

As an instance, the following figure are ‘word clouds’ generated by the keywords subsystem. The keywords are collected from the most negative reviews, serving as the indicators of a coming reputation drop. Your team can monitor the most frequently shown keywords and take measure in advance of the reputation loss.

**Figure 1: hair\_ dryer Figure 2: microwave Figure 3: pacifier**

In additional to that, our keywords subsystem can also be configured to output only nouns and verbs, showing the most popular/hated points of the product. This indicates potentially important design features that would enhance product desirability. Your team are suggested to advertise the advantages or improve the weakness based on the keywords provided.

For example, the word ‘warranty’ in microwave’s word cloud clearly indicates that warranty service is of vital importance and many customers and not satisfied with the current warranty policy. Your team may consider improving and advertising your warranty policy to attract customers to buying your product instead of your competitors’.

We believe that while evaluation of the reviews is of vital importance, it is also crucial to tell if a customer’s review is ‘reliable’, which means whether the customer and the review is trustworthy and objective or a intentionally written lie trying to deceive customers. Our weighting system evaluate every review entry to find out if the review entry, or the customer himself, is ‘reliable’. We believe that our weighted reputation score can accurately evaluate the customer feedback of the product, thus indicate a potentially successful or failing product.

With the help of our weighting system, our visualizing subsystem can visualize the ‘real-time’ and ‘accumulated’ reputation of a selected model into time-based patterns. The ‘real-time’ pattern is the weighted average of a certain period, for example, the following figure on the top is the average reputation score of every 30 days. The ‘accumulated’ pattern is the weighted average of all the scored before the time stamp, as shown the following figure at the bottom.

**Figure 4: B003CK3LDI-Realtime**

**Figure 5: B003CK3LDI-Accumulated**

It is easy to identify the inflection points from the ‘accumulated’ pattern and with the help of the ‘real-time’ pattern we can conclude that the sharp drop of the reputation caused by an extremely negative review DID make the customers tend to write their reviews less positively than before. And your team can work out a plan against any upcoming negative reviews to minimize the potential loss.

That’s all for our model and our findings, we wish our work could help you succeed in your coming online marketplace product offerings.

Yours sincerely

Team 2013868

Real 􀀀 timereputation Weighted average of a certain period.

We deployed an NLP instance to process the text reviews for keywords extraction and sentiment score evaluation.

We designed a scoring system and a weighing system to make a quantitative measurement of the star ratings, reviews, and helpfulness, thus calculate the product’s reputation score.

We wrote several subsystems to help clients take advantage of our findings, the subsystems can emphasize the key factors which indicates the alteration of the product’s reputation, give clients suggestions based on the keywords trend, and visualize the product’s reputation with time-based pattern.

First, we process the text reviews of the three product datasets to extract keywords

and corresponding frequencies. As for the measurement of the success of the product,

we have established a scoring model, which takes the three parameters star ratings,

reviews, and helpfulness ratings as input to get the final score of the product.

1. we use NLP text comment processing on three product datasets to extract keywords

and corresponding frequencies.

2. Build a mask model based on star ratings,reviews,and helpfulness ratings three

parameters to give each product a mask.