# 1 Introduction

## 1.1 Background

With the coming of the information era, billions of pieces of data are produced every day in the world. Fifteen years ago, there were few individuals possessing a computer. But nowadays, everyone shops online. The birth of online shopping impacts people's shopping habits. People tend to try some products in a brick-and-mortar store and buy them online for a better price, which brings a lot of transaction data to the e-commerce industry. It does matter how to mine meaningful information from such a big data pool. At the same time, the development of data analysis tools makes it more convenient for people to process various types of data. From the simplest numerical data to the complicated text-based, images, voices and even video data, data scientists exploit multifarious methods to process data in order to understand the instructive value hidden behind it. On this occasion, the research on people's shopping preferences and purchase history has caught the eyes of big-head companies to introduce more products meeting the customers' personalized needs and gain more profit.

## 1.2 Problem Summary

With the forthcoming of the online shopping era, tons of challenges and opportunities have been generated for retail companies. In order to seize this occasion to help Sunshine Company introduce their three new products: microwave ovens, baby pacifiers, and hairdryers, we make a proper business strategy based on the analysis of sold records such as reviews, customer-supplied ratings for the similar kinds of products on Amazon. In particular, the feedback on the forum will depict the satisfaction level for products, the touch texture and specific using circumstances during the use of the products. Besides, consumers have the right to vote for other comments on the forum if his or her comment is valuable to them. We take the above factors into consideration and are about to find some worthy information through the data within a developmental context. This will contribute to generating a more reasonable online sales strategy and identifying

the potential important features that would enhance product attractiveness. Our goal is to identify potentially important features that lead to a successful product and generate a reasonable online sales strategy.

# 2 Nomenclature

| Symbol | Definition |
|---|---|
| NLTK | Natural Language Toolkit |
| KNN | K-nearest Neighbors Algorithm |
| RBF SVM | Radial Basis Function Support Vector Machine |
| QDA | Linear Discriminant Analysis |
| $A_{n\times m}$ | Matrix of token counts after converting the collection of text documents |
| GBDT | Gradient Boosting Decision Tree |
| $w_{q(x)}$ | The score for leaf $q$ |
| $F$ | Set for all $K$ regression tree |
| $f$ | Regression tree |
| $\hat{y}$ | Prediction value |
| $T$ | Number of leaves |
| PPMCC | Pearson product-moment correlation coefficient |
| PCA | Principal component analysis |
| SVD | Singular value decomposition |

# 3 Our Approach

To identify the key to making a successful product, we utilize a dataset collected from the Amazon Customer Reviews Dataset through the Amazon Simple Storage Service, stored in the following files:

- $hair\_dryer.tsv$: Contains data for hairdryer from 2002 to 2015
- $microwave.tsv$: Contains data for microwave from 2004 to 2015
- $pacifier.tsv$: Contains data for pacifier from 2003 to 2015

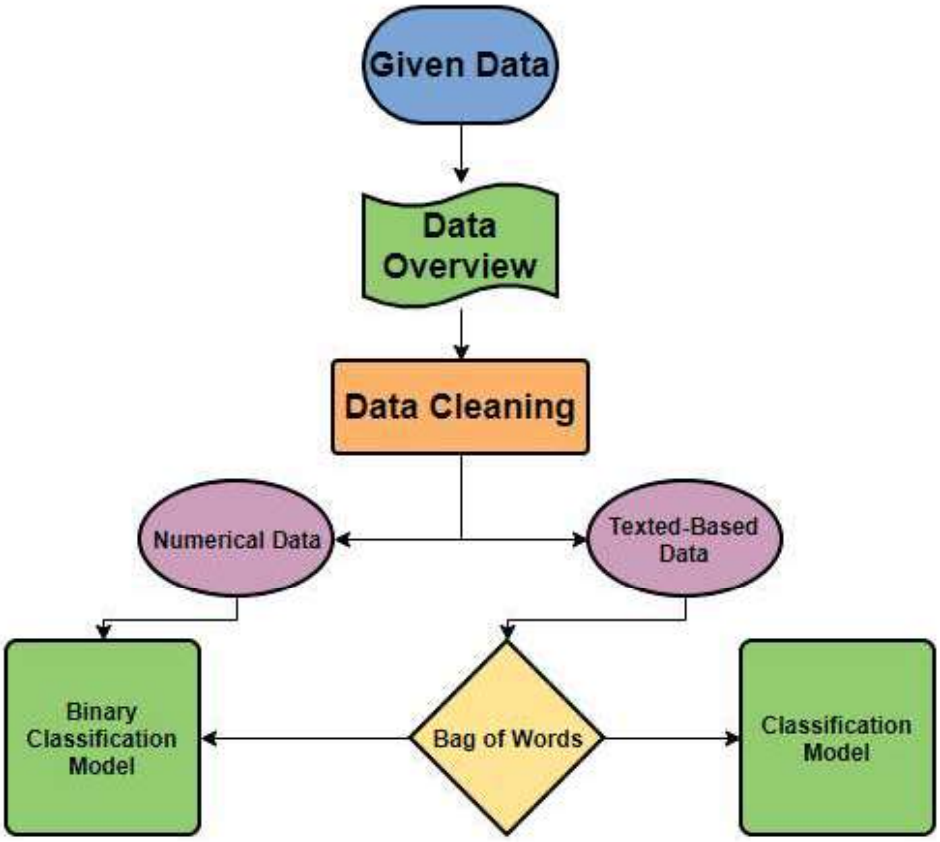


Figure 1: Overview flow chat for our model procedure

An outline for our data mining procedure can be found in Figure 1. The relevant blocks will be described in the rest of this section.

## 3.1 Data Summary

In order to clarify the target features used in our model and mine adequately information from the datasets, data pre-processing and overview of data distribution are essential. These pieces of information can provide important intuitive ideas of the conclusion that will be deduced after further analysis. As the first step, Figure 2 - 7 show the overall distributions of the target products, hairdryer, microwave oven and pacifier. In particular, the pie charts show the star rating proportions for each product, which indicates that hairdryers and pacifiers are more likely to gain high reputation than microwave ovens.

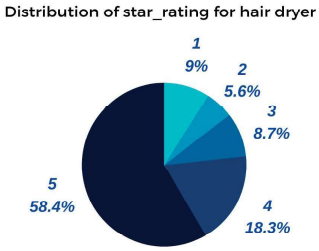


Figure 2: Distribution of star_rating for hairdryer

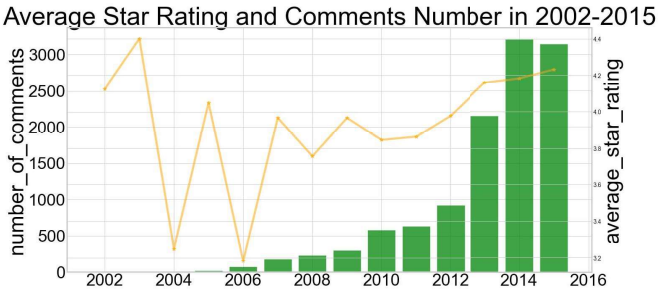


Figure 3: Time-based relation of number_of_comments and avg_star_rating for hairdryer

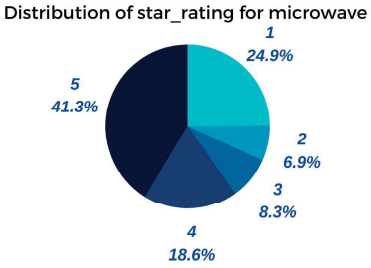


Figure 4: Distribution of star_rating for microwave oven

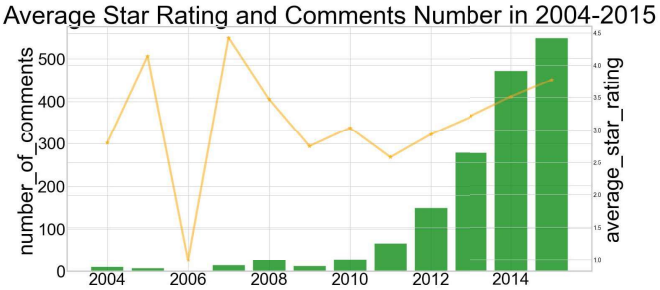


Figure 5: Time-based relation of number_of_comments and avg_star_rating for microwave oven

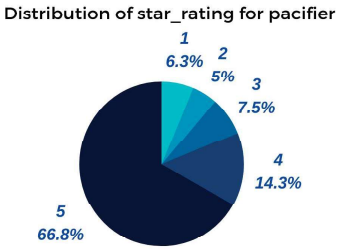


Figure 6: Distribution of star_rating for pacifier

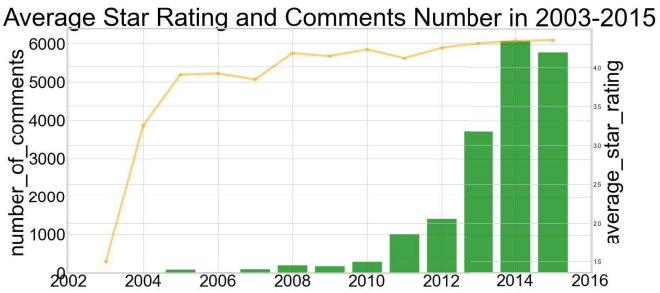


Figure 7: Time-based relation of number_of_comments and avg_star_rating for pacifier

Through the histogram broken lines combination diagrams, we discover that the general trend of the sales volume of each product is steadily increasing up to 2015. The sales volume of the microwave oven is still increasing from 2014-2015 particularly, which shows its potential for future sales. As for the star rating, the average star rating for the pacifier is steadily increasing from 2003 and tends to be stable in recent four years. However, during the previous half of the period, average ratings for hairdryer and microwave oven change in a large degree. It

may be ascribed to the small volume of purchases, i.e. a small number of ratings can have a huge impact on the overall level. Luckily, the recent average star rating for these two products is steadily increasing, which reveals their potential in the future and value for future analysis.

## 3.2 Data Cleaning

The reviews collected contain missing parts and partially filled in data that are challenging to analyze[6]. We have done the following to sanitize the data-set:

- Remove all the blank reviews
- Since the reviews consist of all special characters is relatively few[1]. Without loss of generality, we remove all the reviews that only contain special characters and no English alphabets
- Change all reviews to lowercase to maintain the consistency of features
- Remove stop words, such as $the, a, an, in$ and adopt NLTK to avoid the impact of meaningless words
- Extract word stem and remove morphological affixes from words in reviews using PorterStemmer in order to get rid of the influence of tenses and plurals

## 3.3 Assumption

In the procedure of model designing, one of the most significant steps is to make assumptions with the aim of simplifying the problem into some special cases. In order to facilitate the data adapting to our model and generate more reasonable outcomes, some assumptions have been illustrated as follows:

- All the data provided is reliable, confidential and has not been tampered
- The success of a product can be measured by its total sales volume

[1] 4/11470 in hair_dryer.tsv; 0 / 1615 in microwave.tsv and 4 / 18939 in pacifier.tsv

- The star rating is positively correlated with people's satisfaction of a certain product, i.e., there is no or little reverse rating problem
- The average star rating score is positively related to and can be adopted as a feature to represent the product's reputation
- Every customer comments and writes a review after the purchase, i.e. there is no huge difference between the actual total number of purchases and the number of reviews

## 3.4 Model

### 3.4.1 Bag of Words Model

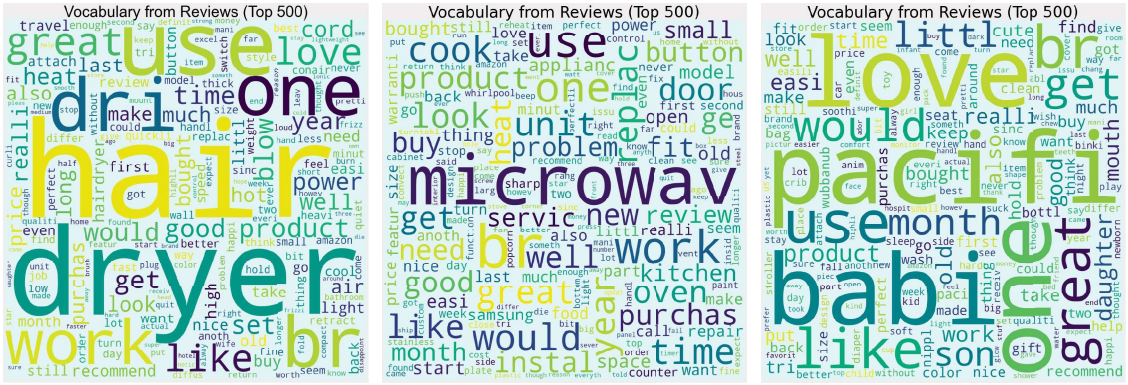


Figure 8: Top 500 review words for hairdryer, microwave and pacifier

In addition to numerical data, text-based data also contains tons of information we need to discover. There are thousands of words contains in reviews as shown in Figure 8.

After data cleaning procedure, the frequency distribution of review words has been computed. However, because of the numerical characteristic of most mathematical models, Bag of Words Model is needed to convert the words into a matrix $A_{n\times m}$, where $n,\ m$ represent the number of reviews and words after data cleaning respectively.

$$A_{i,j} = \text{Number of } j\text{-th word in } i\text{-th review}, \quad \text{where} \quad i = 1, 2, ..., n,\ j = 1, 2, ..., m.$$

$A_{n\times m}$ matrix is obtained using CountVectorizer which converts the words in the reviews into a word frequency matrix and calculate the number of occurrences of each word. We can use this word matrix $A_{n\times m}$ together with other numerical features to do analysis with higher dimensionality and accuracy.

### 3.4.2 Binary Classification Model

In order to indicate the potential correlation between data features we have and successful products, calibrating the standard of which product is successful is essential. Our idea is to train a binary classifier $\hat{S}(W, R) \in \{0, 1\}$ that takes each product's star rating $R_i$ and the words vector $W_i$ generated by the CountVectorizer in Bag of Words model, as model input. The trained classifier minimizes

$$\sum_{i\in\text{product}} \text{error}(\hat{S}(W_i, R_i), S_i),$$

where $S_i \in \{0, 1\}$ indicates the success of product $i$ and the summation is taken over all products in the same category (microwave, hairdryer, pacifier). The above results in a classifier that models the correlation between a product's success and its ratings/reviews. We can then identify the importance of each feature through maximizing the probability that $P(S_i = 1|W, R)$ with respect to $W, R$.

To generate the labels in training data, we define the success (or failure) of a product as

$$S_i = \begin{cases} 1, & \text{if \# sales} \geq \overline{S}_{\text{product}} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $\overline{S}_{\text{product}} = 15, 13, 3$ for the hairdryers, microwave oven, pacifier, respectively. The reason here why we don't apply the same threshold is that different products performed huge diversity in the overall sales volume distribution. For instance, more than 75% of the pacifier brands only sales one pacifier from 2003 to 2015 but the median sales volume for all microwave oven is over 10. In order to keep the precision and generality of our result, custom thresholds are required to guarantee there are enough data for training.

### 3.4.3 Multiclass Classification Model

With the aim of mining the relations between reviews words and star ratings, we introduce a classification model taking all review words as input and mapped to star ratings. Since the star rating takes value from 1 to 5 (5 classes), the multi-class classification model is adopted. Technically, there are two popular ways to handle multiclass classification problem[1]. We adopte a One-Versus-All method there to build our model because of its time-efficiency during the training procedure. Our $k$-class classification model is divided into $5$ small binary class classifiers. Each classifier is responsible for classifying label $y_i,\ i \in \{1, 2, ..., 5\}$, by treating other $4$ (i.e. $k-1$) classes other than $i$ belonging to another same class $\hat{y}_i$, which will generate a perfect two-class problem for the binary classifier to solve. When making a decision, as showed in Algorithm 1, input $X$ of a test sample, respectively substitutes into $k$ classifiers and takes the category with the largest output (i.e. the greatest possibility) as its class standard.

**Algorithm 1** Multiclass Classification Pseudocode

Input:

- $L$, a learner (training algorithm for binary classifiers)
- Samples $X$
- Labels $y$ where $y_i \in \{1, ..., K\}$ is the label for the sample $X_i$

Output:

- a list of classifiers $f_k$ for $k \in \{1, ..., K\}$

Procedure:

- **for** each $k$ in $1, \ldots, K$ **do**
  Construct a new label vector $z$ where $z_i = y_i$ if $y_i = k$ and $z_i = 0$ otherwise
  Apply $L$ to $X$, $z$ to obtain $f_k$
- **end for**

### 3.4.4 Implementation

In order to resolve the above two classification problems, it is necessary to select a suitable prediction model. For every different data set and problem, we take the most effective 10 prediction models into consideration and use their default parameters to train the model. Based on each model's prediction score, training time and scoring time, there is always a trade between the model training time and prediction score. Our choice is to select the model with the best performance with relatively small time consumption to be our decided model. As showed in Appendix A and Appendix B, AdaBoost model [8] [3] and XGBoost model [2] are the models with best performance among all 10 candidates [9]. Therefore, we decides to use these two models to be our model for prediction problems.

### 3.4.5 Algorithm Analysis

AdaBoost Model AdaBoost [4] is a enhancement algorithm which is developed for binary classification [5]. As illustrated in Algorithm 2, AdaBoost is combined with plenty of small short decision trees. After it creates the first decision tree classifier, the samples misclassified by the previous week classifier are strengthened, and the weighted total sample is again used to train the next basic classifier. At the same time, a new weak classifier is added in each round until a predetermined sufficiently small error rate is reached or a predetermined maximum number of iterations is reached. After the construction procedure of every classifier, when a new data point comes in, AdaBoost [7] will utilize the final hypothesis and combine all the weight for each small decision tree according to the training step and do the prediction. Based on this kind of learning process, AdaBoost is able to lower the probability of facing overfitting or generalization problems and maintain high accuracy during training.

**Algorithm 2** AdaBoost Algorithm

Given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in \mathcal{X}, y \in \{-1, +1\}$. Initialize: $D_1(i) = 1/m$ for $i = 1, ..., m$ For $t = 1, ..., T$

- Train weak learner using distribution $D_t$
- Get weak hypothesis $h_t : \mathcal{X} \to \{-1, +1\}$
- Aim: select $h_t$ with low weighted error:

$$\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i] \tag{2}$$

- Choose $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$
- Update, for $i = i, ..., m$ :

$$D_{i+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t} \tag{3}$$

  where $Z_t$ is a normalization factor (chosen so that that $D_{t+1}$ will be a distribution)

Output the final hypothesis

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \tag{4}$$

---

XGBoost Model XGBoost is a supervised learning model based on the GBDT algorithm. The core idea is:

- Keep adding trees and do feature splitting to grow a tree, each tree at a time, it learns a new function to fit the residuals that been predicted last time. Let $\hat{y}$ be the prediction value, $w_{q(x)}$ be the score for leaf $q$, $F$ for the set of all regression tree and $f$ be regression tree, we have,

$$\hat{y} = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i) \quad , where\ F = f(x) = w_{q(x)} (q : \mathbb{R}^m \to T, w \in \mathbb{R}^T)$$

- Whenever we get k trees after training, we have to predict the score of a sample. it will fall to the corresponding leaf node in each tree, and each leaf node will correspond to a score

- The final step is to add up the corresponding scores of each tree, which will serve as the predicted value of the sample

The core algorithm is that the big difference from GBDT is the definition of the target function. As shown in Appendix C, we first pick an f to minimize our target function (here f can be approximated using Taylor's formula). The essence for this step is that optimize the sample distributed a leaf node is also the optimization process of objective function since every lead will inject with an objective function. This is also the reason why parallel processing can be realized, which makes XGBoost has a leap in speed. After introducing and expending the regularization variable $\Omega$, which contributes a lot in avoiding the overfitting problem, we can get our objective function (16). Then we can take derivative respect to $w_j$ and find the optimal value and solution (detail algorithm can be found in Appendix J).

## 3.5 Adjustment

### 3.5.1 PPMCC and PCA

For each individual problem, PPMCC and PCA are adopted to drop some features with high linearity and reduce inputs' dimensionality in order to improve the accuracy and avoid overfitting and underfitting problem.
PPMCC is a way of measuring the similarity of vectors whose output ranges from -1 to +1, where 0 represents no correlation, negative values negative correlation, and positive values positive correlation. It is worthwhile to note that PPMCC is only sensitive to linearity relations. Therefore, we adopt this method to drop the features with high linear correlations before fitting into our model.
PCA calculated the covariance matrix of the data matrix and obtain the eigenvalues and eigenvectors of the corresponding covariance matrix. Then it composed a matrix of the eigenvectors corresponding to the k features with the largest eigenvalue. In this way, the data matrix can be transformed into a new dimensional space to reduce the dimensionality. In addition, We use SVD to implement PCA, the input vectors are then reduced to a $n \times 1$ dimension.

### 3.5.2 Parameter Adjustment

Since the original classifier adopted in the test round only uses the default parameter values. Parameter adjustment is essential for us to improve the model prediction accuracy and generate a more precise result. Our routine is to adjust n_estimators (iteration times) first. We create a list with several possible parameters to gain the accuracy score for each of them and find the most effective one. Then we shrink the parameter interval to find an acceptable value. After fixing n_estimators, we then change a different value on max_depth, which enhance our model's ability to learn about more specific and local samples. The final step is to find a suitable learning rate with the same method as adjust n_estimators.

# 4 Data Analysis

Considering which keyword will decide the success product which we assume that is represented by star rating and how to scheme our blueprint, we use the control a single variable to judge the reflection. For this purpose, we divide the strategy into three parts: handling reviews and star rating feature, combing the relationship between review and star rating, analyzing characteristics of success.

## 4.1 Reviews and Rating Features

In the past several years, various commercial companies endeavor to do such many pieces of big data research that investigate which product satisfied consumers' affection. There is no exception for Sunshine Company to trace and obtain the most useful information based on the ratings and reviews. Serve as a product promotion analyst, we are obligated to widen the road by information reprocessing ahead of time. Hence, the two main sets of data from the table are made to the following Figure 9 about the relation between reviews length and star rating. And the more concrete relationship among reviews, rating and product id in Appendix D.

The x-axis stands for comment length and the y-axis represents the different


Figure 9: Relation between Review Length and Star Rating

star ratings. In particular, one point is one purchase record. From the distribution of the points in the picture, there is no doubt that five star has the longest comment and more sentences in excess of 2000. Compared between three star and one star, the result is also clear, in which three star has longer reviews. Similarly, the graphs about microwave oven and pacifier in Appendix E also present the same idea. From what has been discussed above, we can prove that there exists a positive correlation between length and star rating.

Out of curiosity about the comment keywords, we would like to seek for the top 50 most frequently occurring words of comments to draw Figure 10 as well as microwave oven and pacifier figures in Appendix F.

It's not hard to see that there are almost positive words: 'great', 'good', 'love', 'well' and few a bit of negative words: 'old', 'high' except for commodity name or using method words. The result shows everyone has the chance to express their own voice whatever positive or negative, which implies our data is not an extremely positive example and exemplifies the truth fact that no consensus is a common phenomenon. Especially, what we can't deny is that people are more likely to comment on products they're satisfied with. And people also pay more attention to the price, the appearance of goods, convenient to operation as well as service life from the word like 'price', 'look', 'easy' and 'month' from the graphs

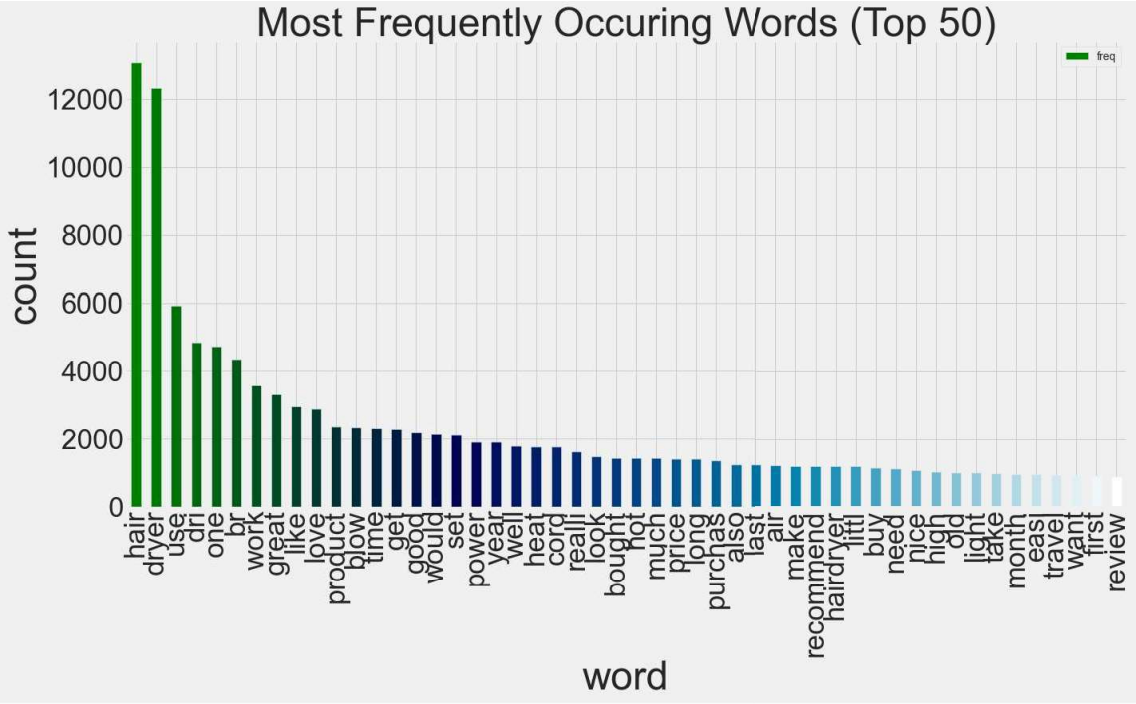


Figure 10: The words frequency in comments

named most frequently occurring words.

In order to predict the potential development of these three products, we exploit the mentioned machine learning model to train delivered data and draw Figure 11 and the same type graph in Appendix G. According to the importance in the figure, the use of feelings will be mentioned several times in the comments for hairdryer and pacifier. What' s more, the appearance of the pacifier is also quite important. However, for the microwave oven, the quality and service seem more essential by our model. We compare this set of figures with the set of figures about the most frequently occurring words. For hairdryers, we find that they both have 'hair', 'dryer', 'use', 'work', 'great' the highest appears rate words, which is also suitable to microwave oven and pacifier. But ought to the small amount of data given by microwave oven, the model may not perform well. Thus, our model is verified by the above compare and can reflect what people most concern.

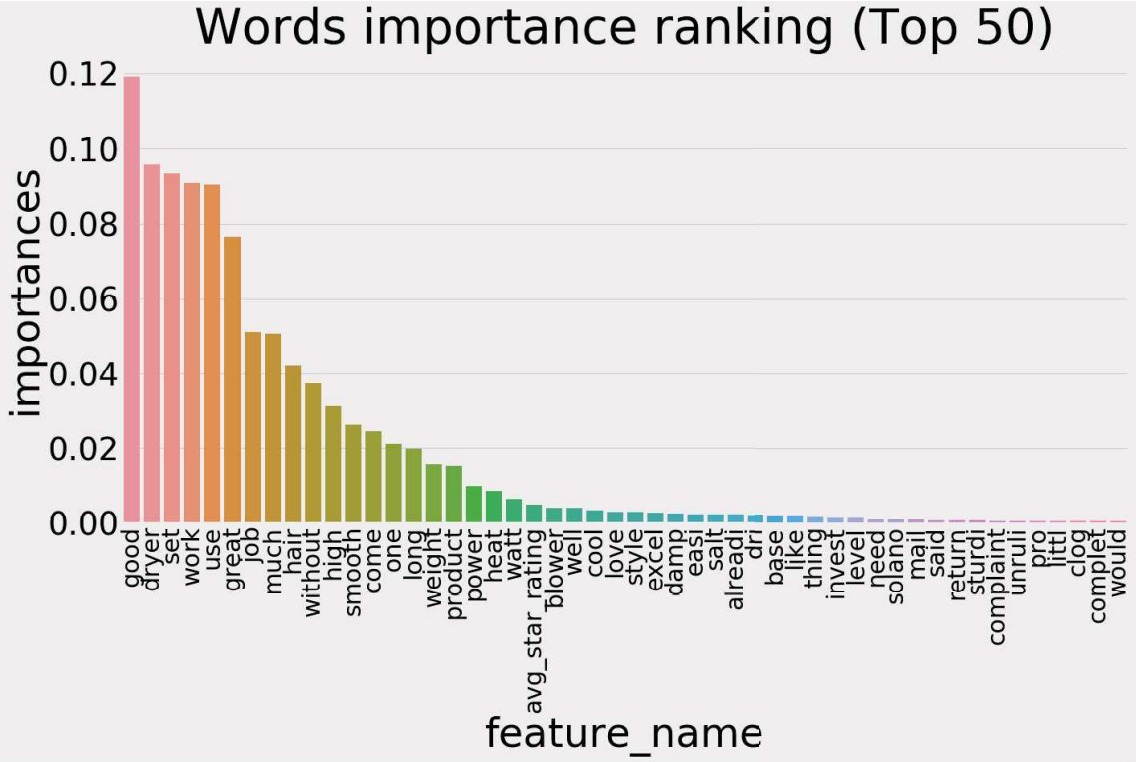


Figure 11: The words importance based on training model for hair_dryer

## 4.2 The Influence to Reputation

For research for time-based measures and patterns within each data set, we assume the reputation has a positive relationship with the average star rating and draw a histogram broken lines combination diagram as figure 2, 4, 6 before. We found the data sample is not enough to summarize the overall tendency that is influenced by the history factor involved before. Faced with oscillating star rating before 2011, single star rating has an inestimable influence on the average star rating on account of a small amount of samples. We decisively abandon the purchase records before 2011 to look for the later regular. For hairdryer and microwave oven, the sales volume gradually rose from 2011 to 2014 and then tended to be a stable stage. Nevertheless, microwave oven potential growth space is obviously larger than the hairdryer, which tends to steady. At the same time, the average star rating of hairdryer and microwave oven grew up all the time, which means this product is more and more popular with persons and accumulates a better reputation than before. Let us divert our attention to the pacifier's diagrams. Pacifier has the similar characteristic of sales volume increase as

well as the hairdryer. As for the average star rating, it always maintained a high level.

We are willing to assist the company to find out the correlation of some type of feedback and popularity. First of all, we attempt to make use of a single variable approach to give the gross tendency of variation. So we determine to set the star rating temporarily fixed to observe the total quantity reviews each year and create the Figure 12 and other graphs in Appendix H.

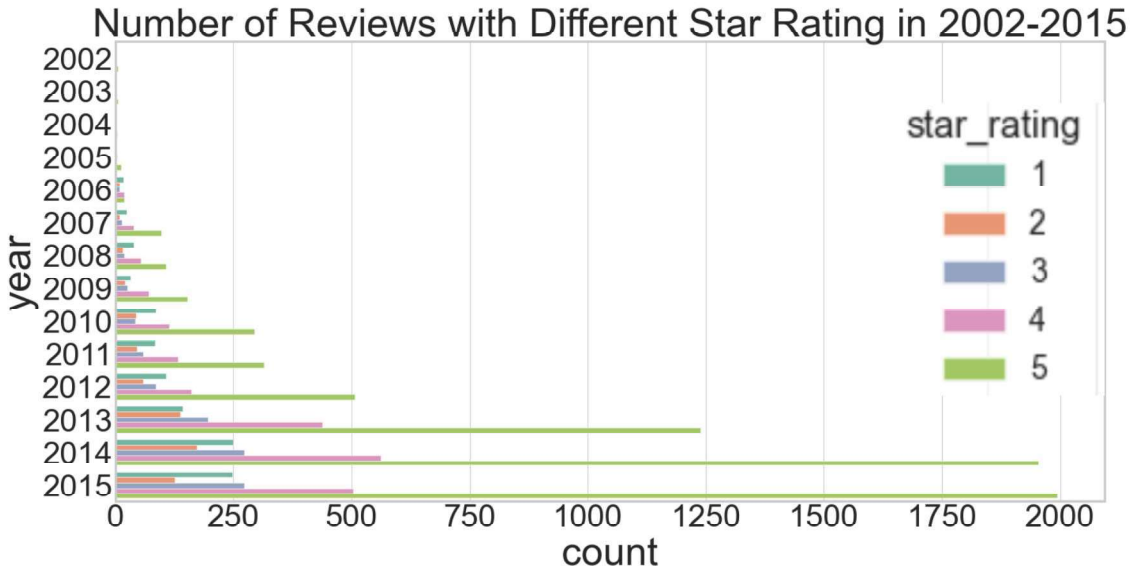


Figure 12: The amount of reviews of different star_rating for hair_dryer

Through browsing the change of cylinder size, before 2007 it is difficult to analyze the situation at that moment by the fewer count of reviews. But for the later years, we observe that the sales volume grows up slowly from 2007 to 2010. However, after 2010, the sales volume surges regardless of star rating until 2014. Then the sales volume leveled off or appear a slight drop. To figure out the reason for changing, from the e-commercial development history, it is inescapable that the common family is lacking computers and other intelligent electronic terminals that restrict the opportunity of online shopping at that period. With the computer widespread use and pursuit a convenient lifestyle, many products are sold online rather than offline. And this business model meets the customers' desire and release the realistic burden of the company. Waiting for 2015, the market demand is basically satisfied, so it presents a stable state.

### 4.3 Characteristics of Successful Products

In terms of the reviews and rating features of three products and our model training, we succeeded in drawing the Figure 13 for hairdryer and other same type diagrams in Appendix I. Because the model refers to reviews word frequency and average star rating which can stand for product reputation, so this result will make specific guidance to the success of the product. We identify the words: 'good', 'work', 'use', 'great', which sum is more than 40 percent, are the key to success for the hairdryer. Thus, consumers take more care about the influence between a product with hair except 'good'. As for people who bought a wonderful microwave oven, they concern more about the machine works well. Finally, pacifiers are laid customers' hope on gratifying kids' demand.

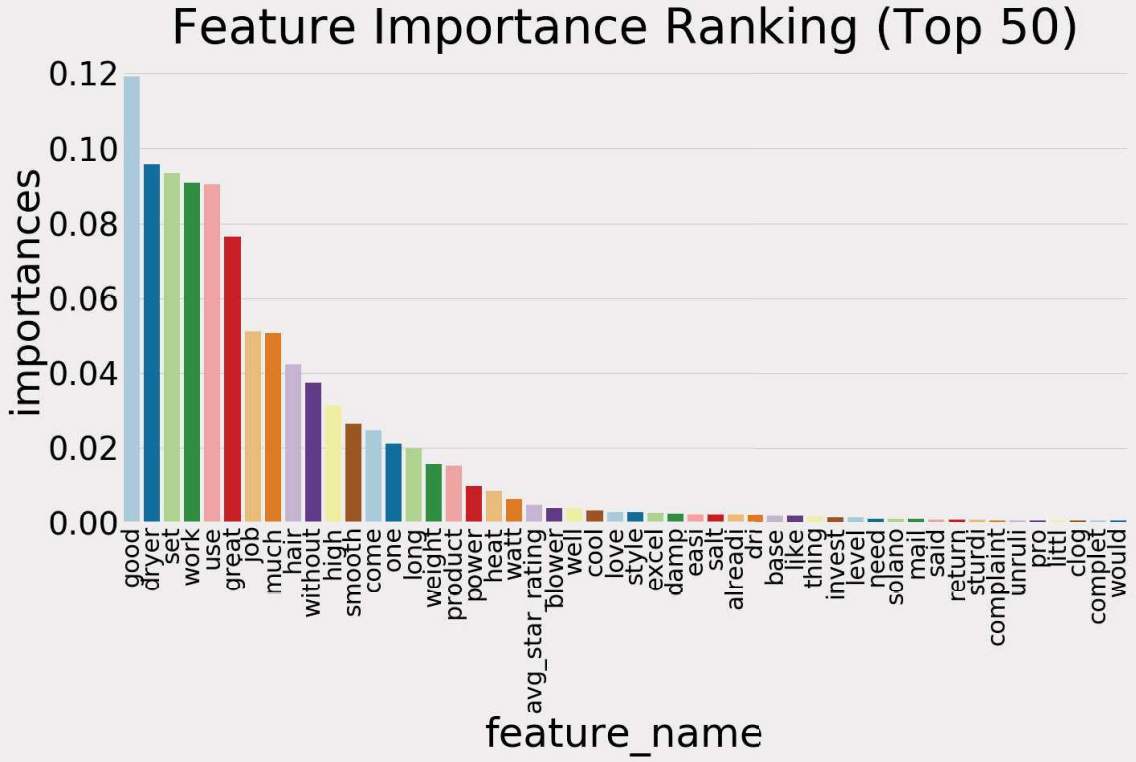


Figure 13: Feature importance graph for success

# 5 Strengths and Weaknesses

## 5.1 Model Strength

- **Quantifies the text-based features**: We use CountVectorizer to make it possible for Our model to utilize text-based data by quantifying them into vectors in order to illustrate how reviews affect the products both positively and negatively and find the connection between star rating levels and review words.
- **Development context**: Our model fully exploits the information of historical data and finds the trend of the product's reputation from the perspective of historical development.
- **Accuracy and stability**: Our model is carefully selected based on the performance of 10 effective prediction models and appropriate parameters are chosen to generate the best outcomes with no overfitting or underfitting problems.
- **1:1 positive and negative data samples**: When we predict the success of certain products based on reviews and star ratings, we randomly choose the same amount of negative data points as the assigned positive data to avoid data imbalance problem.
- **Feature extraction**: After a systematic consideration, we adopt PCA combined with PPMCC to obtain reasonable features.
- **Data cleaning**: Before we apply CountVectorizer to quantifies the reviews, we get rid of all special characters and stop words in order to avoid duplication.

## 5.2 Model Weakness

- **Feature non-linear correlations**: PPMCC method only sensitive to linear correlations, which may cause a problem when our features have non-linear relations.

- **Special reviews**: Emoji and other reviews with special characters are ignored in our model.
- **Data volume**: Our model needs a substantial amount of data to maintain the accuracy of doing prediction. As such, a small dataset like microwave.tsv might not able to be effectively represented like the other two products.
- **Success assumption**: Our prediction for success based on the sales volume of certain products, which may not be sufficient enough.

# 6 Conclusion

In this paper, we first divide the whole problem into three case studies and adopt *WordVectorize* to turn the text-based question into a computational problem. For the purpose of pursuing an approximately perfect model, we reorganize the form and collected the sum of different user ratings for popularity as well as analyze the fluctuation of sales by year. Meanwhile, we also add some sensitive words to measure the frequency after decomposing the customers' reviews. Confronted with the complex purchase psychological factor and credit accumulation of products, our model takes a multi-variate approach and chose Adaboost and XGboost to be the main algorithm in our analysis model. Through the interaction between the purchase frequency and user feedback of the same kind of products, we explain which factors determine the product sales volume and proposed some possible strategies. The biggest success of our model is that it applies a predictive fit method to help the Sunshine Company to resolve how changes in these factors influence the product's reputation. That will make the new products successful on the online market or let company remedy drawback before suffering an even greater economic loss. Finally, the advantages and disadvantages of the model are discussed.

# References

[1] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[2] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[3] Michael Collins, Robert E Schapire, and Yoram Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1-3):253–285, 2002.

[4] Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.

[5] Weiming Hu, Wei Hu, and Steve Maybank. Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(2):577–583, 2008.

[6] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[7] Gunnar Rätsch, Takashi Onoda, and K-R Müller. Soft margins for adaboost. *Machine learning*, 42(3):287–320, 2001.

[8] Robert E Schapire. Explaining adaboost. In *Empirical inference*, pages 37–52. Springer, 2013.

[9] L Torlay, Marcela Perrone-Bertolotti, Elizabeth Thomas, and Monica Baciu. Machine learning–xgboost analysis of language networks to classify patients with epilepsy. *Brain informatics*, 4(3):159–169, 2017.

# Appendices

## Appendix A Test Model Score for Rating Level

hair_dryer.tsv

| Algorithm | Score | Training Time | Scoring Time |
|---|---|---|---|
| Dummy | score = 0.456 | 0.000s | 0.003s |
| KNN(3) | score = 0.596 | 0.019s | 0.153s |
| RBF SVM | score = 0.474 | 0.533s | 0.131s |
| Decision Tree | score = 0.932 | 0.062s | 0.001s |
| Random Forest | score = 0.947 | 0.015s | 0.002s |
| xgboost | score = 0.965 | 9.039s | 0.014s |
| Neural Net | score = 0.860 | 5.478s | 0.003s |
| AdaBoost | score = 0.947 | 1.323s | 0.044s |
| Naive Bayes | score = 0.561 | 0.026s | 0.005s |
| QDA | score = 0.684 | 0.184s | 0.011s |

microwave.tsv

| Algorithm | Score | Training Time | Socring Time |
|---|---|---|---|
| Dummy | score = 0.562 | 0.000s | 0.001s |
| KNN(3) | score = 0.750 | 0.002s | 0.007s |
| RBF SVM | score = 0.312 | 0.023s | 0.006s |
| Decision Tree | score = 0.938 | 0.008s | 0.000s |
| Random Forest | score = 0.875 | 0.006s | 0.001s |
| xgboost | score = 0.938 | 0.723s | 0.002s |
| Neural Net | score = 0.875 | 3.970s | 0.001s |
| AdaBoost | score = 0.946 | 0.278s | 0.009s |
| Naive Bayes | score = 0.438 | 0.007s | 0.001s |
| QDA | score = 0.562 | 0.024s | 0.002s |

pacifier.tsv

| Algorithm | Score | Training Time | Socring Time |
|---|---|---|---|
| Dummy | score = 0.525 | 0.000s | 0.000s |
| KNN(3) | score = 0.543 | 0.545s | 7.159s |
| RBF SVM | score = 0.537 | 21.250s | 5.214s |
| Decision Tree | score = 0.792 | 0.608s | 0.011s |
| Random Forest | score = 0.695 | 0.054s | 0.012s |
| xgboost | score = 0.880 | 19.207s | 0.039s |
| Neural Net | score = 0.748 | 48.188s | 0.014s |
| AdaBoost | score = 0.903 | 19.075s | 0.037s |
| Naive Bayes | score = 0.686 | 0.299s | 0.099s |
| QDA | score = 0.510 | 1.627s | 0.191s |

# Appendix B Test Model Score for Potential Success

hair_dryer.tsv

| Algoeithm | Score | Training Time | Scoring Time |
|---|---|---|---|
| Dummy | score = 0.193 | 0.000s | 0.011s |
| KNN(3) | score = 0.547 | 6.505s | 279.467s |
| RBF SVM | score = 0.583 | 1,401.787s | 170.731s |
| Decision Tree | score = 0.608 | 5.983s | 0.060s |
| Random Forest | score = 0.584 | 0.256s | 0.057s |
| XGBoost | score = 0.636 | 423.858s | 0.324s |
| Neural Net | score = 0.594 | 52.310s | 0.064s |
| AdaBoost | score = 0.633 | 79.139s | 2.731s |
| Naive Bayes | score = 0.174 | 1.429s | 1.139s |
| QDA | score = 0.384 | 75.755s | 3.419s |
| Linear SVC | score = 0.496 | 297.832s | 0.028s |

microwave.tsv

| Algorithm | Score | Training Time | Socring Time |
|---|---|---|---|
| Dummy | score = 0.189 | 0.000s | 0.001s |
| KNN(3) | score = 0.446 | 0.247s | 4.343s |
| RBF SVM | score = 0.424 | 22.489s | 3.000s |
| Decision Tree | score = 0.492 | 0.344s | 0.007s |
| Random Forest | score = 0.443 | 0.041s | 0.009s |
| xgboost | score = 0.576 | 41.644s | 0.026s |
| Neural Net | score = 0.539 | 24.179s | 0.005s |
| AdaBoost | score = 0.514 | 4.749s | 0.208s |
| Naive Bayes | score = 0.331 | 0.115s | 0.084s |
| QDA | score = 0.139 | 0.456s | 0.098s |

pacifier.tsv

| Algorithm | Score | Training Time | Socring Time |
|---|---|---|---|
| Dummy | score = 0.202 | 0.000s | 0.003s |
| KNN(3) | score = 0.620 | 9.171s | 707.446s |
| RBF SVM | score = 0.536 | 3895.557s | 689.226s |
| Decision Tree | score = 0.677 | 13.565s | 0.138s |
| Random Forest | score = 0.665 | 0.535s | 0.122s |
| xgboost | score = 0.684 | 916.255s | 0.712s |
| Neural Net | score = 0.651 | 240.519s | 0.165s |
| AdaBoost | score = 0.686 | 176.781s | 6.171s |
| Naive Bayes | score = 0.180 | 3.143s | 2.352s |
| QDA | score = 0.490 | 316.773s | 12.609s |

# Appendix C XGBoost Objective Function

**Algorithm 3** XGBoost Objective Function

Goal: Find $f_t$ to minimize $Obj^{(t)}$ Let $g_i = \partial_{\hat{y}^{t-1}} l(y_i, \hat{y}^{t-1})$ and $h_i = \partial^2_{\hat{y}^{t-1}} l(y_i, \hat{y}^{t-1})$

$$Obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant \tag{5}$$

$$= \sum_{i=1}^{n} [2(\hat{y}_i^{(t-1)} - y_i) f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + constant$$

$$(By\ Taylor's\ Formula)$$

$$\approx \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + constant \tag{6}$$

Aim: Minimize $Obj^{(t)}$:

$$Define: f_t(x) = w_{q(x)},\ w \in \mathbb{R}^T, q : \mathbb{R}^b \to \{1, 2, ..., T\}$$

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2$$

$$I_j = \{i | q(x_i) = j\}$$

$$Obj^{(t)} \approx \sum_{i=1}^{n} [(g_i f_t(x_i)] + \Omega(f_t) \tag{7}$$

$$= \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \tag{8}$$

$$Define: G_j = \sum_{i \in I_j} g_i \quad H_j = \sum_{i \in I_j} h_i$$

$$Obj^{(t)} = \sum_{j=1}^{T} [G_j wj + \frac{1}{2} (H_j + \lambda) w_j^2] + \gamma T \tag{9}$$

Final outcome

$$w_j^* = -\frac{G_j}{H_j + \lambda} \quad Obj = -\sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{10}$$

# Appendix D Review Length, Star Rating and Product ID Reputation

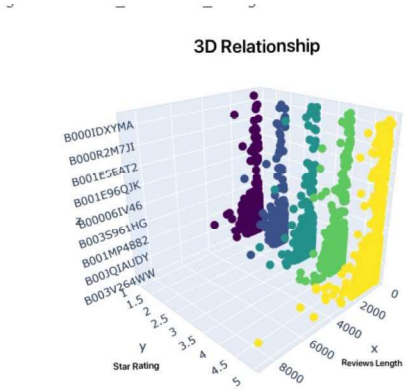


Figure 14: Top 50 review words for hairdryer
3D model for hairdryer

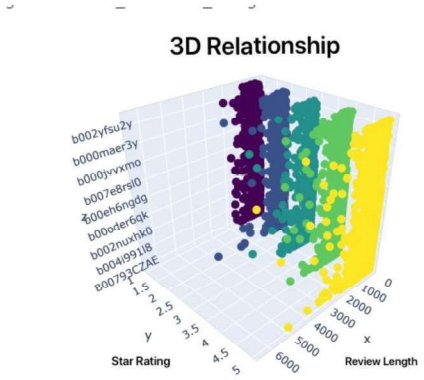


Figure 15: Top 50 review words for hairdryer
3D model for microwave oven

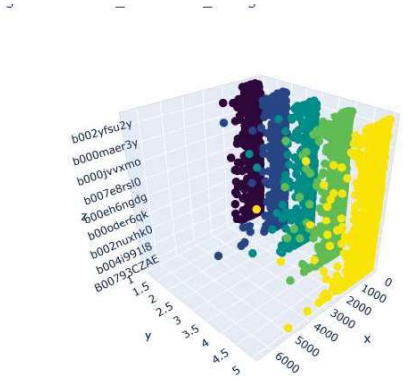


Figure 16: Top 50 review words for hairdryer
3D model for pacifier

# Appendix E Relations Between Review Length and Star Rating

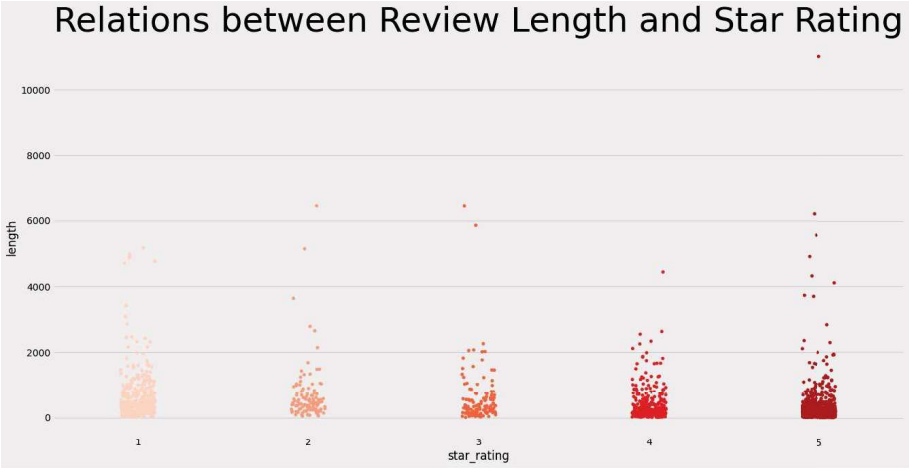


Figure 17: Relations between review length and star rating for microwave_oven

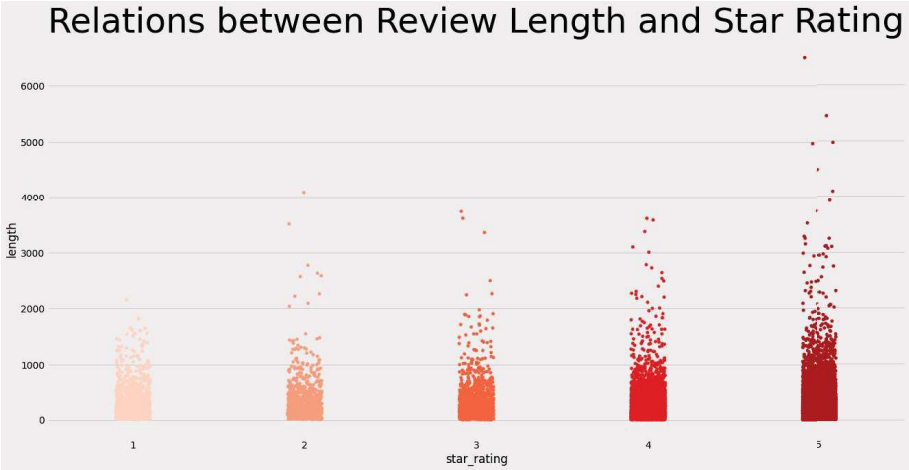


Figure 18: Relations between review length and star rating for pacifier

# Appendix F Words Frequency in Comments

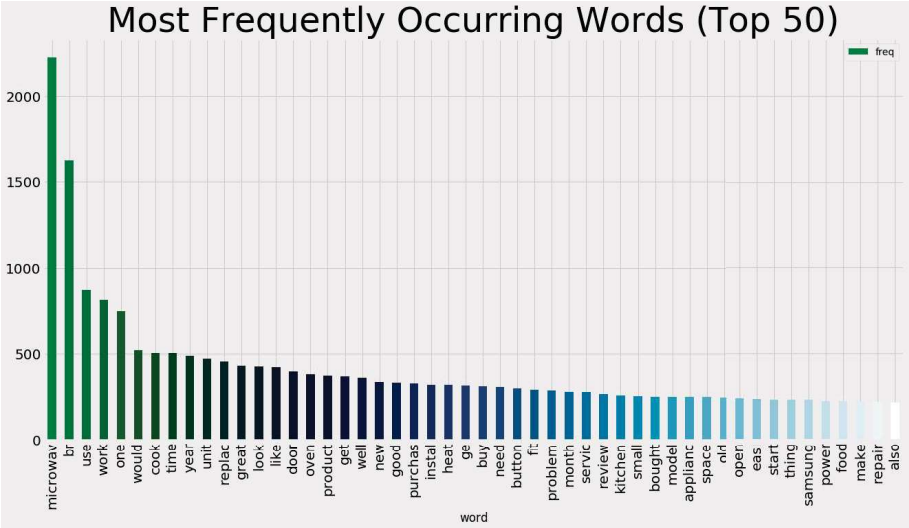


Figure 19: The words frequency in comments for microwave_oven

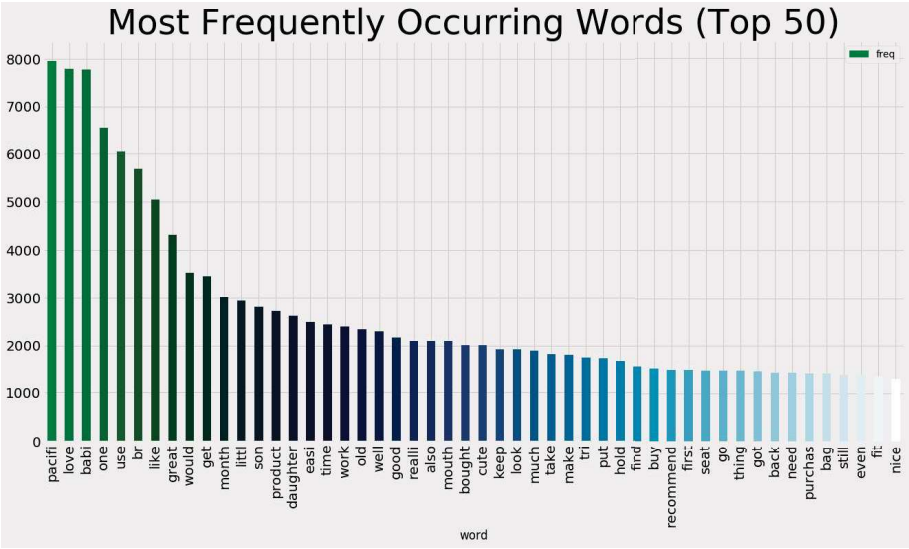


Figure 20: The words frequency in comments for pacifier

# Appendix G Words Importance Based on Training Model

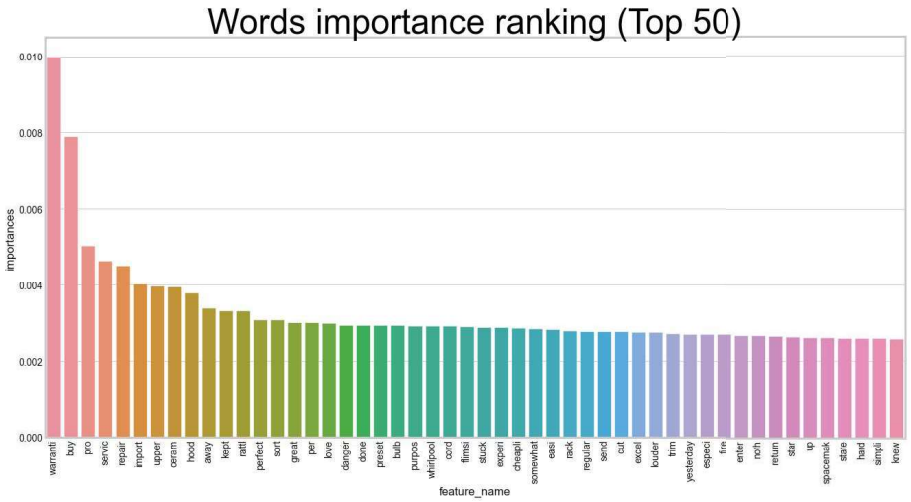


Figure 21: The words importance based on training model for microwave oven

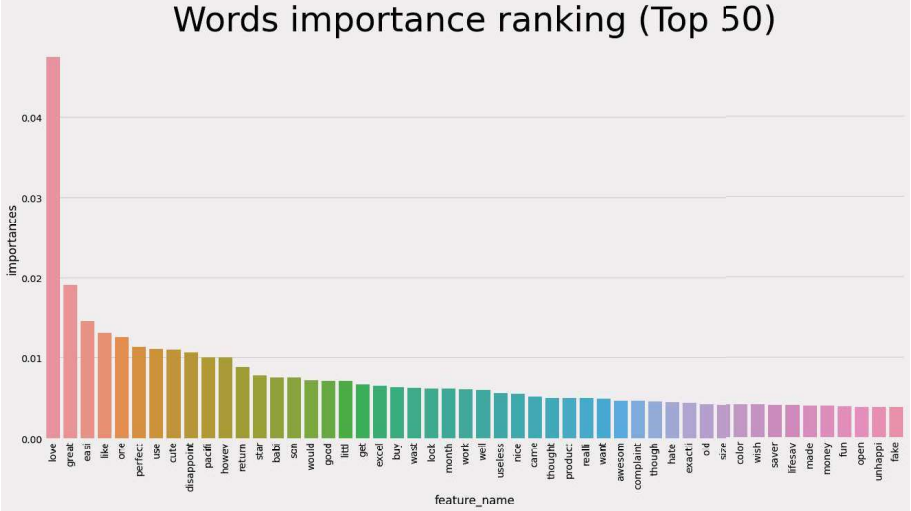


Figure 22: The words importance based on training model for pacifier

# Appendix H Amount of Reviews of Different Star Rating

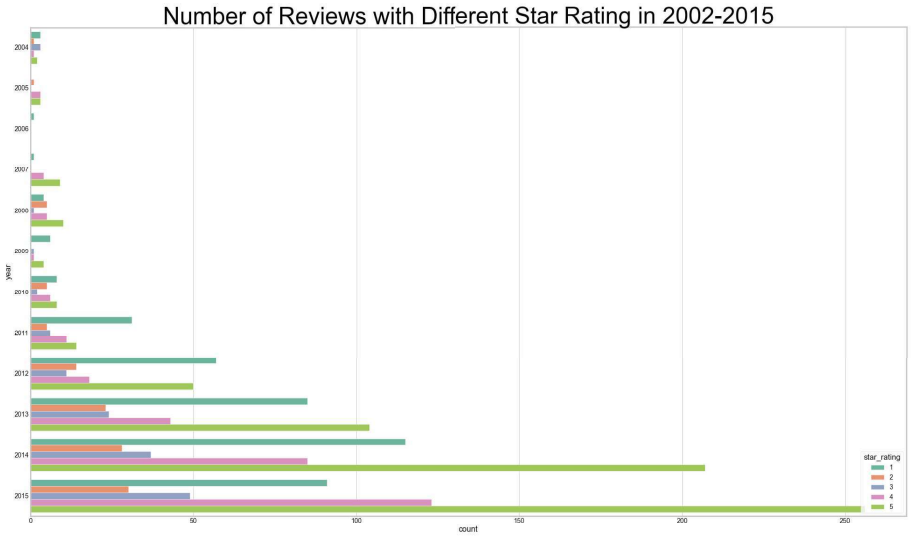


Figure 23: The amount of reviews of different star_rating for microwave_oven

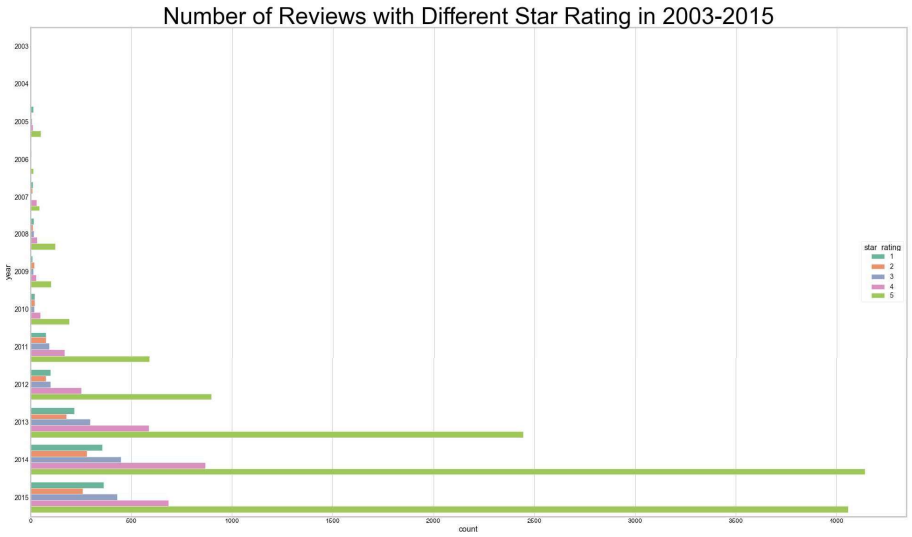


Figure 24: The amount of reviews of different star_rating for pacifier

# Appendix I Words Importance for Success

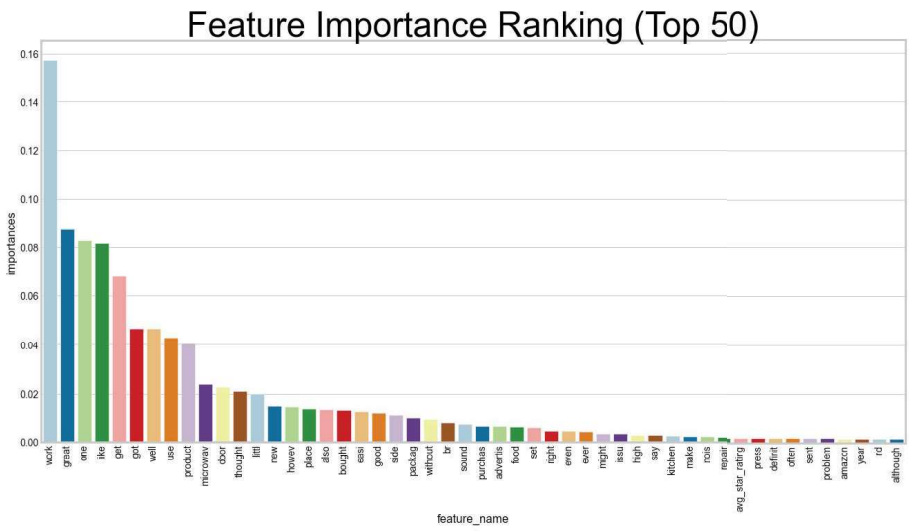


Figure 25: The words importance for success for microwave oven

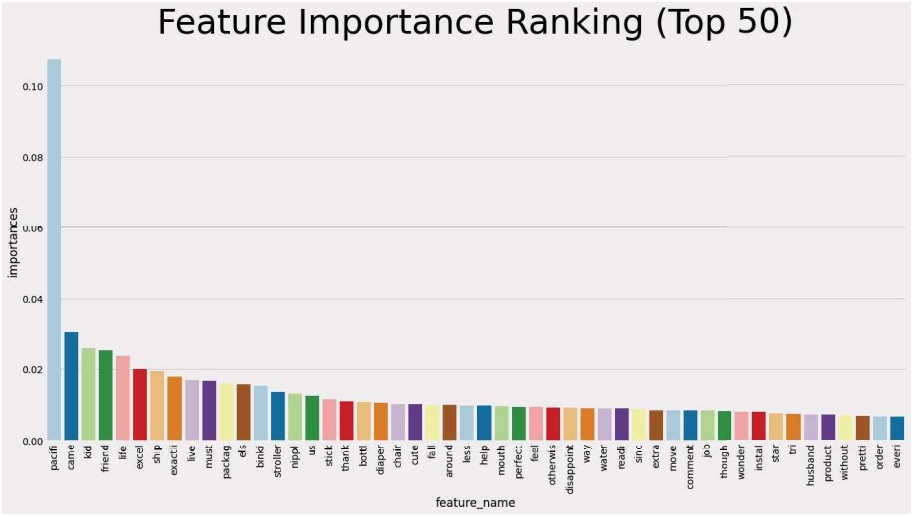


Figure 26: The words importance for success for pacifier

# Appendix J XGBoost Algorithm

**Algorithm 4** XGBoost Algorithm

Input: $I,\ instance\ set\ of\ current\ node$
Input: $d,\ feature\ dimension$
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i,\ H \leftarrow \sum_{i \in I} h_i$
**for** $k = 1\ to\ m$ **do**
  $G \leftarrow 0,\ H \leftarrow 0$
  **for** $j\ in\ sorted(I,\ by\ x_{jk})$ **do**
    $G_L \leftarrow G_L + g_j,\ H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L,\ H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  **end for**
**end for**
Output: Split with max score